JBoss Enterprise Application Platform 6.3

Guide d'administration et de configuration

À utiliser dans Red Hat JBoss Enterprise Application Platform 6

Last Updated: 2017-10-27

JBoss Enterprise Application Platform 6.3 Guide d'administration et de configuration

À utiliser dans Red Hat JBoss Enterprise Application Platform 6

Notice légale

Résumé

Cet ouvrage est un guide d'administration et de configuration de Red Hat JBoss Enterprise Application Platform 6, qui inclut des correctifs publiés.

Table des matières

CHAPITRE 1. INTRODUCTION

1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 6

Red Hat JBoss Enterprise Application Platform 6 (JBoss EAP 6) est une plate-forme middleware construite sur la base de standards ouverts et compatibles avec Java Enterprise Edition 6. Elle intègre JBoss Application Server 7 avec un clustering de haute disponibilité, une messagerie, une mise en cache distribuée et autres technologies.

JBoss EAP 6 comprend une nouvelle structure modulaire qui permet aux services d'être activés seulement si nécessaire, améliorant ainsi la vitesse de démarrage.

La console de gestion et l'interface CLI rendent la modification des fichiers de configuration XML inutile et rajoutent la capacité d'encoder et d'automatiser des tâches.

En plus, JBoss EAP 6 comprend des frameworks de développement et des API pour développer rapidement des applications de Java EE sécurisées et évolutives.

Rapporter un bogue

1.2. LES FONCTIONNALITÉS DE JBOSS EAP 6

Tableau 1.1. Fonctionnalités 6.3.0

Fonctionnalité	Description
Certification Java	JBoss Enterprise Application Platform 6 Full Profil et Web Profile certifiés.
Domaine géré	• Un domaine géré procure une gestion centralisée d'instances de serveurs multiples et d'hôtes physiques, tandis qu'un serveur autonome autorise une instance de serveur unique. • Gestion de groupe de configuration par-serveur, déploiement, liaisons de socket, modules, extensions et propriétés système. • Gestion centralisée et simplifiée de la sécurité des applications (y compris les domaines de sécurité).
console de gestion et interface CLI	Interfaces de gestion de serveur autonome ou nouveaux domaines. L'édition des fichiers de configuration XML n'est plus nécessaire. L'interface CLI comprend également un mode batch qui peut encoder et automatiser les tâches de gestion.

Fonctionnalité	Description
La disposition du répertoire est simplifiée	Le répertoire **`modules`** contient maintenant les modules du serveur d'applications. Les répertoires communs et spécifiques au serveur **`lib`** sont obsolètes. Les répertoires **`domain`** et **`standalone`** contiennent les artefacts et les fichiers de configuration pour les déploiements autonomes et de domaine respectivement.
Mécanisme de chargement de classes modulaire	Les modules sont chargés et déchargés à la demande. Cela améliore la performance et la sécurité, et permet des démarrages et redémarrages plus rapides.
Gestion des sources de données simplifiée	Les pilotes de base de données peuvent être déployés comme tout autre service. En plus, les sources de données sont créées et gérées directement dans la console de gestion ou l'interface CLI.
Utilisation réduite et plus efficace des ressources	JBoss EAP 6 utilise moins de ressources système et les utilise plus efficacement que dans les versions précédentes. Entre autres avantages, JBoss EAP 6 démarre et s'arrête plus rapidement que JBoss EAP 5.

Rapporter un bogue

1.3. JBOSS EAP 6 OPERATING MODES

JBoss EAP 6 fournit deux modes d'opération pour les instances de JBoss EAP 6 : serveur autonome ou domaine géré.

Les deux modes diffèrent dans la façon dont les serveurs sont gérés, pas dans leur capacité à traiter les demandes de l'utilisateur final. Il est important de noter que la fonctionnalité de cluster de haute disponibilité (HA) est disponible avec les deux modes de fonctionnement. Un groupe de serveurs autonomes peut être configuré pour former un cluster HA.

Rapporter un bogue

1.4. LES SERVEURS AUTONOMES

Un mode de serveur autonome est un processus indépendant qui ressemble au mode d'exécution unique des anciennes versions de JBoss EAP.

L'instance de JBoss EAP 6 qui exécute en tant que serveur autonome est une instance unique, qui peut exécuter optionnellement dans une configuration clusterisée.

Rapporter un bogue

1.5. LES DOMAINES GÉRÉS

Le mode d'opération d'un domaine géré permet la gestion de multiples instances de JBoss EAP 6 à partir d'un seul point de contrôle.

Les collections de serveur JBoss EAP 6 centralement gérées sont connues comme membres d'un domaine. Toutes les instances JBoss EAP 6 d'un domaine partagent une stratégie de gestion en commun.

Un domaine consite en un contrôleur de domaine, un ou plusieurs controleur(s) hôte, et zéro ou plusieurs groupes de serveurs par hôte.

Un contrôleur de domaine est un point central à partir duquel le domaine est contrôlé. Il s'assure que chaque serveur est configuré suivant la stratégie de gestion du domaine. Le contrôleur du domaine est également un contrôleur hôte.

Un contrôleur hôte est un hôte physique ou virtuel sur lequel le script **`domain.sh`** ou **`domain.bat`** exécute. Les contrôleurs hôte sont configurés pour déléguer les tâches de gestion de domaine au contrôleur de domaine.

Le contrôleur hôte de chaque hôte interagit avec le contrôleur de domaine pour contrôler le cycle de vie des instances de serveur de l'application exécutant sur son hôte et pour aider le contrôleur de domaine à les gérer. Chaque hôte peut contenir plusieurs groupes de serveurs.

Un groupe de serveurs est un ensemble d'instances de serveurs avec JBoss EAP 6 installé dessus, et qui sont gérées et configurées comme une entité unique. Le contrôleur de domaine gère la configuration et les applications déployées sur les groupes de serveurs. Ainsi, chaque serveur dans un groupe de serveurs partage les mêmes configurations et déploiements.

Il est possible qu'un contrôleur de domaine, un contrôleur hôte unique et plusieurs serveurs s'exécutent dans la même instance de JBoss EAP 6, sur le même système physique.

Les contrôleurs hôtes sont liés à des hôtes physiques (ou virtuels) spécifiques. Vous pouvez exécuter plusieurs contrôleurs hôtes sur le même matériel si vous utilisez différentes configurations, afin d'éviter que les ports et autres ressources n'entrent en conflit.

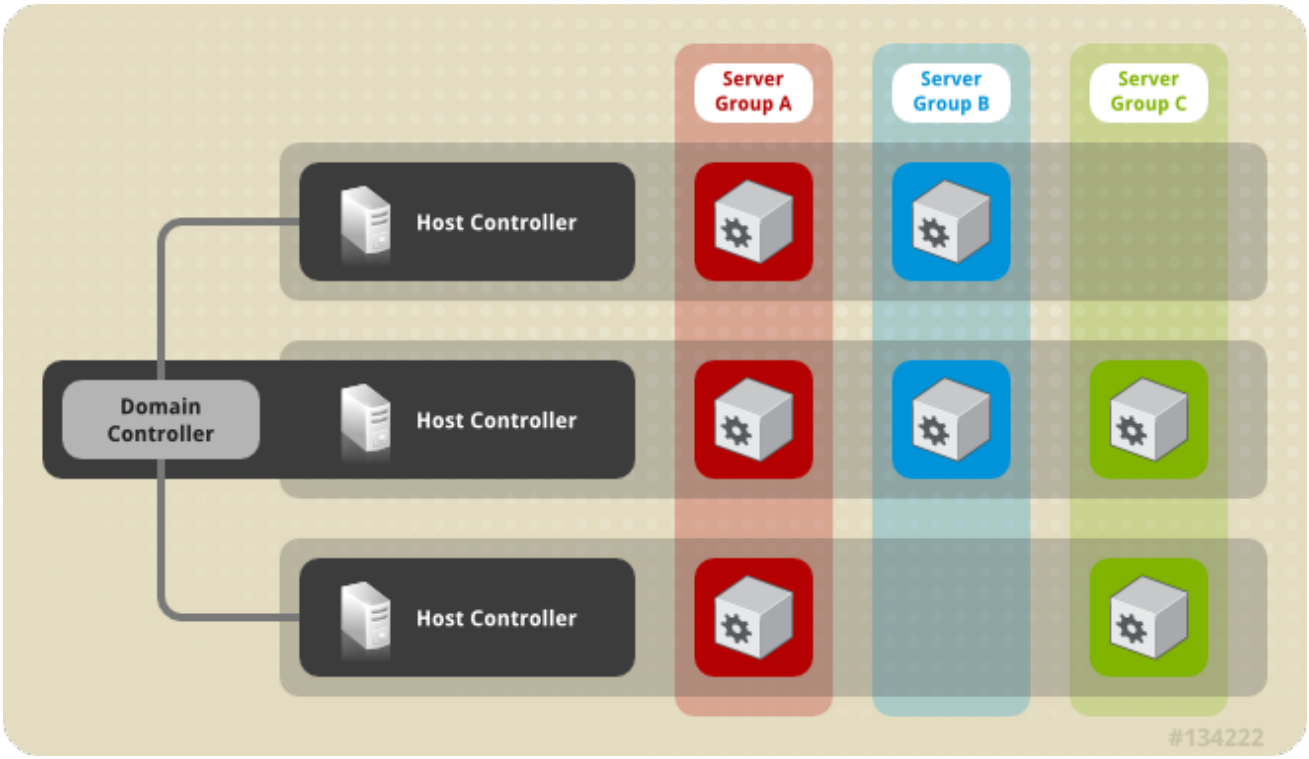


Figure 1.1. Représentation graphique d'un domaine géré

Rapporter un bogue

1.6. CONTRÔLEUR DE DOMAINE

Un contrôleur de domaine est une instance de serveur de JBoss EAP 6 qui agit en tant que point central de gestion pour un domaine. Une instance de contrôleur hôte est configurée pour agir en tant que contrôleur de domaine.

Les responsabilités principales d'un contrôleur de domaine sont les suivantes :

- Maintenir la politique centrale de gestion du domaine.
- S'assurer que tous les contrôleurs soient mis au courant de son contenu actuel.
- Assister tous les contrôleurs pour que toutes les instances en cours de JBoss EAP 6 soient configurées suivant cette politique.

La politique de gestion centrale est stockée par défaut dans le fichier **`domain/configuration/domain.xml`**. Ce fichier est le fichier d'installation JBoss EAP 6 non compressé, qui se trouve sur le système de fichiers de l'hôte du contrôleur de domaines.

Une fichier **`domain.xml`** doit se trouver dans le répertoire **`domain/configuration/`** du contrôleur hôte défini pour exécuter en tant que contrôleur de domaine. Ce fichier n'est pas obligatoire pour les installations sur les contrôleurs hôtes qui ne sont pas sensés exécuter en tant que contrôleurs de domaines. Cependant, la présence d'un fichier **`domain.xml`** sur un tel serveur n'a aucun effet néfaste.

Le fichier **`domain.xml`** contient les configurations de profil qui peuvent être exécutées sur les instances de serveur dans un domaine. Une configuration de profil inclut les paramètres détaillés des différents sous-systèmes qui composent un profil. La configuration de domaine inclut également la définition des groupes de sockets et les définitions de groupes de serveurs.

Rapporter un bogue

1.7. DOMAIN CONTROLLER DISCOVERY ET FAILOVER

Lorsque vous configurez un domaine géré, chaque contrôleur hôte doit être configuré avec les informations nécessaires pour communiquer avec le contrôleur de domaine. Dans JBoss EAP 6.3, chaque contrôleur hôte peut maintenant être configuré avec de multiples options pour trouver le contrôleur de domaine. Les contrôleurs hôtes peuvent parcourir la liste des options jusqu'à ce qu'une d'entre elle réussisse.

Cela permet aux contrôleurs hôtes d'être pré configurés avec des informations de contact d'un contrôleur de domaine secondaire. Un contrôleur hôte de sauvegarde peut être promu pour maîtriser s'il y a un problème avec le contrôleur de domaine principal, permettant aux contrôleurs hôtes de basculer automatiquement vers le nouveau master une fois qu'il a été promu.

Ce qui suit est un exemple sur la façon de configurer un contrôleur hôte avec des options multiples pour trouver le contrôleur de domaine.

```
<domain-controller>
    <remote security-realm="ManagementRealm">
          <discovery-options>
              <static-discovery name="primary" host="172.16.81.100"
port="9999"/>
              <static-discovery name="backup" host="172.16.81.101"
port="9999"/>
          </discovery-options>
    </remote>
</domain-controller>
```

Une option discovery statique inclut les attributs obligatoires suivants :

name

Le nom de cette option discovery de contrôleur de domaine

host

Le nom d'hôte du contrôleur de domaine distant.

Important

Le port du contrôleur de domaine distant.

Dans l'exemple suivant, la première option discovery est celle avec laquelle on attent un résultat positif. La seconde peut être utilisée pour les situations d'échec.

Si un problème survient avec le contrôleur principal de domaine, un contrôleur hôte qui a été démarré avec l'option **--backup** pourra être promu pour agir comme contrôleur de domaine.

NOTE

À partir d'un contrôleur hôte avec l'option **--backup** qui entraînera ce contrôleur à conserver une copie locale de la configuration du domaine. Cette configuration servira si le contrôleur hôte est reconfiguré pour pouvoir agir comme contrôleur de domaine.

Procédure 1.1. Promouvoir un contrôleur hôte comme contrôleur de domaine

1. Assurez-vous que le contrôleur de domaine d'origine a, ou est, arrêté.
2. Utiliser l'interface CLI pour vous connecter au contrôleur hôte qui deviendra le nouveau contrôleur de domaine.
3. Exécutez la commande suivante pour configurer le contrôleur hôte pour qu'il agisse comme nouveau contrôleur de domaine.

```
/host=HOST_NAME:write-local-domain-controller
```

4. Exécutez la commande suivante pour rechercher le contrôleur hôte.

```
reload --host=HOST_NAME
```

Le contrôleur hôte choisi à l'étape 2 agira maintenant en tant que contrôleur de domaine.

Rapporter un bogue

1.8. CONTRÔLEUR HÔTE

Un contrôleur hôte est lancé quand le script **domain.sh** ou **domain.bat** exécute. sur un hôte.

Le principale responsabilité d'un contrôleur hôte est la gestion de serveurs. Il délègue les tâches de gestion de domaines et est chargé de démarrer ou stopper les processus de serveurs d'application individuels qui exécutent sur son hôte.

Il entre en interraction avec le contrôleur de domaines pour gérer la communication entre les serveurs et le contrôleur de domaines. Plusieurs contrôleurs hôtes d'un domaine peuvent interagir avec un contrôleur de domaine unique. Par conséquent, tous les contrôleurs hôtes et les instances de serveurs exécutant en mode de domaine unique ont un contrôleur de domaine unique et doivent appartenir au même domaine.

Chaque contrôleur hôte lit par défaut sa configuration à partir du fichier **domain/configuration/host.xml** situé dans le fichier d'installation de JBoss EAP 6 décompressé sur le système de fichiers de son hôte. Le fichier **host.xml** contient les informations de configuration suivantes spécifiques à l'hôte particulier :

- Les noms des instances de JBoss EAP 6 censées être exécutées à partir de l'installation.
- Une des configurations suivantes :
  - La façon dont le contrôleur contacte le contrôleur de domaines pour s'enregistrer lui-même et pour accéder à la configuration de domaine.
  - La façon de rechercher et contacter un contrôleur de domaines éloigné.
  - Comment le contrôleur hôtes doit se persuader lui-même d'agir en tant que contrôleur de domaines
- Les configuration spécifiques à l'installation physique locale. Ainsi, les définitions d'interfaces nommées déclarées dans **domain.xml** peuvent être mappées vers une adresse IP particulière appartenant à une machine dans **host.xml**. Les noms de chemins d'accès abstraits de domain.xml peuvent être mappés vers les chemins d'accès du système de fichiers dans **host.xml**.

Rapporter un bogue

1.9. LES GROUPES DE SERVEURS

Un groupe de serveurs est un regroupement d'instances des serveurs qui sont gérés et configurés en un. Dans un domaine géré, chaque instance de serveur d'application appartient à un groupe de serveurs, même s'il en est le seul membre. Les instances de serveur d'un groupe partagent la même configuration de profil et le même contenu déployé.

Un contrôleur de domaines et un contrôleur hôte font appliquer la configuration standard sur toutes les instances de serveur de chaque groupe de serveurs sur son domaine.

Un domaine peut se composer de plusieurs groupes de serveurs. Différents groupes de serveurs peuvent être configurés avec des déploiements et des profils différents. Un domaine peut être configuré avec des niveaux de serveurs différents offrant des services différents.

Différents groupes de serveurs peuvent également avoir les mêmes profils et déploiements. Cela permet, par exemple, le cumul des mises à niveau de l'application quand l'application est mise à jour sur un groupe de serveurs, puis mise à jour sur un deuxième groupe de serveurs, évitant ainsi une interruption complète du service.

Voici un exemple de définition de groupe de serveurs :

```
<server-group name="main-server-group" profile="default">
 <socket-binding-group ref="standard-sockets"/>
  <deployments>
   <deployment name="foo.war_v1" runtime-name="foo.war"/>
   <deployment name="bar.ear" runtime-name="bar.ear"/>
  </deployments>
</server-group>
```

Un groupe de serveurs inclut les attributs obligatoires suivants :

- nom : le nom du groupe de serveurs
- profil : le nom du profil du groupe de serveurs
- socket-binding-group : le nom du groupe de liaisons de sockets par défaut à utiliser pour les serveurs dans le groupe. Ce nom peut être remplacé sur la base d'un serveur dans **`host.xml`**. Cependant, c'est un élément obligatoire pour chaque groupe de serveurs et le domaine ne peut pas démarrer s'il n'est pas présent.

Un groupe de serveurs inclut les attributs optionnels suivants :

- deployments : le contenu de déploiement à déployer sur les serveurs du groupe.
- system-properties : les propriétés système à définir sur les serveurs du groupe
- jvm : les paramètres de configuration JMV par défaut de tous les serveurs du groupe. Le contrôleur hôte fait fusionner ces paramètres dans n'importe quelle configuration fournie par **`host.xml`** pour établir les paramètres utilisés dans la JVM du serveur.

Rapporter un bogue

1.10. PROFILS JBOSS EAP 6

Le concept des profils qui ont été utilisés dans les versions précédentes de JBoss EAP n'est plus utilisé. JBoss EAP 6 utilise maintenant un petit nombre de fichiers de configuration simples pour contenir toutes les informations de configuration.

Les modules et les pilotes sont chargés en fonction des besoins, donc le concept du profil par défaut utilisé dans les anciennes versions de JBoss EAP 6 où les profils étaient utilisés pour rendre le démarrage du serveur plus performant n'est pas très utile.

Au moment du déploiement, les dépendances du module sont définies, ordonnancées, et résolues par le serveur ou le contrôleur du domaine, et chargées dans le bon ordre. Les modules sont retirés du chargement quand ils ne sont plus utiles à aucun déploiement.

Il est possible de désactiver les modules ou de décharger les pilotes ou autres services manuellement en retirant les sous-systèmes de la configuration. Cependant, dans la plupart des cas, cela n'est pas utile. Si aucune de vos applications utilisent un module, il ne sera pas chargé.

Rapporter un bogue

CHAPITRE 2. GESTION DE SERVEURS D'APPLICATIONS

2.1. DÉMARRER ET STOPPER JBOSS EAP 6

2.1.1. Démarrer JBoss EAP 6

Démarrer JBoss EAP 6 d'une des manières suivantes :

- Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »
- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »

Rapporter un bogue

2.1.2. Démarrez JBoss EAP 6 comme un serveur autonome

Résumé

Cette rubrique couvre toutes les étapes à couvrir pour démarrer JBoss EAP 6 en tant que serveur autonome.

Procédure 2.1. Démarrer le service de plate-forme comme serveur autonome.

1. **Dans Red Hat Enterprise Linux.**
   Exécuter la commande suivante : ***EAP_HOME*`/bin/standalone.sh`**
2. **Dans Microsoft Windows Server**
   Exécuter la commande suivante : ***EAP_HOME*`\bin\standalone.bat`**
3. **Option : indiquer les paramètres supplémentaires.**
   Pour imprimer une liste de paramètres supplémentaires à passer aux scripts de démarrage, utiliser le paramètre ***-h***.

Résultat

L'instance de serveur autonome JBoss EAP 6 démarre.

Rapporter un bogue

2.1.3. Démarrez JBoss EAP 6 comme domaine géré

Ordre des opérations

Le contrôleur de domaines doit être démarré avant qu'un serveur esclave ne démarre dans des groupes de serveurs du domaine. Utiliser cette procédure sur le contrôleur de domaine pour commencer, puis, sur chaque contrôleur hôte associé et sur chaque hôte associé.

Procédure 2.2. Démarrer le service de plate-forme comme serveur géré

1. **Dans Red Hat Enterprise Linux.**
   Exécutez la commande : ***EAP_HOME*`/bin/domain.sh`**
2. **Dans Microsoft Windows Server**
   Exécutez la commande : ***EAP_HOME*`\bin\domain.bat`**

3. **En option : passez des paramètres supplémentaires au script de démarrage.**
   Pour obtenir une liste de paramètres que vous pourrez passer au script de démarrage, utilisez le paramètre ***-h***.

Résultat

L'instance de domaine géré de JBoss EAP 6 démarre.

Rapporter un bogue

2.1.4. Configuration d'un nom d'hôte dans un domaine géré

Résumé

Chaque hôte exécutant dans un domaine géré doit avoir un nom d'hôte unique. Pour faciliter l'administration et permettre l'utilisation de mêmes fichiers de configuration hôte sur plusieurs hôtes, le serveur utilise la priorité suivante pour déterminer le nom d'hôte.

1. Si défini, l'attribut de **nom** de l'élément **hôte** qui se trouve dans le fichier de configuration **host.xml**.
2. La valeur de la propriété système **jboss.host.name**.
3. La valeur qui suit le caractère (".") dans la propriété système **jboss.qualified.host.name**, ou toute la valeur s'il n'y a pas de point final (".").
4. La valeur qui suit le caractère (".") dans la variable d'environnement **HOSTNAME** pour les systèmes d'exploitation basés POSIX, la variable d'environnement **COMPUTERNAME** dans Microsoft Windows, ou toute la valeur s'il n'y a pas de point final (".")

Pour obtenir des informations sur la façon de définir les variables d'environnement, voir la documentation de votre système d'exploitation. Pour plus d'informations sur la façon de définir les propriétés système, voir Section 3.6.11, « Configurer les propriétés système par l'interface CLI ».

Cette section décrit comment fixer le nom de l'hôte dans le fichier de configuration, à l'aide d'une propriété système ou d'un nom codé en dur.

Procédure 2.3. Configuration d'un nom d'hôte avec une propriété système

1. Ouvrir le fichier de configuration de l'hôte **host.xml** pour le modifier.
2. Cherchez l'élément **host** dans le fichier, comme par exemple :

```
<host name="master" xmlns="urn:jboss:domain:1.6">
```

3. S'il est présent, retirez la déclaration d'attribut **name="*HOST_NAME*"**. L'élément **host** devra ressembler à l'exemple suivant :

```
<host xmlns="urn:jboss:domain:1.6">
```

4. Démarrer le serveur en saisissant **-Djboss.host.name** comme argument de ligne de commande, comme par exemple :

```
-Djboss.host.name=HOST_NAME
```

Procédure 2.4. Configuration d'un nom d'hôte avec un nom spécifique

1. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

   ```
   bin/domain.sh --host-config=HOST_FILE_NAME
   ```

   Par exemple :

   ```
   bin/domain.sh --host-config=host-slave01.xml
   ```

2. Lancer l'interface CLI.

3. Utiliser la syntaxe suivante pour remplacer le nom d'hôte :

   ```
   /host=EXISTING_HOST_NAME:write-
   attribute(name="name",value=UNIQUE_HOST_NAME)
   ```

   Par exemple :

   ```
   /host=master:write-attribute(name="name",value="host-slave01")
   ```

   Vous devriez voir apparaître le résultat suivant.

   ```
    "outcome" => "success"
   ```

   Cela modifie l'attribut **name** de l'hôte dans le fichier **host-slave01.xml** comme suit :

   ```
   <host name="host-slave01" xmlns="urn:jboss:domain:1.6">
   ```

4. Vous devez charger à nouveau la configuration du serveur avec l'ancien nom d'hôte pour terminer le processus.

   ```
   reload --host=EXISTING_HOST_NAME
   ```

   Par exemple :

   ```
   reload --host=master
   ```

Rapporter un bogue

2.1.5. Créer un domaine géré sur deux machines

NOTE

Vous devrez sans doute configurer votre pare-feu pour qu'il puisse exécuter cet exemple.

Vous pouvez créer un domaine géré sur deux machines, avec une machine en tant que contrôleur de domaine, et l'autre en tant qu'hôte. Pour plus d'informations, voir Section 1.6, « Contrôleur de domaine ».

- IP1 = adresse IP du contrôleur de domaine (Machine 1)
- IP2 = adresse IP de l'hôte (Machine 2)

Procédure 2.5. Créer un domaine géré sur deux macchines

1. **Sur la machine 1**

   a. Utiliser le script add-user.sh pour ajouter l'utilisateur de management. Par exemple, **`slave01`**, pour que l'hôte puisse authentifier le contrôleur de domaines. Notez la valeur **`SECRET_VALUE`** de la sortie **`add-user`**.

   b. Démarrer le domaine par le fichier de configuration **`host-master.xml`**, qui est préconfiguré pour un contrôleur de domaines exclusif.

   c. Utiliser **`-bmanagement=$IP1`** pour rendre le contrôleur de domaine visible auprès des autres machines.

```
[$JBOSS_HOME/bin]$ ./domain.sh --host-config=host-master.xml -
bmanagement=$IP1
```

2. **Sur la machine 2**

   a. Mettre à jour le fichier **`$JBOSS_HOME/domain/configuration/host-slave.xml`** avec les identifiants.

```
 <?xml version='1.0' encoding='UTF-8'?>
        <host xmlns="urn:jboss:domain:1.6" name="slave01">
        <!-- add user name here -->
         <management>
            <security-realms>
               <security-realm name="ManagementRealm">
                  <server-identities>
                    <secret value="$SECRET_VALUE" />
                    <!-- use secret value from add-user.sh
output-->
                  </server-identities>
                  ...
```

   b. Démarrer l'hôte.

```
[$JBOSS_HOME/bin]$ ./domain.sh --host-config=host-slave.xml  -
Djboss.domain.master.address=$IP1 -b=$IP2
```

3. **Nous pouvons maintenant gérer le domaine.**
   via le CLI :

```
[$JBOSS_HOME/bin]$ ./jboss-cli.sh -c --controller=$IP1
```

   via la console web :

```
http://$IP1:9990
```

   Accéder à la page d'index du serveur :

```
http://$IP2:8080/
http://$IP2:8230/
```

Rapporter un bogue

2.1.6. Démarrez JBoss EAP 6 avec une configuration différente

Si vous n'indiquez pas de fichier de configuration, le serveur démarrera avec le fichier par défaut. Cependant, quand vous démarrez le serveur, vous pouvez spécifier Configuration manuelle. Le processus varie légèrement, suivant que vous utilisez un Domaine géré ou un Serveur autonome, et suivant le système d'exploitation que vous utilisez.

Conditions préalables

- Avant d'utiliser un fichier de configuration alternatif, préparez-le à l'aide de la configuration par défaut comme modèle. Pour un domaine géré, le fichier de configuration doit être placé dans ***EAP_HOME*/domain/configuration/**. Pour les serveurs autonomes, le fichier de configuration devra être mis dans le répertoire ***EAP_HOME*/standalone/configuration/**.

NOTE

Plusieurs exemples de configurations sont inclus dans les répertoires de configuration ***EAP_HOME*/docs/examples/configs/**. Utiliser ces exemples pour activer des fonctionnalités supplémentaires, comme clustering ou l'API XTS de Transactions.

Procédure 2.6. Démarrage de l'instance par une configuration différente

1. **Serveur autonome**
   Pour un domaine autonome, fournir le nom du fichier de configuration comme option du paramètre ***--server-config***. Le fichier de configuration doit se trouver dans le répertoire ***EAP_HOME*/standalone/configuration/**, et vous devez indiquer le chemin d'accès du fichier de ce répertoire.

   Exemple 2.1. Utiliser un fichier de configuration alternatif pour un serveur autonome Red Hat Enterprise Linux.

   ```
   [user@host bin]$ ./standalone.sh --server-config=standalone-
   alternate.xml
   ```

   Cet exemple utilise le fichier de configuration ***EAP_HOME*/standalone/configuration/standalone-alternate.xml**.

   Exemple 2.2. Utiliser un fichier de configuration alternatif pour un serveur autonome Microsoft Windows.

   ```
   C:\EAP_HOME\bin> standalone.bat --server-config=standalone-
   alternate.xml
   ```

   Cet exemple utilise le fichier de configuration ***EAP_HOME*/standalone/configuration/standalone-alternate.xml**.

2. **Domaine géré**
   Pour un domaine géré, fournir le nom du fichier de configuration comme option du paramètre **--**

domain-config. Le fichier de configuration se trouve dans le répertoire ***EAP_HOME*/domain/configuration/**, et vous devez indiquer le chemin d'accès de ce répertoire.

Exemple 2.3. Utilisation d'un fichier de configuration alternatif pour un domaine géré dans Red Hat Enterprise Linux

```
[user@host bin]$ ./domain.sh --domain-config=domain-alternate.xml
```

Cet exemple utilise le fichier de configuration ***EAP_HOME*/domain/configuration/domain-alternate.xml**.

Exemple 2.4. Utilisation d'un fichier de configuration alternatif pour un domaine géré dans un serveur Microsoft Windows

```
C:\EAP_HOME\bin> domain.bat --domain-config=domain-alternate.xml
```

Cet exemple utilise le fichier de configuration ***EAP_HOME*\domain\configuration\domain-alternate.xml**.

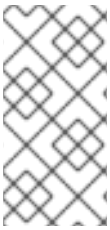
Résultat

La plateforme JBoss Enterprise Application Platform est maintenant en cours d'exécution, avec votre fichier de configuration alternatif.

Rapporter un bogue

2.1.7. Stopper le serveur JBoss EAP 6

La façon dont vous arrêtez la plate-forme JBoss EAP 6 dépend de la façon dont elle a été lancée. Cette tâche couvre l'arrêt d'une instance qui a démarré de manière interactive, faire cesser une instance qui a été démarrée par un service et faire cesser une instance qui a été mise en arrière-plan par un script.



NOTE

Pour obtenir des informations sur la façon de stopper un serveur ou un groupe de serveurs dans un domaine géré, voir Section 2.2.3, « Stopper un serveur qui utilise une console de gestion ». Pour obtenir des informations sur la façon de stopper un serveur par le CLI, voir Section 2.2.1, « Démarrer et arrêter les serveurs par l'interface CLI ».

- **Procédure 2.7. Stopper une instance de JBoss EAP 6**
  - **Stopper une instance qui a été démarrée de façon interactive à partir d'une invite de commande.**
    Appuyez sur **Ctrl-C** dans le terminal où JBoss EAP 6 exécute.

- **Procédure 2.8. Stopper une instance qui a démarré en tant que service de système d'exploitation.**

  Suivant votre système d'exploitation, utiliser une des procédures suivantes :

- **Red Hat Enterprise Linux**
  Dans Red Hat Enterprise Linux, si vous avez écrit un script de service, utiliser sa fonction **`stop`**. Cela devra être inscrit dans le script. Ensuite, vous pourrez utiliser **`service scriptname stop`**, avec *scriptname* comme nom de script.

- **Microsoft Windows Server**
  Dans Microsoft Windows, utiliser la commande **`net service`**, ou bien faites cesser le service à partir de l'applet **`Services`** qui se trouve dans le panneau de contrôle.

- **Procédure 2.9. Stopper une instance qui exécute en arrière-plan (Red Hat Enterprise Linux)**

  1. Chercher l'ID de processus (PID) du processus :

     - **Si une seule instance est en cours d'exécution (mode autonome)**
       N'importe laquelle des commandes suivantes renverront le PID d'une simple instance de JBoss EAP 6 :

       - **`pidof java`**
       - **`jps`**

         (La commande **`jps`** retournera un ID des deux processus ; un pour **`jboss-modules.jar`** et un pour **jps** lui-même. Utiliser l'ID de **`jboss-modules.jar`** pour stopper l'instance EAP)

     - **Si plusieurs instances EAP sont en cours d'exécution (mode de domaine)**
       Identifier le process qui convient pour y mettre un terme si plus d'une instance d'EAP en cours d'exécution nécessitent l'utilisation de commandes plus élaborées.

       - La commande **`jps`** peut être utilisée en mode détaillé (verbose) pour qu'elle puisse fournir davantage d'informations sur les processus java qu'elle trouve.

         Vous trouverez ci-dessous sous une sortie abrégée d'une commande détaillée **`jps`** qui identifie les différents processus d'EAP en cours par PID et rôle :

```
$ jps -v
12155 jboss-modules.jar -D[Server:server-one] -
XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
...

12196 jboss-modules.jar -D[Server:server-two] -
XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
...

12096 jboss-modules.jar -D[Host Controller] -Xms64m -
Xmx512m -XX:MaxPermSize=256m
...

11872 Main -Xms128m -Xmx750m -XX:MaxPermSize=350m -
XX:ReservedCodeCacheSize=96m -XX:+UseCodeCacheFlushing
...

11248 jboss-modules.jar -D[Standalone] -
XX:+UseCompressedOops -verbose:gc
```

```
...

12892 Jps
...

12080 jboss-modules.jar -D[Process Controller] -Xms64m -
Xmx512m -XX:MaxPermSize=256m
...
```

- La commande **ps aux** peut également être utilisée pour renvoyer des informations sur les instances multiples EAP.

  Vous trouverez ci-dessous sous une sortie abrégée d'une commande détaillée **ps aux** qui identifie les différents processus d'EAP en cours par PID et rôle :

```
$ ps aux | grep java
username 12080  0.1  0.9 3606588 36772 pts/0   Sl+  10:09
0:01 /path/to/java -D[Process Controller] -server -Xms128m
-Xmx128m -XX:MaxPermSize=256m
...

username 12096  1.0  4.1 3741304 158452 pts/0  Sl+  10:09
0:13 /path/to/java -D[Host Controller] -Xms128m -Xmx128m -
XX:MaxPermSize=256m
...

username 12155  1.7  8.9 4741800 344224 pts/0  Sl+  10:09
0:22 /path/to/java -D[Server:server-one] -XX:PermSize=256m
-XX:MaxPermSize=256m -Xms1000m -Xmx1000m -server -
...

username 12196  1.8  9.4 4739612 364436 pts/0  Sl+  10:09
0:22 /path/to/java -D[Server:server-two] -XX:PermSize=256m
-XX:MaxPermSize=256m -Xms1000m -Xmx1000m -server
...
```

Dans les exemples ci-dessus, les processus **Process Controller** sont des processus à stopper pour stopper tout le domaine.

L'utilitaire **grep** peut être utilisé avec une de ces commandes pour identifier le **Process Controller** :

```
jps -v | grep "Process Controller"
```

```
ps aux | grep "Process Controller"
```

2. Envoyer le signal **TERM** au processus en exécutant **kill *PID***, quand *PID* est l'ID de processus identifié par une des commandes ci-dessus.

Résultat

Chacune de ces solutions ferme la plate-forme JBoss EAP 6 nettement, ce qui fait qu'aucune donnée n'est perdue.

Rapporter un bogue

2.1.8. Référence aux variables et arguments à passer à l'exécution du serveur

Le script de démarrage du serveur d'applications accepte l'ajout d'arguments et de variables en cours d'exécution. L'utilisation de ces paramètres permettent au serveur d'être démarré sous d'autres configurations que celles qui sont définies dans les fichiers de configuration **`standalone.xml`**, **`domain.xml`** et **`host.xml`**. Cela peut comprendre le démarrage du serveur par un ensemble de liaisons de sockets différent ou une configuration secondaire. Vous pourrez accéder à une liste des paramètres disponibles en passant la variable d'assistance au démarrage.

Exemple 2.5.

L'exemple suivant ressemble au démarrage de serveur expliqué dans Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome » et Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré », avec en plus les variables ***-h*** ou ***--help***>. Les résultats de cette variable d'assistance sont expliqués dans le tableau ci-dessous.

Mode autonome :

```
[localhost bin]$ standalone.sh -h
```

Mode de domaine :

```
[localhost bin]$ domain.sh -h
```

Tableau 2.1. Tableau des arguments et variables du temps d'exécution

Argument ou Variable	Mode	Description
`--admin-only`	Autonome	Définir le type d'exécution du serveur à **`ADMIN_ONLY`**. Cela le fera ouvrir les interfaces administratives et il pourra ainsi accepter les ordres de gestion, mais il ne pourra pas démarrer d'autres services de runtime ou accepter les demandes de l'utilisateur final.
`--admin-only`	Domaine	Définir le type d'exécution du contrôleur hôte à **`ADMIN_ONLY`**, ce qui le fera ouvrir les interfaces administratives et il pourra ainsi accepter les ordres de gestion, mais il ne pourra pas démarrer d'autres serveurs ou, si ce contrôleur hôte est le master du domaine, il pourra accepter les demandes des contrôleurs hôte esclaves.
`-b <value>`, **`-b=<value>`**	Autonome, Serveur	Définir la propriété système **`jboss.bind.address`** à la valeur donnée.
`-b<interface>=<value>`	Autonome, Serveur	Définir la propriété système **`jboss.bind.address.<interface>`** à la valeur donnée.

Argument ou Variable	Mode	Description
--backup	Domaine	Conserver une copie de la configuration de domaine persistante même si cet hôte n'est pas le contrôleur de domaines.
-c <config>, **-c=<config>**	Autonome	Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est **standalone.xml**.
-c <config>, **-c=<config>**	Domaine	Nom du fichier de configuration de serveur à utiliser. La valeur par défaut est **domain.xml**.
--cached-dc	Domaine	Si l'hôte n'est pas le contrôleur de domaine et ne peut pas contacter le contrôleur de domaine au démarrage, puisque le processus de démarrage (booting) utilise une copie de la configuration de domaine mise en cache localement.
--debug [<port>]	Autonome	Active le mode de débogage par un argument en option qui indique le port. Ne fonctionne que si le script de lancement le supporte.
-D<name>[=<value>]	Autonome, Serveur	Définit une propriété système.
--domain-config=<config>	Domaine	Nom du fichier de configuration de serveur à utiliser. La valeur par défaut est **domain.xml**.
-h, **--help**	Autonome, Serveur	Affiche le message d'assistance et sortir.
--host-config=<config>	Domaine	Nom du fichier de configuration hôte à utiliser. La valeur par défaut est **host.xml**.
--interprocess-hc-address=<address>	Domaine	Adresse à laquelle le contrôleur hôte doit écouter la communication en provenance du contrôleur de processus.
--interprocess-hc-port=<port>	Domaine	Port sur lequel le contrôleur hôte doit écouter la communication en provenance du contrôleur de processus.
--master-address=<address>	Domaine	Définit la propriété système **jboss.domain.master.address** à la valeur donnée. Dans une configuration de contrôleur hôte esclave par défaut, c'est utilisé pour configurer l'adresse du contrôleur hôte maître.

Argument ou Variable	Mode	Description
--master-port=<port>	Domaine	Définit la propriété système **jboss.domain.master.port** à la valeur donnée. Dans une configuration de contrôleur hôte esclave par défaut, c'est utilisé pour configurer le port utilisé pour la communication de gestion native du contrôleur hôte maître.
--read-only-server-config=<config>	Autonome	Nom du fichier de configuration du serveur à utiliser. Cela diffère de ***--server-config*** et ***-c*** en ce que le fichier d'origine n'est jamais écrasé.
--read-only-domain-config=<config>	Domaine	Nom du fichier de configuration du domaine à utiliser. Cela diffère de ***--domain-config*** et de ***-c*** en ce que le fichier de départ n'est jamais écrasé.
--read-only-host-config=<config>	Domaine	Nom du fichier de configuration de l'hôte à utiliser. Cela diffère de ***--host-config*** en ce que le fichier de départ n'est jamais écrasé.
-P <url>, **-P=<url>**, **--properties=<url>**	Autonome, Serveur	Télécharge les propriétés système de l'URL donné.
--pc-address=<address>	Domaine	Adresse à laquelle le contrôleur de processus doit écouter les communications en provenance des processus qu'il contrôle.
--pc-port=<port>	Domaine	Port sur lequel le contrôleur de processus doit écouter les communications en provenance des processus qu'il contrôle.
-S<name>[=<value>]	Autonome	Définit une propriété de sécurité.
--server-config=<config>	Autonome	Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est **standalone.xml**.
-u <value>, **-u=<value>**	Autonome, Serveur	Définit la propriété système **jboss.default.multicast.address** à la valeur donnée.
-v, **-V**, **--version**	Autonome, Serveur	Affiche la version du serveur d'application et sortie.

Rapporter un bogue

2.2. DÉMARRER ET ARRÊTER LES SERVEURS

2.2.1. Démarrer et arrêter les serveurs par l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Vous pouvez démarrer et arrêter les serveurs avec l'interface CLI (en mode autonome) ou par la console de gestion. En mode de domaine, vous ne pouvez que démarrer les instances de serveur. Les deux outils de gestion vous permettent de contrôler une seule instance de serveur autonome, ou de gérer sélectivement un ou plusieurs serveurs dans un déploiement de domaine géré. Si vous utilisez la console de gestion en mode de domaine, veuillez consulter Section 2.2.2, « Démarrer un serveur par la console de gestion » pour obtenir des instructions. Si vous utilisez l'interface CLI, le processus varie entre des instances de serveur autonome et de domaine géré.

Démarrer et arrêter un serveur autonome par l'interface CLI

Vous pouvez démarrer une instance de serveur autonome par les scripts de ligne de commande, et le fermer par l'interface CLI par la commande **shutdown**. Si vous avez besoin de l'instance par la suite, exécuter le processus de démarrage à nouveau selon les instructions dans Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome ».

Exemple 2.6. Démarrer et arrêter une instance de serveur autonome par l'interface CLI

```
[standalone@localhost:9999 /] shutdown
```

Démarrer et arrêter un domaine géré par l'interface CLI

Si vous exécutez un domaine géré, la console de gestion va vous permettre de démarrer ou de stopper sélectivement des serveurs spécifiques pour ce domaine. Cela inclut les groupes de serveurs dans tout le domaine, ainsi que les instances de serveur spécifiques sur un hôte.

Exemple 2.7. Stopper un hôte de serveur dans un domaine géré par l'interface CLI

Semblable à une instance de serveur autonome, la commande **shutdown** est utilisée pour fermer un hôte de domaine géré déclaré. Cet exemple stoppe un hôte de serveur nommé master en déclarant le nom d'instance avant d'appeler l'opération de fermeture. Utiliser la touche **tab** pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs hôtes disponibles.

```
[domain@localhost:9999 /] /host=master:shutdown
```

Exemple 2.8. Stopper un hôte de serveur dans un domaine géré par l'interface CLI

Cet exemple montre comment démarrer un groupe de serveurs par défaut nommé ***main-server-group*** en déclarant le groupe avant les opérations **start** et **stop**. Utilisez la touche **tab** pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs de nom de groupes de serveurs disponibles.

```
[domain@localhost:9999 /] /server-group=main-server-group:start-servers
```

```
[domain@localhost:9999 /] /server-group=main-server-group:stop-servers
```

Exemple 2.9. Démarrer et stopper une instance de serveur dans un domaine géré par l'interface CLI

Cet exemple montre comment démarrer et stopper une instance de serveur nommée ***server-one*** sur l'hôte ***master*** en déclarant la configuration de l'hôte et du serveur avant d'invoquer les opérations **start** et **stop**. Utilisez la touche **tab** pour aider à la complétion de chaînes et pour exposer des variables visibles comme les valeurs de configuration de l'hôte et du serveur.

```
[domain@localhost:9999 /] /host=master/server-config=server-one:start
```

```
[domain@localhost:9999 /] /host=master/server-config=server-one:stop
```

Rapporter un bogue

2.2.2. Démarrer un serveur par la console de gestion

Pré-requis

- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »
- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 2.10. Démarrer le serveur d'un domaine géré

1. Sélectionner l'onglet **Runtime** qui se trouve en haut de la console. Étendre le menu **Server** et sélectionner **Overview**.
2. À partir de la liste **Server Instances**, sélectionner le serveur que vous souhaitez démarrer. Les serveurs qui sont en cours d'exécution sont indiqués.

   Placez le curseur sur une instance de cette liste pour afficher les options en bleu dans le texte sous les informations sur le serveur.
3. Pour démarrer une instance, cliquer sur **Start Server** quand vous verrez ce texte. Une boîte de dialogue de confirmation va s'ouvrir. Cliquer sur le bouton **Confirm** pour démarrer le serveur.

Résultat

Le serveur sélectionné démarre et exécute.

Rapporter un bogue

2.2.3. Stopper un serveur qui utilise une console de gestion

Pré-requis

- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »
- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 2.11. Stopper un serveur qui utilise une console de gestion dans un domaine géré

1. Sélectionner l'onglet **`Runtime`** qui se trouve en haut de la console. Étendre le menu **Domain** et sélectionner **Overview**.
2. Une liste d'instances **`Server Instances`** s'affiche dans le tableau **`Hosts, groups and server instances`**. Les serveurs en cours sont cochés.
3. Placez le curseur sur le serveur choisi. Cliquez sur le bouton **`Stop Server`** qui apparaît. Une fenêtre de dialogue de confirmation s'affichera.
4. Cliquer sur le bouton **`Confirm`** pour arrêter le serveur.

Résultat

Le serveur sélectionné est stoppé.

Rapporter un bogue

2.3. CHEMINS D'ACCÈS AUX SYSTÈMES DE FICHIERS

2.3.1. Chemins d'accès aux systèmes de fichiers

JBoss EAP 6 utilise des noms logiques pour les chemins de systèmes de fichiers. Les fichiers de configuration **`domain.xml`**, **`host.xml`** et **`standalone.xml`** incluent tous une section où les chemins d'accès peuvent être déclarés. D'autres sections de la configuration peuvent ensuite référencer ces chemins par leur nom logique, évitant la déclaration du chemin d'accès absolu pour chaque instance. Cela bénéficie aux efforts de configuration et d'administration car cela permet à des configurations hôtes spécifiques de résoudre des noms logiques universels.

Par exemple, la configuration du sous-système de logging comprend une référence au chemin ***`jboss.server.log.dir`*** qui pointe vers le répertoire **`log`** du serveur.

Exemple 2.10. Exemple de chemin d'accès relatif du répertoire de logging

```
<file relative-to="jboss.server.log.dir" path="server.log"/>
```

JBoss EAP 6 fournit un nombre de chemins d'accès standards automatiquement sans que l'utilisateur n'ait besoin de les configurer dans un fichier de configuration.

Tableau 2.2. Chemins d'accès standard

Valeur	Description
`jboss.home.dir`	Le répertoire root de la distribution JBoss EAP 6.
`user.home`	Le répertoire d'accueil de l'utilisateur.
`user.dir`	Le répertoire de travail actuel de l'utilisateur
`java.home`	Le répertoire d'installation de Java

Valeur	Description
jboss.server.base.dir	Le répertoire root d'une instance de serveur individuel.
jboss.server.data.dir	Le répertoire que le serveur va utiliser pour le stockage de fichiers de données persistantes.
jboss.server.config.dir	Le répertoire qui contient la configuration du serveur.
jboss.server.log.dir	Le répertoire que le serveur va utiliser pour le stockage de fichier de journalisation.
jboss.server.temp.dir	Le répertoire que le serveur va utiliser pour le stockage de fichiers temporaires.
jboss.controller.temp.dir	Le répertoire que le contrôleur hôte va utiliser pour le stockage de fichiers temporaires.

Substituer un chemin par un autre

Si vous exécutez en serveur autonome, vous pourrez substituer les chemins **jboss.server.base.dir**, **jboss.server.log.dir**, ou **jboss.server.config.dir** d'une ou de deux manières.

1. Vous pouvez passer des arguments par ligne de commande quand vous démarrez le serveur. Ainsi :

```
bin/standalone.sh -Djboss.server.log.dir=/var/log
```

2. Vous pouvez modifier la variable **JAVA_OPTS** dans le fichier de configuration du serveur. Ouvrir le fichier ***EAP_HOME*/bin/standalone.conf** et ajouter la ligne suivante à la fin du fichier :

```
JAVA_OPTS="$JAVA_OPTS Djboss.server.log.dir=/var/log"
```

La substitution de fichiers n'est pas prise en charge pour les serveurs exécutant dans un domaine géré.

Ajouter un chemin personnalisé

Vous pouvez également créer votre propre chemin personnalisé. Par exemple, vous pouvez définir un chemin relatif à utiliser pour la journalisation :

```
my.relative.path=/var/log
```

Vous pouvez alors modifier le log handler pour qu'il utilise **my.relative.path**,

Rapporter un bogue

2.4. FICHIERS DE CONFIGURATION

2.4.1. Fichiers de configuration de JBoss EAP 6

La configuration de JBoss EAP 6 a considérablement changé depuis les dernières versions. Une des différences principale est l'utilisation d'une structure de fichier de configuration simplifiée, qui comprend un ou plusieurs des fichiers répertoriés ci-dessous :

Tableau 2.3. Emplacements des fichiers de configuration

Mode du serveur	Emplacement	But
domain.xml	***EAP_HOME*/domain/configuration/domain.xml**	Il s'agit du fichier de configuration principal d'un domaine géré. Seul le master du domaine lit ce fichier. Il peut être supprimé pour les autres membres du domaine.
host.xml	***EAP_HOME*/domain/configuration/host.xml**	Ce fichier contient les détails de configuration spécifiques à un hôte physique dans un domaine géré, tels que les interfaces réseau, les liaisons de sockets, le nom de l'hôte et d'autres détails spécifiques à l'hôte. Le fichier **host.xml** contient toutes les fonctionnalités de **hôte-master.xml** et **hôte-slave.xml**, qui sont décrits ci-dessous. Ce fichier n'est pas présent pour les serveurs autonomes.
host-master.xml	***EAP_HOME*/domain/configuration/host-master.xml**	Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur maître de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes.
host-slave.xml	***EAP_HOME*/domain/configuration/host-slave.xml**	Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur esclave de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes.

Mode du serveur	Emplacement	But
standalone.xml	***EAP_HOME*/standalone/configuration/standalone.xml**	C'est le fichier de configuration par défaut pour un serveur autonome. Il contient toutes les informations sur le serveur autonome, y compris les sous-systèmes, réseautage, déploiements, les liaisons de sockets et autres détails configurables. Cette configuration est utilisée automatiquement lorsque vous démarrez votre serveur autonome.
standalone-full.xml	***EAP_HOME*/standalone/configuration/standalone-full.xml**	Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible à l'exception de ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-full.xml**
standalone-ha.xml	***EAP_HOME*/standalone/configuration/standalone-ha.xml**	Cet exemple de fichier de configuration active tous les sous-systèmes par défaut et ajoute les **mod_cluster** et les sous-systèmes JGroups pour un serveur autonome, afin qu'il puisse participer à un cluster de haute disponibilité ou d'équilibrage de la charge. Ce fichier n'est pas applicable à un domaine géré. Pour utiliser cette configuration, arrêtez votre serveur et redémarrez le à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-ha.xml**

Mode du serveur	Emplacement	But
standalone-full-ha.xml	***EAP_HOME*/standalone/configuration/standalone-full-ha.xml**	Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible incluant ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-full-ha.xml**

Ce sont les emplacements par défaut uniquement. Vous pouvez indiquer un fichier de configuration différent en cours d'exécution.

Rapporter un bogue

2.4.2. Remplacement de propriété basée descripteur

La configuration d'application - par exemple, les paramètres de connexion de la source de données - varient normalement entre le développement, les tests, et les déploiements de production. Cette variation est parfois accommodée par les build system scripts, car la spécification Java EE ne contient pas de méthode pour externaliser ces configurations.

Dans JBoss Enterprise Application Platform 6, vous pouvez utiliser *le remplacement de propriété basée descripteur* pour gérer la configuration en externe.

Le *remplacement de propriété basée descripteur* remplace les propriétés basées sur des descripteurs, ce qui vous permet de supprimer les hypothèses concernant l'environnement de l'application et la chaîne de construction. Les configurations spécifiques à l'environnement peuvent être spécifiées dans les descripteurs de déploiement à la place des annotations ou des build system scripts. Vous pouvez fournir la configuration dans les fichiers ou en tant que paramètres en ligne de commande.

Le remplacement de propriétés basées descripteur est activé globalement par **`standalone.xml`** ou **`domain.xml`** :

```
<subsystem xmlns="urn:jboss:domain:ee:1.1">
  <spec-descriptor-property-replacement>
    true
  </spec-descriptor-property-replacement>
  <jboss-descriptor-property-replacement>
    true
  </jboss-descriptor-property-replacement>
</subsystem>
```

Les descripteurs Java EE de **`ejb-jar.xml`** et de **`persistence.xml`** peuvent être remplacés. Désactivé par défaut.

Le remplacement de descripteur spécifique à Java est activé par défaut. Les descripteurs peuvent être remplacés dans :

- **`jboss-ejb3.xml`**

- **jboss-app.xml**
- **jboss-web.xml**
- ***-jms.xml**
- ***-ds.xml**

Par exemple, avec un bean ayant l'annotation suivante :

```
 @ActivationConfigProperty(propertyName = "connectionParameters",
propertyValue = "host=192.168.1.1;port=5445")
```

Avec le remplacement de propriété basée descripteur, **connectionParameters** peut être spécifié en ligne de commande par :

```
./standalone.sh -DconnectionParameters='host=10.10.64.1;port=5445'
```

Pour accomplir la même chose par les propriétés système :

```
<activation-config>
  <activation-config-property>
    <activation-config-property-name>
      connectionParameters
      </activation-config-property-name>
    <activation-config-property-value>
      ${jms.connection.parameters:'host=10.10.64.1;port=5445'}
    </activation-config-property-value>
  </activation-config-property>
</activation-config>
```

${jms.connection.parameters:'host=10.10.64.1;port=5445'} permet aux paramètres de connexion d'être remplacés par un paramètre fourni en ligne de commande, tout en donnant une valeur par défaut.

Rapporter un bogue

2.4.3. Activer/Désactiver un remplacement de propriété basé descripteur

Résumé

Le contrôle précis du remplacement de propriété de descripteur a été introduit dans **jboss-as-ee_1_1.xsd**. Cette tâche couvre les étapes requises pour configurer le remplacement de propriété basé descripteur.

Conditions préalables9

- Section 2.1.1, « Démarrer JBoss EAP 6 »
- Section 3.5.2, « Lancement de l'interface CLI »

Les indicateurs de remplacement de propriété basé descripteur ont les valeurs booléennes suivantes :

- Si défini à **true**, les remplacements de propriété sont activés.

- Si défini à **`false`**, les remplacements de propriété sont désactivés.

Procédure 2.12. `jboss-descriptor-property-replacement`

`jboss-descriptor-property-replacement` est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- **`jboss-ejb3.xml`**
- **`jboss-app.xml`**
- **`jboss-web.xml`**
- **`*-jms.xml`**
- **`*-ds.xml`**

La valeur par défaut de **`jboss-descriptor-property-replacement`** est **`true`**.

1. À l'aide de l'interface CLI, exécuter la commande suivante pour déterminer la valeur de **`jboss-descriptor-property-replacement`**:

```
/subsystem=ee:read-attribute(name="jboss-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="jboss-descriptor-property-replacement",value=VALUE)
```

Procédure 2.13. `spec-descriptor-property-replacement`

`spec-descriptor-property-replacement` est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- **`ejb-jar.xml`**
- **`persistence.xml`**

La valeur par défaut de **`spec-descriptor-property-replacement`** est **`false`**.

1. À l'aide de l'interface CLI, exécuter la commande suivante pour confirmer la valeur de **`spec-descriptor-property-replacement`**:

```
/subsystem=ee:read-attribute(name="spec-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="spec-descriptor-property-replacement",value=VALUE)
```

Résultat

Les indicateurs de remplacement de propriété basé descripteur ont bien été configurés.

Rapporter un bogue

2.4.4. Historique du fichier de configuration

Les fichiers de configuration du serveur d'applications incluent **`standalone.xml`**, ainsi que les fichiers **`domain.xml`** et **`host.xml`**. Alors que ces fichiers sont modifiables directement, la méthode recommandée consiste à configurer le modèle de serveur d'applications par les opérations de gestion disponibles, y compris l'interface CLI et la console de gestion.

Pour aider à la maintenance et à la gestion de l'instance de serveur, le serveur d'applications crée une version horodatée du fichier de configuration original au moment du démarrage. Toutes les modifications de configuration supplémentaires suite aux opérations de gestion résultent à la sauvegarde automatique du fichier d'origine, et une copie de travail de l'instance est alors conservée en tant que référence et pour la restauration. Cette fonctionnalité d'archivage s'étend à l'enregistrement, au chargement et à la suppression des snapshots de configuration du serveur pour autoriser les scénarios de rappel et de restauration.

- Section 2.4.5, « Démarrer le serveur par une ancienne configuration »
- Section 2.4.6, « Sauvegarder un snapshot de configuration par l'interface CLI »
- Section 2.4.7, « Charger un snapshot de configuration par l'interface CLI. »
- Section 2.4.8, « Supprimer un snapshot de configuration par l'interface CLI »
- Section 2.4.9, « Lister tous les snapshots de configuration par l'interface CLI »

Rapporter un bogue

2.4.5. Démarrer le serveur par une ancienne configuration

L'exemple suivant vous montre comment démarrer le serveur d'applications par une ancienne configuration dans un serveur autonome par **`standalone.xml`**. Le même concept s'applique à un domaine géré par **`domain.xml`** et **`host.xml`** respectivement.

Cet exemple nous remémore une ancienne configuration sauvegardée automatiquement par le serveur d'applications tandis que les opérations de gestion modifient le modèle de serveur.

1. Identifier la version de sauvegarde que vous souhaitez démarrer. Cet exemple va rappeler l'instance de modèle de serveur qui précédait la première modification qui a lieu suite à un démarrage réussi.

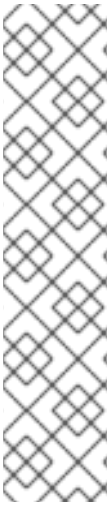
```
EAP_HOME/standalone/configuration/standalone_xml_history/current/sta
ndalone.v1.xml
```

2. Démarrer le serveur par cette configuration du modèle de sauvegarde en passant le nom de fichier relatif sous **`jboss.server.config.dir`**.

```
EAP_HOME/bin/standalone.sh --server-
config=standalone_xml_history/current/standalone.v1.xml
```

Résultat

Le serveur d'applications démarre par la configuration sélectionnée.



NOTE

L'historique de la configuration du domaine se trouve dans ***EAP_HOME*/domain/configuration/domain_xml_history/current/domain.v1.xml**

Démarrer le serveur par cette configuration du modèle de sauvegarde en passant le nom de fichier relatif sous ***jboss.domain.config.dir***.

Pour démarrer le domaine par cette configuration :

```
EAP_HOME/bin/domain.sh --domain-
config=domain_xml_history/current/domain.v1.xml
```

Rapporter un bogue

2.4.6. Sauvegarder un snapshot de configuration par l'interface CLI

Résumé

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier de configuration **standalone.xml**, mais le même processus s'applique aux fichiers de configuration **domain.xml** et **host.xml**.

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 2.14. Télécharger un snapshot de configuration et sauvegardez-le

- **Sauvegarde d'un snapshot**
  Exécuter l'opération **take-snapshot** pour acquérir une copie de la configuration du serveur.

```
[standalone@localhost:9999 /] :take-snapshot
{
    "outcome" => "success",
    "result" =>
"/home/User/EAP_HOME/standalone/configuration/standalone_xml_history
/snapshot/20110630-172258657standalone.xml"
```

Résultat

Un snapshot de la configuration du serveur en cours a été sauvegardé.

Rapporter un bogue

2.4.7. Charger un snapshot de configuration par l'interface CLI.

Les snapshots de configuration représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur. Le processus

de chargement des snapshots ressemble à celui de la méthode pour Section 2.4.5, « Démarrer le serveur par une ancienne configuration », à partir de la ligne de commande et non pas l'interface CLI utilisée pour créer, lister et supprimer les snapshots.

L'exemple suivant utilise le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

Procédure 2.15. Télécharger un snapshot de configuration

1. Identifier le snapshot à télécharger. Cet exemple va rappeler le fichier suivant du répertoire de snapshots. Le chemin par défaut des fichiers de snapshot est le suivant.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20110812-191301472standalone.xml
```

   Les snapshots sont exprimés par leurs chemins relatifs, selon lesquels l'exemple ci-dessus peut être écrit ainsi.

```
jboss.server.config.dir/standalone_xml_history/snapshot/20110812-191301472standalone.xml
```

2. Démarrer le serveur par le snapshot de configuration sélectionné en passant le nom du fichier.

```
EAP_HOME/bin/standalone.sh --server-config=standalone_xml_history/snapshot/20110913-164449522standalone.xml
```

Résultat

Le serveur démarre à nouveau avec la configuration sélectionnée dans le snapshot téléchargé.

Rapporter un bogue

2.4.8. Supprimer un snapshot de configuration par l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

Les exemples suivants utilisent le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

Procédure 2.16. Supprimer un snapshot particulier

1. Identifier le snapshot à effacer. Cet exemple va effacer le fichier suivant du répertoire de snapshots.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20110630-165714239standalone.xml
```

2. Exécuter la commande **`:delete-snapshot`** pour supprimer un snapshot particulier, en spécifiant le nom du snapshot comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="20110630-
165714239standalone.xml")
{"outcome" => "success"}
```

Résultat

Le snapshot a été supprimé.

Procédure 2.17. Supprimer tous les snapshots

- Exécuter la commande **`:delete-snapshot(name="all")`** pour supprimer tous les snapshots comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="all")
{"outcome" => "success"}
```

Résultat

Tous les snapshots ont été supprimés.

Rapporter un bogue

2.4.9. Lister tous les snapshots de configuration par l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

Procédure 2.18. Lister tous les snapshots de configuration

- **Lister tous les snapshots**
  Lister tous les snapshots sauvegardés en exécutant la commande **`:list-snapshots`**.

```
[standalone@localhost:9999 /] :list-snapshots
{
    "outcome" => "success",
    "result" => {
        "directory" =>
"/home/hostname/EAP_HOME/standalone/configuration/standalone_xml_his
tory/snapshot",
        "names" => [
            "20110818-133719699standalone.xml",
            "20110809-141225039standalone.xml",
            "20110802-152010683standalone.xml",
            "20110808-161118457standalone.xml",
```

```
            "20110912-151949212standalone.xml",
            "20110804-162951670standalone.xml"
        ]
    }
}
```

Résultat

Les snapshots sont listés.

Rapporter un bogue

CHAPITRE 3. INTERFACES DE GESTION

3.1. GESTION DU SERVEUR D'APPLICATIONS

JBoss EAP 6 vous propose des outils de gestion multiples pour configurer et administrer votre implémentation suivant les besoins. Ces outils comprennent la nouvelle console de gestion et l'interface CLI, comme exemples d'API de gestion pour permettre aux utilisateurs experts de développer leurs propres outils s'ils le désirent.

Rapporter un bogue

3.2. LES API (DE L'ANGLAIS APPLICATION PROGRAMMING INTERFACES) DE GESTION

Clients de gestion

JBoss EAP 6 offre trois approches différentes pour configurer et gérer des serveurs, étant à la fois une interface web, un client de ligne de commande et un ensemble de fichiers de configuration XML. Tandis que les méthodes recommandées pour la modification du fichier de configuration incluent la console de gestion et l'interface CLI, les modifications de configuration de la part des trois sont toujours synchronisées à travers les différentes vues et sont conservées dans les fichiers XML. Notez que les modifications apportées aux fichiers de configuration XML pendant l'exécution d'une instance de serveur seront remplacées par le modèle de serveur.

HTTP API

La console de gestion est un exemple d'interface web construite avec Google Web Toolkit (GWT). La console de gestion communique avec le serveur à l'aide de l'interface de gestion HTTP. Le point de terminaison HTTP API est le point d'entrée des clients de gestion basés sur le protocole HTTP pour s'intégrer à la couche de gestion. Il utilise un protocole JSON encodé et un API de style RPC de-typed, pour décrire et exécuter des opérations de gestion en fonction d'un domaine géré ou d'un serveur autonome. L'API HTTP est utilisé par la console web, mais offre aussi des possibilités d'intégration à un large éventail d'autres clients.

Le point de terminaison HTTP API est situé soit avec le contrôleur de domaine ou une instance de serveur autonome. Le point de terminaison HTTP API sert deux contextes différents ; un pour l'exécution des opérations de gestion et un autre pour accéder à l'interface web. Par défaut, il s'exécute sur le port 9990.

Exemple 3.1. Exemple de fichier de configuration HTTP API

```
<management-interfaces>
  [...]
  <http-interface security-realm="ManagementRealm">
     <socket-binding http="management-http"/>
  </http-interface>
</management-interfaces>
```

La console web est servie par le même port que l'API de gestion HTTP. Il est important de distinguer entre la console de gestion accessible comme localhost par défaut, la console de gestion accessible à distance par un hôte spécifique et une combinaison de port, et l'API du domaine exposé.

Tableau 3.1. Les URL d'accès à la console de gestion

URL	Description
`http://localhost:9990/console`	La console de gestion à laquelle accéde l'hôte local, et qui contrôle la configuration du domaine géré.
`http://hostname:9990/console`	La console de gestion accédée à distance, qui nomme l'hôte et qui contrôle la configuration du domaine géré.
`http://hostname:9990/management`	L'API de gestion HTTP exécute sur le même port que la console de gestion, affiche les mêmes valeurs et attributs bruts exposés à l'API.

API Natif

Le CLI de gestion est un exemple d'outil d'API Natif. Cet outil de gestion est disponible à une instance de serveur autonome ou à un domaine, permettant ainsi à un utilisateur de se connecter à une instance du serveur autonome ou au contrôleur du domaine, et d'exécuter des opérations de gestion rendues disponibles par le modèle de gestion de-types.

Le point de terminaison de l'API natif est le point d'entrée pour les clients de gestion qui s'appuient sur le protocole natif pour intégrer la couche de gestion. Il utilise un protocole binaire ouvert et une API style-RPC basée sur un très petit nombre de types Java pour décrire et exécuter des opérations de gestion. Il est utilisé par l'outil de gestion Interface CLI, mais offre des capacités d'intégration pour un large éventail d'autres clients également.

Le point de terminaison d'API natif est co-localisé avec un contrôleur hôte ou un serveur autonome. Il doit être activé pour utiliser l'interface CLI. Par défaut, il s'exécute sur le port 9999.

Exemple 3.2. Exemple de fichier de configuration d'API natif

```
<management-interfaces>
  <native-interface security-realm="ManagementRealm">
    <socket-binding native="management-native"/>
  </native-interface>
  [...]
</management-interfaces>
```

Rapporter un bogue

3.3. CONSOLE DE GESTION ET INTERFAC CLI

Dans JBoss EAP 6, toutes les instances de serveurs et toutes les configurations sont gérées par les interfaces de gestion, et non pas par modification de fichiers XML. Malgré que les fichiers de configuration XML puissent toujours être édités, la gestion par les interfaces de gestion fournit une validation supplémentaire et des fonctionnalités avancées pour la gestion persistante des instances de serveurs. Les modifications apportées aux fichiers de configuration XML, tandis que l'instance de serveur est en cours d'exécution, seront remplacées par le modèle de serveur, et des commentaires XML ajoutés disparaîtront ainsi. Seules les interfaces de gestion doivent être utilisées pour modifier les fichiers de configuration pendant l'exécution d'une instance de serveur

Pour gérer les serveurs par une interface utilisateur graphique d'un navigateur web, utiliser la console de gestion.

Pour gérer les serveurs par l'interface de ligne de commande, utiliser l'interface CLI.

Rapporter un bogue

3.4. LA CONSOLE DE GESTION

3.4.1. Console de management

La console de management est un outil administratif basé web pour la plateforme JBoss EAP 6.

Utilisez la console de management pour démarrer et arrêter des serveurs, déployer et annuler le déploiement des applications, régler les paramètres du système et apporter des modifications persistantes à la configuration du serveur. La console de management a également la capacité d'effectuer des tâches administratives, avec des notifications directes lorsque les modifications exigent que l'instance du serveur soit redémarrée ou rechargée.

Dans un domaine géré, les instances de serveur et les groupes de serveurs d'un même domaine peuvent être gérés de façon centralisée à partir de la console de management du contrôleur de domaine.

Rapporter un bogue

3.4.2. Se conncecter à la console de gestion

Dans la page d'accueil de la console de gestion de Red Hat JBoss Enterprise Application Platform, il y a un point d'entrée unique dans l'application pour les serveurs autonomes, ainsi que pour les serveurs exécutant dans un domaine géré. Cette page donne également des liens pour des ressources générales, opérationnelles ou à but de développement.

Conditions préalables

- Vous devez créer un utilisateur administratif comme indiqué ici : Section 4.1.1, « Ajouter un Utilisateur dans les interfaces de gestion ».

  JBoss EAP doit être en cours d'exécution.

Procédure 3.1. Se conncecter à la console de gestion

1. **Naviguer vers la page de démarrage de la console de gestion**
   Naviguez dans la console de gestion avec votre navigateur web. L'emplacement par défaut est http://localhost:9990/console/App.html, où le port 9990 est prédéfini comme liaison de socket de la console de gestion.

2. **Se conncecter à la console de gestion**
   Saisir le nom d'utilisateur et le mot de passe du compte que vous avez créés avant pour vous connecter à l'écran de connexion de la console de gestion.

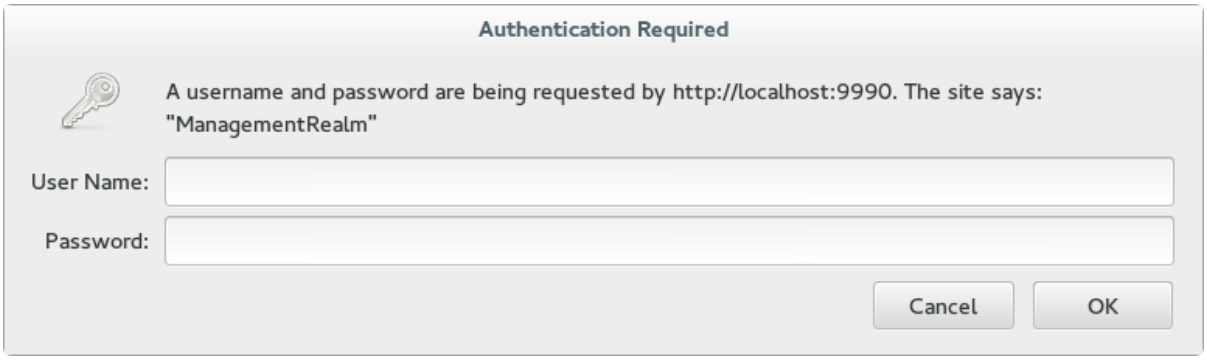


Figure 3.1. Écran de connexion de la console de gestion

Résultat

Une fois connecté, la page d'accueil de la console de gestion apparaîtra : http://localhost:9990/console/App.html#home

Rapporter un bogue

3.4.3. Changer la langue de la console de management

Les paramètres de configuration de la console de management basée web utilisent l'anglais par défaut. Vous pouvez décider d'utiliser une des langues suivantes à la place.

Langues prises en charge

- Allemand (de)
- Chinois simplifié (zn-Hans)
- Portugais brézilien (pt-BR)
- Français (fr)
- Espagnol (es)
- Japonais (ja)

Procédure 3.2. Changer la langue de la console de management basée-web

1. **Connectez-vous à la console de management.**
   Connectez-vous à la console de management basée web.
2. **Ouvrir le dialogue de configuration.**
   Dans le coin gauche de l'écran, il y a une étiquette **`Settings`** de configuration. Cliquer sur cette étiquette pour ouvrir les paramètres de configuration de la console de management.
3. **Sélectionner la langue désirée.**
   Sélectionner la langue désirée à partir de la case **`Locale`**. Puis sélectionner **`Save`**. Une autre case explicative vous demande de charger à nouveau l'application. Cliquer sur **`Confirm`**. Réactualiser votre navigateur pour pouvoir utiliser les nouveaux paramètres régionaux (Locale).

Rapporter un bogue

3.4.4. Données analytiques dans la console EAP

Google Analytics

Google Analytics est un service analytique web gratuit qui fournit des statistiques complètes sur un site Web. Il fournit des données essentielles relatives aux visiteurs d'un site comme leurs visites, pages vues, pages par visite et moyenne de temps passé sur le site. Google Analytics fournit une meilleure visibilité sur la présence d'un site Web et ses utilisateurs.

Google Analytics dans la console d'administration EAP

JBoss EAP 6.3 fournit aux utilisateurs l'option d'activer/de désactiver Google Analytics dans la console de gestion. La fonctionnalité Google Analytics vise à aider les équipes de Red Hat EAP à comprendre comment les clients utilisent la console et quelles parties de la console est particulièrement d'importance aux clients. Cette information aidera l'équipe à adapter l'apparence de la console, ses fonctionnalités et le contenu aux besoins immédiats des clients.

Rapporter un bogue

3.4.5. Activer/désactiver Google Analytics dans la console EAP

Pour activer Google Analytics dans la console d'administration EAP :

- Connectez-vous à la console d'administration
- Cliquer sur le bouton **`Settings`** en bas et à droite de la console

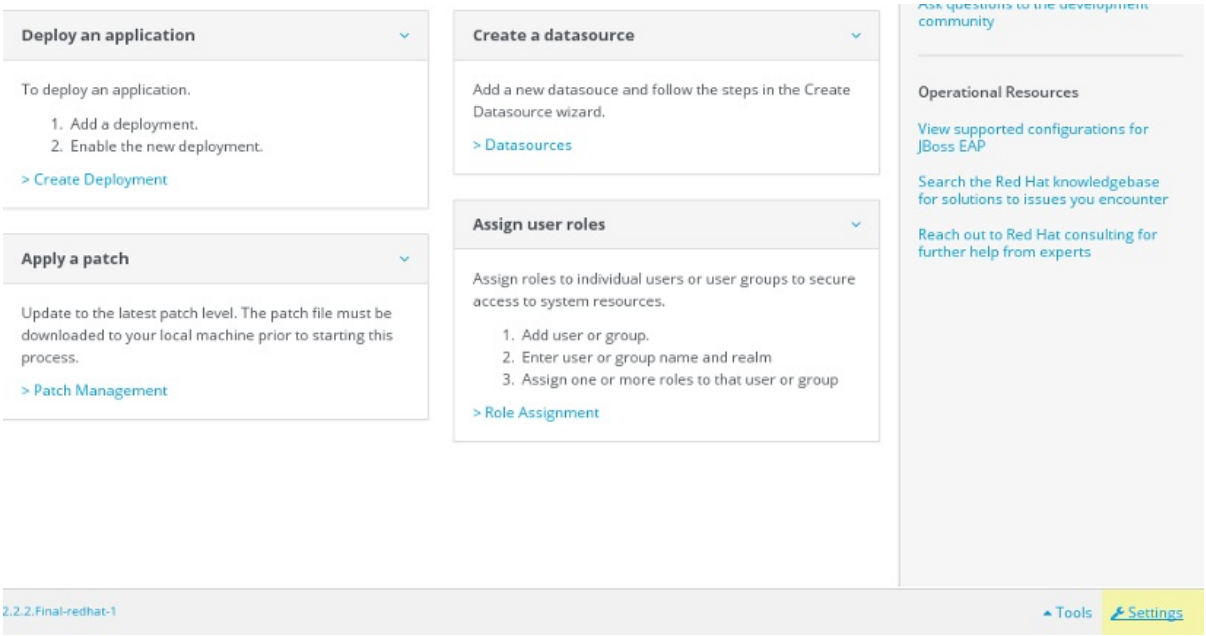


Figure 3.2. Connectez-vous à l'écran de console d'administration

- Sélectionner la case **`Enable Usage Data Collection`** sur la fenêtre **`Settings`** et cliquer sur le bouton **`Save`**. Recharger l'application pour activer les nouveaux paramètres de configuration.

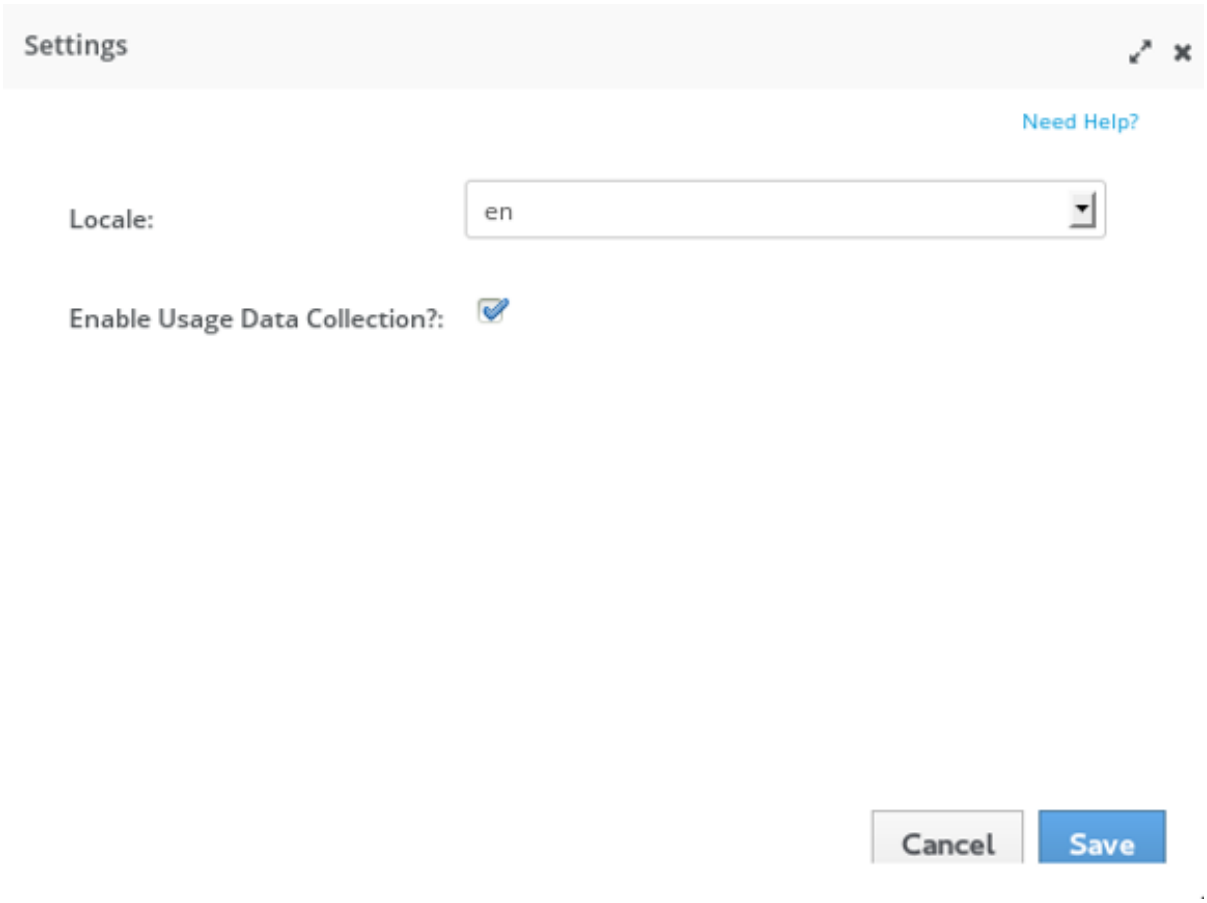


Figure 3.3. Configurer Window (Activer Usage Data Collection)

Pour désactiver Google Analytics dans la console d'administration après l'avoir activé, cliquer sur **`Enable Usage Data Collection`** dans la fenêtre de paramétrage **`Settings`** pour supprimer la sélection, puis cliquer sur **`Save`**.

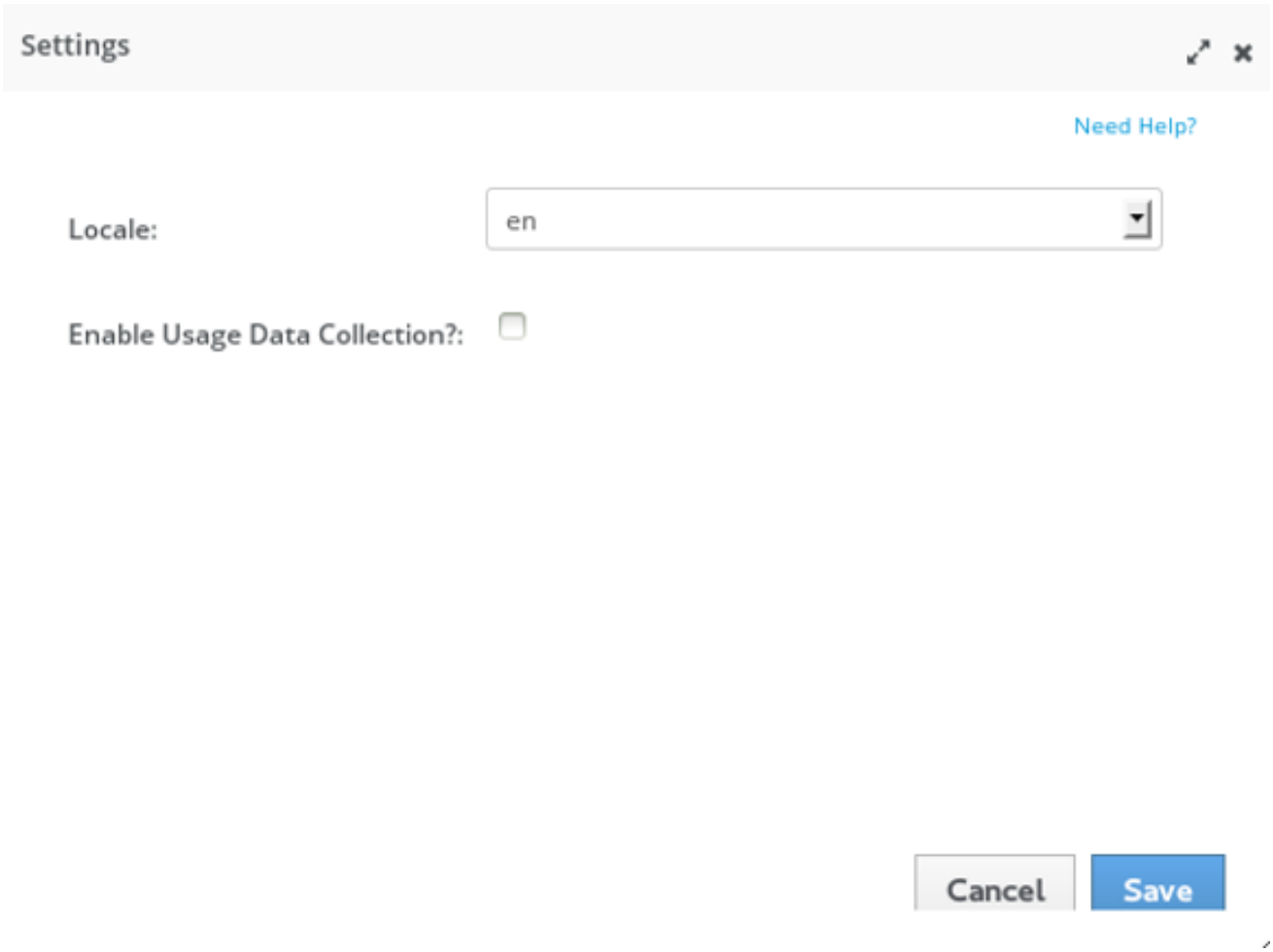
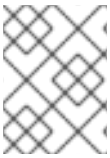


Figure 3.4. Configurer Window (Désactiver Usage Data Collection)



NOTE

Google Analytics est désativé par défaut dans la console d'administration EAP 6.3 et son utilisation est optionnelle

Rapporter un bogue

3.4.6. Configurer un serveur par la console de management

Pré-requis

- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »
- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 3.3. Configurer le serveur

1. Sélectionner l'onglet **`Domain`** en haut de la console. Les instances de serveur disponibles s'afficheront sur un tableau.
2. Sélectionner l'instance de serveur à partir du tableau **`Available Server Configurations`**.
3. Cliquer sur **`Edit`** au dessus des informations sur le serveur choisi.

4. Procédez avec les changements des attributs de configuration.

5. Cliquer sur le bouton **Save** pour terminer.

Résultat

La configuration du serveur a changé, et prendra effet la prochaine fois que le serveur démarrera.

Rapporter un bogue

3.4.7. Ajouter un déploiement dans une console de management

Pré-requis

- Section 3.4.2, « Se conncecter à la console de gestion »

1. Sélectionner l'onglet **Runtime** en haut de la console.

2. Pour un serveur autonome, il vous faudra étendre l'élément de menu **Server** et sélectionner **Manage Deployments**. Pour un domaine géré, étendre l'élément de menu **Domain** et sélectionner **Manage Deployments**. Le panneau **Manage Deployments** apparaîtra.

3. Sélectionner **Add** dans l'onglet **Content Repository**. Une boîte de dialogue **Create Deployment** apparaîtra.

4. Dans la boîte de dialogue, cliquer sur **Browse**. Cherchez le fichier que vous souhaitez déployer, et sélectionnez-le en vue de son chargement. Cliquer sur **Next** pour continuer.

5. Vérifier le nom du déploiement et le nom du runtime qui apparaît dans la boîte de dialogue **Create Deployments**. Sélectionner **Save** pour charger le fichier une fois que les noms auront été vérifiés.

Résultat

Le contenu sélectionné est téléchargé dans le serveur et est maintenant prêt à être déployé.

Rapporter un bogue

3.4.8. Créer un nouveau serveur dans la console de management

Pré-requis

- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »
- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 3.4. Créer une nouvelle configuration de serveur

1. **Naviguez sur la page Server Configuration qui se trouve dans la console de management**
   Sélectionner l'onglet **Domain** au haut de la console.

2. **Créer une nouvelle configuration**

   a. Sélectionner le bouton **Add** qui se trouve en haut du tableau **Available Server Configuration**.

b. Saisir les paramètres de base du serveur dans la boîte de dialogue **Create Server Configuration**.

c. Cliquez sur le bouton **Save** pour enregistrer la nouvelle configuration de votre serveur.

Résultat

Le nouveau serveur créé est listé dans **Server Configurations**.

Rapporter un bogue

3.4.9. Modifier les niveaux de journalisation par défaut en utilisant la console de management

Procédure 3.5. Modifier les niveaux de journalisation

1. **Naviguer dans le panneau Logging de la console de gestion.**

   a. Si vous travaillez dans un domaine géré, sélectionner l'onglet **Configuration** en haut de la console, puis, sélectionner le profil qui convient dans la liste déroulante à gauche de la console.

   b. Pour un domaine géré ou un serveur autonome, étendre le menu **Core** qui se trouve à gauche de la console et cliquer sur **Logging**.

   c. Cliquer sur l'onglet **Log Categories** en haut de la console.

2. **Modifier les informations du créateur du journal**
   Modifer les informations des entrées du tableau **Log Categories**.

   a. Sélectionner une entrée dans le tableau **Log Categories**, puis cliquer sur **Edit** dans la section **Details** ci-dessous.

   b. Définir le niveau de journalisation de la catégorie dans la zône déroulante **Log Level**. Cliquer sur le bouton **Save** quand c'est fait.

Résultat

Les niveaux de journalisation des catégories qui conviennent sont maintenant mis à jour.

Rapporter un bogue

3.4.10. Créer un nouveau groupe de service dans la console de gestion

Conditions préalables

- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 3.6. Configurer et ajouter un groupe de serveurs

1. **Se rendre dans la vue Server Groups**
   Sélectionner l'onglet **Domain** au haut de la console.

2. Étendre l'étiquette **Server** dans le menu qui se trouve en haut de la colonne. Sélectionner **Server Groups**.

3. **Ajouter un groupe de serveurs**
   Cliquer sur le bouton **Add** pour ajouter un nouveau groupe de serveurs.

4. **Configurer le groupe de serveurs**

   a. Saisir un nom pour le groupe de serveurs

   b. Sélectionner le profil d'un groupe de serveurs.

   c. Sélectionner la liaison de socket du groupe de serveurs.

   d. Cliquer sur le bouton **Save** pour sauvegarder le nouveau groupe.

Résultat

Le nouveau groupe de serveurs est visible dans la console de gestion.

Rapporter un bogue

3.5. L'INTERFACE CLI

3.5.1. Gestion par interface en ligne de commande (CLI)

La gestion par interface en ligne de commande (CLI) est un outil d'administration en ligne de commande pour JBoss EAP 6.

Utiliser l'interface CLI pour démarrer et stopper les serveurs, déployer et retirer les déploiements d'applications, configurer les paramètres du système, ou encore, effectuer d'autre tâches administratives. Les opérations peuvent être effectuées par mode de lots, ce qui permet à plusieurs tâches d'être exécutées en groupe.

Rapporter un bogue

3.5.2. Lancement de l'interface CLI

Conditions préalables :

- Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »
- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »

Procédure 3.7. Lancement du CLI dans Linux ou Windows

- **Lancement du CLI dans Linux**
  Exécutez le fichier ***EAP_HOME*/bin/jboss-cli.sh** en saisissant ce qui suit dans la ligne de commande :

  ```
  $ EAP_HOME/bin/jboss-cli.sh
  ```

- **Lancement du CLI dans Windows**
  Exécutez le fichier ***EAP_HOME*\bin\jboss-cli.bat** en cliquant deux fois, ou en saisissant ce qui suit dans la ligne de commande :

  ```
  C:\>EAP_HOME\bin\jboss-cli.bat
  ```

Rapporter un bogue

3.5.3. Quitter l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.8. Quitter l'interface CLI

- **Exécutez la commande `quit`**
  À l'aide de l'interface CLI, saisir la commande **`quit`** :

```
[domain@localhost:9999 /] quit
```

Rapporter un bogue

3.5.4. Se connecter à une instance de serveur géré par l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.9. Se connecter à une instance de serveur gérée

- **Exécutez la commande `connect`**
  À l'aide de l'interface CLI, saisir la commande **`connect`** :

```
[disconnected /] connect
Connected to domain controller at localhost:9999
```

  - Sinon, pour vous connecter à un serveur géré quand vous démarrez une interface CLI sur un système Linux, utiliser le paramètre ***`--connect`*** :

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

  - Le paramètre ***`--connect`*** peut être utilisé pour indiquer l'hôte et le port du serveur. Pour connecter l'adresse **`192.168.0.1`** à la valeur du port **`9999`** ce qui suit s'applique :

```
$ EAP_HOME/bin/jboss-cli.sh --connect --
controller=192.168.0.1:9999
```

Rapporter un bogue

3.5.5. Comment obtenir de l'aide par l'interface CLI

Résumé

L'interface CLI dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. Les commandes d'assistance qui dépendent du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne

seront pas affichées dans la liste, sauf si la connexion a été établie.

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.10. Aide au niveau options Générales et options Sensibles au contexte

1. **Exécuter la commande `help`**
   À l'aide de l'interface CLI, saisir la commande **`help`** :

   ```
   [standalone@localhost:9999 /] help
   ```

2. **Obtenez de l'aide au niveau sensibilité du contexte**
   À l'aide de l'interface CLI, saisir la commande étendue **`help -commands`** :

   ```
   [standalone@localhost:9999 /] help --commands
   ```

3. Pour plus d'informations sur une commande particulière, exécuter la commande **`help`** avec comme argument **`'--help'`**.

   ```
   [standalone@localhost:9999 /]  deploy --help
   ```

Résultat

L'information d'assistance du CLI s'affichera.

Rapporter un bogue

3.5.6. Utiliser l'interface CLI en mode par lot

Résumé

Le traitement par lot permet à un certain nombre de requêtes d'être groupées par séquences et exécutées ensemble par unité. Si une des demandes opérationnelles d'une séquence échoue, tout le groupe d'opérations sera annulé.

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »

Procédure 3.11. Commandes et opérations en mode par lot

1. **Saisir un mode par lot**
   Saisir le mode par lot par la commande **`batch`**.

   ```
   [standalone@localhost:9999 /] batch
   [standalone@localhost:9999 / #]
   ```

   Le mode par lot est indiqué par le signe (#) dans l'invite.

2. **Ajouter les demandes opérationnelles au lot**
   Une fois que vous serez en mode par lot, saisir les demandes opérationnelles comme d'habitude. Les demandes opérationnelles sont ajoutées au lot dans l'ordre de saisie.

   Voir Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI » pour obtenir des informations sur la façon de formater les demandes opérationnelles.

3. **Exécuter le lot**
   Une fois que toute la séquence de demandes opérationnelles est saisie, exécuter le lot avec la commande **`run-batch`**.

   ```
   [standalone@localhost:9999 / #] run-batch
   The batch executed successfully.
   ```

   Voir Section 3.5.7, « Commandes CLI Mode Lot » pour obtenir une liste complète des commandes disponibles pour pouvoir travailler avec des lots.

4. **Les commandes de lots sont stockées dans des fichiers externes**
   Les commandes fréquemment exécutées peuvent être stockées dans un fichier texte externe et être chargées en passant le chemin d'accès complet au fichier comme argument à la commande **`batch`** ou exécutées directement en étant un argument à la commande **`run-batch`**.

   Vous pouvez créer un fichier de commande lot avec l'éditeur de texte. Chaque commande doit figurer sur une ligne par elle-même et l'interface CLI doit être en mesure d'y accéder.

   La commande suivante chargera un fichier **`myscript.txt`** en mode lot. Toutes les commandes de ce fichier seront alors être prêtes à la modification ou à la suppression. De nouvelles commandes pourront être ajoutées. Les modifications effectuées au cours de cette session de lot ne persistera pas dans le fichier **`myscript.txt`**.

   ```
   [standalone@localhost:9999 /] batch --file=myscript.txt
   ```

   Ce qui suit exécutera instantanément les commandes lot stockées dans le fichier **`myscript.txt`**

   ```
   [standalone@localhost:9999 /] run-batch --file=myscript.txt
   ```

Résultat

La séquence de demandes opérationnelles saisies est effectuée sous forme de lot.

Rapporter un bogue

3.5.7. Commandes CLI Mode Lot

Ce tableau vous fournit une liste de commandes lot valides pouvant être utilisées dans JBoss EAP 6 CLI. Ces commandes ne peuvent être utilisées que pour travailler en lots.

Tableau 3.2. Commandes CLI Mode Lot

Command Name	Description
list-batch	Liste des commandes et des opérations du lot en cours.

Command Name	Description
edit-batch-line line-number edited-command	Modifier une ligne du lot en cours en donnant un numéro de ligne à modifier et la commande éditée. Exemple: **`edit-batch-line 2 data-source disable --name=ExampleDS`**.
move-batch-line fromline toline	Réorganiser les lignes dans le lot en spécifiant le numéro de ligne à déplacer comme premier argument et sa nouvelle position comme deuxième argument. Exemple : **`move-batch-line 3 1`**.
remove-batch-line linenumber	Supprimer la commande de lot à la ligne indiquée. Exemple: **`remove-batch-line 3`**.
holdback-batch [batchname]	Vous pouvez reporter à plus tard ou stocker un lot en cours à l'aide de cette commande. Utiliser cette option si vous voulez soudainement exécuter quelque chose dans la CLI en dehors du lot. Pour revenir à ce lot en attente, tapez simplement **`batch`** à nouveau à la ligne de commande CLI. Si vous fournissez un nom de lot en utilisant la commande **`holdback-batch`**, le lot sera stocké sous ce nom. Pour retourner au lot nommé, utilisez la commande **`batch batchname`**. L'appel de la commande **`batch`** sans un nom de lot va commencer un nouveau lot (sans nom). Il peut y avoir qu'un seul lot suspendu sans nom. Pour voir une liste de tous les lots suspendus, utiliser la commande **`batch -l`**.
discard-batch	Rejète le lot actif en cours.

Rapporter un bogue

3.5.8. Utiliser les opérations et les commandes de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »

Procédure 3.12. Créer, configurer et exécuter les requêtes

1. **Construire la demande opérationnelle**
   Les demandes opérationnelles facilitent une interaction de bas niveau dans le modèle de gestion. Il s'agit d'une façon contrôlée de modifier les configurations du serveur. Une demande opérationnelle se présente en trois parties :
   - une *adresse*, avec une barre oblique devant (/).

- un *nom d'opération*, avec deux points (`:`).
- un groupe optionnel de *paramètres*, entre parenthèses (`()`).

a. **Déterminer l'adresse**
La configuration est présentée sous forme de ressources auxquelles s'adresser et de façon hiérarchique. Chaque noeud de ressources procure un groupe différent d'opérations. Une adresse utilise la syntaxe suivante :

```
/node-type=node-name
```

- *node-type* correspond au type de noeud de ressource. Cela correspond à un nom d'élément dans la configuration XML.
- *node-name* correspond au nom du nœud. Cela correspond à l'attribut **nom** de l'élément dans la configuration XML.
- Séparer chaque niveau de l'arborescence de ressources par une barre oblique (`/`).

Voir les fichiers de configuration XML pour déterminer l'adresse qui convient. Le fichier ***EAP_HOME*/standalone/configuration/standalone.xml** contient la configuration d'un serveur autonome et les fichiers ***EAP_HOME*/domain/configuration/domain.xml** et ***EAP_HOME*/domain/configuration/host.xml** contiennent la configuration d'un domaine géré.

Exemple 3.3. Exemple d'adresses d'opérations

Pour procéder à une opération dans le sous-système de journalisation, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=logging
```

Pour effectuer une opération sur la source de données Java, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=datasources/data-source=java
```

b. **Déterminer l'opération**
Les opérations diffèrent avec chaque type de nœud de ressource. Une opération utilise la syntaxe suivante :

```
:operation-name
```

- *operation-name* correspond au nom de l'opération à demander.

Utiliser l'opération **read-operation-names** sur une adresse de ressources d'un serveur autonome pour lister les opérations disponibles.

Exemple 3.4. Opérations disponibles

Pour énumérer toutes les opérations disponibles du sous-système de journalisation, saisir la requête suivante dans un serveur autonome :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-names
{
    "outcome" => "success",
    "result" => [
        "add",
        "read-attribute",
        "read-children-names",
        "read-children-resources",
        "read-children-types",
        "read-operation-description",
        "read-operation-names",
        "read-resource",
        "read-resource-description",
        "remove",
        "undefine-attribute",
        "whoami",
        "write-attribute"
    ]
}
```

c. **Déterminer un paramètre**
Chaque opération a sans doute besoin de paramètres différents.

Les paramètres utilisent la syntaxe suivante :

```
(parameter-name=parameter-value)
```

- *parameter-name* correspond au nom du paramètre.
- *parameter-value* correspond à la valeur du paramètre.
- Les différents paramètres sont séparés par des virgules (`,`).

Afin de déterminer les paramètres qui conviennent, exécutez la commande **`read-operation-description`** sur un nœud de ressource, en faisant passer le nom de l'opération en tant que paramètre. Voir Exemple 3.5, « Déterminer les paramètres des opérations » pour plus de détails.

Exemple 3.5. Déterminer les paramètres des opérations

Afin de déterminer les paramètres qui conviennent pour l'opération **`read-children-types`** sur le sous-système de journalisation, saisir la commande **`read-operation-description`** comme suit :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-description(name=read-children-types)
{
    "outcome" => "success",
    "result" => {
        "operation-name" => "read-children-types",
        "description" => "Gets the type names of all the
children under the selected resource",
        "reply-properties" => {
```

```
            "type" => LIST,
            "description" => "The children types",
            "value-type" => STRING
        },
        "read-only" => true
    }
}
```

2. **Saisir toute la demande opérationnelle**
   Une fois que l'adresse, l'opération et tous les paramètres auront été sélectionnés, saisir la demande opérationnelle complète.

Exemple 3.6. Exemple de demande opérationnelle

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(recursive=true)
```

Résultat

L'interface de gestion effectue la demande opérationnelle sur la configuration du serveur.

Rapporter un bogue

3.5.9. Options de configuration de Management CLI

Le fichier de configuration de l'interface CLI - **`jboss-cli.xml`** - est chargé à chaque fois que le CLI démarre. Il doit se situer dans le répertoire ***`$EAP_HOME`*`/bin`** ou dans un système spécifié dans la propriété système **`jboss.cli.config`**.

`default-controller`

Configuration du contrôleur auquel il faut vous connecter si la commande **`connect`** est exécutée sans paramètre.

Paramètres de contrôleur par défaut

`host`

Nom d'hôte du contrôleur. La valeur par défaut est : **`localhost`**.

`port`

Numéro de port sur lequel il faut se connecter au contrôleur. La valeur par défaut est 9999.

`validate-operation-requests`

Indique si la liste des paramètres des demandes d'opération doit être validée avant que les demandes ne soient envoyées au contrôleur pour être exécutées. Type : Booléen. Valeur par défaut : **`true`**.

`history`

Cet élément contient la configuration pour les commandes et pour le journal de l'historique des opérations.

Paramètres `history`

`enabled`

Indique si **`history`** est activé ou non. Type : booléen. Valeur par défaut : **`true`**.

file-name

Nom du fichier dans lequel l'historique doit être stocké. La valeur par défaut est **`.jboss-cli-history`**.

file-dir

Répertoire dans lequel l'historique doit être stocké. La valeur par défaut est = **`$USER_HOME`**

max-size

Taille maximum du fichier d'historique. La valeur par défaut : 500.

resolve-parameter-values

Indique si l'on doit résoudre les propriétés système qui sont spécifiées comme argument de commande (ou paramètres d'opérations) avant d'envoyer la demande d'opération au contrôleur ou laisser la résolution avoir lieu côté serveur. Type : Booléen. Valeur par défaut : **`true`**.

connection-timeout

Durée autorisée pour établir une connexion avec le contrôleur. Type : entier relatif. Valeur par défaut : 5000 secondes.

ssl

Cet élément contient la configuration pour les stores Trust et Key utilisés par SSL.

Paramètres `ssl`

archivage sécurisé

Type : **`vaultType`**

key-store

Type : string.

key-store-password

Type : string.

alias

Type : string

key-password

Type : string

trust-store

Type : string.

trust-store-password

Type : string.

modify-trust-store

Si défini à **`true`**, le CLI invitera l'utilisateur quand des certificats non reconnus auront été reçus et les autorisera à les stocker dans le truststore. Type : Booléen. Valeur par défaut : **`true`**.

`vaultType`

Quand ni **`code`** ou **`module`** ne sont précisés. l'implémentation par défaut sera utilisée. Si le **`code`** est spécifié, mais pas le **`module`** il cherchera la classe spécifique dans le module de Picketbox. Si le **`module`** **et** le **`code`** sont spécifiés, il cherchera la classe spécifiée par le **`code`** dans le module spécifié par le 'module'.

code

Type : String.

module

Type : string

`silent`

Indique si les messages d'erreur et d'information doivent sortir dans le terminal. Même si la valeur **`false`** est indiquée, les messages continueront d'être enregistrés par le logger si sa configuration le permet et/ou si la cible de sortie est spécifiée dans la ligne de commande par >. La valeur par défaut est : **`False`**.

Rapporter un bogue

3.5.10. Références de commandes d'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

La section Section 3.5.5, « Comment obtenir de l'aide par l'interface CLI » décrit comment accéder aux fonctionnalités d'assistance de l'interface CLI. L'interface de gestion dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. Les commandes d'assistance dépendant du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne seront pas affichées dans la liste, sauf si la connexion a été établie.

Tableau 3.3.

Commande	Description
batch	Démarrer le mode par lot en créant un nouveau lot ou, selon les lots existants retenus, réactiver l'un d'entre eux. Si il n'y a pas de lot existant retenu, cette commande sans argument va démarrer un nouveau lot. S'il y a un lot sans nom existant retenu, cette commande le réactivera. S'il y a des lots existants retenus, mais avec nom, il peuvent être activés en exécutant cette commande avec ce nom comme argument.
cd	Change le chemin du nœud en cours à l'argument. Le chemin du nœud en cours est utilisé comme adresse pour les requêtes opérationnelles qui ne contiennent pas la partie adresse. Si une opération n'inclut pas d'adresse, l'adresse incluse sera considérée comme relative au chemin du nœud en cours. Le chemin du nœud en cours peut finir en type de nœud. Dans ce cas, exécuter une opération en spécifiant un nom de nœud est suffisant, comme logging:read-resource.
clear	Efface l'écran
command	Vous permet d'ajouter, de supprimer ou de lister des commandes existantes de type standard. Une commande de type standard est une commande qui est assignée à un type de nœud spécifique et qui vous permet d'effectuer une opération disponible à une instance de ce type. Elle vous permet de modifier n'importe quelle propriété exposée par le type de n'importe quelle instance existante.
connect	Connecte le contrôleur à l'hôte et au port spécifiés.
connection-factory	Définit une usine de connexions
data-source	Gère les configurations de sources de données JDBC dans le sous-système de la source de données.
deploy	Déploie l'application désignée par le chemin d'accès au fichier ou bien, active une application qui est pré-existante, mais désactivée dans le référentiel. Si elle est exécutée sans argument, cette commande énumérera tous les déploiements existants.
help	Affiche le message d'asistance. Peut être utilisé avec l'argument ***--commands*** pour fournir aux commandes données des résultats sensibles au contexte.
history	Affiche l'historique en mémoire de la commande CLI et affiche un statut pour savoir si l'expansion de l'historique est activée ou non. Peut être utilisé avec des arguments pour effacer, désactiver, ou activer l'expansion de l'historique selon les besoins.
jms-queue	Définit une file d'attente JMS dans le sous-système de messagerie.
jms-topic	Définit un topic dans le sous-système de messagerie.

Commande	Description
ls	Lister les contenus du chemin d'accès au nœud. Par défaut, le résultat est imprimé dans des colonnes qui utilisent toute la largeur du terminal. Utiliser ***-1*** affichera les résultats sur la base d'un nom par ligne.
pwd	Affiche le chemin d'accès du nœud pour le nœud en cours
quit	Termine l'interface de ligne de commande.
read-attribute	Affiche la valeur et, suivant les arguments, la description de l'attribut d'une ressource gérée.
read-operation	Affiche la description d'une opération particulière, ou bien liste toutes les opérations si aucune n'est spécifiée.
undeploy	Annule une demande lorsque celle-ci est exécutée avec le nom de l'application prévue. Peut être exécuté avec des arguments pour supprimer également l'application du référentiel. Imprime la liste de tous les déploiements existants si exécutée sans application spécifiée.
version	Affiche la version de serveur d'application et les informations d'environnement.
xa-data-source	Gére la configuration de la source de données JDBC XA du sous-système de la source de données.

Rapporter un bogue

3.5.11. Référence aux opérations d'interface CLI

Exposer les opérations d'interface CLI

Les opérations d'interface CLI peuvent être exposées par l'opération **read-operation-names** décrite dans la rubrique Section 3.6.5, « Afficher les noms de l'opération en utilisant l'interface CLI ». Les descriptions des opérations peuvent être exposées par l'opération **read-operation-descriptions** décrite dans la rubrique Section 3.6.4, « Affiche une description d'opération en utilisant l'interface CLI ».

Tableau 3.4. Les opérations d'interface CLI

Nom de l'opération	Description
add-namespace	Ajoute un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom.
add-schema-location	Ajoute un schéma de mappage d'emplacement à la mappe d'attribut schema-locations.
delete-snapshot	Efface un snapshot de la configuration serveur dans le répertoire de snapshots.

Nom de l'opération	Description
full-replace-deployment	Ajoute le contenu précédemment téléchargé de déploiement à la liste de contenu disponible, remplace le contenu existant du même nom dans le runtime et supprime le contenu remplacé dans la liste de contenu disponible. Voir lien pour plus de renseignements.
list-snapshots	Liste les snapshots de la configuration du serveur sauvegardée dans le répertoire des snapshots.
read-attribute	Affiche la valeur d'un attribut d'une ressource sélectionnée.
read-children-names	Affiche le nom de tous les enfants d'une ressource donnée ayant le type donné.
read-children-resources	Affiche des informations sur tous les enfants d'une ressource d'un type donné.
read-children-types	Affiche les noms de types de tous les enfants pour la ressource sélectionnée.
read-config-as-xml	Lit la configuration actuelle et l'affiche en format XML.
read-operation-description	Affiche les détails d'une opération de la ressource donnée.
read-operation-names	Affiche les noms de toutes les opérations de la ressource donnée.
read-resource	Affiche les valeurs des attributs d'un modèle de ressource avec des informations complètes ou de base sur n'importe quelle ressource enfant.
read-resource-description	Indique la description des attributs d'une ressource, les types de dépendants et les opérations.
reload	Charge le serveur à nouveau en fermant tous les services et en redémarrant.
remove-namespace	Supprime un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom.
remove-schema-location	Supprime un schéma de mappage d'emplacement à la mappe d'attribut schema-locations.
replace-deployment	Remplace le contenu existant du runtime par un contenu nouveau. Le nouveau contenu doit avoir été chargé auparavant dans le référentiel du contenu de déploiement.

Nom de l'opération	Description
`resolve-expression`	Opération qui accepte une expression comme entrée ou un string pouvant être compris comme une expression, et résolu en fonction du système local de propriétés et des variables d'environnement.
`resolve-internet-address`	Prend un ensemble de critères de résolution d'interface et trouve une adresse IP sur une machine locale qui correspond au critère, ou échoue si aucune adresse IP correspondante n'est trouvée.
`server-set-restart-required`	Met le serveur en mode «restart-required»
`shutdown`	Ferme le serveur via un appel à **`System.exit(0)`**.
`start-servers`	Démarre tous les serveurs configurés dans un domaine géré qui n'est pas actuellement en cours d'exécution.
`stop-servers`	Arrête tous les serveurs actuellement en cours d'exécution dans un domaine géré.
`take-snapshot`	Prend un snapshot de la configuration du serveur et la sauvegarde dans le répertoire des snapshots.
`upload-deployment-bytes`	Indique si le contenu de déploiement du tableau d'octets inclus doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.
`upload-deployment-stream`	Indique si le contenu de déploiement disponible dans l'index des flux entrants doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.
`upload-deployment-url`	Indique si le contenu de déploiement disponible dans l'URL doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime.
`validate-address`	Valide l'adresse de l'opération
`write-attribute`	Indique la valeur d'un attribut d'une ressource sélectionnée.

Rapporter un bogue

3.6. OPÉRATIONS DE L'INTERFACE CLI

3.6.1. Afficher les attributs d'une ressource par l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

L'opération **`read-attribute`** est une opération globale utilisée pour lire la valeur d'exécution d'un attribut sélectionné. Peut être utilisée pour exposer uniquement les valeurs qui ont été définies par l'utilisateur, en ignorant toute valeur par défaut ou non définie. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

`name`

Le nom de l'attribut pour obtenir la valeur sous la ressource sélectionnée.

`include-defaults`

Un paramètre booléen qui peut être défini à **`false`** pour limiter les résultats de l'opération aux attributs qui ont été définis par l'utilisateur uniquement, et ignorer les valeurs par défaut.

Procédure 3.13. Affiche la valeur de runtime en cours pour un attribut sélectionné.

- **Exécuter l'opération `read-attribute`**
  À l'aide de l'interface CLI, utiliser l'opération **`read-attribute`** pour afficher la valeur d'un attribut de ressource. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».

```
[standalone@localhost:9999 /]:read-attribute(name=name-of-attribute)
```

Un avantage de l'opération **`read-attribute`** est la possibilité d'exposer la valeur d'exécution actuelle d'un attribut spécifique. Des résultats similaires peuvent être obtenus avec l'opération **`read-attribute`**, mais seulement avec l'addition de la propriété de requête **`include-runtime`**, et uniquement dans le cadre d'une liste de toutes les ressources disponibles pour ce nœud. L'opération **`read-attribute`** est destinée aux requêtes d'attribut précises, comme le montre l'exemple suivant.

Exemple 3.7. Exécuter l'opération `read-attribute` pour exposer l'IP d'interface publique.

Si vous connaissez le nom de l'attribut que vous souhaitez exposer, vous pouvez utiliser la commande **`read-attribute`** pour renvoyer la valeur exacte dans le runtime en cours.

```
[standalone@localhost:9999 /] /interface=public:read-
attribute(name=resolved-address)
{
    "outcome" => "success",
    "result" => "127.0.0.1"
}
```

L'attribut **`resolved-address`** est une valeur de runtime, donc ne s'affiche pas dans les résultats de l'opération **`read-resource`** standard.

```
[standalone@localhost:9999 /] /interface=public:read-resource
{
    "outcome" => "success",
    "result" => {
```

```
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

Pour afficher **`resolved-address`** ou des autres valeurs de runtime, vous devrez utiliser la propriété de requête **`include-runtime`**.

```
[standalone@localhost:9999 /] /interface=public:read-resource(include-
runtime=true)
{
    "outcome" => "success",
    "result" => {
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "resolved-address" => "127.0.0.1",
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

Résultat

La valeur de l'attribut du runtime en cours est affichée.

Rapporter un bogue

3.6.2. Afficher l'utilisateur qui est actif dans l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

L'opération **`whoami`** est une opération globale utilisée pour identifier les attributs de l'utilisateur actif. L'opération expose l'identité de nom d'utilisateur et le domaine qui leur est attribué. L'opération **`whoami`** est utile pour les administrateurs qui gèrent plusieurs comptes d'utilisateurs dans plusieurs domaines, ou pour aider à assurer le suivi des utilisateurs actifs à travers les instances de domaine avec plusieurs sessions de terminal et les comptes utilisateurs.

Procédure 3.14. Affiche l'utilisateur qui est actif dans l'interface CLI par l'opération `whoami`

- **Exécuter l'opération `whoami`**
  À partir de l'interface CLI, utiliser l'opération **`whoami`** pour afficher le compte utilisateur actif.

  ```
  [standalone@localhost:9999 /] :whoami
  ```

  L'exemple suivant utilise la commande **`whoami`** dans une instance de serveur autonome pour montrer que l'utilisateur actif est le *username*, et que l'utilisateur est assigné au domaine **`ManagementRealm`**.

  Exemple 3.8. Utiliser la commande `whoami` dans une instance autonome

  ```
  [standalone@localhost:9999 /]:whoami
  {
      "outcome" => "success",
      "result" => {"identity" => {
          "username" => "username",
          "realm" => "ManagementRealm"
      }}
  }
  ```

Résultat

Votre compte d'utilisateur actif en cours est affiché

Rapporter un bogue

3.6.3. Affiche les informations système et serveur dans l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.15. Affiche les informations système et serveur dans l'interface CLI

- **Exécuter la commande `version`**
  À partir de l'interface CLI, saisir la commande **`version`** :

```
[domain@localhost:9999 /] version
```

Résultat

Les informations sur la version de votre serveur d'applications et sur votre environnement s'afficheront.

Rapporter un bogue

3.6.4. Affiche une description d'opération en utilisant l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.16. Éxécuter la commande CLI suivante

- **Exécuter l'opération `read-operation-description`**
  À partir de l'interface CLI, utiliser **`read-operation-description`** pour afficher des informations sur l'opération. L'opération requiert des paramètres supplémentaires dans le format d'une paire clé-valeur pour indiquer quelle opération afficher. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».

```
[standalone@localhost:9999 /]:read-operation-description(name=name-of-operation)
```

Exemple 3.9. Afficher la description de l'opération «list-snapshots»

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération **`list-snapshots`**.

```
[standalone@localhost:9999 /] :read-operation-description(name=list-snapshots)
{
    "outcome" => "success",
    "result" => {
        "operation-name" => "list-snapshots",
        "description" => "Lists the snapshots",
        "request-properties" => {},
        "reply-properties" => {
            "type" => OBJECT,
            "value-type" => {
                "directory" => {
                    "type" => STRING,
                    "description" => "The directory where the snapshots are stored",
                    "expressions-allowed" => false,
                    "required" => true,
                    "nillable" => false,
                    "min-length" => 1L,
```

```
                    "max-length" => 2147483647L
                },
                "names" => {
                    "type" => LIST,
                    "description" => "The names of the snapshots within
the snapshots directory",
                    "expressions-allowed" => false,
                    "required" => true,
                    "nillable" => false,
                    "value-type" => STRING
                }
            }
        },
        "access-constraints" => {"sensitive" => {"snapshots" => {"type"
=> "core"}}},
        "read-only" => false
    }
}
```

Résultat

La description est affichée pour l'opération choisie.

Rapporter un bogue

3.6.5. Afficher les noms de l'opération en utilisant l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.17. Éxécuter la commande CLI suivante

- **Exécuter l'opération `read-operation-names`**
  À partir de l'interface CLI, utiliser l'opération **`read-operation-names`** pour afficher les noms des opérations disponibles. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».

  ```
  [standalone@localhost:9999 /]:read-operation-names
  ```

Exemple 3.10. Afficher les noms de l'opération en utilisant l'interface CLI

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération **`read-operation-names`**.

```
[standalone@localhost:9999 /]:read-operation-names
{
    "outcome" => "success",
    "result" => [
        "add-namespace",
        "add-schema-location",
        "delete-snapshot",
        "full-replace-deployment",
```

```
            "list-snapshots",
            "read-attribute",
            "read-children-names",
            "read-children-resources",
            "read-children-types",
            "read-config-as-xml",
            "read-operation-description",
            "read-operation-names",
            "read-resource",
            "read-resource-description",
            "reload",
            "remove-namespace",
            "remove-schema-location",
            "replace-deployment",
            "resolve-expression",
            "resolve-internet-address",
            "server-set-restart-required",
            "shutdown",
            "take-snapshot",
            "undefine-attribute",
            "upload-deployment-bytes",
            "upload-deployment-stream",
            "upload-deployment-url",
            "validate-address",
            "validate-operation",
            "whoami",
            "write-attribute"
        ]
    }
```

Résultat

Les noms d'opérations disponibles sont affichés.

Rapporter un bogue

3.6.6. Afficher les ressources disponibles en utilisant l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

L'opération **`read-resource`** est une opération globale utilisée pour lire la valeur des ressources. Peut être utilisée pour exposer des informations complètes ou de base sur les ressources des nœuds en cours ou des nœuds enfants, ainsi qu'un ensemble de propriétés de requêtes qui peuvent étendre ou limiter l'étendue des résultats de l'opération. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

`recursive`

Pour savoir si on doit inclure récursivement des informations complètes sur les ressources enfant.

`recursive-depth`

La précision des informations de ressources enfant incluses

`proxies`

Si on doit inclure des ressources éloignées pour une recherche récursive. Par exemple, si on doit inclure les ressources niveau hôte à partir des contrôleurs hôtes esclave pour une demande de contrôleur de domaines.

`include-runtime`

Si on doit inclure des attributs de runtime dans la réponse, comme des valeurs d'attributs qui ne proviennent pas de la configuration persistante. Cette propriété de requête est définie sur false par défaut.

`include-defaults`

Une propriété de demande booléenne qui sert à activer ou à désactiver la lecture des attributs par défaut. Si définie sur false, seuls les attributs définis par l'utilisateur seront renvoyés, ignorant ainsi ceux qui sont non définis.

Procédure 3.18. Éxécuter la commande CLI suivante

1. **Exécuter l'opération `read-resource`**
   Avec l'interface CLI, faites l'opération **`read-resource`** pour afficher les ressources disponibles.

   ```
   [standalone@localhost:9999 /]:read-resource
   ```

   L'exemple suivant vous montre comment il est possible d'utiliser l'opération **`read-resource`** dans une instance de serveur autonome pour exposer les informations de ressources générales. Les résultats ressemblent au fichier de configuration **`standalone.xml`**, qui affiche les ressources de système, les extensions, les interfaces et les sous-systèmes installés ou configurés pour l'instance du serveur. Ces résultats peuvent être interrogés directement.

   Exemple 3.11. Exécuter l'opération `read-resource` niveau racine

   ```
   [standalone@localhost:9999 /]:read-resource
   {
       "outcome" => "success",
       "result" => {
           "deployment" => undefined,
           "deployment-overlay" => undefined,
           "management-major-version" => 1,
           "management-micro-version" => 0,
           "management-minor-version" => 4,
           "name" => "longgrass",
           "namespaces" => [],
           "product-name" => "EAP",
           "product-version" => "6.3.0.GA",
           "release-codename" => "Janus",
           "release-version" => "7.2.0.Final-redhat-3",
           "schema-locations" => [],
           "system-property" => undefined,
           "core-service" => {
   ```

```
            "management" => undefined,
            "service-container" => undefined,
            "server-environment" => undefined,
            "platform-mbean" => undefined
        },
        "extension" => {
            "org.jboss.as.clustering.infinispan" => undefined,
            "org.jboss.as.connector" => undefined,
            "org.jboss.as.deployment-scanner" => undefined,
            "org.jboss.as.ee" => undefined,
            "org.jboss.as.ejb3" => undefined,
            "org.jboss.as.jaxrs" => undefined,
            "org.jboss.as.jdr" => undefined,
            "org.jboss.as.jmx" => undefined,
            "org.jboss.as.jpa" => undefined,
            "org.jboss.as.jsf" => undefined,
            "org.jboss.as.logging" => undefined,
            "org.jboss.as.mail" => undefined,
            "org.jboss.as.naming" => undefined,
            "org.jboss.as.pojo" => undefined,
            "org.jboss.as.remoting" => undefined,
            "org.jboss.as.sar" => undefined,
            "org.jboss.as.security" => undefined,
            "org.jboss.as.threads" => undefined,
            "org.jboss.as.transactions" => undefined,
            "org.jboss.as.web" => undefined,
            "org.jboss.as.webservices" => undefined,
            "org.jboss.as.weld" => undefined
        },
        "interface" => {
            "management" => undefined,
            "public" => undefined,
            "unsecure" => undefined
        },
        "path" => {
            "jboss.server.temp.dir" => undefined,
            "user.home" => undefined,
            "jboss.server.base.dir" => undefined,
            "java.home" => undefined,
            "user.dir" => undefined,
            "jboss.server.data.dir" => undefined,
            "jboss.home.dir" => undefined,
            "jboss.server.log.dir" => undefined,
            "jboss.server.config.dir" => undefined,
            "jboss.controller.temp.dir" => undefined
        },
        "socket-binding-group" => {"standard-sockets" =>
undefined},
        "subsystem" => {
            "logging" => undefined,
            "datasources" => undefined,
            "deployment-scanner" => undefined,
            "ee" => undefined,
            "ejb3" => undefined,
            "infinispan" => undefined,
            "jaxrs" => undefined,
```

```
            "jca" => undefined,
            "jdr" => undefined,
            "jmx" => undefined,
            "jpa" => undefined,
            "jsf" => undefined,
            "mail" => undefined,
            "naming" => undefined,
            "pojo" => undefined,
            "remoting" => undefined,
            "resource-adapters" => undefined,
            "sar" => undefined,
            "security" => undefined,
            "threads" => undefined,
            "transactions" => undefined,
            "web" => undefined,
            "webservices" => undefined,
            "weld" => undefined
        }
    }
}
```

2. **Exécuter l'opération `read-resource` pour un nœud enfant**
   L'opération **`read-resource`** peut être exécutée pour chercher les nœuds enfants à partir de la racine. La structure de l'opération commence par définir le nœud à exposer, puis s'ajoute à l'opération pour exécuter à ses côtés.

```
[standalone@localhost:9999 /]/subsystem=web/connector=http:read-
resource
```

   Dans l'exemple suivant, on peut exposer des informations de ressources spécifiques sur un composant de sous-système en redirigeant l'opération **`read-resource`** vers un nœud de sous-système web particulier.

Exemple 3.12. Exposer les ressources de nœud enfant à partir d'un nœud racine

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
        "enable-lookups" => false,
        "enabled" => true,
        "executor" => undefined,
        "max-connections" => undefined,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
```

```
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

Les mêmes résultats sont possibles en utilisant la commande **cd** pour naviguer dans les nœuds enfants et en exécutant l'opération **read-resource** directement.

Exemple 3.13. Exposer les ressources de nœud enfant en changeant de répertoire

```
[standalone@localhost:9999 /] cd subsystem=web
```

```
[standalone@localhost:9999 subsystem=web] cd connector=http
```

```
[standalone@localhost:9999 connector=http] :read-resource
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
        "enable-lookups" => false,
        "enabled" => true,
        "executor" => undefined,
        "max-connections" => undefined,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

3. **Utiliser le paramètre récursif pour inclure des valeurs actives dans les résultats**
   Le paramètre récursif peut être utilisé pour exposer les valeurs de tous les attributs, y compris les valeurs non persistantes, celles qui sont passées au démarrage, ou les autres attributs normalement actifs du modèle d'exécution.

```
[standalone@localhost:9999 /]/interface=public:read-
resource(include-runtime=true)
```

Par rapport à l'exemple précédent, l'inclusion de la propriété de requête **include-runtime** expose des attributs actifs supplémentaires, comme des octets envoyés ou des octets reçus par le connecteur HTTP.

Exemple 3.14. Exposer des valeurs actives et supplémentaires par le paramètre `include-runtime`

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(include-runtime=true)
{
    "outcome" => "success",
    "result" => {
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression
"${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "resolved-address" => "127.0.0.1",
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

Rapporter un bogue

3.6.7. Afficher les descriptions de ressources disponibles en utilisant l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.19. Éxécuter la commande CLI suivante

1. **Exécuter l'opération `read-resource-description`**
   À partir du CLI, utiliser l'opération **`read-resource-description`** pour lire et afficher les noms des ressources disponibles. Pour plus d'informations sur les requêtes d'opérations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».

   ```
   [standalone@localhost:9999 /]:read-resource-description
   ```

2. **Utiliser les paramètres en option**

L'opération **`read-resource-description`** autorise l'utilisation de paramètres supplémentaires.

a. Utiliser le paramètre **`operations`** pour inclure les descriptions des opérations de la ressource.

```
[standalone@localhost:9999 /]:read-resource-
description(operations=true)
```

b. Utiliser le paramètre **`inherited`** pour inclure ou pour exclure des descriptions des opérations héritées de ressource. L'état par défaut est true.

```
[standalone@localhost:9999 /]:read-resource-
description(inherited=false)
```

c. Utiliser le paramètre **`recursive`** pour inclure les descriptions récursives des ressources dépendantes.

```
[standalone@localhost:9999 /]:read-resource-
description(recursive=true)
```

d. Utiliser le paramètre **`locale`** pour obtenir la description des ressources. Si « null », la locale régionale par défaut sera utilisée.

```
[standalone@localhost:9999 /]:read-resource-
description(locale=true)
```

Résultat

Les descriptions des ressources disponibles sont affichées.

Rapporter un bogue

3.6.8. Charger à nouveau le serveur d'applications à l'aide du CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Avec l'interface CLI, utiliser l'opération **`reload`** pour fermer tous les servcies et démarrer le runtime à nouveau. Une fois que la commande **`reload`** sera complétée, l'interface CLI se reconnectera automatiquement.

Pour obtenir plus d'informations sur les demandes d'opérations, voir Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».

Exemple 3.15. Charger à nouveau le serveur d'applications

```
[standalone@localhost:9999 /]:reload
{"outcome" => "success"}
```

Rapporter un bogue

3.6.9. Fermer le serveur d'applications par l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.20. Fermer le serveur d'applications

- **Exécuter l'opération `shutdown`**
  - À partir du CLI, utiliser l'opération **`shutdown`** pour fermer le serveur via l'appel de système **`System.exit(0)`**. Pour plus d'informations sur les requêtes d'opérations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes de l'interface CLI ».
    - En mode autonome, utiliser la commande suivante :

      ```
      [standalone@localhost:9999 /]:shutdown
      ```

    - En mode domaine, utiliser la commande suivante avec le nom d'hôte qui convient :

      ```
      [domain@localhost:9999 /]/host=master:shutdown
      ```

  - Pour vous connecter à une instance CLI détachée et pour fermer le serveur, exécuter la commande suivante :

    ```
    jboss-cli.sh --connect command=:shutdown
    ```

  - Pour vous connecter à une instance CLI éloignée et pour fermer le serveur, exécuter la commande suivante :

    ```
    [disconnected /] connect IP_ADDRESS
    Connected to IP_ADDRESS:PORT_NUMBER
    [192.168.1.10:9999 /] :shutdown
    ```

    Remplacer l'*IP_ADDRESS* par l'adresse IP de votre instance.

Résultat

Le serveur d'applications se ferme. Le CLI sera déconnecté car le runtime n'est pas disponible.

Rapporter un bogue

3.6.10. Configurer un attribut à l'aide du CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

L'opération **`write-attribute`** est une opération globale utilisée pour écrire ou modifier un attribut de

la ressource sélectionnée. Vous pouvez utiliser l'opération pour rendre les modifications persistantes ou pour modifier les paramètres de configuration de vos instances de serveur géré. Les propriétés de la requête incluent les paramètres suivants.

Propriétés de requêtes

`name`

Le nom de l'attribut pour définir la valeur sous la ressource sélectionnée.

`value`

La valeur désirée de l'attribut sous la ressource sélectionnée. Peut être « null » si le modèle sous-jacent supporte des valeurs « null ».

Procédure 3.21. Configurer un attribut de ressource par l'interface CLI

- **Exécuter l'opération`write-attribute`**
  À partir de l'interface CLI, utiliser l'opération **`write-attribute`** pour modifier la valeur d'un attribut de ressource. L'opération peut être exécutée dans le nœud dépendant de la ressource, ou bien dans le nœud root du CLI où le chemin de ressource complet est spécifié.

Exemple 3.16. Désactiver le scanner de déploiement par l'opération `write-attribute`

L'exemple suivant utilise l'opération **`write-attribute`** pour désactiver le scanner de déploiement. L'opération est exécutée à partir d'un nœud root, en utilisant l'onglet de complétion de tâche pour pouvoir peupler le chemin de ressources qui convient.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:write-attribute(name=scan-enabled,value=false)
{"outcome" => "success"}
```

Le résultat de l'opération peut être confirmé directement par l'opération **`read-attribute`**.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:read-attribute(name=scan-enabled)
{
    "outcome" => "success",
    "result" => false
}
```

Les résultats peuvent également être confirmés en listant tous les attributs de ressources du nœud disponibles par l'opération **`read-resource`**. Dans l'exemple suivant, cette configuration particulière vous montre que l'attribut **`scan-enabled`** est maintenant défini à **`false`**.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:read-resource
{
    "outcome" => "success",
    "result" => {
        "auto-deploy-exploded" => false,
        "auto-deploy-xml" => true,
        "auto-deploy-zipped" => true,
        "deployment-timeout" => 600,
```

```
        "path" => "deployments",
        "relative-to" => "jboss.server.base.dir",
        "scan-enabled" => false,
        "scan-interval" => 5000
    }
}
```

Résultat

L'attribut de ressource est mis à jour.

Rapporter un bogue

3.6.11. Configurer les propriétés système par l'interface CLI

Procédure 3.22. Configurer les propriétés système par l'interface CLI

1. Démarrer le serveur JBoss EAP.
2. Lancer l'interface CLI par la commande pour votre système d'exploitation.

   Dans Linux :

   ```
   EAP_HOME/bin/jboss-cli.sh --connect
   ```

   Dans Windows :

   ```
   EAP_HOME\bin\jboss-cli.bat --connect
   ```

3. Ajouter une propriété système.

   La commande que vous utilisez dépend de savoir si vous utilisez un serveur autonome ou un domaine géré. Si vous utilisez un domaine géré, vous pouvez ajouter des propriétés système à un ou à plusieurs serveurs exécutant sur ce domaine.

   - Ajouter une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

     ```
     /system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
     ```

     Exemple 3.17. Ajouter une propriété système sur un serveur autonome

     ```
     [standalone@localhost:9999 /] /system-
     property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
     e)
     {"outcome" => "success"}
     ```

   - Ajouter une propriété système sur tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

     ```
     /system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
     ```

Exemple 3.18. Ajouter une propriété système à tous les serveurs d'un domaine géré

```
[domain@localhost:9999 /] /system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
        "server-one" => {"response" => {"outcome" =>
"success"}},
        "server-two" => {"response" => {"outcome" =>
"success"}}
    }}}}
}
```

- Ajouter une propriété système à un hôte et à ses instances de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

Exemple 3.19. Ajouter une propriété système à un hôte et à ses serveurs dans un domaine

```
[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
        "server-one" => {"response" => {"outcome" =>
"success"}},
        "server-two" => {"response" => {"outcome" =>
"success"}}
    }}}}
}
```

- Ajouter une propriété système à une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

Exemple 3.20. Ajouter une propriété système à une instance de serveur d'un domaine géré

```
[domain@localhost:9999 /] /host=master/server-config=server-
one/system-
```

```
property=property.mybean.queue:add(value=java:/queue/MyBeanQueu
e)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {"server-one" => {"response" => {"outcome" =>
"success"}}}}}}
}
```

4. Lire une propriété système.

   La commande que vous utilisez dépend de savoir si vous exécutez sur un serveur autonome ou un domaine géré.

   - Lire une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

     ```
     /system-property=PROPERTY_NAME:read-resource
     ```

     Exemple 3.21. Lire une propriété système sur un serveur autonome

     ```
     [standalone@localhost:9999 /] /system-
     property=property.mybean.queue:read-resource
     {
         "outcome" => "success",
         "result" => {"value" => "java:/queue/MyBeanQueue"}
     }
     ```

   - Lire une propriété système sur tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

     ```
     /system-property=PROPERTY_NAME:read-resource
     ```

     Exemple 3.22. Lire une propriété système de tous les serveurs dans un domaine géré

     ```
     [domain@localhost:9999 /] /system-
     property=property.mybean.queue:read-resource
     {
         "outcome" => "success",
         "result" => {
             "boot-time" => true,
             "value" => "java:/queue/MyBeanQueue"
         }
     }
     ```

   - Lire une propriété système d'un hôte et de ses instances de serveur d'un domaine géré en utilisant la syntaxe suivante :

     ```
     /host=master/system-property=PROPERTY_NAME:read-resource
     ```

Exemple 3.23. Lire une propriété système d'un hôte et de ses serveurs dans un domaine

```
[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {
        "boot-time" => true,
        "value" => "java:/queue/MyBeanQueue"
    }
}
```

- Lire une propriété système d'une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/host=master/server-config=server-one/system-
property=PROPERTY_NAME:read-resource
```

Exemple 3.24. Lire une propriété système d'une instance de serveur dans un domaine géré

```
[domain@localhost:9999 /] /host=master/server-config=server-
one/system-property=property.mybean.queue:read-resource
{
    "outcome" => "success",
    "result" => {
        "boot-time" => true,
        "value" => "java:/queue/MyBeanQueue"
    }
}
```

5. Supprimer une propriété système.

   La commande que vous utilisez dépend de savoir si vous exécutez sur un serveur autonome ou un domaine géré.

   - Supprimer une propriété système sur un serveur autonome en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:remove
```

Exemple 3.25. Supprimer une propriété système sur un serveur autonome

```
[standalone@localhost:9999 /] /system-
property=property.mybean.queue:remove
{"outcome" => "success"}
```

- Supprimer une propriété système de tous les hôtes et serveurs d'un domaine géré en utilisant la syntaxe suivante :

```
/system-property=PROPERTY_NAME:remove
```

Exemple 3.26. Supprimer une propriété système de tous les hôtes et de ses serveurs dans un domaine.

```
[domain@localhost:9999 /] /system-
property=property.mybean.queue:remove
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
        "server-one" => {"response" => {"outcome" =>
"success"}},
        "server-two" => {"response" => {"outcome" =>
"success"}}
    }}}}
}
```

- Supprimer une propriété système d'un hôte et de ses instances de serveur dans un domaine géré en utilisant la syntaxe suivante :

```
/host=master/system-property=PROPERTY_NAME:read-resource
```

Exemple 3.27. Supprimer une propriété système d'un hôte et de ses serveurs dans un domaine

```
[domain@localhost:9999 /] /host=master/system-
property=property.mybean.queue:remove
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {
        "server-one" => {"response" => {"outcome" =>
"success"}},
        "server-two" => {"response" => {"outcome" =>
"success"}}
    }}}}
}
```

- Supprimer une propriété système d'une instance de serveur d'un domaine géré en utilisant la syntaxe suivante :

```
/host=master/server-config=server-one/system-
property=PROPERTY_NAME:remove
```

Exemple 3.28. Supprimer une propriété système d'un serveur dans un domaine géré

```
[domain@localhost:9999 /] /host=master/server-config=server-
one/system-property=property.mybean.queue:remove
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" =>
{"master" => {"server-one" => {"response" => {"outcome" =>
"success"}}}}}}
}
```

Rapporter un bogue

3.7. HISTORIQUE DES COMMANDES DE L'INTERFACE CLI

3.7.1. Utiliser l'historique de commande à l'aide de l'interface CLI.

L'interface CLI contient une fonctionnalité d'historique de commandes qui est activée par défaut dans l'installation du serveur d'applications. L'historique est conservé en tant qu'archive dans la mémoire volatile de la session CLI active, et est ajouté le fichier de journalisation qui sauvegarde automatiquement dans le répertoire d'accueil de l'utilisateur sous le nom **`.jboss-cli-history`**. Le fichier de l'historique est configuré par défaut pour enregistrer un maximum de 500 commandes CLI.

La commande **`history`** elle-même renverra l'historique de la session en cours, ou si accompagnée d'arguments, elle pourra désactiver, activer ou supprimer l'historique de la mémoire de session. L'interface CLI vous donne également la possibilité d'utiliser les flèches de votre clavier pour naviguer dans l'historique des commandes et des opérations.

Fonctions de l'historique de l'interface CLI

- Section 3.7.2, « Afficher l'historique de commandes par interface CLI. »
- Section 3.7.3, « Effacer l'historique de commandes par interface CLI. »
- Section 3.7.4, « Désactiver l'historique de commandes par l'interface CLI. »
- Section 3.7.5, « Activer l'historique des commandes de l'interface CLI »

Rapporter un bogue

3.7.2. Afficher l'historique de commandes par interface CLI.

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.23. Afficher l'historique de commandes par l'interface CLI.

- **Exécuter la commande `history`**
  À partir de l'interface CLI, saisir la commande **`history`** :

```
[standalone@localhost:9999 /] history
```

Résultat

L'historique de la commande CLI stocké en mémoire depuis le démarrage du CLI ou la commande de suppression de l'historique est affiché.

Rapporter un bogue

3.7.3. Effacer l'historique de commandes par interface CLI.

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.24. Effacer l'historique de commandes par interface CLI.

- **Exécuter la commande `history --clear`**
  À partir de l'interface CLI, saisir la commande **`history--clear`** :

```
[standalone@localhost:9999 /] history --clear
```

Résultat

L'historique des commandes enregistré depuis le démarrage du CLI est supprimé de la mémoire de session. L'historique de la commande est toujours présent dans le fichier **`.jboss-cli-history`** qui est sauvegardé dans le répertoire d'accueil de l'utilisateur.

Rapporter un bogue

3.7.4. Désactiver l'historique de commandes par l'interface CLI.

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.25. Désactiver l'historique de commandes par l'interface CLI.

- **Exécuter la commande `history --disable`**
  À partir de l'interface CLI, saisir la commande **`history --disable`** :

```
[standalone@localhost:9999 /] history --disable
```

Résultat

Les commandes passées dans le CLI ne seront pas enregistrées dans la mémoire ou dans un fichier **`.jboss-cli-history`** sauvegardé dans le répertoire d'accueil de l'utilisateur.

Rapporter un bogue

3.7.5. Activer l'historique des commandes de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 3.26. Activer l'historique des commandes de l'interface CLI

- **Exécuter la commande `history --enable`**
  À partir de l'interface CLI, saisir la commande **`history --enable`** :

```
[standalone@localhost:9999 /] history --enable
```

Résultat

Les commandes passées dans le CLI seront enregistrées dans la mémoire ou dans un fichier **`.jboss-cli-history`** sauvegardé dans le répertoire d'accueil de l'utilisateur.

Rapporter un bogue

3.8. LA JOURNALISATION D'AUDITING DE L'INTERFACE DE GESTION

3.8.1. La journalisation d'auditing de l'interface de gestion

La journalisation de l'auditing peut être activée pour que les opérations effectuées via l'API de gestion puissent être enregistrées dans un journal d'audit. Qu'elles soient conduites via la console de gestion, l'interface CLI, ou une interface écrite à la main, elles sont toutes assujetties à la journalisation d'auditing. Les logs peuvent être envoyés dans un fichier, transmis à un serveur syslog ou les deux à la fois. Par défaut, audit logging est désactivé.

Les données de journalisation sont en format JSON, avec plusieurs options de configuration pour pouvoir influencer les opérations incluses dans le log et le format des entrées de journalisation.

Avant d'être stockées, les entrées de journalisation passent par un formateur et un gestionnaire (handler). Le formateur spécifie le format des entrées de journalisation, tandis que le gestionnaire envoie les enregistrements vers une destination (ou plusieurs) particulière(s). Un seul formateur est actuellement disponible ; il crée des entrées en format JSON.

NOTE

La journalisation de l'auditing peut être configurée uniquement via l'interface CLI.

Rapporter un bogue

3.8.2. Activer la journalisation d'auditing de l'interface de gestion par le CLI

Pour activer la Journalisation de l'auditing par le CLI, utiliser la commande suivante.

```
/core-service=management/access=audit/logger=audit-log:write-
attribute(name=enabled,value=true)
```

La Journalisation d'auditing est rendue dans une sortie préfigurée qui va dans le fichier *EAP_HOME*/standalone/data/audit-log.log.

Rapporter un bogue

3.8.3. Formateur pour la journalisation d'auditing de l'interface de gestion

Le formateur spécifie le format des entrées de journalisation.

Tableau 3.5. Champs de formateur JSON

Attribut	Description
include-date	Valeur booléenne qui définit si l'horodatage est oui ou non inclus dans les archives log formatées.
date-separator	Chaîne contenant des caractères pour séparer la date du reste des messages log formatés. Sera ignoré si ***include-date***=**false**.
date-format	Format de date utilisé pour les horodatages compris par java.text.SimpleDateFormat. Ignoré si ***include-date***=**false**.
compact	Si défini sur **true**, formatera le JSON sur une seule ligne. Il y a sans doute encore des valeurs contenant de nouvelles lignes, donc s'il est important pour vous d'avoir l'enregistrement total sur une seule ligne, définir ***escape-new-line*** ou ***escape-control-characters*** à **true**.
escape-control-characters	Si défini sur **true**, échapera tous les caractères de contrôle (entrées ASCII à valeur décimales < 32) avec le code ASCII en octale ; par exemple, une nouvelle ligne devient '#012'. Si défini sur **true**, remplacera ***escape-new-line***=**false**.
escape-new-line	Si défini sur true, échappera toutes les nouvelles lignes avec le code ASCII en octale, comme par exemple **#012**.

Rapporter un bogue

3.8.4. Gestionnaire de fichiers de journalisation de l'auditing de l'interface de gestion

Un gestionnaire de fichiers indique les paramètres par lesquels les enregistrements de journalisation sont mis dans les fichiers. Il indique plus particulièrement le formateur, le nom du fichier et le chemin d'accès du fichier.

Tableau 3.6. Champs de journalisation d'audit de gestionnaire de fichiers.

Attribut	Description	Lecture seule

Attribut	Description	Lecture seule
formateur	Le nom d'un formateur JSON à utiliser pour formater les enregistrements de journalisation.	False
path	Le chemin d'accès du fichier de journalisation de l'auditing	False
relative-to	Le nom d'un chemin d'accès déjà nommé, ou d'un des chemins standard fournis par le système. Si ***`relative-to`*** est donné, la valeur de l'attribut du chemin d'accès sera traité comme relative au chemin spécifié par cet attribut.	False
failure-count	Le nombre d'échecs de journalisation depuis l'initialisation du gestionnaire.	True
max-failure-count	Le nombre maximum d'échecs de journalisation avant de désactiver ce gestionnaire.	False
disabled-due-to-failure	**`true`** si ce gestionnaire était désactivé à cause des échecs de journalisation.	True

Rapporter un bogue

3.8.5. Gestionnaire de fichier Syslog de journalisation de l'auditing de l'interface de gestion

Un gestionnaire de syslog indique les paramètres par lesquels les entrées de journalisations d'auditing sont envoyées à un serveur syslog, comme le nom d'hôte du serveur syslog et le nom du port sur lequel le serveur syslog écoute.

Envoyer une journalisation d'auditing à un serveur de syslog fournit davantage d'options de sécurité que de journaliser dans un fichier local ou sur un serveur syslog local. On peut définir plusieurs gestionnaires syslog.

Les serveurs syslog peuvent varier dans leur implémentation, donc tous les paramètres ne sont pas forcément applicables à tous les serveurs syslog. Le testing a été suivi par l'implémentation syslog *rsyslog*. Les RFC référencés sont les suivants :

- **`http://www.ietf.org/rfc/rfc3164.txt`**
- **`http://www.ietf.org/rfc/rfc5424.txt`**
- **`http://www.ietf.org/rfc/rfc6587.txt`**

Tableau 3.7. Champs de gestionnaire Syslog

Champ	Description	Lecture seule
formatter	Le nom du formateur à utiliser pour formater les enregistrements de journalisation.	False
failure-count	Le nombre d'échecs de journalisation depuis l'initialisation du gestionnaire	True
max-failure-count	Le nombre maximum d'échecs de journalisation avant de désactiver ce gestionnaire.	False
disabled-due-to-failure	**True** si ce gestionnaire était désactivé à cause des échecs de journalisation.	True
syslog-format	Format de syslog : *RFC-5424* ou *RFC-3164*.	False
max-length	La taille maximum d'un message de journalisation (en octets), comprenant l'en-tête. Si non défini, la valeur par défaut sera de **1024** octets si le **syslog-format** est **RFC3164**, ou **2048** octets si le **syslog-format** est **RFC5424**.	False.
truncate	Indique si un message doit ou non tronquer le message quand la longueur en octets est supérieure à la longueur maximale. Si la valeur est définie sur false, les messages seront divisés et envoyés avec les mêmes valeurs d'en-tête.	False

Rapporter un bogue

3.8.6. Activer la journalisation d'auditing de l'interface de gestion par le serveur syslog.

NOTE

Ajouter le préfixe **/host=*HOST_NAME*** aux commandes **/core-service** si vous devez appliquer les changements à un domaine géré.

Procédure 3.27. Activer la journalisation sur un serveur syslog

1. **Créer un syslog handler nommé `msyslog`**

```
[standalone@localhost:9999 /]batch
[standalone@localhost:9999 /]/core-
service=management/access=audit/syslog-
handler=mysyslog:add(formatter=json-formatter)
[standalone@localhost:9999 /]/core-
service=management/access=audit/syslog-
handler=mysyslog/protocol=udp:add(host=localhost,port=514)
[standalone@localhost:9999 /]run-batch
```

2. **Ajouter une référence au syslog handler.**

```
[standalone@localhost:9999 /]/core-
service=management/access=audit/logger=audit-
log/handler=mysyslog:add
```

Résultat

Les entrées de journalisation de l'auditing de l'interface de gestion sont alors enregistrées sur le serveur syslog.

Rapporter un bogue

3.8.7. Options de journalisation d'auditing de l'interface de gestion

En plus d'activer ou de désactiver la journalisation de l'auditing de l'interface de gestion, il existe d'autres options de journalisation.

Options de configuration

log-boot

Si défini sur **`true`**, quand on démarre le serveur, les opérations de gestion seront incluses dans le log d'auditing, sinon **`false`**. La valeur par défaut est **`false`**.

log-read-only

Si défini sur **`true`**, toutes les opérations d'auditing seront journalisées. Si défini sur **`false`**, seules les opérations qui changeront le modèle seront journalisées. La valeur par défaut est : **`false`**.

Rapporter un bogue

3.8.8. Champs de journalisation d'auditing de l'interface de gestion

Tableau 3.8. Champs de journalisation d'auditing de l'interface de gestion

Nom de champ	Description

Nom de champ	Description
type	Peut avoir pour valeur **`core`**, indiquant ainsi qu'il s'agit d'une opération de gestion, ou **`jmx`** indiquant une provenance de sous-système JMX (voir le sous-système JMX pour la configuration de la journalisation de l'auditing du sous-système JMX).
r/o	**`true`** si l'opération ne modifie pas le modèle de gestion, sinon **`false`**.
booting	**`true`** si l'opération a été exécutée à l'amorçage, **`false`** si elle a été exécutée une fois que le serveur était en route.
version	Numéro de version de l'instance JBoss EAP.
user	Nom d'utilisateur de l'utilisateur authentifié. Si l'opération a été journalisée via le CLI sur le même ordinateur que le serveur en cours d'exécution, l'utilisateur spécial **`$local`** sera utilisé.
domainUUID	Identifiant pour relier toutes les opérations tandis qu'elles sont propagées du contrôleur de domaines vers ses serveurs, ses contrôleurs hôtes, et ses serveurs de contrôleurs hôtes esclaves.
access	Cela peut avoir une des valeurs suivantes : NATIVE, HTTP, JMX. NATIVE - l'opération provient de l'interface de gestion native, par exemple le CLI. HTTP - l'opération provient de l'interface HTTP de domaine, par exemple la console d'administration. JMX - l'opération provient du sous-système de JMX. Voir JMX pour savoir comment configurer la journalisation de l'auditing pour JMX.
remote-address	L'adresse du client qui exécute l'opération.
success	**`true`** si l'opération a réussi, **`false`** si non.
ops	Les opérations en cours. Liste des opérations sérialisées dans JSON. Au démarrage, cela correspondra à toutes les opérations résultant du traitement XML. Une fois démarré, la liste contiendra normalement une seule entrée.

Rapporter un bogue

CHAPITRE 4. GESTION DES UTILISATEURS

4.1. CRÉATION D'UTILISATEUR

4.1.1. Ajouter un Utilisateur dans les interfaces de gestion

Aperçu

Les interfaces de gestion sont sécurisées par défaut dans JBoss EAP 6 car il n'y a aucun compte utilisateur initialement disponible, sauf si vous avez installé la plateforme à l'aide de l'installateur graphique. Il s'agit d'une précaution de sécurité pour empêcher les violations de sécurité des systèmes distants pour cause d'erreur de configuration simple. L'accès non-HTTP local est protégé par un mécanisme SASL, avec une négociation entre le client et le serveur la première fois que le client se connecte à partir de l'hôte local.

Cette tâche décrit comment créer l'utilisateur d'administration d'origine, qui peut utiliser la console de gestion basée web et les instances éloignées de l'interface CLI pour configurer et administrer la plate-forme JBoss EAP 6 à partir de systèmes distants.

NOTE

La communication HTTP avec la plate-forme JBoss EAP 6 est considérée « communication à distance », même si le trafic prend sa source sur l'hôte local. Par conséquent, vous devez créer au moins un utilisateur afin de pouvoir utiliser la console de gestion. Si vous essayiez d'accéder à la console de gestion avant d'ajouter un utilisateur, vous receviez une erreur parce qu'il n'y a pas de déploiement tant que l'utilisateur n'a pas été créé.

Procédure 4.1. Créer l'utilisateur administratif d'origine pour les interfaces de gestion distantes

1. **Invoquer le script `add-user.sh` ou `add-user.bat`.**
   Passez au répertoire ***`EAP_HOME`*`/bin/`**. Invoquer le script qui convient à votre système d'exploitation.

   Red Hat Enterprise Linux

   ```
   [user@host bin]$ ./add-user.sh
   ```

   Microsoft Windows Server

   ```
   C:\bin>  add-user.bat
   ```

2. **Choisissez d'utiliser un utilisateur Management.**
   Appuyer sur **`ENTER`** pour sélectionner l'option **`a`** pour ajouter un utilisateur Management. Cet utilisateur sera ajouté au domaine **`ManagementRealm`** et il sera autorisé à effectuer des opérations de gestion par la console de gestion basée web ou par l'interface CLI basée ligne de commande. Autre alternative, **`b`**, ajoutera l'utilisateur au domaine **`ApplicationRealm`**, et ne fournit aucune permission particulière. Ce domaine est fourni pour être utilisé avec des applications.

3. **Saisir le nom d'utilisateur et le mot de passe que vous souhaitez.**

Quand on vous y invite, saisir le nom d'utilisateur et le nom de passe. On vous demandera de saisir le mot de passe une seconde fois pour confirmer.

4. **Saisissez les informations sur votre groupe.**
   Ajouter le groupe ou les groupes auxquels l'utilisateur appartient. Si l'utilisateur appartient à plusieurs groupes, saisir une liste séparée par des virgules. Laisser vide s'il n'y a pas de groupes pour l'utilisateur.

5. **Vérifier les informations et confirmer.**
   On vous invitera à confirmer les informations. Quand vous serez satisfait, saisir **`yes`**.

6. **Décidez si l'utilisateur représente une instance de serveur de JBoss EAP 6 à distance.**
   En plus des administrateurs, un autre type d'utilisateur qui a parfois besoin d'être ajouté à JBoss EAP 6 dans le domaine **`ManagementRealm`** est un utilisateur qui représente une autre instance de JBoss EAP 6, et qui a besoin d'être authentifié pour rejoindre un groupement en tant que membre. L'invite suivante vous permet de désigner votre utilisateur supplémentaire dans ce but. Si vous sélectionnez **`yes`**, on vous donnera une valeur **`secret`** de hachage, qui représentera le mot de passe de l'utilisateur, que vous aurez besoin d'ajouter dans un fichier de configuration différent. Dans le but de cette tâche, répondre **no** à cette question.

7. **Saisir des utilisateurs supplémentaires.**
   Vous pouvez saisir des utilisateurs supplémentaires si vous le souhaitez, en répétant la procédure. Vous pouvez également les ajouter à tout moment sur un système en cours d'exécution. Au lieu de choisir le domaine de sécurité par défaut, vous pouvez ajouter des utilisateurs d'autres domaines afin d'ajuster leurs autorisations.

8. **Créer des utilisateurs en mode non interactif.**
   Vous pouvez créer des utilisateurs en mode non interactif, en l'indiquant dans chaque paramètre de ligne de commande. Cette approche n'est pas recommandée sur les systèmes partagés, parce que les mots de passe seront visibles dans les fichiers de journalisation (log) et dans les fichiers d'historique. La syntaxe de la commande, pour le domaine de gestion, est la suivante :

   ```
   [user@host bin]$ ./add-user.sh username password
   ```

   Pour utiliser le domaine d'application, utiliser le paramètre **`-a`**.

   ```
   [user@host bin]$ ./add-user.sh -a username password
   ```

9. Vous pouvez supprimer la sortie normale du script d'ajout d'utilisateur en passant le paramètre **`--silent`**. Cela s'applique uniquement si un minimum de paramètres, **`nom d'utilisateur`** et **`mot de passe`**, ont été indiqués. Le message d'erreur apparaîtra toujours.

Résultat

Tout utilisateur que vous ajoutez est activé dans les domaines de sécurité que vous avez indiqués. Les utilisateurs actifs dans le domaine **`ManagementRealm`** sont en mesure de gérer la plateforme JBoss EAP 6 à partir de systèmes éloignés.

Voir également :

- Section 11.8.1, « Configuration de la sécurité utilisateur par défaut »

4.1.2. Passer des arguments au script add-user de la gestion utilisateur

Vous pouvez exécuter la commande **`add-user.sh`** ou **`add-user.bat`** interactivement ou vous pouvez passer des arguments par la ligne de commande. Cette section décrit les options qui se présentent pour passer des arguments en ligne de commande au script add-user.

Pour obtenir une liste d'arguments en ligne de commandes disponibles pour **`add-user.sh`** ou **`add-user.bat`**, consulter Section 4.1.3, « Arguments pour la commande Add-user » .

Pour plus d'informations sur la façon d'indiquer un autre fichier de propriétés et son emplacement, consulter Section 4.1.4, « Spécifier des fichiers de propriétés alternatifs pour les informations de gestion des utilisateurs » .

Pour obtenir des exemples qui montrent comment passer des arguments sur la commande **`add-user.sh`** ou **`add-user.bat`**, consulter Section 4.1.5, « Exemples de lignes de commande de script Add-user » .

Rapporter un bogue

4.1.3. Arguments pour la commande Add-user

Le tableau suivant décrit les arguments disponibles pour la commande **`add-user.sh`** ou **`add-user.bat`**.

Tableau 4.1. Arguments pour la commande Add-user

Argument de ligne de commande	Valeur d'argument	Description
-a	S/O	Cet argument demande de créer un utilisateur dans le domaine de l'application. S'il est omis, un utilisateur sera créé par défaut dans le domaine de gestion.
-dc	*DOMAIN_CONFIGURATION_DIRECTORY*	Cet argument spécifie le répertoire de configuration de domaine qui contient les fichiers de propriétés. S'il est omis, le répertoire par défaut sera ***`EAP_HOME`*`/domain/configuration/`**.
-sc	*SERVER_CONFIGURATION_DIRECTORY*	Cet argument spécifie un répertoire de configuration de serveur autonome différent qui contient les fichiers de propriétés. S'il est omis, le répertoire par défaut sera ***`EAP_HOME`*`/standalone/configuration/`**.
-up --user-properties	*USER_PROPERTIES_FILE*	Cet argument spécifie le nom d'un autre fichier de propriétés utilisateur. Il peut correspondre à un chemin absolu ou il peut correspondre à un nom de fichier utilisé en conjonction avec l'argument **`-sc`** ou **`-dc`** qui spécifie le répertoire de configuration alternatif.
-g --group	*GROUP_LIST*	Une liste séparée par des virgules de groupes à assigner à cet utilisateur.

Argument de ligne de commande	Valeur d'argument	Description
-gp --group-properties	*GROUP_PROPERTIES_FILE*	Cet argument spécifie le nom d'un autre fichier de propriétés de groupe. Il peut correspondre à un chemin absolu ou il peut correspondre à un nom de fichier utilisé en conjonction avec l'argument **`-sc`** ou **`-dc`** qui spécifie le répertoire de configuration alternatif.
-p --password	*PASSWORD*	Le mot de passe utilisateur. Le mot de passe doit remplir les critères suivants : • Il doit contenir 8 caractères au moins. • Il doit contenir au moins un caractère de l'alphabet. • Il doit contenir un chiffre au moins. • Il doit contenir au moins un symbole non alphanumérique
-u --user	*USER_NAME*	Le nom de l'utilisateur. Il ne peut contenir que des caractères alphanumériques.
-r --realm	*REALM_NAME*	Le nom du domaine utilisé pour sécuriser les interfaces de gestion. S'il est omis, la valeur par défaut sera **`ManagementRealm`**.
-s --silent	S/O	Exécuter le script add-user sans sortie vers la console.
-h --help	S/O	Afficher les informations d'utilisation du script add--user.

Rapporter un bogue

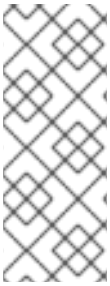
4.1.4. Spécifier des fichiers de propriétés alternatifs pour les informations de gestion des utilisateurs

Aperçu

Par défaut, les informations utilisateurs et rôles créés à l'aide du script **`add-user.sh`** ou **`Add-user.bat`** sont stockées dans des fichiers de propriétés situés dans le répertoire de configuration de serveur. Les informations de configuration du serveur sont stockées dans le répertoire **`EAP_HOME/standalone/configuration/`** et les informations de configuration de domaine sont stockées dans le répertoire **`EAP_HOME/domaine/configuration/`**. Cette rubrique décrit comment substituer les noms de fichier et emplacements par défaut.

Procédure 4.2. Spécifier des fichiers de propriétés alternatifs

- Pour spécifier un autre répertoire pour la configuration du serveur, utilisez l'argument **-sc**. Cet argument spécifie un autre répertoire qui contiendra les fichiers de propriétés de configuration de serveur.
- Pour spécifier un répertoire alternatif de configuration de domaine, utiliser l'argument **-dc**. Cet argument spécifie un répertoire alternatif qui contient les fichiers de propriétés de configuration de domaines.
- Pour spécifier un fichier de configuration utilisateur différent, utiliser l'argument **-up** ou **--user-properties**. Peut correspondre à un chemin complet ou à un nom de fichier utilisé en conjonction avec **-sc** ou **-dc** spécifiant le répertoire de configuration alternatif.
- Pour spécifier un fichier de configuration de groupe différent, utiliser l'argument **-up** ou **--group-properties**. Peut correspondre à un chemin complet ou à un nom de fichier utilisé en conjonction avec **-sc** ou **-dc** indiquant le répertoire de configuration alternatif.



NOTE

La commande **add-user** a pour but d'opérer sur des fichiers de propriétés existants. Tout fichier de propriété alternatif spécifié dans un argument de ligne de commande devra sortir là où vous verrez l'erreur suivante :

```
JBAS015234: No appusers.properties files found
```

Pour obtenir plus d'informations sur les arguments de commande, consulter Section 4.1.3, « Arguments pour la commande Add-user » .

Pour obtenir plus d'exemples sur les commandes add-user, consulter Section 4.1.5, « Exemples de lignes de commande de script Add-user » .

Rapporter un bogue

4.1.5. Exemples de lignes de commande de script Add-user

Les exemples suivants montrent comment passer des arguments à la commande **add-user.sh** ou **add-user.bat**. À moins que cela soit notifié, ces commandes supposent la configuration d'un serveur autonome.

Exemple 4.1. Créer un utilisateur qui appartienne à un groupe unique en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -a -u 'appuser1' -p 'password1!' -g 'guest'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **appuser1** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/application-users.properties**

  ***EAP_HOME*/domain/configuration/application-users.properties**

- L'utilisateur **appuser1** ayant pour groupe **guest** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/application-roles.properties**

  ***EAP_HOME*/domain/configuration/application-roles.properties**

Exemple 4.2. Créer un utilisateur qui appartienne à plusieurs groupes en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -a -u 'appuser1' -p 'password1!' -g
'guest,app1group,app2group'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **appuser1** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/application-users.properties**

  ***EAP_HOME*/domain/configuration/application-users.properties**

- L'utilisateur **appuser1** ayant pour groupes **guest**, **app1group**, et **app2group** est ajouté aux fichiers de propriétés suivantes par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/application-roles.properties**

  ***EAP_HOME*/domain/configuration/application-roles.properties**

Exemple 4.3. Créer un utilisateur ayant des privilèges admin dans le domaine par défaut en utilisant les fichiers de propriétés par défaut.

```
EAP_HOME/bin/add-user.sh -u 'adminuser1' -p 'password1!' -g 'admin'
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **adminuser1** est ajouté aux fichiers de propriétés suivantes par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/mgmt-users.properties**

  ***EAP_HOME*/domain/configuration/mgmt-users.properties**

- L'utilisateur **adminuser1** ayant pour groupe **admin** est ajouté aux fichiers de propriétés suivants par défaut qui contiennent les informations utilisateur.

  ***EAP_HOME*/standalone/configuration/mgmt-groups.properties**

  ***EAP_HOME*/domain/configuration/mgmt-groups.properties**

Exemple 4.4. Créer un utilisateur qui appartienne à un groupe unique en utilisant des fichiers de propriétés alternatifs pour stocker des informations.

```
EAP_HOME/bin/add-user.sh -a -u appuser1 -p password1! -g app1group -sc
/home/someusername/userconfigs/ -up appusers.properties -gp
appgroups.properties
```

La commande ci-dessus produit les résultats suivants.

- L'utilisateur **appuser1** est ajouté aux fichiers de propriétés suivants et ce fichier est maintenant le fichier par défaut qui contient les informations utilisateur.

  /home/someusername/userconfigs/appusers.properties

- L'utilisateur **appuser1** ayant pour groupe **app1group** est ajouté aux fichiers de propriétés suivants et ce fichier est maintenant le fichier par défaut qui contient les informations utilisateur.

  /home/someusername/userconfigs/appgroups.properties

Rapporter un bogue

CHAPITRE 5. RÉSEAU ET CONFIGURATION DE PORT

5.1. INTERFACES

5.1.1. Les interfaces

Le serveur d'applications utilise des références d'interface nommées dans la configuration. Cela permet à la configuration de référencer les déclarations d'interface individuelle avec des noms logiques, au lieu de référencer toutes les informations sur l'interface à chaque utilisation. L'utilisation de noms logiques permet également de conserver une homogénéité des références de groupe pour les interfaces nommées, quand les instances de serveur d'un domaine géré peut contenir des informations d'interface diverses à travers les machines. En utilisant cette méthodologie, chaque instance de serveur peut correspondre à un groupe de nom logique, ce qui facilite l'administration du groupe d'interfaces dans son ensemble.

Une interface réseau est déclarée en spécifiant un nom logique et un critère de sélection pour l'interface physique. Le serveur d'applications est fourni avec une configuration par défaut pour une gestion et un nom d'interface publique. Dans cette configuration, le groupe de l'interface publique est destiné à être utilisé par toute communication de réseau liée à l'application comme Web ou Messaging. Le groupe interface de gestion est destiné à tous les composants et les services requis par la couche de gestion, y compris HTTP Management Endpoint. Les noms d'interface sont fournis en tant que suggestions uniquement, alors que n'importe quel nom de groupe peut être substitué ou créé selon les besoins.

Les fichiers de configuration **`domain.xml`**, **`host.xml`** et **`standalone.xml`** comprennent tous des déclarations d'interface. Les critères de déclaration peuvent référencer une adresse générique ou spécifier un ensemble de caractéristiques qu'une interface ou une adresse doit avoir pour pouvoir établir une correspondance valide. Les exemples suivants illustrent plusieurs configurations possibles de déclarations d'interface, généralement définies soit dans le fichier de configuration **`standalone.xml`** ou **`host.xml`**. Cela permet à des groupes d'hôtes distants de maintenir leurs propres attributs d'interfaces spécifiques, tout en permettant une référence aux groupes d'interfaces dans le fichier de configuration **`domain.xml`** du contrôleur de domaine.

Le premier exemple montre une valeur d'**`inet-address`** indiquée pour le groupes de noms relatifs **`management`** et **`public`**.

Exemple 5.1. Un groupe d'interfaces créé avec une adresse inet

```
<interfaces>
  <interface name="management">
   <inet-address value="127.0.0.1"/>
  </interface>
  <interface name="public">
   <inet-address value="127.0.0.1"/>
  </interface>
</interfaces>
```

Dans l'exemple suivant, un groupe global interface utilise l'élément **`any-address`** pour déclarer une adresse générique.

Exemple 5.2. Un groupe global créé avec une déclaration générique

```
<interface name="global">
   <!-- Use the wild-card address -->
   <any-address/>
</interface>
```

L'exemple suivant déclare une carte d'interface de réseau sous un groupe relatif avec un nom **externe**.

Exemple 5.3. Un groupe externe créé avec un NIC

```
<interface name="external">
   <nic name="eth0"/>
</interface>
```

Dans l'exemple suivant, une déclaration est créée comme groupe par défaut pour un besoin spécifique. Dans ce cas, les caractéristiques des éléments supplémentaires définissent les conditions pour que l'interface soit une correspondance valide. Cela permet la création de groupes de déclaration d'interface très spécifique, avec la possibilité de les référencer de manière prédéfinie, réduisant ainsi le temps de configuration et d'administration sur plusieurs instances de serveur.

Exemple 5.4. Un groupe par défaut créé avec des valeurs conditionnelles spécifiques

```
<interface name="default">
   <!-- Match any interface/address on the right subnet if it's
        up, supports multicast, and isn't point-to-point -->
   <subnet-match value="192.168.0.0/16"/>
   <up/>
   <multicast/>
   <not>
      <point-to-point/>
   </not>
</interface>
```

Alors que les déclarations d'interface peuvent être faites et modifiées dans les fichiers de configuration des sources, l'interface CLI et la console de gestion fournissent un environnement sécurisé, contrôlé et persistant pour les modifications de configuration.

Rapporter un bogue

5.1.2. Configurer les interfaces

Les configurations de l'interface par défaut des fichiers de configuration **standalone.xml** et **host.xml** offrent trois interfaces nommées avec jetons d'interfaces relatives pour chacune. Vous pouvez utiliser la console de gestion ou l'interface CLI pour configurer des attributs et valeurs supplémentaires indiquées dans le tableau ci-dessous. Vous pouvez également remplacer les liaisons

d'interfaces relatives par des valeurs spécifiques selon les besoins, mais notez que si vous le faites, vous serez incapable de passer une valeur d'interface en cours d'exécution de serveur, car **-b**> peut seulement substituer une valeur relative.

Exemple 5.5. Configuration d'interface par défaut

```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address:127.0.0.1}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="${jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>
```

Tableau 5.1. Attributs et valeurs d'interface

Élément d'interface	Description
any	Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection.
any-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique. L'adresse générique IPv6 (::) sera utilisée à moins que la propriété système java.net.preferIpV4Stack soit définie sur true, dans lequel cas, l'adresse générique (0.0.0.0) IPv4 sera utilisée. Si un socket est lié à une adresse anylocal IPv6 sur une machine dual-stack, il pourra accepter le trafic IPv6 et IPv4 ; si lié à l'adresse IPv4 anylocal (mappées IPv4), il ne peut accepter que le trafic IPv4.
any-ipv4-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (0.0.0.0) IPv4.
any-ipv6-address	Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (::). IPv6.
inet-address	Soit une adresse IP en notation à points IPV6 ou IPV4, ou un nom d'hôte qui puisse être résolu.
link-local-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit si oui ou non il a une adresse associée local-link.
loopback	Élément vide indiquant qu'une partie du critère de sélection d'une interface est de savoir s'il s'agit oui ou non d'une interface de loopback.

Élément d'interface	Description
loopback-address	Une adresse de loopback qui ne peut pas réellement être configurée sur l'interface de loopback de la machine. Diffère d'inet-addressType car la valeur donnée sera utilisée même si aucune carte réseau possédant l'adresse IP associée ne peut être trouvée.
multicast	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être si oui ou non il y a un support multi-diffusion.
nic	Le nom d'une interface de réseau (e.g. eth0, eth1, lo).
nic-match	Une expression standard à laquelle faire correspondre les noms des interfaces de réseau disponibles sur la machine pour trouver une interface qui convienne.
not	Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection.
point-to-point	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être de savoir si elle a oui ou non une interface d'un point à un autre.
public-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être de savoir si elle a oui ou non une adresse publiquement routable.
site-local-address	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une adresse associée à à son site-local
subnet-match	Une adresse IP réseau et le nombre de bits dans le préfixe de réseau de l'adresse, sous la forme « / » ; par exemple"192.168.0.0/16".
up	Élément vide indiquant qu'une partie du critère de sélection d'une interface est active ou non.
virtual	Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une interface virtuelle.

- **Configuration des attributs d'une interface**
  Vous pouvez utiliser la fonction de saisie semie-automatique pour saisir la commande au fur et à mesure que vous saisissez, et afin d'exposer les attributs disponibles.

  - **Configurer les attributs d'interface par l'interface CLI**
    Utiliser l'interface CLI pour ajouter de nouvelles interfaces et pour écrire des nouvelles valeurs dans dans les attributs de l'interface.

    a. **Ajouter une nouvelle interface**
    L'opération **add** (ajouter) crée de nouvelles interfaces selon les besoins. Vous pouvez exécuter cette commande **add** à partir de la racine de la session CLI, qui, dans

l'exemple suivant, crée un nouveau titre de nom d'interface *interfacename*, avec une **`inet-address`** déclarée *12.0.0.2*.

```
/interface=interfacename/:add(inet-address=12.0.0.2)
```

b. **Modifier les attributs d'une interface**
L'opération **`write-attribute`** écrit de nouvelles valeurs sur un attribut. L'exemple suivant met à jour la valeur de **`inet-address`** à *12.0.0.8*.

```
/interface=interfacename/:write-attribute(name=inet-address,
value=12.0.0.8)
```

c. **Vérifier les attributs d'interface.**
Confirmer que les valeurs d'attribut ont changé en exécutant l'opération **`read-resource`** accompagnée du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur. Par exemple :

```
[standalone@localhost:9999 interface=public] :read-
resource(include-runtime=true)
```

- **Configurer les attributs d'interface par la console de gestion**

a. **Connectez-vous à la console de management.**
Connectez-vous à la console de gestion de votre instance de serveur autonome ou de domaine géré.

b. **Naviguer dans l'écran d'interfaces**

i. **Naviguer dans l'onglet de configuration.**
Sélectionner **`Configuration`** qui se trouve en haut de l'écran.

ii. **Mode domaine uniquement**
Sélectionner un profil à partir du menu déroulant **`Profile`** qui se situe en haut et à gauche de l'écran.

c. **Sélectionner `Interfaces` à partir du menu de navigation.**
Étendre le menu **`General Configuration`**. Sélectionner l'élément de menu **`Interfaces`** à partir du menu de navigation.

d. **Ajouter une nouvelle interface**

i. Cliquer sur **`Ajouter`**.

ii. Saisir les valeurs qui conviennent pour les **`Name`** (Nom), **`Inet Address`** (Adresse Inet) et **`Address Wildcard`** (Adresse générique).

iii. Cliquer sur le bouton **`Save`**.

e. **Modifier les attributs d'une interface**

i. Sélectionner l'interface que vous souhaitez modifier dans la liste **`Available Interfaces`** et cliquer sur le bouton **`Edit`**.

ii. Saisir les valeurs qui conviennent pour les **Name** (Nom), **Inet Address** (Adresse Inet) et **Address Wildcard** (Adresse générique).

iii. Cliquer sur le bouton **Save**.

Rapporter un bogue

5.2. GROUPES DE LIAISONS DE SOCKETS

5.2.1. Groupes de liaisons de sockets

Les liaisons de sockets et les groupes de liaisons de sockets vous permettent de définir les ports de réseau et leur relation aux interfaces de réseau requises pour votre configuration JBoss EAP 6.

Une liaison de socket est une configuration nommée pour un socket. Les déclarations pour ces configurations nommées se trouvent dans les deux fichiers de configuration **domain.xml** et **standalone.xml**. D'autres sections de la configuration peuvent alors faire référence à ces sockets par leur nom logique, plutôt que d'avoir à inclure tous les détails de la configuration de socket. Cela permet de référencer des configurations de sockets relatives qui peuvent varier sur des machines différentes.

Les liaisons de sockets sont rassemblées sous un groupe de liaisons du sockets. Un groupe de liaisons de sockets est une collection de déclarations de liaisons de sockets qui sont regroupées sous un nom logique. Le groupe peut ensuite être référencé dans l'ensemble de la configuration. Un serveur autonome contient uniquement un de ces groupes, alors qu'une instance managée de domaine peut contenir plusieurs groupes. Vous pouvez créer un groupe de liaison de socket pour chaque groupe de serveurs dans le domaine géré, ou partager un groupe de liaisons de sockets entre plusieurs groupes de serveurs.

Les groupes de noms permettent à des références simplifiées d'être utilisées pour des groupes particuliers de liaisons de sockets lors de la configuration des groupes de serveurs, dans le cas d'un domaine géré. Une autre utilisation courante est pour la configuration et la gestion de plusieurs instances du serveur autonome sur un seul système. Les exemples suivants montrent les groupes de liaison de sockets par défaut dans les fichiers de configuration des instances du domaine et du serveur autonome.

Exemple 5.6. Liaisons de sockets par défaut pour la configuration du serveur autonome.

Les groupes de liaisons de sockets par défaut dans le fichier de configuration **standalone.xml** sont groupés sous **standard-sockets**. Ce groupe est aussi référencé dans l'interface **publique** , en utilisant la même méthodologie de référencement logique.

```
<socket-binding-group name="standard-sockets" default-
interface="public">
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="jacorb" port="3528"/>
    <socket-binding name="jacorb-ssl" port="3529"/>
    <socket-binding name="jmx-connector-registry" port="1090"
interface="management"/>
    <socket-binding name="jmx-connector-server" port="1091"
interface="management"/>
    <socket-binding name="jndi" port="1099"/>
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-throughput" port="5455"/>
```

```
    <socket-binding name="osgi-http" port="8090"
interface="management"/>
    <socket-binding name="remoting" port="4447"/>
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
</socket-binding-group>
```

Exemple 5.7. Liaisons de sockets par défaut pour la configuration du serveur autonome.

Les groupes de liaisons de sockets par défaut dans le fichier de configuration **`domain.xml`** contiennent quatre groupes : **`standard-sockets`**, **`ha-sockets`**, **`full-sockets`** et **`full-ha-sockets`**. Ces groupes sont également référencés dans une interface appelée **`public`**.

```
<socket-binding-groups>
    <socket-binding-group name="standard-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'default' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="ha-sockets" default-interface="public">
        <!-- Needed for server groups using the 'ha' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45700"/>
        <socket-binding name="jgroups-tcp" port="7600"/>
        <socket-binding name="jgroups-tcp-fd" port="57600"/>
        <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45688"/>
        <socket-binding name="jgroups-udp-fd" port="54200"/>
        <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
```

```
    <socket-binding-group name="full-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'full' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jacorb" interface="unsecure" port="3528"/>
        <socket-binding name="jacorb-ssl" interface="unsecure"
port="3529"/>
        <socket-binding name="messaging" port="5445"/>
        <socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
        <socket-binding name="messaging-throughput" port="5455"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="full-ha-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'full-ha' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jacorb" interface="unsecure" port="3528"/>
        <socket-binding name="jacorb-ssl" interface="unsecure"
port="3529"/>
        <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45700"/>
        <socket-binding name="jgroups-tcp" port="7600"/>
        <socket-binding name="jgroups-tcp-fd" port="57600"/>
        <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45688"/>
        <socket-binding name="jgroups-udp-fd" port="54200"/>
        <socket-binding name="messaging" port="5445"/>
        <socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
        <socket-binding name="messaging-throughput" port="5455"/>
        <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
```

```
        </outbound-socket-binding>
    </socket-binding-group>
</socket-binding-groups>
```

Les instances de liaisons de sockets peuvent être créées et modifiées dans les fichiers source **`standalone.xml`** et **`domain.xml`** du répertoire de l'application. La méthode recommandée pour gérer les liaisons consiste à utiliser la console de gestion ou l'interface CLI. Les avantages à utiliser la console de gestion est l'interface utilisateur graphique avec un écran de groupes de liaisons de sockets dédié dans la section **`Configuration générale`**. L'interface CLI propose un API et un flux de travail basés ligne de commande qui permettent le traitement par lots et l'utilisation de scripts aux niveaux supérieurs et inférieurs de la configuration de serveur d'applications. Les deux interfaces permettent la persistance des modifications ou bien leur enregistrement dans la configuration du serveur.

Rapporter un bogue

5.2.2. Configurer les liaisons de sockets

Les liaisons de sockets peuvent être définies dans des groupes de liaisons sockets uniques. Le serveur autonome contient un de ces groupes, le groupe **`standard-sockets`**, et ne peut pas créer de groupes supplémentaires. À la place, vous pouvez créer des fichiers de configuration de serveur autonome alternatif. Pour le domaine géré cependant, vous pouvez créer des groupes de liaisons de sockets et configurer les liaisons de sockets qu'ils contiennent selon vos besoins. Le tableau suivant montre les attributs disponibles pour chaque liaison de sockets.

Tableau 5.2. Attributs de liaisons de sockets

Attribut	Description	Rôle
`name`	Nom logique de la configuration de socket qui doit être utilisée ailleurs dans la configuration.	Requis
`port`	Port de base auquel un socket basé sur cette configuration doit être lié. Notez que les serveurs peuvent être configurés pour substituer cette valeur de base en appliquant une incrémentation ou décrémentation à toutes les valeurs de port.	Requis
`interface`	Nom logique de l'interface à laquelle un socket basé sur cette configuration doit être lié. Si non défini, la valeur de l'attribut **`default-interface`** du groupe de liaison de sockets enveloppant servira.	Option
`multicast-address`	Si le socket doit être utilisé en multi-diffusion, c'est l'adresse multi-diffusion qu'il vous faut.	Option

Attribut	Description	Rôle
multicast-port	Si le socket doit être utilisé en multi-diffusion, c'est le port multi-diffusion qu'il vous faut.	Option
fixed-port	Si défini sur **true**, déclare que la valeur de **port** doit toujours être utilisée par le socket et ne doit pas être remplacée par l'ajout d'un incrément ou d'un décrément.	Option

- **Configurer des liaisons de sockets dans des groupes de liaisons de sockets**
  Sélectionner le CLI ou la console de gestion pour configurer vos liaisons de sockets selon les besoins.
  - **Configurer les liaisons de sockets par l'interface CLI**
    Utiliser l'interface CLI pour configurer les liaisons de sockets.
    a. **Ajouter un nouvelle liaison de sockets**
    Utiliser l'opération **add** (ajouter) pour créer une nouvelle configuration d'adresse si nécessaire. Vous pouvez exécuter cette commande à partir de la racine de la session de CLI, qui, dans l'exemple suivant, crée une nouvelle liaison de sockets intitulée *newsocket*, avec un attribut **port** déclaré comme *1234*. Les exemples s'appliquent à la fois pour la modification des serveurs autonomes et des serveurs gérés sur la liaison de sockets **standard-sockets** comme montré ci-dessous.

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:add(port=1234)
```

    b. **Modifier les attributs de modèle**
    Utiliser l'opération **write-attribute** pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour aider à compléter la chaîne de commande que vous saisissez, et pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur **port** à *2020*

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:write-
attribute(name=port,value=2020)
```

    c. **Confirmer les attributs de modèle**
    Confirmer que les valeurs ont changé en exécutant **read-resource** accompagné du paramètre **include-runtime=true** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:read-resource
```

  - **Configurer les liaisons de sockets par la console de gestion**
    Utiliser la console de gestion pour configurer les liaisons de sockets.

a. **Connectez-vous à la console de management.**
Connectez-vous à la console de gestion de votre domaine géré ou de votre serveur autonome.

b. **Naviguer dans `Configuration`.**
Sélectionner **`Configuration`** qui se trouve en haut de l'écran.

c. **Sélectionner l'élément `Socket Binding` à partir du menu de navigation.**
Étendre le menu **`General Configuration`**. Sélectionner **`Socket Binding`**. Si vous utilisez un domaine géré, sélectionner le groupe désiré dans la liste **`Socket Binding Groups`**.

d. **Ajouter un nouvelle liaison de sockets**

i. Cliquer sur le bouton **`Add`** (ajouter).

ii. Saisir les valeurs qui conviennent pour les **`Name`** (Nom), **`Port`** (Port) et **`Binding Group`** (Groupe de liaisons).

iii. Cliquer sur le bouton **`Save`** pour terminer.

e. **Modifier la liaison de socket**

i. Sélectionner la liaison de sockets de la liste et cliquer sur le bouton **`Edit`**.

ii. Saisir les valeurs qui conviennent pour les **`Name`** (Nom), **`Interface`** (Interface) ou **`Port`** (Port).

iii. Cliquer sur le bouton **`Save`** pour terminer.

Rapporter un bogue

5.2.3. Ports de réseau utilisés par JBoss EAP 6

Les ports utilisés par la configuration JBoss EAP 6 par défaut sont liés à plusieurs facteurs :

- Le fait que vos groupes de serveurs utilisent le groupe de liaisons de sockets par défaut, ou un groupe de liaisons de sockets personnalisé.
- Les exigences de vos déploiements individuels.

NOTE

Un décalage de port numérique peut être configuré pour atténuer les conflits de ports lorsque vous exécutez plusieurs serveurs sur un même serveur physique. Si votre serveur utilise un décalage de port numérique, ajoutez la valeur de décalage au numéro de port par défaut pour le groupe de liaisons de sockets de son groupe de serveurs. Par exemple, si le port HTTP du groupe de liaisons de socket est **`8080`** et si votre serveur utilise un décalage de port de **`100`**, son port HTTP sera **`8180`**.

À moins d'instruction particulière, les ports utilisent le protocole TCP.

Groupes de liaison de sockets par défaut

- `full-ha-sockets`
- `full-sockets`
- `ha-sockets`
- `standard-sockets`

Tableau 5.3. Référence aux groupes de liaisons de sockets par défaut

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
`ajp`	8009		Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges.	Oui	Oui	Oui	Oui
`http`	8080		Le port par défaut des applications déployées.	Oui	Oui	Oui	Oui
`https`	8443		Connexion cryptée-SSL entre les applications déployées et les clients.	Oui	Oui	Oui	Oui
`jacorb`	3528		Services CORBA pour les transactions JTS et autres services dépendants-ORB.	Oui	Oui	Non	Non
`jacorb-ssl`	3529		Services CORBA cryptés-SSL.	Oui	Oui	Non	Non
`jgroups-diagnostics`		7500	Multidiffusion. Utilisée pour découvrir des homologues dans les groupements HA. Non configurable par les interfaces de gestion.	Oui	Non	Oui	Non

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
jgroups-mping		45700	Multidiffusion. Utilisée pour découvrir l'appartenance de groupe d'origine dans un cluster HA.	Oui	Non	Oui	Non
jgroups-tcp	7600		Découverte d'homoplogues unicastes dans les groupements HA avec TCP.	Oui	Non	Oui	Non
jgroups-tcp-fd	57600		Utilisé pour la détection des échecs en TCP.	Oui	Non	Oui	Non
jgroups-udp	55200	45688	Découverte d'homologues unicastes dans les groupements HA avec UDP.	Oui	Non	Oui	Non
jgroups-udp-fd	54200		Utilisé pour la détection des échecs par UDP.	Oui	Non	Oui	Non
messaging	5445		Service JMS.	Oui	Oui	Non	Non
messaging-group			Référencé par la diffusion HornetQ JMS et les groupes Discovery	Oui	Oui	Non	Non
messaging-throughput	5455		Utilisé par JMS Remoting.	Oui	Oui	Non	Non
mod_cluster		23364	Port de multidiffusion pour la communication entre JBoss EAP 6 et l'équilibreur de charges HTTP.	Oui	Non	Oui	Non

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
`osgi-http`	8090		Utilisé par les composants internes qui utilisent le sous-système OSGi. Non configurable par les interfaces de gestion.	Oui	Oui	Oui	Oui
`remoting`	4447		Utilisé pour l'invocation EJB.	Oui	Oui	Oui	Oui
`txn-recovery-environment`	4712		Gestionnaire de recouvrement des transactions JTA.	Oui	Oui	Oui	Oui
`txn-status-manager`	4713		Gestionnaire des transactions JTA / JTS.	Oui	Oui	Oui	Oui

Ports de gestion

En plus des groupes de liaisons de socket, chaque contrôleur hôte ouvre deux ports supplémentaires pour la gestion :

- **`9990`** - Le port de console de gestion web
- **`9999`** - Le port utilisé par la console de gestion et par l'API de gestion.

De plus, si HTTPS est activé pour la console de gestion, 9443 sera également ouvert en tant que valeur de port par défaut.

Rapporter un bogue

5.2.4. Valeurs de décalage des ports pour les groupes de liaison de sockets

Les valeurs de décalage des ports (Port offsets) est un décalage chiffré qui vient s'ajouter aux valeurs de port données par le groupe de liaisons de sockets pour ce serveur. Cela permet à un seul serveur d'hériter les liaisons de sockets du groupe de serveurs auquel il appartient, avec un décalage pour veiller à ce qu'il n'entre pas en conflit avec les autres serveurs du groupe. Par exemple, si le port HTTP du groupe de liaisons de socket est 8080 et si votre serveur utilise un port offset de 100, son port HTTP sera 8180.

Rapporter un bogue

5.2.5. Configurer Port Offset (valeurs de décalage de ports)

- **Configurer Port Offset (valeurs de décalage de ports)**
  Sélectionner l'interface CLI ou la console de gestion pour configurer vos valeurs de décalage de ports.

  - **Configurer Port Offsets par l'interface CLI**
    Sélectionner l'interface CLI pour configurer vos ports offsets.

    a. **Modifier les Ports Offsets**
    Utiliser l'opération **`write-attribut`** pour écrire une nouvelle valeur de référence de port. L'exemple suivant met à jour la valeur du **`socket-binding-port-offset`** de *server-two* à *250*. Ce serveur est membre du groupe hôte local par défaut. Un redémarrage est nécessaire pour que les modifications puissent prendre effet.

    ```
    [domain@localhost:9999 /] /host=master/server-config=server-
    two/:write-attribute(name=socket-binding-port-
    offset,value=250)
    ```

    b. **Confirmer les attributs d'un Port Offset**
    Confirmer que les valeurs ont changé en exécutant **`read-resource`** accompagné du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur.

    ```
    [domain@localhost:9999 /] /host=master/server-config=server-
    two/:read-resource(include-runtime=true)
    ```

  - **Configurer les valeurs de décalage de ports par la console de gestion**
    Sélectionner la console de gestion pour configurer vos valeurs de décalage de ports.

    a. **Connectez-vous à la console de gestion.**
    Connectez-vous à la console de gestion de votre domaine géré.

    b. **Sélectionner l'onglet `Domain`**
    Sélectionner l'onglet **`Domain`** en haut de l'écran.

    c. **Modifier les attributs de Port Offset**

    i. Sélectionner le serveur dans la liste **`Available Server Configurations`** et cliquer sur **`Edit`** en haut de la liste des attributs en dessous.

    ii. Saisir les valeurs que vous désirez dans le champ **`Port Offset`**.

    iii. Cliquer sur le bouton **`Save`** pour terminer.

Rapporter un bogue

5.2.6. Configuration de la taille d'un message à distance

Le sous-système de communication à distance (Remoting) offre la possibilité de limiter la taille des messages pour des protocoles de communication à distance. Vous pouvez définir la taille maximale des messages entrants (**`MAX_INBOUND_MESSAGE_SIZE`**) et la taille maximale des messages sortants (**`MAX_OUTBOUND_MESSAGE_SIZE`**) pour s'assurer que les messages sont reçus et envoyés dans la limite des tailles appropriées.

La configuration de la taille des messages dans des protocoles de communication à distance contribue à une utilisation efficace de la mémoire système et l'empêche d'atteindre un état « out of memory » pendant l'exécution des opérations importantes.

Si l'expéditeur envoie un message qui dépasse la limite maximale admissible (**`MAX_OUTBOUND_MESSAGE_SIZE`**), le serveur lèvera une exception et annulera la transmission des données. Toutefois, la connexion restera ouverte et l'expéditeur pourra choisir de fermer le message si nécessaire.

Si un message reçoit la limite permise maximum (**`MAX_INBOUND_MESSAGE_SIZE`**), le message sera fermé de façon asynchrone avec la connexion restée ouverte.

Rapporter un bogue

5.3. IPV6

5.3.1. Configurer les préférences de JVM Stack d'IPv6 Networking

Résumé

Cette section couvre l'activation du réseau IPv6 de l'installation JBoss EAP 6

Procédure 5.1. Désactiver la propriété IPv4 Stack Java

1. Ouvrir le fichier qui convient à l'installation :
   - **Pour le serveur autonome :**
     Ouvrir ***`EAP_HOME`*`/bin/standalone.conf`**.
   - **Pour un domaine géré :**
     Ouvrir ***`EAP_HOME`*`/bin/domain.conf`**.
2. Modifier la propriété IPv4 Stack Java à false :

   ```
   -Djava.net.preferIPv4Stack=false
   ```

   Par exemple :

   ```
   # Specify options to pass to the Java VM.
   #
   if [ "x$JAVA_OPTS" = "x" ]; then
      JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
   Djava.net.preferIPv4Stack=false
      -Dorg.jboss.resolver.warning=true -
   Dsun.rmi.dgc.client.gcInterval=3600000
      -Dsun.rmi.dgc.server.gcInterval=3600000 -
   Djava.net.preferIPv6Addresses=true"
   fi
   ```

Rapporter un bogue

5.3.2. Configurer les déclarations d'interface du réseautage IPv6

Résumé

Suivre ces étapes de configuration de l'adresse inet d'interface dans IPv6 par défaut:

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »
- Section 3.4.2, « Se conncecter à la console de gestion »

Procédure 5.2. Configurer l'interface du réseautage IPv6

1. Sélectionner **`Configuration`** qui se trouve en haut de l'écran.
2. Étendre le menu **`General Configuration`** et sélectionnez **`Interfaces`**.
3. Sélectionner l'interface dans la liste **`Available Interfaces`**.
4. Cliquer sur **`Edit`** dans la liste d'informations.
5. Définir l'adresse inet à :

```
${jboss.bind.address.management:[ADDRESS]}
```

6. Cliquer sur le bouton **`Save`** pour terminer.
7. Démarrer le serveur à nouveau pour implémenter les changements.

Rapporter un bogue

5.3.3. Configurer les Préférences JVM Stacks des adresses IPv6

Résumé

Cette rubrique couvre la configuration de l'installation de JBoss EAP 6 pour qu'elle préfère les adresses IPv6 dans les fichiers de configuration.

Procédure 5.3. Configuration de l'installation JBoss EAP 6 pour qu'elle préfère les adresses IPv6

1. Ouvrir le fichier qui convient à l'installation :
   - **Pour le serveur autonome :**
     Ouvrir ***`EAP_HOME`*`/bin/standalone.conf`**.
   - **Pour un domaine géré :**
     Ouvrir ***`EAP_HOME`*`/bin/domain.conf`**.
2. Ajouter la propriété Java suivante aux options de la Java VM

```
-Djava.net.preferIPv6Addresses=true
```

   Par exemple :

```
# Specify options to pass to the Java VM.
```

```
#
if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=false
   -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000
   -Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.net.preferIPv6Addresses=true"
fi
```

Rapporter un bogue

CHAPITRE 6. GESTION DES SOURCES DE DONNÉES

6.1. INTRODUCTION

6.1.1. JDBC

L'API JDBC est la norme qui définit comment les bases de données sont accessibles par les applications Java. Une application configure une source de données qui référence un pilote JDBC. Le code de l'application peut alors s'inscrire au pilote, à la place de la base de données. Le pilote convertit le code dans le langage de base de données. Cela signifie que si le pilote qui convient est installé, une application pourra être utilisée sans base de données supportée.

La norme JDBC 4.0 est définie ici : http://jcp.org/en/jsr/detail?id=221.

Pour débuter dans JDBC et avec les sources de données, voir la section JDBC Driver du Guide d'administration et de configuration de JBoss EAP 6.

Rapporter un bogue

6.1.2. Bases de données prises en charge par JBoss EAP 6

Liste des bases de données compatibles JDBC prises en charge par JBoss EAP 6 : https://access.redhat.com/site/articles/111663.

Rapporter un bogue

6.1.3. Types de sources de données

Les deux grands types de ressources sont nommés **`les sources de données`** et **`les sources de données XA`**.

Les sources de données non-XA sont utilisées pour les applications qui n'utilisent pas de transactions, ou les applications qui utilisent des transactions avec une base de données simple.

Les sources de données XA sont utilisées par les applications dont les transactions sont réparties à travers plusieurs bases de données. Les sources de données XA rajoutent un niveau supplémentaire.

Vous n'avez qu'à indiquer le type de source de données quand vous la créez dans la console de gestion ou l'interface CLI.

Rapporter un bogue

6.1.4. L'exemple de source de données

JBoss EAP 6 inclut une base de données H2. Il s'agit d'un système de gestion de bases de données relationnelles léger, qui donne aux développeurs la possibilité de construire des applications rapidement, et qui représente la source de données de référence pour la plate forme.

AVERTISSEMENT

Toutefois, elle ne devrait pas être utilisée dans un environnement de production. C'est une source de données très petite, autonome, qui prend en charge toutes les normes nécessaires pour le test et la création d'applications, mais qui n'est pas fiable ou suffisamment évolutive pour une utilisation en production.

Pour obtenir une liste de sources de données certifiées ou prises en charges, consulter Section 6.1.2, « Bases de données prises en charge par JBoss EAP 6 ».

Rapporter un bogue

6.1.5. Déploiement des fichiers -ds.xml

Dans JBoss EAP 6, les sources de données sont définies comme une ressource du sous-système de serveur. Dans les versions précédentes, un fichier de configuration de source de données **`*-ds.xml`** était requis dans le répertoire de déploiement de la configuration du serveur. Les fichiers **`*-ds.xml`** peuvent toujours être déployés dans JBoss EAP 6, suivant le schéma de sources de données 1.1 disponible sous *Schemas* : http://www.ironjacamar.org/documentation.html.

AVERTISSEMENT

Cette fonctionnalité doit être utilisée pour le développement uniquement. Elle n'est pas conseillée en production car non supportée par les outils de gestion et d'admin de JBoss.

IMPORTANT

Il est obligatoire d'utiliser une référence à une entrée de <driver> (pilote) déjà déployé / défini quand on déploie les fichiers **`*-ds.xml`**.

Rapporter un bogue

6.2. PILOTES JDBC

6.2.1. Installer un pilote JDBC avec la console de gestion

Résumé

Avant que votre application puisse se connecter à une source de données JDBC, vos pilotes JDBC du fournisseur de votre source de données doivent être installés dans un endroit où JBoss EAP 6 puisse les utiliser. JBoss EAP 6 vous permettra de déployer ces pilotes comme tout autre déploiement. Cela signifie que vous pouvez les déployer sur plusieurs serveurs dans un groupe de serveurs, si vous utilisez un domaine géré.

Conditions préalables

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre fournisseur de base de données.

NOTE

Tout pilote conforme à JDBC 4 est automatiquement reconnu et installé dans le système avec son nom et de sa version. Un JAR JDBC utilisant le mécanisme de fournisseur de service Java est identifié. Ces JAR contiennent un fichier-texte nommé META-INF/services/java.sql.Driver qui indique les nom(s) de classes de pilote de ce JAR.

Procédure 6.1. Modifier le JAR du pilote JDBC

Si le JAR de pilote JDBC n'est pas conforme à JDBC 4, il peut être rendu déployable à l'aide de la commande suivante.

1. Modifier ou créer un répertoire temporaire vide.
2. Créer un sous-répertoire META-INF.
3. Créer un sous-répertoire META-INF/services.
4. Créer un fichier META-INF/services/java.sql.Driver qui contienne une ligne indiquant le nom de classe complet du pilote JDBC.
5. Utiliser l'outil de ligne de commande du JAR pour mettre à jour le JAR ainsi :

```
jar \-uf jdbc-driver.jar META-INF/services/java.sql.Driver
```

Procédure 6.2. Déployer le pilote JDBC

1. **Accéder à la console de gestion**
   Section 3.4.2, « Se conncecter à la console de gestion »
2. **Déployez le fichier JAR dans votre serveur ou groupe de serveurs.**
   Si vous utilisez un domaine géré, déployez le fichier JAR dans un groupe de serveurs. Sinon, déployez-le sur votre serveur. Voir Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion ».

Résultat :

Le pilote JDBC est déployé, et est disponible pour vos applications.

Rapporter un bogue

6.2.2. Installer un pilote JDBC comme core module

Conditions préalables

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre base de données fournisseur. Voici les adresses de téléchargement pour le pilote JDBC : Section 6.2.3, « Adresses des téléchargements de pilotes JDBC ».

- En extraire l'archive

Procédure 6.3. Installer un pilote JDBC comme core module

1. Créer une structure de chemin d'accès de fichier sous le répertoire ***EAP_HOME*/modules/**. Ainsi, pour un pilote MySQL JDBC, créer une structure de répertoire comme suit : ***EAP_HOME*/modules/com/mysql/main/**.

2. Copier le pilote JDBC dans le sous-répertoire **main/**.

3. Dans le sous-répertoire **main/**, créer un fichier **module.xml** ressemblant à l'exemple qui se trouve Section 7.1.1, « Modules ». Le **module** XSD est défini dans le fichier ***EAP_HOME*/docs/schema/module-1_2.xsd**.

4. Démarrer le serveur.

5. Démarrer l'interface CLI.

6. Exécuter la commande CLI pour ajouter le module de pilote JDBC à la configuration du serveur.

   La commande que vous avez choisie dépend du nombre de classe listées dans le fichier **/META-INF/services/java.sql.Driver** qui se situe dans le JAR du pilote JDBC. Par exemple, le fichier **/META-INF/services/java.sql.Driver** du JAR JDBC MySQL 5.1.20 liste deux classes :

   - **com.mysql.jdbc.Driver**
   - **com.mysql.fabric.jdbc.FabricMySQLDriver**

   Quand il n'y a plus d'une entrée, vous devez également spécifier le nom de la classe de pilote. Sinon, vous provoquerez une erreur semblable à ce qui suit :

   ```
   JBAS014749: Operation handler failed: Service jboss.jdbc-
   driver.mysql is already registered
   ```

   - Exécuter la commande CLI pour les JAR JDBC qui contiennent une entrée de classe.

     ```
     /subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-
     name=DRIVER_NAME,driver-module-name=MODULE_NAME,driver-xa-
     datasource-class-name=XA_DATASOURCE_CLASS_NAME)
     ```

     Exemple 6.1. Exemple de commande CLI en mode autonome pour les JAR JDBC ayant une classe de pilote.

     ```
     /subsystem=datasources/jdbc-driver=mysql:add(driver-
     name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
     class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
     ```

     Exemple 6.2. Exemple de commande CLI en mode de domaine pour les JAR JDBC ayant une classe de pilote.

```
/profile=ha/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
```

- Exécuter la commande CLI pour les JAR JDBC qui contiennent plusieurs entrées de classe.

```
/subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-
name=DRIVER_NAME,driver-module-name=MODULE_NAME,driver-xa-
datasource-class-name=XA_DATASOURCE_CLASS_NAME, driver-class-
name=DRIVER_CLASS_NAME)
```

Exemple 6.3. Exemple de commande CLI en mode autonome pour les JAR JDBC ayant plusieurs classes de pilote.

```
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource,
driver-class-name=com.mysql.jdbc.Driver)
```

Exemple 6.4. Exemple de commande CLI en mode de domaine pour les JAR JDBC ayant plusieurs classes de pilote.

```
/profile=ha/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource,
driver-class-name=com.mysql.jdbc.Driver)
```

Résultat

Le pilote JDBC est maintenant installé et configuré comme core module. Il est maintenant prêt à être référencé par les sources de données d'application.

Rapporter un bogue

6.2.3. Adresses des téléchargements de pilotes JDBC

Le tableau suivant donne les adresses de téléchargement standard pour les pilotes JDBC de bases de données utilisées avec JBoss EAP 6. Ces liens pointent vers des sites tiers qui ne sont pas contrôlés ou surveillés activement par Red Hat. Pour les pilotes les plus à jour de votre base de données, consultez le site Web et la documentation de votre fournisseur de base de données.

Tableau 6.1. Adresses des téléchargements du pilote JDBC

Fournisseur	Adresse de téléchargement
MySQL	http://www.mysql.com/products/connector/
PostgreSQL	http://jdbc.postgresql.org/

Fournisseur	Adresse de téléchargement
Oracle	http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html
IBM	http://www-306.ibm.com/software/data/db2/java/
Sybase	http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect
Microsoft	http://msdn.microsoft.com/data/jdbc/

Rapporter un bogue

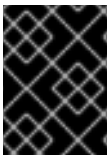
6.2.4. Accès aux classes spécifiques à un fournisseur

Résumé

Cette section couvre les étapes à suivre pour utiliser les classes spécifiques JDBC. Cela est utile quand une application a besoin d'utiliser une fonctionnalité spécifique à un fournisseur ne faisant pas partie de l'API JDBC.

AVERTISSEMENT

Ceci est une procédure d'utilisation avancée. Seules les applications qui ont besoin d'une fonctionnalité que l'on ne peut pas trouver dans l'API JDBC doivent implémenter cette procédure.



IMPORTANT

Ce processus est requis pour le mécanisme de ré-authentification et pour accéder aux classes spécifiques au fournisseur.

IMPORTANT

Suivre les directives de l'API spécifiques au fournisseur de près, car la connexion est contrôlée par le conteneur IronJacamar.

Pré-requis

- Section 6.2.2, « Installer un pilote JDBC comme core module ».

Procédure 6.4. Ajouter une dépendance à l'application

- **Configurer le fichier `MANIFEST.MF`**

a. Ouvrir le fichier **`META-INF/MANIFEST.MF`** de l'application dans un éditeur de texte.

b. Ajouter une dépendance au module JDBC et sauvegarder le fichier.

```
Dépendences : MODULE_NAME
```

Exemple 6.5. Exemple de dépendance

```
Dépendences : com.mysql
```

- a. **Créer un fichier `jboss-deployment-structure.xml`**
  Créer un fichier **`jboss-deployment-structure.xml`** dans le dossier **`META-INF/`** ou **`WEB-INF`** de l'application.

Exemple 6.6. Exemple de fichier `jboss-deployment-structure.xml`

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="com.mysql" />
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

Exemple 6.7. Accèder à l'API spécifique du fournisseur

L'exemple ci-dessous accède à l'API MySQL.

```
import java.sql.Connection;
import org.jboss.jca.adapters.jdbc.WrappedConnection;

  Connection c = ds.getConnection();
  WrappedConnection wc = (WrappedConnection)c;
  com.mysql.jdbc.Connection mc = wc.getUnderlyingConnection();
```

Rapporter un bogue

6.3. SOURCES DE DONNÉES NON-XA

6.3.1. Créer une source de données non-XA avec les interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour créer une source de données non-XA, en utilisant la console de gestion ou l'interface CLI.

Pré-requis

- Le serveur JBoss EAP 6 doit être en cours d'exécution.

NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles aurait créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

Procédure 6.5. Créer une source de données en utilisant l'interface CLI ou la console de gestion

- **L'interface CLI**

  a. Lancer le CLI et connectez-vous à votre serveur.

  b. Exécuter la commande suivante pour créer une source de données non-XA, et configurer les variables comme il se doit :

  ```
  data-source add --name=DATASOURCE_NAME --jndi-name=JNDI_NAME --driver-name=DRIVER_NAME  --connection-url=CONNECTION_URL
  ```

  c. Activer la source de données :

  ```
  data-source enable --name=DATASOURCE_NAME
  ```

- **Console de management**

  a. Connectez-vous à la console de gestion.

  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**

  i. Sélectionner **`Configuration`** qui se trouve en haut de la console.

  ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

  iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.

  iv. Sélectionner **Datasources** à partir du menu à gauche de la console.

  c. **Créer une nouvelle source de données**

  i. Cliquer sur **`Add`** qui se trouve en haut du panneau **`Datasources`**.

  ii. Saisir les attributs de la nouvelle source de données de l'assistant **`Create Datasource`** et appuyez sur **`Next`**.

  iii. Saisir les informations sur le pilote JDBC dans l'assistant **`Create Datasource`** et cliquer sur **`Next`** pour continuer.

  iv. Saisir les paramètres de connexion dans l'assistant **`Create Datasource`**.

  v. Cliquer sur le bouton **`Test Connection`** pour tester la connexion à la ressource de données et vérifier que les paramètres de configuration sont corrects.

vi. Cliquer sur **Done** pour terminer.

Résultat

La source de données non-Xa a été ajoutée au serveur. Elle est maintenant visible dans le fichier `standalone.xml` ou le fichier `domain.xml`, ainsi que dans les interfaces de gestion.

Rapporter un bogue

6.3.2. Modifier une source de données non-XA par les interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour modifier une source de données non-XA, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 ».

NOTE

Les sources de données non-XA peuvent être intégrées dans les transactions JTA. Pour intégrer la source de données dans JTA, veillez à ce que le paramètre `jta` soit défini à `true`.

Procédure 6.6. Modifier une source de données non-XA

- **Interface CLI**
  a. Section 3.5.2, « Lancement de l'interface CLI ».
  b. Utiliser la commande `write-attribute` pour configurer un attribut de source de données :

```
/subsystem=datasources/data-source=DATASOURCE_NAME:write-attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

  c. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

- **Console de gestion**
  a. Section 3.4.2, « Se conncecter à la console de gestion ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**
    i. Sélectionner `Configuration` qui se trouve en haut de la console.
    ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.
    iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.

iv. Sélectionner **Datasources** à partir du menu déroulant.

c. **Modifier la source de données**

i. Sélectionner une source de données dans la liste **`Available Datasources`**. Les attributs de source de données sont affichés ci-dessous.

ii. Cliquer sur **`Edit`** pour modifier les attributs de la source de données.

iii. Cliquer sur le bouton **`Save`** pour terminer.

Résultat

La source de données non-Xa a été configurée. Elle est maintenant visible dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données. voir : Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion ».
- Pour supprimer la source de données, voir : Section 6.3.3, « Supprimer une source de données non-XA par les interfaces de gestion ».

Rapporter un bogue

6.3.3. Supprimer une source de données non-XA par les interfaces de gestion

Résumé

Ce sujet couvre les étapes requises pour supprimer une source de données non-XA de JBoss EAP 6, en utilisant la console de gestion ou l'interface CLI.

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 ».

Procédure 6.7. Supprimer une source de données non-XA

- **Interface CLI**

a. Section 3.5.2, « Lancement de l'interface CLI ».

b. Exécuter la commande suivante pour supprimer une source de données non-XA :

```
data-source remove --name=DATASOURCE_NAME
```

- **Console de gestion**

a. Section 3.4.2, « Se conncecter à la console de gestion ».

b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**

i. Sélectionner **`Configuration`** qui se trouve en haut de la console.

ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.

iv. Sélectionner **Datasources**.

c. Sélectionner la source de données enregistrée à supprimer, et cliquer sur **Remove**.

Résultat

La source de données non-XA a été supprimée dans le serveur.

- Pour créer une nouvelle source de données, voir : Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion ».

Rapporter un bogue

6.4. SOURCES DE DONNÉES XA

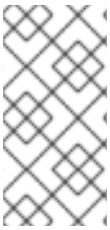
6.4.1. Créer une source de données XA par les interfaces de gestion

Conditions préalables :

- Section 2.1.1, « Démarrer JBoss EAP 6 »

Résumé

Cette section explique les étapes à suivre pour créer une source de données XA, en utilisant la console de gestion ou l'interface CLI.



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

Procédure 6.8. Créer une source de données XA en utilisant l'interface CLI ou la console de gestion

- **L'interface CLI**

a. Section 3.5.2, « Lancement de l'interface CLI ».

b. Exécuter la commande suivante pour créer une source de données XA, et configurer les variables comme il se doit:

```
xa-data-source add --name=XA_DATASOURCE_NAME --jndi-
name=JNDI_NAME --driver-name=DRIVER_NAME --xa-datasource-
class=XA_DATASOURCE_CLASS
```

c. **Configurer les propriétés de la source de données XA**

i. **Définir le nom du serveur**
Exécuter la commande suivante pour configurer le nom du serveur de l'hôte :

```
/subsystem=datasources/xa-data-
source=XA_DATASOURCE_NAME/xa-datasource-
properties=ServerName:add(value=HOSTNAME)
```

ii. **Définir le nom de la base de données**
Exécuter la commande suivante pour configurer le nom de la base de données :

```
/subsystem=datasources/xa-data-
source=XA_DATASOURCE_NAME/xa-datasource-
properties=DatabaseName:add(value=DATABASE_NAME)
```

d. Activer la source de données :

```
xa-data-source enable --name=XA_DATASOURCE_NAME
```

- **Console de management**

a. Section 3.4.2, « Se conncecter à la console de gestion ».

b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**

i. Sélectionner **`Configuration`** qui se trouve en haut de la console.

ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.

iv. Sélectionner **Datasources**.

c. Sélectionner l'onglet **`XA Datasource`**.

d. **Créer une nouvelle source de données XA**

i. Cliquer sur **`Add`**.

ii. Saisir les attributs de la nouvelle source de données XA de l'assistant **`Create XA Datasource`** et cliquez sur **`Next`**.

iii. Saisir les informations sur le pilote JDBC dans l'assistant **`Create XA Datasource`** et cliquer sur **`Next`**.

iv. Saisir les propriétés XA et cliquer sur **`Next`**.

v. Saisir les paramètres de connexion dans l'assistant **`Create XA Datasource`**.

vi. Cliquer sur le bouton **`Test Connection`** pour tester la connexion à la ressource de données XA et vérifier que les paramètres de configuration soient corrects.

vii. Cliquer sur **`Done`** pour terminer.

Résultat

La source de données XA a été ajoutée au serveur. Elle est maintenant visible dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

Voir également :

- Section 6.4.2, « Modifier une base de données XA par les interfaces de gestion »
- Section 6.4.3, « Supprimer une base de données XA par les interfaces de gestion »

Rapporter un bogue

6.4.2. Modifier une base de données XA par les interfaces de gestion

Résumé

Cette section explique les étapes à suivre pour modifier une source de données XA, en utilisant la console de gestion ou l'interface CLI.

Pré-requis

- Section 2.1.1, « Démarrer JBoss EAP 6 ».

Procédure 6.9. Modifier une source de données XA en utilisant l'interface CLI ou la console de gestion

- **L'interface CLI**
  a. Section 3.5.2, « Lancement de l'interface CLI ».
  b. **Configurer les attributs de source de données XA**
  Utiliser la commande **`write-attribute`** pour configurer un attribut de source de données :

```
/subsystem=datasources/xa-data-source=XA_DATASOURCE_NAME:write-attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

  c. **Configurer les propriétés de la source de données XA**
  Exécuter la commande suivante pour configurer une sous-ressource de source de données XA :

```
/subsystem=datasources/xa-data-source=DATASOURCE_NAME/xa-datasource-properties=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

  d. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

- **Console de gestion**
  a. Section 3.4.2, « Se conncecter à la console de gestion ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**
    i. Sélectionner **`Configuration`** qui se trouve en haut de la console.
    ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector** .

iv. Sélectionner **Datasources**.

c. Sélectionner l'onglet **`XA Datasource`**.

d. **Modifier la source de données**

i. Sélectionner la source de données XA qui convient à partir de la liste **`Available XA Datasources`**. Les attributs de la source de données XA sont affichés dans le panneau **`Attributes`** ci-dessous.

ii. Sélectionner le bouton **`Edit`** pour modifier les attributs de la source de données.

iii. Modifier les attributs de la source de données XA et sélectionner le bouton **`Save`** quand c'est fait.

Résultat

La source de données XA a été configurée. Les changements sont maintenant visibles dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données, voir : Section 6.4.1, « Créer une source de données XA par les interfaces de gestion ».
- Pour supprimer la source de données, voir : Section 6.4.3, « Supprimer une base de données XA par les interfaces de gestion ».

Rapporter un bogue

6.4.3. Supprimer une base de données XA par les interfaces de gestion

Résumé

Ce sujet couvre les étapes requises pour supprimer une source de données XA de JBoss EAP 6, en utilisant la console de gestion ou l'interface CLI.

Pré-requis

- Section 2.1.1, « Démarrer JBoss EAP 6 ».

Procédure 6.10. Supprimer une source de données XA en utilisant l'interface CLI ou la console de gestion

- **Management CLI**

a. Section 3.5.2, « Lancement de l'interface CLI ».

b. Exécuter la commande suivante pour supprimer une source de données :

```
xa-data-source remove --name=XA_DATASOURCE_NAME
```

- **Console de gestion**

a. Section 3.4.2, « Se conncecter à la console de gestion ».

b. **Naviguez dans le panneau `Datasources` qui se trouve dans la console de gestion**

   i. Sélectionner **`Configuration`** qui se trouve en haut de la console.

   ii. En mode de domaine uniquement, sélectionner un profil à partir du menu déroulant qui se trouve en haut et à gauche.

   iii. Étendre le menu **Subsystems** qui se trouve à gauche de la console, puis étendre le menu **Connector**.

   iv. Sélectionner **Datasources**.

c. Sélectionner l'onglet **`XA Datasource`**.

d. Sélectionner la source de données XA enregistrée à supprimer, et cliquer sur le bouton **`Remove`** pour supprimer la source de données XA de façon permanente.

Résultat

La source de données XA a été supprimée dans le serveur.

- Pour ajouter une nouvelle source de données, voir : Section 6.4.1, « Créer une source de données XA par les interfaces de gestion ».

Rapporter un bogue

6.4.4. XA Recovery

6.4.4.1. Les modules de recouvrement XA

Chaque ressource XA a besoin d'un module de recouvrement associé avec sa configuration. Le module de recouvrement doit étendre la classe **`com.arjuna.ats.jta.recovery.XAResourceRecovery`**.

JBoss EAP 6 fournit des modules de recouvrement pour les ressources JDBC et JMS XA. Pour ces types de ressources, les modules de recouvrement sont automatiquement enregistrés. Si vous avez besoin d'utiliser un module personnalisé, vous pouvez l'enregistrer dans votre source de données.

Rapporter un bogue

6.4.4.2. Configurer les modules de recouvrement

Pour la plupart des ressources JDBC et JMS, le module de recouvrement est automatiquement associé à la ressource. Dans de tels cas, vous aurez uniquement besoin de configurer les options qui permettent au module de recouvrement de se connecter à vos ressources afin de procéder au recouvrement.

Pour les ressources personnalisées qui ne sont ni JDBC, ni JMS, contacter Red Hat Global Services pour obtenir des informations sur les configurations qui sont prises en charge.

Chacun de ces attributs de configuration peut être défini lors de la création de la source de données, ou par la suite. Vous pouvez les définir en utilisant la console de gestion basée web ou l'interface CLI par ligne de commande. Se référer à Section 6.4.1, « Créer une source de données XA par les interfaces de gestion » and Section 6.4.2, « Modifier une base de données XA par les interfaces de gestion » pour obtenir des informations sur la façon de configurer des sources de données XA.

Voir les tableaux suivants pour les attributs de configuration de sources de données, et pour obtenir des informations sur la configuration spécifique à certains fournisseurs de bases de données.

Tableau 6.2. Attributs de configuration générale

Attribut	Description
recovery-username	Le nom d'utilisateur qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-password	Le mot de passe qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-security-domain	Le domaine de sécurité qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement.
recovery-plugin-class-name	Si vous devez utiliser un module de recouvrement personnalisé, définissez cet attribut au nom complet de classe du module. Le module doit étendre la classe **`com.arjuna.ats.jta.recovery.XAResourceRecovery`**.
recovery-plugin-properties	Si vous utilisez un module de récupération personnalisée qui requiert des propriétés à définir, définissez cet attribut à la liste de paires *key=value* séparée par des virgules pour les propriétés.

Informations de configuration spécifiques au fournisseur

Oracle

Si la source de données Oracle n'est pas configurée correctement, vous apercevrez sans doute les erreurs suivantes dans votre sortie de journalisation :

```
WARN  [com.arjuna.ats.jta.logging.loggerI18N]
[com.arjuna.ats.internal.jta.recovery.xarecovery1] Local
XARecoveryModule.xaRecovery  got XA exception
javax.transaction.xa.XAException, XAException.XAER_RMERR
```

Pour résoudre cette erreur, veillez à ce que l'utilisateur Oracle configuré dans ***`recovery-username`*** ait bien accès aux tableaux utiles au recouvrement. L'énoncé SQL suivant affiche les attributions d'instances correctes Oracle 11g ou Oracle 10g R2 corrigées pour le bogue 5945463 d'Oracle.

```
GRANT SELECT ON sys.dba_pending_transactions TO recovery-username;
GRANT SELECT ON sys.pending_trans$ TO recovery-username;
GRANT SELECT ON sys.dba_2pc_pending TO recovery-username;
GRANT EXECUTE ON sys.dbms_xa TO recovery-username;
```

Si vous utilisez une version Oracle 11g antérieure à 11g, modifier l'énoncé final **EXECUTE** ainsi:

```
GRANT EXECUTE ON sys.dbms_system TO recovery-username;
```

PostgreSQL

Voir la documentation PostgreSQL pour obtenir des instructions sur la façon d'activer des transactions (ex. XA) préparées. La version 8.4-701 du pilote JDBC de PostgreSQL a un bogue dans **`org.postgresql.xa.PGXAConnection`** qui empêche le recouvrement dans certaines situations. Cela a été résolu dans les nouvelles versions.

MySQL

Basé sur http://bugs.mysql.com/bug.php?id=12161, le recouvrement de transactions XA ne fonctionnait pas dans certaines versions de MySQL 5. Cela a été corrigé dans MySQL 6.1. VOir l'URL du bogue ou la documentation MySQL pour obtenir davantage d'informations.

IBM DB2

IBM DB2 s'attend à ce que la méthode **`XAResource.recover`** soit appelée uniquement pendant la phase de resynchronisation désignée qui se produit lorsque le serveur d'applications est redémarré après un accident ou une panne. Il s'agit d'une décision de conception pour l'implémentation de DB2, qui est en dehors du dessein de cette documentation.

Sybase

Sybase s'attend à ce que les transactions XA soient activées sur la base de données. Sans configuration correcte de la base de données, les transactions XA ne fonctionneront pas. **`enable xact coordination`** active ou désactive les services de coordination de transaction Adaptive Server. Lorsque ce paramètre est activé, Adaptive Server garantit que les mises à jour de données Adaptive Server distantes soient validées ou annulées avec la transaction d'origine. Pour permettre la coordination de transaction, utilisez :

```
sp_configure 'enable xact coordination', 1
```

.

Rapporter un bogue

6.5. SÉCURITÉ DES BASES DE DONNÉES

6.5.1. Sécurité des bases de données

La meilleure solution de sécuriser les bases de données est soit l'utilisation des domaines de sécurité, soit les mots de passe. Vous trouverez un exemple de chacun ci-dessous. Pour plus d'informations, consulter :

- Domaines de sécurité : Section 11.6.1, « Les domaines de sécurité ».
- Mots de passe : Section 11.13.1, « Sécurisation des chaînes confidentielles de fichiers en texte clair ».

Exemple 6.8. Exemple de domaine de sécurité

```
<security>
   <security-domain>mySecurityDomain</security-domain>
</security>
```

Exemple 6.9. Exemple de mots de passe

```
<security>
  <user-name>admin</user-name>
```

```
<password>${VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0LWE4M
mEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3ZhdWx0}</password>
</security>
```

Rapporter un bogue

6.6. CONFIGURATION DES SOURCES DE DONNÉES

6.6.1. Paramètres de source de données

Tableau 6.3. Les paramètres de source de données communs aux sources XA ou non-XA

Paramètre	Description
jndi-name	Le nom JNDI unique pour la source de données.
pool-name	Le nom du pool de gestion de la source de données.
enabled	Indique si la source de données est activée.
use-java-context	Indique si on doit relier la source de données au JNDI global.
spy	Activer la fonctionnalité **spy** sur la couche JDBC. Cela journalisera tout le trafic JDBC dans la source de données. Notez que la catégorie de journalisation **jboss.jdbc.spy** doit également être définie au niveau **DEBUG** dans le sous-système de journalisation.
use-ccm	Activer le gestionnaire de connexion cache.
new-connection-sql	Un énoncé SQL qui exécute quand la connexion est ajoutée au pool de connexion.
transaction-isolation	Un parmi ces enoncés : • TRANSACTION_READ_UNCOMMITTED • TRANSACTION_READ_COMMITTED • TRANSACTION_REPEATABLE_READ • TRANSACTION_SERIALIZABLE • TRANSACTION_NONE
url-delimiter	Le délimiteur d'URLs d'une connexion url pour les bases de données clusterisées HA (Haute disponibilité).

Paramètre	Description
url-selector-strategy-class-name	Une classe qui implémente l'interface **`org.jboss.jca.adapters.jdbc.URLSelectorStrategy`**.
sécurité	Contient des éléments dépendants en tant que paramètres de sécurité. Voir Tableau 6.8, « Paramètres de sécurité ».
validation	Contient des éléments dépendants en tant que paramètres de validation. Voir Tableau 6.9, « Paramètres de validation ».
timeout	Contient des éléments dépendants en tant que paramètres de timeout. Voir Tableau 6.10, « Paramètres de timeout ».
énoncé	Contient des éléments dépendants en tant que paramètres d'énoncés. Voir Tableau 6.11, « Paramètres d'instruction ».

Tableau 6.4. Paramètres de source de données non-xa

Paramètre	Description
jta	Active l'intégration JTA pour les sources de données non-XA. Ne s'applique pas aux sources de données XA.
connection-url	L'URL de connexion du pilote JDBC.
driver-class	Le nom complet de la classe de pilote JDBC.
connection-property	Propriétés de connexion arbitraires passées à la méthode **`Driver.connect(url,props)`**. Chaque connection-property indique une paire name/value. Le nom de la propriété provient du nom, et la valeur provient du contenu de l'élément.
pool	Contient des éléments dépendants en tant que paramètres de pooling. Voir Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA ».

Tableau 6.5. Paramètres de source de données XA

Paramètre	Description

Paramètre	Description
xa-datasource-property	Une propriété pour assigner la classe d'implémentation **`XADataSource`**. Spécifié par *name=value*. Si une méthode setter existe, dans le format **`set`*`Name`***, la propriété sera définie en appelant une méthode setter sous le format **`set`*`Name`*`(`*`value`*`)`**.
xa-datasource-class	Le nom complet de la classe d'implémentation de **`javax.sql.XADataSource`**.
pilote	Unique référence au module de chargeur de classe qui contient le pilote JDBC. Le format accepté est *driverName#majorVersion.minorVersion*.
xa-pool	Contient des éléments dépendants en tant que paramètres de pooling. Voir Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA » et Tableau 6.7, « Paramètres du pool XA »
recouvrement	Contient des éléments dépendants en tant que paramètres de recouvrement. Voir Tableau 6.12, « Paramètres de recouvrement ».

Tableau 6.6. Les paramètres de pool communs aux sources XA ou non-XA

Paramètre	Description
min-pool-size	Le nombre minimum de connexions contenues par un pool.
max-pool-size	Le nombre maximum de connexions qu'un pool peut contenir
Pré-remplissage	Indique si l'on doit essayer de pré-remplir un pool de connexions. Un élément vide indique une valeur **`true`**. La valeur par défaut est **`false`**.
use-strict-min	Indique si la taille du pool est stricte. **`false`** par défaut.
flush-strategy	Indique si le pool est vidé en cas d'erreur. Les valeurs acceptées sont : • FailingConnectionOnly • IdleConnections • EntirePool La valeur par défaut est **`FailingConnectionOnly`**.

Paramètre	Description
allow-multiple-users	Indique si plusieurs utilisateurs pourront avoir accès à la source de données par la méthode getConnection(user, password), et si les types de pools internes ont une influence sur ce comportement.

Tableau 6.7. Paramètres du pool XA

Paramètre	Description
is-same-rm-override	Indique si la classe **`javax.transaction.xa.XAResource.isSameRM(XAResource)`** retourne **`true`** ou **`false`**.
entrelacement	Indique si on doit activer l'entrelacement pour les fabriques de connexion XA.
no-tx-separate-pools	Indique si on doit créer des sous-répertoires distincts pour chaque contexte. Cela est nécessaire pour les sources de données Oracle, qui ne permettent pas aux connexions XA d'être utilisées à la fois à l'intérieur et à l'extérieur d'une transaction de JTA Utiliser cette option entraînera la multiplication par deux de la taille du pool **`max-pool-size`**, car en fait, deux pools seront créés.
pad-xid	Indique si on doit remplir le Xid.
wrap-xa-resource	Indique si on doit inclure XAResource dans une instance **`org.jboss.tm.XAResourceWrapper`**.

Tableau 6.8. Paramètres de sécurité

Paramètre	Description
user-name	Le nom d'utilisation pour créer une nouvelle connexion.
password	Le mot de passe à utiliser pour créer une nouvelle connexion
security-domain	Contient le nom d'un gestionnaire de sécurité JAAS, qui gère l'authentification. Ce nom correspond à l'attribut application-policy/name de la configuration de connexion JAAS.

Paramètre	Description
reauth-plugin	Définit un plugin d'authentification à nouveau pour la réauthentification de connexions physiques.

Tableau 6.9. Paramètres de validation

Paramètre	Description
valid-connection-checker	Une mise en œuvre d'interface **`org.jboss.jca.adaptors.jdbc.ValidConnectionChecker`** qui fournit une méthode **`SQLException.isValidConnection(Connection e)`** pour valider une connexion. Une exception signifie que la connexion est détruite. Cela remplace le paramètre **`check-valid-connection-sql`** s'il est présent.
check-valid-connection-sql	Un énoncé SQL pour vérifier la validité d'un pool de connexion. Peut être appelé quand une connexion gérée est tirée d'un pool.
validate-on-match	Indique si la validation des niveaux connexion est exécutée lorsqu'une fabrique de connexions essaie de correspondre à une connexion gérée pour un ensemble donné. Indiquer "true" pour **`validate-on-match`** n'est pas normalement fait en conjonction avec "true" pour **`background-validation`**. **`Validate-on-match`** est utile quand un client doit avoir une connexion vallidée avant utilisation. Ce paramètre est à false par défaut.
background-validation	Indique que les connexions sont validées sur un thread d'arrière-plan. La validation de l'arrière-plan (background validation) est une optimisation de performances lorsque non utilisé avec **`validate-on-match`**. Si **`Validate-on-match`** est sur true, l'utilisation de **`background-validation`** pourrait entraîner des contrôles redondants. La validation de l'arrière-plan pourrait provoquer une mauvaise connexion (une connexion qui irait mal entre le moment de l'analyse de validation et avant d'être donnée au client), l'application cliente doit par conséquent tenir compte de cette possibilité.
background-validation-millis	La durée, en millisecondes, pendant laquelle la validation d'arrière-plan exécute.
use-fast-fail	Si défini sur true, échoue une allocation de connexion lors de la première tentative, si la connexion est non valide. La valeur par défaut est **`false`**.

Paramètre	Description
stale-connection-checker	Une instance de **`org.jboss.jca.adapters.jdbc.StaleConnectionChecker`** qui produit une méthode booléenne **`isStaleConnection(SQLException e)`**. Si cette méthode renvoie un **`true`**, l'exception sera contenue dans **`org.jboss.jca.adapters.jdbc.StaleConnectionException`**, qui correspond à une sous-classe de **`SQLException`**.
exception-sorter	Une instance de **`org.jboss.jca.adapters.jdbc.ExceptionSorter`** qui fournit une méthode booléenne **`isExceptionFatal(SQLException e)`**. Cette méthode valide le fait qu'une exception soit envoyée à toutes les instances d'un **`javax.resource.spi.ConnectionEventListener`** en tant que message **`connectionErrorOccurred`**.

Tableau 6.10. Paramètres de timeout

Paramètre	Description
use-try-lock	Utiliser **`tryLock()`** au lieu de **`lock()`**. Vous essayerez ainsi d'obtenir un verrou pour le nombre de secondes configurées, avant le timeout, au lieu d'échouer immédiatement quand le verrou n'est pas disponible. La valeur par défaut est de **`60`** secondes. Par exemple, pour définir un timeout de 5 minutes, définir **`<use-try-lock>`**300**`</use-try-lock>`**.
blocking-timeout-millis	La durée maximale, en millisecondes, de blocage lorsque vous attendez une connexion. Après ce délai, une exception sera levée. Cela bloque uniquement pendant que vous attendez un permis de connexion et ne lève pas d'exception si la création d'une nouvelle connexion prend beaucoup de temps. Par défaut, à 30000, ce qui correspond à 30 secondes.
idle-timeout-minutes	La durée maximale, en minutes, avant qu'une connexion inactive soit fermée. La durée maximale réelle dépend de la durée d'analyse de l'idleRemover, qui correspond à la moitié du plus petit **`idle-timeout-minutes`** de n'importe quel pool.

Paramètre	Description
set-tx-query-timeout	Indique si on doit définir le timeout d'interrogation par rapport au temps qui reste avant le timeout de transaction. Si aucune transaction n'existe, on utilisera le timeout de recherche qui a été configuré. La valeur par défaut est **`false`**.
query-timeout	Timeout pour les recherches, en secondes. La valeur par défaut est « no timeout ».
allocation-retry	Le nombre de tentatives de connexions avant d'envoyer une connexion. La valeur par défaut est **`0`**, pour qu'une exception puisse être envoyée à la première défaillance.
allocation-retry-wait-millis	Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est 5000, soit 5 secondes.
xa-resource-timeout	Si la valeur est non nulle, elle passe à la méthode **`XAResource.setTransactionTimeout`**.

Tableau 6.11. Paramètres d'instruction

Paramètre	Description
track-statements	Indique si l'on doit vérifier les énoncés non fermés lorsqu'une connexion est renvoyée à un pool ou qu'un énoncé est retourné dans le cache de l'énoncé préparé. Si défini à false, les énoncés ne seront pas suivis. **Valeurs valides** • **`true`** : les énoncés et les ensembles de résultats sont suivis, et un avertissement sera émis s'ils ne sont pas fermés. • **`false`** : ni les énoncés, ni les ensembles de résultats ne seront suivis. • **`nowarn`** : les énoncés seront suivies, mais il n'y a aucun avertissement. Valeur par défaut.
prepared-statement-cache-size	Le nombre d'énoncés préparés par connexion, dans le cache LRU (Least Recently Used / Utilisé le moins souvent récemment).

Paramètre	Description
share-prepared-statements	Indique si le fait de demander un même énoncé deux fois sans le fermer utilise le même énoncé préparé sous-jacente. La valeur par défaut est **`false`**.

Tableau 6.12. Paramètres de recouvrement

Paramètre	Description
recover-credential	Une paire nom d'utilisateur/mot de passe ou domaine de sécurité pour le recouvrement.
recover-plugin	Une mise en œuvre de la classe **`org.jboss.jca.core.spi.recoveryRecoveryPlugin`** à utiliser pour le recouvrement.

Rapporter un bogue

6.6.2. Les URL de connexion de sources de données

Tableau 6.13. Les URL de connexion de sources de données

Source de données	URL de connexion
PostgreSQL	jdbc:postgresql://*SERVER_NAME*:*PORT*/*DATABASE_NAME*
MySQL	jdbc:mysql://*SERVER_NAME*:*PORT*/*DATABASE_NAME*
Oracle	jdbc:oracle:thin:*@ORACLE_HOST*:*PORT*:*ORACLE_SID*
IBM DB2	jdbc:db2://*SERVER_NAME*:*PORT*/*DATABASE_NAME*
Microsoft SQLServer	jdbc:microsoft:sqlserver://*SERVER_NAME*:*PORT*;DatabaseName=*DATABASE_NAME*

Rapporter un bogue

6.6.3. Extensions de sources de données

Les déploiements de sources de données peuvent utiliser plusieurs extensions de l'adaptateur de ressources JDBC pour améliorer la validation de la connexion, et vérifier si l'exception doit rétablir la connexion. Ces extensions sont les suivantes :

Tableau 6.14. Extensions de sources de données

Extension de sources de données	Paramètre de configuration	Description
org.jboss.jca.adapters.jdbc.spi.ExceptionSorter	<exception-sorter>	Vérifie si SQLException est fatale pour la connexion sur laquelle l'exception a été lancée
org.jboss.jca.adapters.jdbc.spi.StaleConnection	<stale-connection-checker>	Wraps stale SQLExceptions in a **`org.jboss.jca.adapters.jdbc.StaleConnectionException`**
org.jboss.jca.adapters.jdbc.spi.ValidConnection	<valid-connection-checker>	Vérifie si une connexion est valide pour être utilisée par l'application

JBoss EAP 6 comprend également des implémentations de ces extensions pour plusieurs bases de données prises en charge.

Implémentations des extensions

Générique

- org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.novendor.NullStaleConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.novendor.NullValidConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker

PostgreSQL

- org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker

MySQL

- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLReplicationValidConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker

IBM DB2

- org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker

Sybase

- org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectionChecker

Microsoft SQLServer

- org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker

Oracle

- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker

Rapporter un bogue

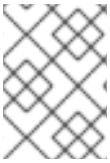
6.6.4. Voir les statistiques de sources de données

Vous pourrez voir des statistiques de sources de données définies dans **`jdbc`** et **`pool`** qui utilisent des versions modifiées des commandes ci-dessous :

Procédure 6.11.

- ```
  /subsystem=datasources/data-source=ExampleDS/statistics=jdbc:read-resource(include-runtime=true)
  ```

  ```
  /subsystem=datasources/data-source=ExampleDS/statistics=pool:read-resource(include-runtime=true)
  ```



NOTE

Veillez à spécifier l'argument ***`include-runtime=true`*** car tous les statistiques sont en runtime uniquement et la valeur par défaut est **`false`**.

Rapporter un bogue

6.6.5. Statistiques de bases de données

Statistiques principaux

Le tableau suivant montre une liste des statistiques principaux de sources de données pris en charge :

Tableau 6.15. Statistiques principaux

Nom	Description
`ActiveCount`	Le nombre de connexions actives. Chacune de ces connexions est soit utilisée par une application, ou disponible via pool
`AvailableCount`	Le nombre de connexions disponibles dans le pool
  ```

Nom	Description
AverageBlockingTime	Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes.
AverageCreationTime	Le durée moyenne passée à créer une connexion. La valeur est en millisecondes.
CreatedCount	Le nombre de connexions créées.
DestroyedCount	Le nombre de connexions détruites.
InUseCount	Le nombre de connexions actuellement utilisées.
MaxCreationTime	La durée maximum pour créer une connexion. La valeur est en millisecondes.
MaxUsedCount	Le nombre maximum de connexions utilisées
MaxWaitCount	Le nombre maximum de requêtes attendant une connexion en même temps.
MaxWaitTime	Le durée maximum à attendre un verrou exclusif sur le pool.
TimedOut	Le nombre de connexions expirées
TotalBlockingTime	Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes.
TotalCreationTime	La durée passée à créer des connexions. La valeur est en millisecondes.
WaitCount	Le nombre de requêtes en attente de connexion.

Statistiques JDBC

Le tableau suivant montre une liste des statistiques JDBC de sources de données pris en charge :

Tableau 6.16. Statistiques JDBC

Nom	Description
PreparedStatementCacheAccessCount	Le nombre de fois qu'un cache d'énoncé a été accédé.
PreparedStatementCacheAddCount	Le nombre d'énoncés ajoutés au cache de l'énoncé.

Nom	Description
PreparedStatementCacheCurrentSize	Le nombre d'énoncés préparés et que l'on peut appeler, actuellement mis en cache dans un cache d'énoncé.
PreparedStatementCacheDeleteCount	Le nombre d'énoncés rejetés du cache.
PreparedStatementCacheHitCount	Le nombre de fois que des énoncés de cache ont été utilisés.
PreparedStatementCacheMissCount	Le nombre de fois qu'une requête d'énoncé a pu être réglée par un énoncé d'un cache.

Vous pouvez activer les statistiques **Core** et **JDBC** en utilisant des versions appropriément modifiées des commandes suivantes :

- ```
  /subsystem=datasources/data-source=ExampleDS/statistics=pool:write-attribute(name=statistics-enabled,value=true)
  ```

  ```
  /subsystem=datasources/data-source=ExampleDS/statistics=jdbc:write-attribute(name=statistics-enabled,value=true)
  ```

Rapporter un bogue

6.7. EXEMPLES DE SOURCES DE DONNÉES

6.7.1. L'exemple de source de données PostgreSQL

Exemple 6.10.

L'exemple ci-dessous est une configuration de source de données PostgreSQL. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/PostgresDS" pool-name="PostgresDS">
    <connection-
url>jdbc:postgresql://localhost:5432/postgresdb</connection-url>
    <driver>postgresql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker"></valid-connection-checker>
```
```

```
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données PostgreSQL ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.2. Exemple de source de données PostgreSQL XA

Exemple 6.11.

L'exemple ci-dessous est une configuration de source de données PostgreSQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/PostgresXADS" pool-
name="PostgresXADS">
    <driver>postgresql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">5432</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">postgresdb</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker">
      </valid-connection-checker>
```

```
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter">
      </exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données PostgreSQL XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.3. Exemple de source de données MySQL

Exemple 6.12.

L'exemple ci-dessous est une configuration de source de données MySQL. La source de données a été activée, un utilisateur a été ajouté, et des options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/MySqlDS" pool-name="MySqlDS">
    <connection-url>jdbc:mysql://mysql-
localhost:3306/jbossdb</connection-url>
    <driver>mysql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"
></exception-sorter>
```

```
    </validation>
  </datasource>
  <drivers>
    <driver name="mysql" module="com.mysql">
      <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données MySQL ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.4. Exemple de source de données MySQL XA

Exemple 6.13.

L'exemple ci-dessous est une configuration de source de données MySQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/MysqlXADS" pool-
name="MysqlXADS">
  <driver>mysql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mysqldb</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"
></exception-sorter>
    </validation>
  </xa-datasource>
```

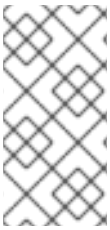
```
  <drivers>
    <driver name="mysql" module="com.mysql">
      <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données MySQL XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.5. L'exemple de source de données Oracle



NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

Exemple 6.14.

L'exemple ci-dessous est une configuration de source de données Oracle. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/OracleDS" pool-name="OracleDS">
    <connection-url>jdbc:oracle:thin:@localhost:1521:XE</connection-
url>
    <driver>oracle</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio
nChecker"></valid-connection-checker>
      <stale-connection-checker class-
```

```
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="oracle" module="com.oracle">
      <xa-datasource-
class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Oracle ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.6. Exemple de source de données d'Oracle XA

NOTE

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

IMPORTANT

Les paramètres de configuration doivent être appliqués pour un utilisateur qui accède à une source de données Oracle XA pour que le recouvrement XA fonctionne correctement. La valeur **user** est une valeur définie par l'utilisateur pour pouvoir se connecter à partir de JBoss à Oracle :

- GRANT SELECT ON sys.dba_pending_transactions TO user;
- GRANT SELECT ON sys.pending_trans$ TO user;
- GRANT SELECT ON sys.dba_2pc_pending TO user;
- GRANT EXECUTE ON sys.dbms_xa TO user; (If using Oracle 10g R2 (patched) or Oracle 11g)

  OU

  GRANT EXECUTE ON sys.dbms_system TO user; (If using an unpatched Oracle version prior to 11g)

Exemple 6.15.

L'exemple ci-dessous est une configuration de source de données Oracle XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/XAOracleDS" pool-name="XAOracleDS">
    <driver>oracle</driver>
    <xa-datasource-property name="URL">jdbc:oracle:oci8:@tc</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <no-tx-separate-pools />
    </xa-pool>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio
nChecker"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="oracle" module="com.oracle">
      <xa-datasource-
class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
```

```
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données Oracle XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.7. Exemple de source de données Microsoft SQLServer

Exemple 6.16.

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/MSSQLDS" pool-name="MSSQLDS">
    <connection-
url>jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=MyDatabase</c
onnection-url>
    <driver>sqlserver</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionC
hecker"></valid-connection-checker>
    </validation>
  </datasource>
  <drivers>
    <driver name="sqlserver" module="com.microsoft">
      <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données Microsoft SQLServer ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.8. Exemple de source de données Microsoft SQLServer XA

Exemple 6.17.

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/MSSQLXADS" pool-name="MSSQLXADS">
    <driver>sqlserver</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mssqldb</xa-datasource-
property>
    <xa-datasource-property name="SelectMethod">cursor</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionC
hecker"></valid-connection-checker>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="sqlserver" module="com.microsoft">
      <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Microsoft SQLServer XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.9. Exemple de source de données IBM DB2

Exemple 6.18.

L'exemple ci-dessous est une configuration de source de données IBM DB2. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/DB2DS" pool-name="DB2DS">
    <connection-url>jdbc:db2:ibmdb2db</connection-url>
    <driver>ibmdb2</driver>
    <pool>
      <min-pool-size>0</min-pool-size>
      <max-pool-size>50</max-pool-size>
    </pool>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></e
xception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="ibmdb2" module="com.ibm">
      <xa-datasource-class>com.ibm.db2.jdbc.DB2XADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données IBM DB2 ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.10. Exemple de source de données IBM DB2 XA

Exemple 6.19.

L'exemple ci-dessous est une configuration de source de données IBM DB2 XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/DB2XADS" pool-name="DB2XADS">
    <driver>ibmdb2</driver>
    <xa-datasource-property name="DatabaseName">ibmdb2db</xa-
datasource-property>
    <xa-datasource-property name="ServerName">hostname</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">port</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></e
xception-sorter>
    </validation>
    <recovery>
      <recover-plugin class-
name="org.jboss.jca.core.recovery.ConfigurableRecoveryPlugin">
```

```
        <config-property name="EnableIsValid">false</config-property>
        <config-property name="IsValidOverride">false</config-property>
        <config-property name="EnableClose">false</config-property>
      </recover-plugin>
    </recovery>
  </xa-datasource>
  <drivers>
    <driver name="ibmdb2" module="com.ibm">
      <xa-datasource-class>com.ibm.db2.jcc.DB2XADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données IBM DB2 XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc4.jar"/>
    <resource-root path="db2jcc_license_cisuz.jar"/>
    <resource-root path="db2jcc_license_cu.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.11. L'exemple de source de données Sybase

Exemple 6.20.

L'exemple ci-dessous est une configuration de source de données Sybase. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/SybaseDB" pool-name="SybaseDB"
enabled="true">
    <connection-url>jdbc:sybase:Tds:localhost:5000/DATABASE?
JCONNECT_VERSION=6</connection-url>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
nChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
r"></exception-sorter>
```

```
    </validation>
  </datasource>
  <drivers>
    <driver name="sybase" module="com.sybase">
      <datasource-
class>com.sybase.jdbc4.jdbc.SybDataSource</datasource-class>
      <xa-datasource-class>com.sybase.jdbc4.jdbc.SybXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Sybase ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

6.7.12. L'exemple de source de données Sybase

Exemple 6.21.

L'exemple ci-dessous est une configuration de source de données Sybase XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/SybaseXADS" pool-
name="SybaseXADS" enabled="true">
    <xa-datasource-property name="NetworkProtocol">Tds</xa-datasource-
property>
    <xa-datasource-property name="ServerName">myserver</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">4100</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mydatabase</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <background-validation>true</background-validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
nChecker"></valid-connection-checker>
```

```
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="sybase" module="com.sybase">
      <datasource-
class>com.sybase.jdbc4.jdbc.SybDataSource</datasource-class>
      <xa-datasource-class>com.sybase.jdbc4.jdbc.SybXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Sybase XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Rapporter un bogue

CHAPITRE 7. CONFIGURATION DES MODULES

7.1. INTRODUCTION

7.1.1. Modules

Un module est un regroupement logique des classes utilisées pour le chargement de classes et pour la gestion de dépendances. JBoss EAP 6 identifie deux types de modules, parfois appelés modules statiques et dynamiques. Cependant, la seule différence entre les deux est la façon dont ils sont emballés. Tous les modules offrent les mêmes caractéristiques.

Modules statiques

Les modules statiques sont prédéfinis dans le répertoire ***EAP_HOME*/modules/** du serveur d'applications. Chaque sous-répertoire représente un module et contient un un fichier de configuration (**module.xml**) et tout fichier JAR requis. Le nom du module est défini dans le fichier **module.xml**. Tous les API fournis par le serveur de l'application sont des modules statiques, y compris les API Java EE, et les autres API comme JBoss Logging.

Exemple 7.1. Exemple de fichier module.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.15.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Le nom du module, **com.mysql**, doit correspondre à la structure du répertoire du module.

Les modules fournis dans les distributions JBoss EAP se trouvent dans un répertoire **system** se trouvant lui-même dans le répertoire ***JBOSS_HOME*/modules**. Cela les rend séparés de tout module fourni par une tierce partie.

Tout produit mis en couche fourni par Red Hat, se superposant sur JBoss EAP 6.1 ou version supérieure installera également leurs modules dans le répertoire **system**.

La création de modules statiques personnalisés peut être utile si plusieurs applications sont déployées sur un même serveur utilisant les mêmes bibliothèques de tierce partie. Au lieu d'un regroupement de ces bibliothèques pour chaque application, un module contenant ces bibliothèques peut être créé et installé par l'administrateur JBoss. Les applications peuvent ensuite déclarer une dépendance explicite sur les modules statiques personnalisés.

Les utilisateurs doivent s'assurer que les modules personnalisés soient installés dans le répertoire ***JBOSS_HOME*/modules**, en utilisant un répertoire par couche de modules. Cela garantit que les versions personnalisées de modules qui existent déjà dans le répertoire **system** soient bien chargées à la place des versions fournies. Ainsi, les modules personnalisés utilisateur auront la priorité sur les modules fournis par le système.

Si vous utilisez la variable d'environnement **JBOSS_MODULE_PATH** pour changer les emplacements où JBoss EAP cherche les modules, le produit ira chercher dans une structure de sous-répertoire **system** dans un des emplacements spécifiés. Une structure de sous-répertoire **system** doit exister quelquepart dans les emplacements spécifiés dans **JBOSS_MODULE_PATH**.

Modules dynamiques

Les modules dynamiques sont créés et chargés par le serveur d'application pour chaque déploiement JAR ou WAR (ou sous-déploiement d'un EAR). Le nom d'un module dynamique est dérivé du nom de l'archive déployée. Comme les déploiements sont chargés sous forme de modules, ils peuvent configurer des dépendances et peuvent être utilisés comme dépendances par d'autres déploiements.

Les modules ne sont chargés qu'en fonction des besoins. Cela a généralement lieu quand une application est déployée avec des dépendances implicites ou explicites.

Rapporter un bogue

7.1.2. Modules globaux

Un module global est un module que JBoss EAP 6 fournit comme dépendance pour chaque application. Chaque module peut être composé en l'ajoutant à la liste du serveur d'applications des modules globaux. Il n'est nul besoin de faire des changements au module.

Rapporter un bogue

7.1.3. Les dépendances de modules

Une dépendance de module est une déclaration qui indique qu'un module a besoin des classes d'un autre module pour pouvoir fonctionner. Les modules peuvent déclarer leurs dépendances sur un certain nombre d'autres modules. Quand le serveur d'applications charge un module, le chargeur de classes de module traite les dépendances de ce module et ajoute les classes de chaque dépendance à son chemin de classe. Si une dépendance particulière est introuvable, le module ne pourra pas charger.

Les applications déployées (JAR et WAR) sont chargées sous forme de modules dynamiques et utilisent des dépendances pour accéder aux API fournies par JBoss EAP 6.

Il y a deux types de dépendances : explicite et implicite.

Les dépendances explicites sont déclarées dans la configuration par le développeur. Les modules statiques peuvent déclarer des dépendances dans le fichier modules.xml. Les modules dynamiques peuvent avoir des dépendances déclarées dans les descripteurs de déploiement MANIFEST.MF ou jboss-deployment-structure.xml.

Les dépendances explicites peuvent être spécifiées comme étant optionnelles. Une erreur de chargement de dépendance optionnelle n'entraînera pas l'annulation d'un chargement de module. Cependant, si la dépendance est rendue disponible par la suite, elle ne sera PAS ajoutée au chemin de classe du module. Les dépendances doivent être rendues disponibles quand le module est chargé.

Des dépendances implicites sont ajoutées automatiquement par le serveur d'applications quand certaines conditions ou meta-données se trouvent dans un déploiement. Les API Java EE 6 fournies avec JBoss EAP 6 sont des exemples de modules ajoutés par détection de dépendances implicites dans les déploiements.

Les déploiements peuvent également être configurés de façon à exclure des dépendances implicites

particulières. Il vous faut pour cela le fichier de déploiement jboss-deployment-structure.xml. C'est normalement le cas quand une application empaquète une version spécifique de bibliothèque que le serveur d'applications tentera d'ajouter comme dépendance implicite.

Un chemin de classe de module ne contient que ses propres classes et celles de ses dépendances immédiates. Un module n'est pas en mesure d'accéder aux classes des dépendances. Cependant, un module peut indiquer quand une dépendance explicite est exportée. Une dépendance exportée est fournie à tout module qui dépend du module qui l'exporte.

Exemple 7.2. Les dépendances de module

Le Module A dépend du Module B et le Module B dépend su Module C. Le Module A peut accéder aux classes du Module B, et le Module B peut accéder aux classes du Module C. Le Module C ne peut pas accéder aux classes du Module C à moins que :

- Le Module A déclare une dépendance explicite sur le Module C, ou bien
- Le Module B exporte ses dépendances sur le Module C.

Rapporter un bogue

7.1.4. Isolement du chargeur de classes d'un sous-déploiement

Chaque sous-déploiement d'EAR (Archive Enterprise) est un module dynamique possédant son propre chargeur de classe. Par défaut, un sous-déploiement peut accéder aux ressources d'autres sous-déploiements.

Si un sous-déploiement ne doit pas accéder aux ressources d'autres sous-déploiements (une isolation de sous-déploiement est alors requise), alors cela pourra être activé.

Rapporter un bogue

7.2. DÉSACTIVER L'ISOLEMENT DE MODULE DE SOUS-DÉPLOIEMENT POUR TOUS LES DÉPLOIEMENTS

Cette tâche montre aux administrateurs du serveur comment désactiver l'isolement du module de sous-déploiement dans le serveur d'applications. Cela affecte tous les déploiements.

AVERTISSEMENT

Cette tâche requiert que vous éditiez les fichiers de configuration XML du serveur. Le serveur doit être arrêté avant cela. Ce n'est que temporaire car les outils administratifs de version finale supporteront ce type de configuration.

1. **Arrêter le serveur**
   Mettre en halte le serveur JBoss EAP 6.

2. **Ouvrir le fichier de configuration du serveur**

Ouvrir le fichier de configuration du serveur dans un éditeur de texte

Ce fichier sera différent pour un domaine géré ou un serveur autonome. De plus, des emplacements et des noms de fichiers non-défauts peuvent être utilisés. Les fichiers de configuration par défaut sont **`domain/configuration/domain.xml`** et **`standalone/configuration/standalone.xml`** pour les domaines gérés et les serveurs autonomes respectivement.

3. **Chercher la configuration de sous-système EE**
   Chercher la configuration de sous-système EE dans le fichier de configuration. L'élément **`<profile>`** du fichier de configuration contient plusieurs éléments du sous-système. L'élément du sous-système EE a comme espace-nom **`urn:jboss:domain:ee:1.1`**.

   ```
   <profile>

      ...

      <subsystem xmlns="urn:jboss:domain:ee:1.1" />

      ...
   ```

   La configuration par défaut a une balise en fermeture automatique unique mais une configuration personnalisée peut avoir des balises d'ouverture ou de fermeture distinctes (éventuellement avec d'autres éléments à l'intérieur) comme ceci :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.1" ></subsystem>
   ```

4. **Remplacer les balises en fermeture automatique si nécessaire**
   Si l'élément de sous-système EE est une balise en fermeture automatique unique, remplacez-la par les balises d'ouverture ou de fermeture qui conviennent ainsi :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.1" ></subsystem>
   ```

5. **Ajouter l'élément ear-deployments-isolated**
   Ajouter l'élément **`ear-subdeployments-isolated`** comme dépendant de l'élément du sous-système EE et ajouter le contenu de **`false`** comme suit :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.1" ><ear-subdeployments-
   isolated>false</ear-subdeployments-isolated></subsystem>
   ```

6. **Démarrer le serveur**
   Lancer à nouveau le serveur JBoss EAP 6 pour qu'il commence à exécuter avec la nouvelle configuration.

Résultat :

Le serveur va maintenant exécuter avec l'isolement de module de sous-déploiement désactivé pour tous les déploiements.

Rapporter un bogue

7.3. AJOUTER UN MODULE À TOUS LES DÉPLOIEMENTS

Cette tâche montre comment les administrateurs JBoss peuvent définir une liste de modules globaux.

Conditions préalables

1. Vous devez connaître le nom des modules qui ont été ajoutés comme modules globaux. Voir Section 7.5.1, « Les modules inclus » pour obtenir la liste des modules statiques inclus dans JBoss EAP 6. Si le module est dans un autre déploiement, voir Section 7.5.2, « Nommage de modules dynamiques » pour déterminer le nom du module.

Procédure 7.1. Ajouter un module à la liste des modules globaux

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion »
2. Naviguez dans le panneau **`EE Subsystem`**.
   a. Sélectionner **`Configuration`** qui se trouve en haut de la console.
   b. **Mode Domaine uniquement**
      i. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
   c. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
   d. Sélectionner **Container** → **EE** à partir du menu à gauche de la console.
3. Cliquer sur **`Add`** dans la section **`Subsystem Defaults`**. La boîte de dialogue **`Create Module`** apparaîtra.
4. Saisir alors le nom du module et le slot de module, en option.
5. Cliquez sur **`Enregistrer`** pour ajouter le nouveau module global, ou bien cliquer sur **`Annuler`** pour annuler.
   - Si vous cliquez sur **`Enregistrer`**, la boîte de dialogue va se fermer et le module spécifié sera ajouté à la liste des modules globaux.
   - Si vous cliquez sur le bouton **`Cancel`**, la boîte de dialogue se fermera et il n'y aura aucun changement.

Résultat

Les modules ajoutés à la liste des modules globaux seront ajoutés en tant que dépendances à chaque déploiement.

Rapporter un bogue

7.4. DÉFINIR UN RÉPERTOIRE DE MODULES JBOSS EXTERNE

Résumé

Par défaut, JBoss EAP recherche les modules dans le répertoire ***`EAP_HOME`*`/modules/`**. Vous pouvez demander à JBoss EAP de regarder dans un ou plusieurs modules répertoires externes en définissant une variable d'environnement **`JBOSS_MODULEPATH`** ou en définissant la variable dans le fichier de configuration de démarrage. Cette section décrit les deux méthodes.

Procédure 7.2. Définissez la variable d'environnement JBOSS_MODULEPATH

- Pour sépécifier un ou plusieurs répertoires de module externes, définir la variable d'environnement **JBOSS_MODULEPATH**.

  Dans Linux, utiliser les deux-points pour délimiter une liste de répertoires. Par exemple :

  ```
  export
  JBOSS_MODULEPATH=EAP_HOME/modules/:/home/username/external/modules/d
  irectory/
  ```

  Dans Linux, utiliser un point-virgule pour délimiter une liste de répertoires. Par exemple :

  ```
  SET JBOSS_MODULEPATH=EAP_HOME\modules\;D:\JBoss-Modules\
  ```

Procédure 7.3. Définissez la variable JBOSS_MODULEPATH dans le fichier de configuration de démarrage.

- Si vous choisissez de ne pas définir une variable d'environnement globale, vous pouvez définir la variable **JBOSS_MODULEPATH** dans le fichier de configuration de démarrage de JBoss EAP. Si vous exécutez dans un serveur autonome, il s'agira du fichier ***EAP_HOME*/bin/standalone.conf**. Si le serveur exécute dans dans un domaine géré, il s'agira du fichier ***EAP_HOME*/bin/domain.conf**.

  Vous trouverez ci-dessous un exemple de la commande qui définit la variable **JBOSS_MODULEPATH** dans le fichier **standalone.conf**

  ```
  JBOSS_MODULEPATH="EAP_HOME/modules/:/home/username/external/modules/
  directory/"
  ```

Rapporter un bogue

7.5. RÉFÉRENCE

7.5.1. Les modules inclus

Un tableau énumérant les modules JBoss EAP 6 inclus et indiquant s'ils sont pris en charge se trouve dans le portail clients à https://access.redhat.com/articles/1122333.

Rapporter un bogue

7.5.2. Nommage de modules dynamiques

Tous les déploiements sont chargés en tant que modules par JBoss EAP 6 et sont nommés en fonction des conventions suivantes :

1. Les déploiements des fichiers WAR et JAR sont nommés selon le format suivant :

   ```
   deployment.DEPLOYMENT_NAME
   ```

   Par exemple, **inventory.war** et **store.jar** auront les mêmes noms de module que **deployment.inventory.war** et **deployment.store.jar** respectivement.

2. Les sous-déploiements des archives Enterprise sont nommés selon le format suivant :

```
deployment.EAR_NAME.SUBDEPLOYMENT_NAME
```

Ainsi, le sous-déploiement **reports.war**, qui se trouve dans l'archive enterprise **accounts.ear**, aura le nom de module du **deployment.accounts.ear.reports.war**.

Rapporter un bogue

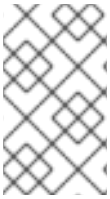
CHAPITRE 8. JSVC

8.1. INTRODUCTION

8.1.1. Jsvc

Jsvc est un ensemble de bibliothèques et d'applications qui permettent aux applications Java exécutées sur UNIX ou sur des plateforme style UNIX en arrière-plan. Jsvc permet à une application d'effectuer des opérations en tant qu'utilisateur privilégié, puis de changer identité en tant qu'utilisateur non privilégié.

Jsvc utilise trois processus : un processus de lanceur, un processus de contrôleur et un processus contrôlé. Le processus contrôlé est également le principal thread Java. Si la JVM plante le processus du contrôleur, il redémarrera dans les 60 secondes. Jsvc est un processus de démon et pour JBoss EAP 6, il doit être lancé par un utilisateur privilégié.



NOTE

Jsvc est peut être utilisé sur Red Hat Enterprise Linux, Solaris et HP-UX. Pour une fonctionnalité similaire dans Microsoft Windows, voir **`prunsrv.exe`** dans **`Native Utilities for Windows Server`** disponibles depuis le portail clients de Red Hat.

Rapporter un bogue

8.1.2. Démarrer et arrêter JBoss EAP par Jsvc

Les instruction de démarrage et d'arrêt de JBoss EAP par Jsvc varient, selon le mode opérationnel : autonome ou domaine. Sachez que si JBoss EAP est exécuté en mode de domaine, Jsvc ne gérera que le processus de contrôleur de domaine. Quelle que soit la commande que vous utilisez pour démarrer JBoss EAP par Jsvc, elle doit être gérée par un utilisateur privilégié.

Conditions préalables

- Si JBoss EAP était installé par la méthode Zip :
  - Installez le package *Native Utilities* de votre système d'exploitation, disponible à partir du portail clients de Red Hat *Install Native Components and Native Utilities (Zip, Installer)* dans *Installation Guide*.
  - Créez le compte d'utilisateur sous lequel s'exécute l'instance de JBoss EAP 6. Le compte utilisé pour démarrer et arrêter le serveur doit posséder un accès lecture et écriture dans le répertoire où JBoss EAP a été installé.
- Si JBoss EAP était installé par la méthode RPM, installez le package *apache-commons-daemon-jsvc-eap6*. Voir *Install Native Components and Native Utilities (RPM Installation)* dans le guide *Installation Guide*.

Les commandes suivantes sont utilisées pour démarrer et arrêter JBoss EAP en mode autonome ou domaine. Notez que les emplacements de fichiers sont différents selon la méthode utilisée pour installer Jsvc dans JBoss EAP 6. Utilisez les tableaux ci-dessous pour déterminer les fichiers à utiliser pour résoudre les variables des commandes.

Mode autonome

Les instructions suivantes sont utilisées pour démarrer ou pour stopper JBoss EAP en mode autonome.

Tableau 8.1. Emplacements des fichiers Jsvc pour les installations zip - Mode autonome

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	${eap-installation-location}/jboss-eap-${version}
JSVC-BIN	EAP_HOME/modules/system/layers/base/native/sbin/jsvc
JSVC-JAR	EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar
CONF-DIR	EAP_HOME/standalone/configuration
LOG-DIR	EAP_HOME/standalone/log

Tableau 8.2. Emplacements des fichiers Jsvc pour les installations RPM - Mode autonome

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	/usr/share/jbossas
JSVC-BIN	/usr/bin/jsvc-eap6/jsvc
JSVC-JAR	EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar
CONF-DIR	/etc/jbossas/standalone
LOG-DIR	/var/log/jbossas/standalone

Démarrez JBoss en serveur autonome

- ```
  JSVC_BIN \
   -outfile LOG_DIR/jsvc.out.log   \
   -errfile LOG_DIR/jsvc.err.log   \
   -pidfile LOG_DIR/jsvc.pid  \
   -user jboss \
   -D[Standalone] -XX:+UseCompressedOops -Xms1303m \
   -Xmx1303m -XX:MaxPermSize=256m \
   -Djava.net.preferIPv4Stack=true
   -Djboss.modules.system.pkgs=org.jboss.byteman \
   -Djava.awt.headless=true \
   -Dorg.jboss.boot.log.file=LOG_DIR/server.log \
   -Dlogging.configuration=file:CONF_DIR/logging.properties \
   -Djboss.modules.policy-permissions \
   -cp EAP_HOME/jboss-modules.jar:JSVC_JAR \
   -Djboss.home.dir=EAP_HOME \
   -Djboss.server.base.dir=EAP_HOME/standalone   \
   @org.jboss.modules.Main -start-method main \
  ```
  ```

```
 -mp EAP_HOME/modules \
 -jaxpmodule javax.xml.jaxp-provider \
 org.jboss.as.standalone
```

Stopper JBoss en serveur autonome

- ```
  JSVC_BIN \
   -stop \
   -outfile LOG_DIR/jsvc.out.log   \
   -errfile LOG_DIR/jsvc.err.log   \
   -pidfile LOG_DIR/jsvc.pid  \
   -user jboss \
   -D[Standalone] -XX:+UseCompressedOops -Xms1303m \
   -Xmx1303m -XX:MaxPermSize=256m \
   -Djava.net.preferIPv4Stack=true \
   -Djboss.modules.system.pkgs=org.jboss.byteman \
   -Djava.awt.headless=true \
   -Dorg.jboss.boot.log.file=LOG_DIR/server.log \
   -Dlogging.configuration=file:CONF_DIR/logging.properties \
   -Djboss.modules.policy-permissions \
   -cp EAP_HOME/jboss-modules.jar:JSVC_JAR \
   -Djboss.home.dir=EAP_HOME \
   -Djboss.server.base.dir=EAP_HOME/standalone   \
  @org.jboss.modules.Main -start-method main \
   -mp EAP_HOME/modules \
   -jaxpmodule javax.xml.jaxp-provider \
   org.jboss.as.standalone
  ```

Mode Domaine

Les instructions suivantes sont utilisées pour démarrer ou pour stopper JBoss EAP en mode autonome. Notez qu'en mode de domaine, vous devrez remplacer la variable *JAVA_HOME* par le répertoire d'accueil de Java.

Tableau 8.3. Emplacements des fichiers Jsvc pour les installations zip - Mode de domaine

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	${eap-installation-location}/jboss-eap-${version}
JSVC-BIN	EAP_HOME/modules/system/layers/base/native/sbin/jsvc
JSVC-JAR	EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar
CONF-DIR	EAP_HOME/domain/configuration
LOG-DIR	EAP_HOME/domain/log

Tableau 8.4. Emplacements des fichiers Jsvc pour les installations RPM - Mode de domaine
  ```

Référence Fichier en Instructions	Emplacement fichier
EAP-HOME	/usr/share/jbossas
JSVC-BIN	/usr/bin/jsvc-eap6/jsvc
JSVC-JAR	EAP_HOME/modules/system/layers/base/native/sbin/commons-daemon.jar
CONF-DIR	/etc/jbossas/domain
LOG-DIR	/var/log/jbossas/domain

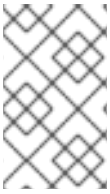
Démarrez JBoss EAP en mode de domaine

- ```
  JSVC_BIN \
   -outfile LOG_DIR/jsvc.out.log   \
   -errfile LOG_DIR/jsvc.err.log   \
   -pidfile LOG_DIR/jsvc.pid  \
   -user jboss \
   -nodetach -D"[Process Controller]" -server -Xms64m \
   -Xmx512m -XX:MaxPermSize=256m \
   -Djava.net.preferIPv4Stack=true  \
   -Djboss.modules.system.pkgs=org.jboss.byteman \
   -Djava.awt.headless=true  \
   -Dorg.jboss.boot.log.file=LOG_DIR/process-controller.log \
   -Dlogging.configuration=file:CONF_DIR/logging.properties \
   -Djboss.modules.policy-permissions \
   -cp "EAP_HOME/jboss-modules.jar:JSVC_JAR" \
   org.apache.commons.daemon.support.DaemonWrapper \
   -start org.jboss.modules.Main -start-method main \
   -mp EAP_HOME/modules org.jboss.as.process-controller \
   -jboss-home EAP_HOME -jvm $JAVA_HOME/bin/java \
   -mp EAP_HOME/modules -- \
   -Dorg.jboss.boot.log.file=LOG_DIR/host-controller.log \
   -Dlogging.configuration=file:CONF_DIR/logging.properties \
   -Djboss.modules.policy-permissions \
   -server -Xms64m -Xmx512m -XX:MaxPermSize=256m \
   -Djava.net.preferIPv4Stack=true \
   -Djboss.modules.system.pkgs=org.jboss.byteman \
   -Djava.awt.headless=true -- -default-jvm $JAVA_HOME/bin/java
  ```

Stopper JBoss EAP en mode de domaine

- ```
  JSVC_BIN \
   -stop \
   -outfile LOG_DIR/jsvc.out.log   \
   -errfile LOG_DIR/jsvc.err.log   \
   -pidfile LOG_DIR/jsvc.pid  \
   -user jboss \
   -nodetach -D"[Process Controller]" -server -Xms64m \
  ```

```
-Xmx512m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true  \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true  \
-Dorg.jboss.boot.log.file=LOG_DIR/process-controller.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-cp "EAP_HOME/jboss-modules.jar:JSVC_JAR" \
org.apache.commons.daemon.support.DaemonWrapper \
-start org.jboss.modules.Main -start-method main \
-mp EAP_HOME/modules org.jboss.as.process-controller \
-jboss-home EAP_HOME -jvm $JAVA_HOME/bin/java \
-mp EAP_HOME/modules -- \
-Dorg.jboss.boot.log.file=LOG_DIR/host-controller.log \
-Dlogging.configuration=file:CONF_DIR/logging.properties \
-Djboss.modules.policy-permissions \
-server -Xms64m -Xmx512m -XX:MaxPermSize=256m \
-Djava.net.preferIPv4Stack=true \
-Djboss.modules.system.pkgs=org.jboss.byteman \
-Djava.awt.headless=true -- -default-jvm $JAVA_HOME/bin/java
```



NOTE

Si JBoss EAP 6 est arrêté de façon anormale, comme un crash de la JVM, Jsvc le démarrera à nouveau automatiquement. Si JBoss EAP 6 s'arrête normalement, Jsvc s'arrêtera également.

Rapporter un bogue

CHAPITRE 9. VALVES GLOBALES

9.1. VALVES

Une valve est une classe Java insérée dans le pipeline de traitement de demande d'une application. Elle est insérée dans le pipeline avant les filtres servlet. Les valves peuvent apporter des modifications à la demande avant de les passer ou d'effectuer tout autre traitement comme l'authentification ou annuler la demande.

Les valves peuvent être configurées au niveau du serveur ou au niveau de l'application. La seule différence est la façon dont elles sont configurées ou empaquetées.

- Les valves globales sont configurées au niveau serveur et s'appliquent à toutes les applications déployées dans le serveur. Les instructions sur la façon de configurer les valves globales se trouvent dans le guide *Administration and Configuration Guide* de JBoss EAP.
- Les valves configurées au niveau de l'application sont empaquetées dans le déploiement de l'application et n'affectent que l'application en question. Des instructions sur la façon de configurer des valves au niveau de l'application se trouvent dans le guide *Development Guide* de JBoss EAP.

Les versions 6.1.0 et supérieures prennent en charge les valves globales.

Rapporter un bogue

9.2. VALVES GLOBALES

Une valve globale est une valve insérée dans le pipeline de traitement de requête de toutes les applications déployées. Une valve est rendue globale lorsqu'elle est mise en paquetage et qu'elle est installée comme module static dans JBoss EAP 6. Les valves globales sont configurées dans le sous-système web.

Seules les versions 6.1.0 et supérieures prennent en charge les valves globales.

Pour obtenir des instructions sur la façon de configurer les valves globales, voir le chapitre qui s'intitule *Global Valves* dans le guide *Administration and Configuration Guide for JBoss EAP*.

Rapporter un bogue

9.3. LES VALVES D'AUTHENTIFICATION

Une valve d'authentification est une valve qui authentifie les informations d'identification d'une requête. Cette valve est une sous-classe de **`org.apache.catalina.authenticator.AuthenticatorBase`** et elle remplace la méthode **`authenticate(Request request, Response response, LoginConfig config)`**.

Elle peut être utilisée pour implémenter des schémas d'authentification supplémentaires.

Rapporter un bogue

9.4. INSTALLATION D'UNE VALVE GLOBALE

Les valves globales doivent être empaquetées et installées sous forme de modules statics dans JBoss EAP 6. Cette tâche vous montre comment installer le module.

Conditions préalables :

- La valve doit déjà avoir été créée et empaquetée dans un fichier JAR.
- Un fichier **`module.xml`** doit déjà avoir été créé pour le module.

  Voir Section 7.1.1, « Modules » pour un exemple de fichier **`module.xml`**.

Procédure 9.1. Installer un module global

1. **Créer un répertoire d'installation de module**
   Vous devrez créer un répertoire pour le module à installer dans le répertoire de modules du serveur d'applications.

   ```
   EAP_HOME/modules/system/layers/base/MODULENAME/main
   ```

   ```
   $ mkdir -P EAP_HOME/modules/system/layers/base/MODULENAME/main
   ```

2. **Copier les fichiers**
   Copier le JAR et les fichiers **`module.xml`** dans le répertoire créé dans l'étape 1.

   ```
   $ cp MyValves.jar module.xml
   EAP_HOME/modules/system/layers/base/MODULENAME/main
   ```

Les classes de valve déclarées dans le module sont maintenant disponibles et peuvent être configurées dans le sous-système web.

Rapporter un bogue

9.5. CONFIGURATION D'UNE VALVE GLOBALE

Les valves globales sont activées et configurées dans le sous-système web par l'intermédiaire de l'interface CLI de JBoss.

Procédure 9.2. Configuration d'une valve globale

1. **Activer la valve**
   Utiliser l'opération **add** pour ajouter une nouvelle saisie de valve.

   ```
   /subsystem=web/valve=VALVENAME:add(module="MODULENAME",class-
   name="CLASSNAME")
   ```

   Vous devrez indiquer les valeurs suivantes :

   - **`VALVENAME`**, le nom utilisé pour cette valve dans la configuration de l'application.
   - **`MODULENAME`**, le module qui contient la valeur en cours de configuration.
   - **`CLASSNAME`**, le nom de classe de la valve spécifique du module.

   ```
   /subsystem=web/valve=clientlimiter:add(module="clientlimitermodule",
   class-name="org.jboss.samplevalves.RestrictedUserAgentsValve")
   ```

2. **En option : spécifier les paramètres**
   Si la valve a des paramètres de configuration, spécifier les dans l'opération **`add-param`**.

```
/subsystem=web/valve=testvalve:add-param(param-name="NAME", param-
value="VALUE")
```

```
/subsystem=web/valve=testvalve:add-param(
   param-name="restrictedUserAgents",
   param-value="^.*MS Web Services Client Protocol.*$"
)
```

Cette valve est maintenant activée et configurée pour toutes les applications déployées.

Rapporter un bogue

CHAPITRE 10. DÉPLOIEMENT D'APPLICATIONS

10.1. LES DÉPLOIEMENTS D'APPLICATIONS

JBoss EAP 6 dispose d'une gamme d'options de déploiement et de configuration d'application pour répondre à la fois aux environnements administratifs et de développement. Pour les administrateurs, la console de gestion et l'interface CLI offrent un graphisme et des interfaces de ligne de commande idéals pour gérer le déploiement des applications dans un environnement de production. Pour les développeurs, la gamme des options de testing de déploiement d'application incluent un scanner de déploiement hautement configurable de système de fichiers, l'utilisation d'un IDE comme JBoss Developer Studio, ou le déploiement et l'annulation du déploiement via Maven.

Administration

- **Console de management**
  - Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »
  - Section 10.2.3, « Désactiver une application déployée à l'aide de la console de gestion »
- **Interface CLI**
  - Section 10.3.4, « Déployer une application dans un domaine géré à l'aide de l'interface CLI »
  - Section 10.3.2, « Déployer une application dans un serveur autonome à l'aide de l'interface CLI »
  - Section 10.3.5, « Supprimer le déploiement d'une application dans un domaine géré à l'aide de l'interface CLI »
  - Section 10.3.3, « Supprimer le déploiement d'une application dans un serveur autonome à l'aide de l'interface CLI »
  - Section 10.3.1, « Gérer le déploiement d'une application à l'aide de l'interface CLI »

Développement

- **Scanner de déploiement**
  - Section 10.5.7, « Configurer le scanneur de déploiement »
  - Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »
  - Section 10.5.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement »
  - Section 10.5.4, « Redéploiement d'une application dans une instance de serveur autonome par le scanneur de déploiement »
  - Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI »
  - Section 10.5.6, « Référence pour attributs de scanneur de déploiement »

- Section 10.5.5, « Référence pour les fichiers de marquage de scanneur de déploiement »

- **Maven**
  - Section 10.6.2, « Déployer une application dans Maven »
  - Section 10.6.3, « Supprimer le déploiement d'une application dans Maven »

Rapporter un bogue

10.2. DÉPLOYER AVEC LA CONSOLE DE GESTION

10.2.1. Gérer le déploiement d'application à l'aide de la console de gestion

Le déploiement d'applications par l'intermédiaire de la console de gestion vous donne l'avantage d'une interface graphique facile à utiliser. Vous pouvez voir en un coup d'œil quelles applications sont déployées sur votre serveur ou les groupes de serveurs, et vous pouvez désactiver ou supprimer des applications dans le référentiel de contenu selon les besoins.

Rapporter un bogue

10.2.2. Activer une application déployée à l'aide de la console de gestion

Conditions préalables

- Section 3.4.2, « Se conncecter à la console de gestion »
- Section 3.4.7, « Ajouter un déploiement dans une console de management »

Procédure 10.1. Activer une application déployée à l'aide de la console de gestion

1. Sélectionner l'onglet **`Runtime`** en haut de la console.
2. - Pour une domaine géré, étendre le menu **`Domain`**.
   - Pour une domaine autonome, étendre le menu **`Server`**.
3. Sélectionner **`Manage Deployments`**.
4. La méthode de déploiement des applications variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.
   - **Activer une application sur une instance de serveur autonome**
     Le tableau **`Available Deployments`** affiche tous les déploiements d'applications disponibles et leurs statuts.
     a. Pour activer une application dans une instance de serveur autonome, sélectionner l'application, puis cliquer sur **`En/Disable`**.
     b. Cliquer sur **`confirm`** pour confirmer que l'application puisse être activée dans l'instance du serveur.
   - **Activer une application dans un domaine géré**

L'onglet **`Content Repository`** contient un tableau **`Available Deployment Content`** qui montre tous les déploiements d'applications disponibles et leur statut.

a. Pour activer une application dans un domaine géré, sélectionner l'application à déployer. Cliquer sur **`Assign`** au dessus du tableau **`Available Deployment Content`**.

b. Cochez les cases pour chaque groupe de serveurs dans lesquels vous souhaitez ajouter l'application et cliquer sur le bouton **`Save`** pour terminer.

c. Sélectionner l'onglet **`Server Groups`** pour afficher le tableau **`Server Groups`**. Votre application sera maintenant déployée dans les groupes de serveurs que vous avez sélectionnés.

Résultat

L'application est déployée sur le serveur qui convient ou dans le groupe de serveurs qui convient.

Rapporter un bogue

10.2.3. Désactiver une application déployée à l'aide de la console de gestion

Conditions préalables

- Section 3.4.2, « Se conncecter à la console de gestion »
- Section 3.4.7, « Ajouter un déploiement dans une console de management »
- Section 10.2.2, « Activer une application déployée à l'aide de la console de gestion »

Procédure 10.2. Désactiver une application déployée à l'aide de la console de gestion

1. a. Sélectionner l'onglet **`Runtime`** en haut de la console.

   b. ■ Pour une domaine géré, étendre le menu **Domain**.

   ■ Pour une domaine autonome, étendre le menu **Server**.

   c. Sélectionner **`Manage Deployments`**.

2. La méthode de suppression d'une application variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.

   - **Désactiver l'application déployée dans une instance de serveur autonome**
     Le tableau **`Available Deployments`** affiche tous les déploiements d'applications disponibles et leurs statuts.

     a. Sélectionner l'application à désactiver. Cliquer sur **`En/Disable`** pour désactiver l'application sélectionnée.

     b. Cliquer sur **`Confirm`** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.

   - **Désactiver l'application déployée dans un domaine géré**
     L'affichage écran **`Manage Deployment Content`** contient un onglet **`Content Repository`**. Le tableau **`Available Deployment Content`** affiche tous les déploiements d'applications disponibles et leur statut.

a. Sèlectionner l'onglet **`Server Groups`** pour afficher les groupes de serveurs et le statut de leurs applications déployées.

b. Sélectionner le nom du serveur dans le tableau **`Server Group`** pour supprimer un déploiement. Cliquer sur **`View`** pour voir les applications.

c. Sélectionner l'application à désactiver et cliquer sur **`En/Disable`** pour désactiver l'application sur le serveur sélectionné.

d. Cliquer sur **`Confirm`** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.

e. Répéter au besoin pour d'autres groupes de serveur. Le statut de l'application est confirmé pour chaque groupe de serveurs dans le tableau **`Group Deployments`** de ce groupe de serveurs.

Résultat

L'application n'est pas déployée à partir d'un serveur ou d'un groupe de serveurs qui convient.

Rapporter un bogue

10.3. DÉPLOYER PART L'INTERFACE DE COMMANDES CLI

10.3.1. Gérer le déploiement d'une application à l'aide de l'interface CLI

Le déploiement d'applications par l'intermédiaire de l'interface CLI vous donne l'avantage d'une interface à ligne de commande facile à utiliser. Vous pouvez utiliser les capacités de scripting pour configurer le déploiement d'applications spécifiques et des scénarios de gestion. Vous pouvez gérer le statut de déploiement d'un serveur dans le cas d'une instance autonome, ou d'un réseau entier de serveurs dans le cas d'un domaine géré.

Rapporter un bogue

10.3.2. Déployer une application dans un serveur autonome à l'aide de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »

Procédure 10.3. Déployer une application dans un serveur autonome

- **Exécuter la commande `deploy`**
  À l'aide de l'interface CLI, saisir la commande **`deploy`** avec le chemin d'accès vers le déploiement de l'application.

  ```
  [standalone@localhost:9999 /] deploy /path/to/test-application.war
  ```

  Veuillez noter qu'un déploiement réussi ne crée pas de sortie vers le CLI.

Résultat

L'application indiquée est maintenant déployée dans un serveur autonome.

Rapporter un bogue

10.3.3. Supprimer le déploiement d'une application dans un serveur autonome à l'aide de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »
- Section 10.3.2, « Déployer une application dans un serveur autonome à l'aide de l'interface CLI »

Procédure 10.4. Supprimer le déploiement d'une application dans un serveur autonome

- **Exécuter la commande `undeploy`**
  À l'aide de l'interface CLI, saisir la commande **`undeploy`** avec le nom du fichier du déploiement de l'application.

  ```
  [standalone@localhost:9999 /] undeploy test-application.war
  ```

  Veuillez noter qu'une suppression de déploiement réussie ne crée pas de sortie vers le CLI.

Résultat

L'application spécifiée n'est désormais plus déployée.

Rapporter un bogue

10.3.4. Déployer une application dans un domaine géré à l'aide de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »

Procédure 10.5. Déployer une application dans un domaine géré

- **Exécuter la commande `deploy`**
  Par l'intermédiaire de l'interface CLI, saisir la commande **`deploy`** ainsi que le chemin d'accès vers le déploiement de l'application. Inclure le paramètre ***`--all-server-groups`*** afin de déployer tous les groupes de serveurs.

  ```
  [domain@localhost:9999 /] deploy /path/to/test-application.war --
  all-server-groups
  ```

  - Sinon, définir des groupes de serveurs particuliers de déploiement avec le paramètre ***`--server-groups`***.

```
[domain@localhost:9999 /] deploy /path/to/test-application.war --
server-groups=server_group_1,server_group_2
```

Veuillez noter qu'un déploiement réussi ne crée pas de sortie vers le CLI.

Résultat

L'application indiquée est maintenant déployée dans un groupe de serveurs dans votre domaine géré.

Rapporter un bogue

10.3.5. Supprimer le déploiement d'une application dans un domaine géré à l'aide de l'interface CLI

Pré-requis

- Section 3.5.2, « Lancement de l'interface CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »
- Section 10.3.4, « Déployer une application dans un domaine géré à l'aide de l'interface CLI »

Procédure 10.6. Supprimer le déploiement d'une application dans un domaine géré

- **Exécuter la commande `undeploy`**
  Par l'intermédiaire de l'interface CLI, saisir la commande **`undeploy`** ainsi que le nom de fichier du déploiement de l'application. On peut retirer le déploiement de l'application à partir de n'importe quel groupe de serveur dans lequel elle a été déployée à l'origine en ajoutant le paramètre ***`--all-relevant-server-groups`***.

  ```
  [domain@localhost:9999 /] undeploy test-application.war --all-
  relevant-server-groups
  ```

  Veuillez noter qu'une suppression de déploiement réussie ne crée pas de sortie vers le CLI.

Résultat

L'application spécifiée n'est désormais plus déployée.

Rapporter un bogue

10.4. DÉPLOYER PAR L'API HTTP

10.4.1. Déployer une application par l'API HTTP

Résumé

Les applications peuvent être déployées via l'API HTTP en suivant les instructions suivantes.

Procédure 10.7. Déployer une application par `DeployDmrToJson.java`

1. Utiliser **`DeployDmrToJson.java`** pour créer une requête à JSON de déploiement de votre application.

Exemple 10.1. DeployDmrToJson.java class

```
import org.jboss.dmr.ModelNode;
import java.net.URL;

public class DeployDmrToJson
{
  public static void main(String[] args) throws Exception
  {
    if(args.length < 1)
      throw new IllegalArgumentException("The first argument must
be a URL");

    URL url = new URL(args[0]);
    String[] pathElements = url.getFile().split("/");
    String name = pathElements[pathElements.length-1];

    ModelNode deploy = getDeploy(url.toExternalForm(), name);
    ModelNode undeploy = getUndeploy(name);

    System.out.println("Deploy\n----------------------------
\n");
    System.out.println("Formatted:\n" +
deploy.toJSONString(false));
    System.out.println("Unformatted:\n" +
deploy.toJSONString(true));
    System.out.println("\nUneploy\n----------------------------
-\n");
    System.out.println("Formatted:\n" +
undeploy.toJSONString(false));
    System.out.println("Unformatted:\n" +
undeploy.toJSONString(true));
  }

  public static ModelNode getUndeploy(String name)
  {
    ModelNode undeployRequest = new ModelNode();
    undeployRequest.get("operation").set("undeploy");
    undeployRequest.get("address", "deployment").set(name);

    ModelNode removeRequest = new ModelNode();
    removeRequest.get("operation").set("remove");
    removeRequest.get("address", "deployment").set(name);

    ModelNode composite = new ModelNode();
    composite.get("operation").set("composite");
    composite.get("address").setEmptyList();
    final ModelNode steps = composite.get("steps");
    steps.add(undeployRequest);
    steps.add(removeRequest);
    return composite;
  }

  public static ModelNode getDeploy(String url, String name)
  {
    ModelNode deployRequest = new ModelNode();
```

```
        deployRequest.get("operation").set("deploy");
        deployRequest.get("address", "deployment").set(name);

        ModelNode addRequest = new ModelNode();
        addRequest.get("operation").set("add");
        addRequest.get("address", "deployment").set(name);
        addRequest.get("content").get(0).get("url").set(url);

        ModelNode composite = new ModelNode();
        composite.get("operation").set("composite");
        composite.get("address").setEmptyList();
        final ModelNode steps = composite.get("steps");
        steps.add(addRequest);
        steps.add(deployRequest);
        return composite;
    }
}
```

2. Exécuter la classe par une commande basée sur les instructions suivantes :

Exemple 10.2. Commande Execute

```
java -cp .:$JBOSS_HOME/modules/org/jboss/dmr/main/jboss-
dmr-1.1.1.Final-redhat-1.jar DeployDmrToJson \
file:///Users/username/support/helloWorld.war/dist/helloWorld.war
```

3. Quand la classe est exécutée, les formats de commande suivants seront affichés. Utiliser la commande **`deploy`** ou la commande **`undeploy`** selon vos besoins.

Exemple 10.3.

```
Deploy
------------------------------

Formatted:
{
    "operation" : "composite",
    "address" : [],
    "steps" : [
        {
            "operation" : "add",
            "address" : {"deployment" : "helloWorld.war"},
            "content" : [{"url" :
"file:/Users/username/support/helloWorld.war/dist/helloWorld.war"}
]
        },
        {
            "operation" : "deploy",
            "address" : {"deployment" : "helloWorld.war"}
        }
    ]
}
```

```
Unformatted:
{"operation" : "composite", "address" : [], "steps" :
[{"operation" : "add", "address" : {"deployment" :
"helloWorld.war"}, "content" : [{"url" :
"file:/Users/username/support/helloWorld.war/dist/helloWorld.war"}
]},{"operation" : "deploy", "address" : {"deployment" :
"helloWorld.war"}}]}

Uneploy
------------------------------

Formatted:
{
    "operation" : "composite",
    "address" : [],
    "steps" : [
        {
            "operation" : "undeploy",
            "address" : {"deployment" : "helloWorld.war"}
        },
        {
            "operation" : "remove",
            "address" : {"deployment" : "helloWorld.war"}
        }
    ]
}
Unformatted:
{"operation" : "composite", "address" : [], "steps" :
[{"operation" : "undeploy", "address" : {"deployment" :
"helloWorld.war"}},{"operation" : "remove", "address" :
{"deployment" : "helloWorld.war"}}]}
```

4. Utiliser la commande suivante pour déployer ou supprimer le déploiement d'une application. Remplacer **`json request`** par la requête décrite ci-dessus.

Exemple 10.4. Commande Execute

```
curl -f --digest -u "<user>:<pass>" -H Content-Type:\
application/json -d '<json request>'
"http://localhost:9990/management"
```

Rapporter un bogue

10.5. DÉPLOYER AVEC LE SCANNEUR DE DÉPLOIEMENT

10.5.1. Gérer le déploiement d'applications dans le scanneur de déploiement

Déployer des applications dans une instance de serveur autonome par l'intermédiaire d'un scanneur de déploiement vous permet de créer et de tester des applications d'une manière adaptée aux cycles de développement rapides. Vous pouvez configurer le scanneur de déploiement en fonction de vos besoins de fréquence de déploiement et de comportement pour une variété de types d'applications.

Rapporter un bogue

10.5.2. Déployer une application dans une instance de serveur autonome par un scanneur de déploiement

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »

Résumé

Cette tâche présente une méthode de déploiement des applications dans une instance de serveur autonome par le scanneur de déploiement. Comme il est indiqué dans la section Section 10.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.

Procédure 10.8. Utiliser le scanneur de déploiement pour déployer les applications.

1. **Copier le contenu dans le dossier de déploiement**
   Copier le fichier de l'application dans un dossier de déploiement qui se situe ***EAP_HOME*/standalone/deployments/**.

2. **Modes d'analyses de déploiements**
   Il existe deux méthodes de déploiement. Vous pouvez choisir entre les modes de traitement manuel ou automatique. Avant de démarrer une des méthodes de déploiement, lisez ceci Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI ».

   - **Déploiement automatique**
     Le scanner de déploiement saisit un changement d'état d'un dossier et crée un fichier de marquage, comme expliqué dans la section Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI ».

   - **Déploiement manuel**
     Le scanneur de déploiement a besoin d'un fichier de marqueurs pour déclencher le processus de déploiement. L'exemple suivant utilise la commande Unix **touch** pour créer un nouveau fichier **.dodeploy**.

     Exemple 10.5. Déploiement par la commande `touch`

     ```
     [user@host bin]$ touch
     $EAP_HOME/standalone/deployments/example.war.dodeploy
     ```

Résultat

Le fichier de l'application est déployé sur le serveur d'applications. Un fichier de marquage est créé dans le dossier de déploiement pour indiquer la réussite du déploiement, et l'application est marquée comme **Enabled** dans la console de gestion.

Exemple 10.6. Contenu du dossier de déploiement après le déploiement.

```
example.war
example.war.deployed
```

Rapporter un bogue

10.5.3. Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »
- Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »

Résumé

Cette tâche présente une méthode de retrait de déploiement d'applications en provenance d'une instance de serveur autonome qui ont été déployées par le scanneur de déploiement. Comme il est indiqué dans la section Section 10.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.

NOTE

Le scanneur de déploiement ne devrait pas servir en conjonction avec d'autres méthodes de déploiement pour la gestion des applications. Les applications supprimées du serveur d'applications par la console de gestion seront retirées du runtime sans affecter les fichiers de marquage ou l'application contenue dans le répertoire de déploiement. Pour minimiser le risque de redéploiement accidentel ou autre erreur, utilisez l'interface CLI et la console de gestion pour l'administration dans des environnements de production.

Procédure 10.9. Retirer le déploiement d'une application à l'aide d'une des méthodes suivantes

- **Supprimer le déploiement de l'application**
  Il existe deux méthodes pour supprimer un déploiement d'application suivant que vous souhaitez supprimer l'application d'un dossier de déploiement ou bien que vous souhaitez uniquement modifier son statut de déploiement.
  - **Retirer un déploiement par suppression du fichier de marquage**
    Supprimer le fichier de marquage **`example.war.deployed`** de l'application qui est déployée pour commencer le retrait de déploiement de l'application du runtime.

    Résultat

    Le scanneur de déploiement retire l'application et crée un fichier de marquage **`example.war.undeployed`**. L'application demeure dans le dossier de déploiement.

  - **Retirer le déploiement en supprimant l'application**
    Supprimer l'application depuis le répertoire de déploiement pour encourager le scanneur de déploiement à commencer le retrait du déploiement de l'application du runtime.

    Résultat

    Le scanneur de déploiement retire l'application et crée un fichier de marquage **`filename.filetype.undeployed`**. L'application n'est plus présente dans le dossier de déploiement.

Résultat

Le fichier de l'application n'est plus déployé dans le serveur d'applications et n'est plus visible dans l'écran **`Deployments`** de la console de gestion.

Rapporter un bogue

10.5.4. Redéploiement d'une application dans une instance de serveur autonome par le scanneur de déploiement

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »
- Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »

Résumé

Cette tâche présente une méthode de retrait de déploiement d'applications dans une instance de serveur autonome qui a été déployée par le scanneur de déploiement. Comme il est indiqué dans la section Section 10.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de console de gestion ou d'interface CLI sont recommandées pour la gestion des applications dans les environnements de production.

Procédure 10.10. Déployer à nouveau une application dans un serveur autonome

- **Redéploiement de l'application**
  Il existe trois méthodes possibles pour redéployer une application déployée par le scanner de déploiement. Ces méthodes déclenchent le scanneur de déploiement pour initier un cycle de déploiement, et peuvent être choisies en fonction des préférences personnelles.
  - **Redéploiement par modification du fichier de marquage**
    Déclencher le redéploiement du scanneur de déploiement en modifiant l'horodatage d'accès du fichier. Dans l'exemple Linux suivant, on utilise une commande Unix **`touch`**.

    Exemple 10.7. Redéployer par la commande Unix `touch`

    ```
    [user@host bin]$ touch
    EAP_HOME/standalone/deployments/example.war.dodeploy
    ```

    Résultat

    Le scanneur de déploiement a détecté un changement dans le fichier de marquage et a déployé à nouveau l'application. Un nouveau fichier de marquage **`.deployed`** remplace le précédent.
  - **Déployer à nouveau en créant un nouveau fichier de marquage `.dodeploy`**
    Déclencher le redéploiement du scanneur de déploiement en créant un nouveau fichier de marquage **`.dodeploy`**. Voir les instructions de déploiement du manuel qui se trouvent dans Section 10.5.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement ».
  - **Déployer à nouveau en supprimant le fichier de marquage**

Comme décrit dans Section 10.5.5, « Référence pour les fichiers de marquage de scanneur de déploiement », la suppression du fichier de marquage **.deployed** va déclencher un retrait de déploiement et créera un marqueur **.undeployed**. Supprimer le marqueur de suppression de déploiement déclenchera le cycle de déploiement à nouveau. Voir Section 10.5.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement » pour obtenir des informations supplémentaires.

Résultat

Le fichier de l'application est déployé à nouveau.

Rapporter un bogue

10.5.5. Référence pour les fichiers de marquage de scanneur de déploiement

Les fichiers de marquage

Les fichiers de marquage font partie du sous-système du scanneur de déploiement. Ces fichiers indiquent le statut d'une application dans un répertoire de déploiement de l'instance du serveur autonome. Un fichier de marquage a le même nom que l'application, avec un suffixe de fichier qui indique l'état du déploiement de l'application. Le tableau suivant définit les types et les réponses de chaque fichier de marquage.

Exemple 10.8. Exemple de fichier de marquage

L'exemple suivant montre un fichier de marquage d'une instance déployée réussie pour une application nommée **testapplication.war**.

```
testapplication.war.deployed
```

Tableau 10.1. Définitions de types de fichiers de marquage

Suffixe du nom de fichier	Origine	Description
.dodeploy	Utilisateur généré	Indique que le contenu doit être déployé ou redéployé dans le runtime.
.skipdeploy	Utilisateur généré	Désactive l'autodéploiement d'une application si présent. Utile pour bloquer temporairement l'auto-déploiement d'un contenu explosé, empêchant ainsi le risque de modifications de contenu imcomplètes d'être rendues live. Peut être utile pour le contenu compressé, bien que le scanneur détecte les changements en cours dans le contenu compressé et attend qu'ils soient terminés.
.isdeploying	Système généré	Indique l'initiation du déploiement. Le fichier de marquage sera effacé quand le processus de déploiement sera complété.
.deployed	Système généré	Indique que le contenu a été déployé. Le déploiement du contenu sera supprimé si le fichier est effacé.

Suffixe du nom de fichier	Origine	Description
.failed	Système généré	Indique les échecs de déploiement. Le fichier de marquage contient des informations sur la cause de l'échec. Si le fichier de marquage est supprimé, le contenu sera rendu visible dans l'auto-déploiement à nouveau.
.isundeploying	Système généré	Indique une réponse suite à la suppression d'un fichier **.deployed**. Le déploiement du contenu sera supprimé et le marqueur sera effacé automatiquement dès complétion.
.undeployed	Système généré	Indique si le contenu a été déployé. La suppression du fichier de marquage n'a pas d'impact sur le re-déploiement du contenu.
.pending	Système généré	Indique que les instructions de déploiement devront être envoyées au serveur suite à la résolution d'un problème qui a été detecté. Ce marqueur sert de blocage de déploiement global. Le scanneur ne demandera pas au serveur de déployer ou de supprimer un déploiement de contenu tant que cette condition existe.

Rapporter un bogue

10.5.6. Référence pour attributs de scanneur de déploiement

Le scanneur de déploiement contient les attributs suivants, qui sont exposés dans l'interface CLI et qui peuvent être configurés par l'opération **write-attribute**. Pour plus d'informations sur les options de configuration, se référer à la section suivante Section 10.5.8, « Configurer le scanneur de déploiement avec l'interface CLI ».

Tableau 10.2. Attributs de scanneur de déploiement

Nom	Description	Type	Valeur par défaut
auto-deploy-exploded	Permet le déploiement automatique d'un contenu éclaté sans nécessiter un fichier de marquage **.dodeploy**. Recommandé pour les scénarios de développement de base uniquement pour empêcher le déploiement d'applications éclatées de se produire lors de changements du développeur ou du système d'exploitation.	Booléen	False
auto-deploy-xml	Autorise le déploiement automatique d'un contenu XML sans besoin de fichier de marquage **.dodeploy**.	Booléen	True

Nom	Description	Type	Valeur par défaut
auto-deploy-zipped	Autorise le déploiement automatique d'un contenu compressé sans besoin de fichier de marquage **.dodeploy**.	Booléen	True
deployment-timeout	La durée nécessaire en secondes pour que le scanneur de déploiement puisse permettre un déploiement avant annulation.	Long	600
path	Définit le chemin du système de fichier à scanner. Si l'attribut **relative-to** est spécifié, la valeur **path** agira comme un ajout relatif à ce répertoire ou chemin d'acccès.	String	déploiements
relative-to	Référence à un chemin de système de fichier défini dans la section **paths** du fichier de configuration XML du serveur.	String	jboss.server.base.dir
scan-enabled	Autorise le scanning automatique des applications par **scan-interval** au démarrage.	Booléen	True
scan-interval	L'intervalle en millisecondes entre les balayages de référentiels. Une valeur inférieure à 1 empêche le scanneur d'opérer au démarrage.	Int	5000

Rapporter un bogue

10.5.7. Configurer le scanneur de déploiement

Le scanner de déploiement peut être configuré à l'aide de la console de gestion ou l'interface CLI. Vous pouvez créer un nouveau scanneur de déploiement ou bien gérer les attributs existants de scanneur, c'est à dire l'intervalle de balayage, l'emplacement du dossier de déploiement et les types de fichiers d'application qui déclencheront un déploiement.

Rapporter un bogue

10.5.8. Configurer le scanneur de déploiement avec l'interface CLI

Conditions préalables

- Section 3.5.2, « Lancement de l'interface CLI »

Résumé

Bien qu'il existe plusieurs méthodes de configuration du scanneur de déploiement, l'interface CLI permet d'exposer et de modifier les attributs par l'utilisation de scripts de lots ou en temps réel. Vous pouvez modifier le comportement du scanneur de déploiement par l'utilisation de l'attribut lecture **read-**

attribute et des opérations de ligne de commande **write-attribute**. Davantage d'informations sur les attributs de scanneur de déploiement sont définis dans la rubrique Section 10.5.6, « Référence pour attributs de scanneur de déploiement ».

Le scanneur de déploiement est un sous-système de JBoss EAP 6, que vous pouvez voir dans **standalone.xml**.

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
    <deployment-scanner path="deployments" relative-
to="jboss.server.base.dir" scan-interval="5000"/>
</subsystem>
```

Procédure 10.11. Configurer le scanneur de déploiement

1. **Déterminer les attributs de scanner de déploiement à configurer**
   Pour configurer le scanneur de déploiement par l'interface CLI, vous devrez tout d'abord exposer les noms d'attribut qui conviennent. Vous pouvez faire cela grâce à l'opération **read-resources** au nœud root, ou bien par la commande **cd** pour passer au nœud dépendant du sous-système. Vous pouvez également afficher les attributs par la commande **ls** à ce niveau.

   - **Exposer les attributs de scanneur de déploiement par l'opération `read-resource`**
     Utiliser l'opération **read-resource** pour exposer les attributs définis par la ressource de scanneur de déploiement par défaut.

     ```
     [standalone@localhost:9999 /]/subsystem=deployment-
     scanner/scanner=default:read-resource
     {
         "outcome" => "success",
         "result" => {
             "auto-deploy-exploded" => false,
             "auto-deploy-xml" => true,
             "auto-deploy-zipped" => true,
             "deployment-timeout" => 600,
             "path" => "deployments",
             "relative-to" => "jboss.server.base.dir",
             "scan-enabled" => true,
             "scan-interval" => 5000
         }
     }
     ```

   - **Exposer les attributs de scanneur de déploiement par la commande `ls`**
     Utiliser la commande **ls** avec l'argument **-l** en option pour afficher une table de résultats qui incluent des attributs de nœud, des valeurs et types de sous-système. Vous pouvez en apprendre davantage sur la commande **ls** et ses arguments en exposant l'entrée **ls --help**. Pour plus d'informations sur le menu help du Management CLI, voir la section Section 3.5.5, « Comment obtenir de l'aide par l'interface CLI ».

     ```
     [standalone@localhost:9999 /] ls -l /subsystem=deployment-
     scanner/scanner=default
     ATTRIBUTE            VALUE                TYPE
     auto-deploy-exploded false                BOOLEAN
     auto-deploy-xml      true                 BOOLEAN
     auto-deploy-zipped   true                 BOOLEAN
     deployment-timeout   600                  LONG
     ```

```
path                 deployments           STRING
relative-to          jboss.server.base.dir STRING
scan-enabled         true                  BOOLEAN
scan-interval        5000                  INT
```

2. **Configurer le scanneur de déploiement par l'opération `write-attribute`**
   Une fois que vous avez déterminé le nom de l'attribut à modifier, utiliser la commande **`write-attribute`** pour spécifier le nom de l'attribut et la nouvelle valeur à indiquer. Les exemples suivants sont tous exécutés au niveau du nœud dépendant, qui peut être accédé en utilisant la commande **cd** et la saisie semi-automatique via la touche TAB pour passer au nœud de scanneur par défaut.

```
[standalone@localhost:9999 /] cd subsystem=deployment-
scanner/scanner=default
```

   a. **Active le déploiement automatique du contenu explosé.**
      Utiliser l'opération **`write-attribute`** pour activer le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-exploded,value=true)
{"outcome" => "success"}
```

   b. **Désactiver le déploiement automatique du contenu XML**
      Utiliser l'opération **`write-attribute`** pour désactiver le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-xml,value=false)
{"outcome" => "success"}
```

   c. **Désactiver le déploiement automatique du contenu compressé**
      Utiliser la commande **`write-attribute`** pour désactiver le déploiement automatique du contenu d'applications compressé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-zipped,value=false)
{"outcome" => "success"}
```

   d. **Configurer l'attribut du chemin d'accès**
      Utiliser l'opération **`write-attribute`** pour modifier l'attribut de chemin d'accès, pour substituer la valeur de l'exemple **`newpathname`** par un nouveau nom de chemin d'accès que le scanneur de déploiement puisse surveiller. Noter que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=path,value=newpathname)
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
```

```
        "process-state" => "reload-required"
    }
}
```

e. **Configurer l'attribut du chemin relatif**
Utiliser l'opération **`write-attribute`** pour modifier la référence relative du chemin du système de fichier ainsi définie dans la section des chemins d'accès du fichier de configuration XML. Notez que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=relative-to,value=new.relative.dir)
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

f. **Désactiver le scanneur de déploiement**
Utiliser l'opération **`write-attribute`** pour désactiver le scanneur de déploiement en définissant la valeur **`scan-enabled`** à false.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=scan-enabled,value=false)
{"outcome" => "success"}
```

g. **Changer l'intervalle de balayage**
Utiliser l'opération **`write-attribute`** pour modifier l'intervalle de balayage de 5 000 millisecondes à 10 000 millisecondes.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=scan-interval,value=10000)
{"outcome" => "success"}
```

Résultat

Vos modifications de configuration sont sauvegardées dans le scanneur de déploiement.

Rapporter un bogue

10.6. DÉPLOYER AVEC MAVEN

10.6.1. Gestion du déploiement d'applications dans Maven

Le déploiement d'applications dans Maven vous permet d'incorporer un cycle de déploiement dans votre flux de développement existant.

Rapporter un bogue

10.6.2. Déployer une application dans Maven

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »

Résumé

Cette tâche vous montre une méthode pour déployer des applications dans Maven. L'exemple fourni utilise l'application **`jboss-helloworld.war`** qui se trouve dans la collection Jboss EAP 6 Quickstarts. Le projet **`helloworld`** contient un fichier POM qui initialise le **`jboss-as-maven-plugin`**. Ce plugin fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance du serveur d'applications.

Procédure 10.12. Déployer une application dans Maven.

1. Ouvrir la session de terminal et naviguez dans le répertoire qui contient les exemples Quickstart.

   Exemple 10.9. Passer au répertoire d'application helloworld

   ```
   [localhost]$ cd /QUICKSTART_HOME/helloworld
   ```

2. Exécuter la commande de déploiement Maven pour déployer l'application. Si l'application est déjà en cours d'exécution, elle sera déployée à nouveau.

   ```
   [localhost]$ mvn package jboss-as:deploy
   ```

3. Afficher les résultats.

   - Le déploiement peut être confirmé si vous regardez les entrées de journalisation de l'opération dans la fenêtre du terminal.

     Exemple 10.10. Confirmation Maven pour l'application helloworld

     ```
     [INFO] ------------------------------------------------------------------
     [INFO] BUILD SUCCESS
     [INFO] ------------------------------------------------------------------
     [INFO] Total time: 32.629s
     [INFO] Finished at: Fri Mar 14 09:09:50 EDT 2014
     [INFO] Final Memory: 23M/204M
     [INFO] ------------------------------------------------------------------
     ```

   - Le déploiement peut également être confirmé dans le flux de statut de l'instance du serveur d'applications actives.

     Exemple 10.11. Confirmation du serveur d'applications pour l'application helloworld

     ```
     09:09:49,167 INFO  [org.jboss.as.repository] (management-
     ```

```
handler-thread - 1) JBAS014900: Content added at location
/home/username/EAP_HOME/standalone/data/content/32/4b4ef9a4bbe7
206d3674a89807203a2092fc70/content
09:09:49,175 INFO  [org.jboss.as.server.deployment] (MSC
service thread 1-7) JBAS015876: Starting deployment of "jboss-
helloworld.war" (runtime-name: "jboss-helloworld.war")
09:09:49,563 INFO  [org.jboss.weld.deployer] (MSC service
thread 1-8) JBAS016002: Processing weld deployment jboss-
helloworld.war
09:09:49,611 INFO  [org.jboss.weld.deployer] (MSC service
thread 1-1) JBAS016005: Starting Services for CDI deployment:
jboss-helloworld.war
09:09:49,680 INFO  [org.jboss.weld.Version] (MSC service thread
1-1) WELD-000900 1.1.17 (redhat)
09:09:49,705 INFO  [org.jboss.weld.deployer] (MSC service
thread 1-2) JBAS016008: Starting weld service for deployment
jboss-helloworld.war
09:09:50,080 INFO  [org.jboss.web] (ServerService Thread Pool -
- 55) JBAS018210: Register web context: /jboss-helloworld
09:09:50,425 INFO  [org.jboss.as.server] (management-handler-
thread - 1) JBAS018559: Deployed "jboss-helloworld.war"
(runtime-name : "jboss-helloworld.war")
```

Résultat

L'application est déployée dans le serveur d'applications.

Rapporter un bogue

10.6.3. Supprimer le déploiement d'une application dans Maven

Conditions préalables

- Section 2.1.1, « Démarrer JBoss EAP 6 »

Résumé

Cette tâche vous montre une méthode pour supprimer le déploiement des applications dans Maven. L'exemple fourni utilise l'application **`jboss-helloworld.war`** qui se trouve dans la collection Jboss EAP 6 Quickstarts. Le projet **`helloworld`** contient un fichier POM qui initialise le **`jboss-as-maven-plugin`**. Ce plugin fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance du serveur d'applications.

Procédure 10.13. Supprimer le déploiement d'une application dans Maven

1. Ouvrir la session de terminal et naviguez dans le répertoire qui contient les exemples Quickstart.

   Exemple 10.12. Passer au répertoire d'application helloworld

   ```
   [localhost]$ cd /QUICKSTART_HOME/helloworld
   ```

2. Exécuter la commande de suppression de déploiement.

```
[localhost]$ mvn jboss-as:undeploy
```

3. Afficher les résultats.

   - La suppression du déploiement peut être confirmée si on observe les entrées de journalisation de la fenêtre de terminal.

     Exemple 10.13. Confirmation Maven pour la suppression du déploiement de l'application helloworld.

     ```
     [INFO] ------------------------------------------------------------------
     [INFO] BUILD SUCCESSFUL
     [INFO] ------------------------------------------------------------------
     [INFO] Total time: 1 second
     [INFO] Finished at: Mon Oct 10 17:33:02 EST 2011
     [INFO] Final Memory: 11M/212M
     [INFO] ------------------------------------------------------------------
     ```

   - La suppression du déploiement peut également être confirmée dans le flux de statut de l'instance du serveur d'applications actives.

     Exemple 10.14. Confirmation du serveur d'applications pour la suppression du déploiement de l'application helloworld.

     ```
     09:51:40,512 INFO  [org.jboss.web] (ServerService Thread Pool -
     - 69) JBAS018224: Unregister web context: /jboss-helloworld
     09:51:40,522 INFO  [org.jboss.weld.deployer] (MSC service
     thread 1-3) JBAS016009: Stopping weld service for deployment
     jboss-helloworld.war
     09:51:40,536 INFO  [org.jboss.as.server.deployment] (MSC
     service thread 1-1) JBAS015877: Stopped deployment jboss-
     helloworld.war (runtime-name: jboss-helloworld.war) in 27ms
     09:51:40,621 INFO  [org.jboss.as.repository] (management-
     handler-thread - 10) JBAS014901: Content removed from location
     /home/username/EAP_HOME/jboss-eap-
     6.3/standalone/data/content/44/e1f3c55c84b777b0fc201d69451223c0
     9c9da5/content
     09:51:40,621 INFO  [org.jboss.as.server] (management-handler-
     thread - 10) JBAS018558: Undeployed "jboss-helloworld.war"
     (runtime-name: "jboss-helloworld.war")
     ```

Résultat

La suppression du déploiement de l'application provient du serveur d'applications.

Rapporter un bogue

10.7. CONTRÔLER L'ORDRE DES APPLICATIONS DÉPLOYÉES DANS JBOSS EAP 6

JBoss EAP 6 offre un contrôle à grain fin sur l'ordre de déploiement d'applications lors du démarrage du serveur. L'ordre strict de déploiement d'applications présentes dans plusieurs fichiers ear peut être activé avec la persistance de cet ordre après un redémarrage.

Procédure 10.14. Contrôle de l'ordre de déploiement dans EAP 6.0.X

1. Crée des scripts CLI qui déploient et retirent les déploiements d'applications dans un ordre séquentiel quand le serveur est à l'arrêt/démarrage.

2. CLI prend également en charge le concept de mode batch qui permet de grouper les commandes et les opérations et les exécuter ensemble comme une unité atomique. Si au moins une des commandes ou opérations échoue, toutes les autres commandes et opérations exécutées avec succès dans le lot seront annulées.

Procédure 10.15. Contrôler l'ordre de déploiement dans EAP 6.1.X

La nouvelle fonctionnalité nommée Inter Deployment Dependencies d'EAP 6.1.X vous permet de déclarer des dépendances entre les niveaux supérieurs de déploiement.

1. Créer (s'il n'existe pas encore) un fichier **`jboss-all.xml`** dans le dossier **`app.ear/META-INF`** où **`app.ear`** est l'archive d'application qui dépend d'une autre archive d'application à déployer avant.

2. Effectuer une saisie **`jboss-deployment-dependencies`** dans ce fichier comme indiqué ci-dessous. Notez que dans la liste ci-dessous, **`framework.ear`** est l'application de dépendance qui doit être déployée avant que l'archive d'application **`app.ear`** ne le soit.

```
<jboss umlns="urn:jboss:1.0">
  <jboss-deployment-dependencies xmlns="urn:jboss:deployment-
dependencies:1.0">
    <dependency name="framework.ear" />
  </jboss-deployment-dependencies>
</jboss>
```

Rapporter un bogue

10.8. REMPLACEMENT DU DESCRIPTEUR DE DÉPLOIEMENT

Une nouvelle fonctionnalité de EAP 6.1 x vous permet de remplacer des descripteurs de déploiement, des JAR, des classes, des pages JSP, et d'autres fichiers lors du temps d'exécution. Un *deployment overlay* représente un ensemble de règles de fichiers devant être remplacés dans les archives. Il fournit également des liens vers les nouveaux fichiers qui doivent être utilisés au lieu de ceux ayant été remplacés. Si le fichier remplacé n'est pas présent dans les archives de déploiement, il sera ajouté au déploiement.

Procédure 10.16. Remplacer le descripteur de déploiement en utilisant le Management CLI

Les étapes suivantes partent du principe que vous possédez déjà une application déployée appelée **`app.war`** et que vous souhaitez remplacer son fichier **`WEB-INF/web.xml`** avec un autre fichier **`web.xml`** situé dans **`/home/user/web.xml`**.

1. Ajouter une superposition de déploiement (deployment overlay) et y ajouter du contenu. Vous pouvez y parvenir de deux façons :

   - **Utilisation d'une arborescence DMR**

     a.
     ```
     /deployment-overlay=myoverlay:add
     ```

     b.
     ```
     /deployment-overlay=myoverlay/content=WEB-
     INF\/web.xml:add(content={url=file:///home/user/web.xml})
     ```

     Vous pouvez ajouter des règles de contenu supplémentaires en utilisant le deuxième code.

   - **Utilisation des méthodes de facilité**

     ```
     deployment-overlay add --name=myoverlay --content=WEB-
     INF/web.xml=/home/user/web.xml
     ```

2. Relie la superposition à une archive de déploiement. Vous pouvez y parvenir de deux façons :

   - **Utilisation d'une arborescence DMR**

     ```
     /deployment-overlay=myoverlay/deployment=app.war:add
     ```

   - **Utilisation des méthodes de facilité**

     ```
     deployment-overlay link --name=myoverlay --deployments=app.war
     ```

     Vous pouvez spécifier plusieurs noms d'archives séparés par des virgules.

   Veuillez noter que le nom d'archive de déploiement n'a pas besoin d'exister sur le serveur. Vous devez indiquer le nom, mais il ne sera pas encore lié à un déploiement.

3. **Redéploiement de l'application**

   ```
   /deployment=app.war:redeploy
   ```

Rapporter un bogue

CHAPITRE 11. SÉCURISER JBOSS EAP 6

11.1. LA SÉCURITÉ DU SOUS-SYSTÈME

Le sous-système de **`sécurité`** fournit l'infrastructure de sécurité pour toutes les fonctionnalités de sécurité de la plateforme Red Hat JBoss Enterprise Application en utilisant les services de sécurité fournis par PicketBox. Le sous-système utilise un contexte de sécurité associé à la demande en cours pour exposer les fonctionnalités du gestionnaire d'authentification, du gestionnaire d'autorisations, du gestionnaire d'auditing et du gestionnaire de mappage (mises en correspondance) pour le conteneur approprié.

Par défaut, le sous-système **`security`** est préconfiguré dans l'installation out-of-the-box, donc les éléments de sécurité ont rarement besoin d'être changés. Le seul élément de sécurité qui ait besoin d'être changé est pour savoir si on doit utiliser *deep-copy-subject-mode*. Dans la plupart des cas, les administrateurs se concentreront sur la configuration des *security domains*.

Mode Deep Copy

Voir Section 11.4, « Mode de sujet Deep Copy » pour obtenir des détails supplémentaires sur le mode Deep Copy.

Domaine de sécurité

Un domaine de sécurité est un ensemble de configurations de sécurité déclarative *Java Authentication and Authorization Service (JAAS)* qu'une ou plusieurs applications utilisent pour contrôler l'authentification, l'autorisation, l'auditing et le mapping. Trois domaines de sécurité sont inclus par défaut : **`jboss-ejb-policy`**, **`jboss-web-policy`**, et **`other`**. Vous pouvez créer autant de domaines de sécurité que vous souhaitez pour accommoder les besoins de vos applications. Voir Section 11.6.12, « Utiliser un domaine de sécurité dans votre application » pour obtenir des détails sur le domaine de sécurité.

Rapporter un bogue

11.2. STRUCTURE DU SOUS-SYSTÈME DE SÉCURITÉ

Le sous-système de sécurité est configuré dans le domaine géré ou le fichier de configuration autonome. La plupart des éléments de configuration peuvent être configurés à l'aide de la console de gestion via web ou de l'interface CLI sur console. Voici le code XML qui représente un exemple de sous-système de sécurité.

Exemple 11.1. Exemple de configuration de sous-système de sécurité

```
<subsystem xmlns="urn:jboss:domain:security:1.2">
 <security-management>
  ...
 </security-management>
 <security-domains>
        <security-domain name="other" cache-type="default">
            <authentication>
                <login-module code="Remoting" flag="optional">
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
                <login-module code="RealmUsersRoles" flag="required">
                    <module-option name="usersProperties"
```

```
value="${jboss.domain.config.dir}/application-users.properties"/>
                    <module-option name="rolesProperties"
value="${jboss.domain.config.dir}/application-roles.properties"/>
                    <module-option name="realm"
value="ApplicationRealm"/>
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
            </authentication>
        </security-domain>
        <security-domain name="jboss-web-policy" cache-type="default">
            <authorization>
                <policy-module code="Delegating" flag="required"/>
            </authorization>
        </security-domain>
        <security-domain name="jboss-ejb-policy" cache-type="default">
            <authorization>
                <policy-module code="Delegating" flag="required"/>
            </authorization>
        </security-domain>
    </security-domains>
    <vault>
      ...
    </vault>
</subsystem>
```

Les éléments **<security-management>**, **<subject-factory>** et **<security-properties>** ne sont pas présents dans la configuration par défaut. Les éléments **<subject-factory>** et **<security-properties>** ont été désapprouvés sur JBoss EAP 6.1 et versions ultérieures.

Rapporter un bogue

11.3. CONFIGURER LE SOUS-SYSTÈME DE SÉCURITÉ

Vous pouvez configurer le sous-système de sécurité par l'interface CLI ou par la console de gestion basée web.

Chaque élément de niveau supérieur du sous-système de sécurité contient des informations sur un aspect différent de la configuration de la sécurité. Voir Section 11.2, « Structure du sous-système de sécurité » pour obtenir un exemple de configuration de sous-système.

<security-management>

Cette section remplace les comportements de haut niveau du sous-système de sécurité. Chaque paramètre est optionnel. Il est rare de modifier ces paramètres sauf pour le mode de sujet Deep Copy.

Option	Description
deep-copy-subject-mode	Indiquez si l'on doit copier ou lier les tokens de sécurité pour la sécurité des threads.

Option	Description
authentication-manager-class-name	Indiquer un nom de classe d'implémentation AthentificationManager alternatif à utiliser.
authorization-manager-class-name	Indiquer un nom de classe d'implémentation AthorizationManager alternatif à utiliser.
audit-manager-class-name	Indiquer un nom de classe d'implémentation Audit Manager alternatif à utiliser.
identity-trust-manager-class-name	Indiquer un nom de classe d'implémentation IdentityTrustManager alternatif à utiliser.
mapping-manager-class-name	Indiquer le nom de classe d'implémentation MappingManager à utiliser.

<subject-factory>

La fabrique de sujets contrôle la création d'instances de sujets. Elle utilise sans doute le gestionnaire d'authentification pour vérifier l'appelant. L'utilisation principale du sujet est la création d'un sujet par les composants JCA. Il est rare que l'on ait besoin de modifier la fabrique de sujets.

<security-domains>

Un élément de conteneur qui contient plusieurs domaines de sécurité. Un domaine de sécurité peut contenir des informations sur des modules d'authentification, d'autorisation, de mappage, et d'auditing, ainsi qu'une autorisation JASPI et une configuration JSSE. Votre application indique ainsi un domaine de sécurité pour gérer ses informations de sécurité.

<security-properties>

Contient des noms et des valeurs définis dans la classe java.security.Security

Rapporter un bogue

11.4. MODE DE SUJET DEEP COPY

Si le mode de sujet Deep Copy (*deep copy subject mode*) est désactivé (par défaut), copier une structure de données de sécurité fait référence à la structure d'origine, au lieu de copier toute la structure de données. Ce comportement est plus efficace, mais enclin à la corruption des données si plusieurs threads possédant la même identité effacent le sujet lors d'un vidage ou d'une déconnexion.

Le mode de sujet Deep Copy entraîne la copie totale de la structure des données et de toutes les données associées possibles, quand elles sont marquées « clonables ». C'est plus sûr au niveau thread, mais moins efficace.

Le mode de sujet Deep Copy est configuré dans le cadre du sous-système de sécurité.

Rapporter un bogue

11.5. ACTIVER LE MODE DE SUJET DEEP COPY

Vous pouvez activer le mode de sécurité Deep Copy à partir de la console de gestion basée web ou de l'interface CLI.

Procédure 11.1. Activez le mode de sécurité Deep Copy à partir de la console de gestion

1. **Connectez-vous à la console de management.**
   Voir Section 3.4.2, « Se conncecter à la console de gestion ».

2. **Domaine géré : sélectionnez le profil qui convient.**
   Dans un domaine géré, le sous-système de sécurité est configuré par profil, ou vous pouvez activer ou désactiver le mode de sécurité Deep Copy de manière indépendante dans chaque profil.

   Pour sélectionner un profil, cliquez sur **`Configuration`** en haut et à droite de l'écran, puis sélectionnez un profil à partir du menu déroulant **`Profile`** en haut et à gauche.

3. **Ouvrir le menu de configuration `Security Subsystem`.**
   Étendre l'item de menu **`Security`**, puis cliquez sur le lien **`Security Subsystem`**.

4. **Activez le mode `Deep Copy Subject`.**
   Cliquez sur **`Edit`**. Cochez la case qui se trouve à côté de **`Deep Copy Subjects`** pour activer le mode Copier Sujet (copy subject).

Activez Deep Copy Subject Mode par la Console CLI

Si vous préférez utiliser l'interface CLI pour activer cette option, utilisez une des commandes suivantes.

Exemple 11.2. Domaine géré

```
/profile=full/subsystem=security/:write-attribute(name=deep-copy-
subject-mode,value=TRUE)
```

Exemple 11.3. Serveur autonome

```
/subsystem=security/:write-attribute(name=deep-copy-subject-
mode,value=TRUE)
```

Rapporter un bogue

11.6. DOMAINES DE SÉCURITÉ

11.6.1. Les domaines de sécurité

Les domaines de sécurité font partie du sous-système de sécurité de JBoss EAP. La configuration de la sécurité est désormais gérée de façon centralisée, ou par le contrôleur de domaines d'un domaine géré ou par le serveur autonome.

Un domaine de sécurité se compose de configurations d'authentification, d'autorisation, de mappage de sécurité et d'audit. Il met en place la sécurité déclarative *Java Authentication and Authorization Service (JAAS)*.

L'authentification est impliquée dans la vérification de l'identité d'un utilisateur. Dans la terminologie de la sécurité, cet utilisateur est appelé un *principal*. Bien que l'authentification et l'autorisation soient différentes, de nombreux modules d'authentification intégrés gèrent également l'autorisation.

Une *authorization* est un process par lequel le serveur détermine si un utilisateur authentifié a la permission d'accéder à des privilèges ou ressources particulières dans le système ou l'opération.

Security mapping se rapporte à la possibilité d'ajouter, de modifier ou de supprimer des informations d'un principal, rôle ou attribut avant de passer les informations à votre application.

L'Auditing Manager vous permet de configurer les *provider modules* pour contrôler la façon dont les événements de sécurité sont rapportés.

Si vous utilisez des domaines de sécurité, vous pouvez supprimer toutes les configurations de sécurité spécifiques à votre application proprement dite. Cela permet de modifier les paramètres de sécurité de façon centralisée. Un scénario courant qui bénéficie de ce type de structure de configuration est le processus de déplacement des applications entre les environnements de test et de production.

Rapporter un bogue

11.6.2. Picketbox

Picketbox est le cadre de sécurité de base qui fournit l'authentification, l'autorisation, la vérification et des fonctions de mappage à des applications Java exécutant dans JBoss EAP 6. Il fournit les fonctions suivantes, dans un cadre unique avec une configuration simple :

- Section 11.6.3, « Authentification »
- Section 11.6.5, « L'autorisation » et contrôle d'accès
- Section 11.6.7, « Security Auditing »
- Section 11.6.10, « Security Mapping » des principaux, rôles et attributs

Rapporter un bogue

11.6.3. Authentification

L'authentification consiste à identifier un sujet et à vérifier l'authenticité de l'identification. Le mécanisme d'authentification le plus commun est une combinaison de nom d'utilisateur et de mot de passe. D'autres mécanismes d'authentification communs utilisent des clés partagées, des smart cards (cartes à puce) ou des empreintes digitales. Le résultat d'une authentification réussie s'appelle un Principal, en termes de sécurité déclarative Java Enterprise Edition.

JBoss EAP utilise un système pouvant se connecter à des modules d'authentification en vue de flexibilité et d'intégration avec les systèmes d'authentification que vous utilisez dans votre organisation. Chaque domaine de sécurité contient un ou plusieurs modules d'authentification configurés. Chaque module comprend des paramètres de configuration supplémentaires pour personnaliser son comportement. Le plus simple consiste à configurer le sous-système d'authentification dans la console de gestion web.

L'authentification n'est pas la même chose que l'autorisation, même si elles sont souvent liées. Bon nombre des modules d'authentification intégrés peuvent également gérer des autorisations.

Rapporter un bogue

11.6.4. Configurer l'authentification dans un domaine de sécurité

Pour configurer les paramètres d'authentification d'un domaine de sécurité, vous connecter dans la console de gestion et suivre la procédure suivante :

Procédure 11.2. Configurez l'authentification dans un domaine de sécurité

1. **Ouvrir l'affichage détaillé du domaine de sécurité.**
   a. Cliquez sur l'étiquette **`Configuration`** qui se trouve en haut de la console de gestion.
   b. Sélectionnez le profil à modifier à partir du menu de sélection **`Profile`** qui se trouve en haut et à gauche de la vue 'Profile'.
   c. Étendre l'item de menu **`Security`**, puis sélectionnez **`Security Domains`**.
   d. Cliquez sur le lien **`View`** pour obtenir le domaine de sécurité que vous voulez modifier.
2. **Naviguez dans la configuration du sous-système d'authentification.**
   Sélectionnez l'étiquette **`Authentification`** en haut de l'affichage si elle n'a pas déjà été sélectionnée.

   La zone de configuration est divisée en deux : **`Login Modules`** et **`Details`**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune peut inclure plusieurs attributs et options.
3. **Ajoutez un module de connexion**
   Cliquez sur le bouton **`Add`** pour ajouter un module d'authentification JAAS. Remplissez les informations pour votre module.

   Le **`Code`** est le nom de classe de votre module. Les **`Flags`** controlent la façon dont le module est lié à d'autres modules de polices d'authentification du même domaine de sécurité.

   Explication des indicateurs

   La spécification Java Enterprise Edition 6 fournit l'explication suivante sur les marqueurs des modules de sécurité. La liste suivante provient de http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html#Appendix Consultez ce document pour obtenir des informations plus détaillées.

Marqueur	Détails
requis	Le LoginModule est requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule.
pré-requis	Le LoginModule est requis pour la réussite. En cas de succès, l'authentification continue vers le bas de la liste LoginModule. En cas d'échec, le contrôle est renvoyé immédiatement à l'application (l'authentification ne continue pas son chemin le long de la liste LoginModule).

Marqueur	Détails
suffisant	Le LoginModule n'est pas nécessaire pour aboutir. S'il ne réussit pas, le contrôle retournera immédiatement à l'application (l'authentification ne se déroule pas vers le bas de la liste LoginModule). S'il échoue, l'authentification se poursuit vers le bas de la liste LoginModule.
option	Le LoginModule n'est pas requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule.

4. **Modifiez les paramètres d'authentification**
   Une fois que vous aurez ajouté votre module, vous pourrez modifier son **Code** ou ses **Flags** (indicateurs) en cliquant **Edit** dans la section **Details** de l'écran. Veillez à ce que l'onglet **Attributes** soit bien sélectionné.

5. **Option : ajouter ou supprimer un module**
   Pour ajouter des options à votre module, cliquez sur leurs entrées dans la liste **Login Modules**, et sélectionnez l'onglet **Module Options** dans la section **Details** qui se trouve en bas de la page. Cliquez sur le bouton **Add**, et fournir la clé et la valeur de l'option. Utilisez le bouton **Remove** pour supprimer l'option.

Résultat

Votre module d'authentification est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

L'option de module `jboss.security.security_domain`

Par défaut, chaque module de connexion défini dans un domaine de sécurité a l'option de module **`jboss.security.security_domain`** ajoutée automatiquement. Cette option cause des problèmes avec les modules de connexion, qui vérifient que seules les options connues soient définies. Le module de connexion Kerberos d'IBM **`com.ibm.security.auth.module.Krb5LoginModule`** est l'un d'entre eux.

Vous pouvez désactiver le comportement d'ajout de cette option de module, en définissant la propriété de système sur **`true`** en démarrant la plate-forme JBoss EAP 6. Ajoutez ce qui suit pour démarrer les paramètres.

```
-Djboss.security.disable.secdomain.option=true
```

Vous pourrez également définir cette propriété par la console de gestion. Dans un serveur autonome, vous pouvez définir les propriétés de système dans la section **Profile** de configuration. Dans un domaine géré, vous pouvez définir les propriétés du système pour chaque groupe de serveur.

Rapporter un bogue

11.6.5. L'autorisation

L'autorisation est un mécanisme d'octroi ou de refus de permission d'accéder à une ressource, basé sur l'identité. Ce mécanisme est implémenté sous forme de rôles de sécurité déclarative, qui peuvent être donnés à des principaux.

JBoss EAP utilise un système modulaire pour configurer l'autorisation. Chaque domaine de sécurité peut contenir une ou plusieurs stratégies d'autorisation. Chaque stratégie possède un module de base qui définit son comportement. Il est configuré via les attributs et indicateurs spécifiques. La façon la plus simple consiste à configurer le sous-système de l'autorisation à l'aide de la console de gestion basée web.

L'authentification est différente de l'autorisation, et a lieu, en général, après l'authentification. La plupart des modules d'authentification gèrent également l'autorisation.

Rapporter un bogue

11.6.6. Configurer l'autorisation pour un domaine de sécurité

Pour configurer les paramètres d'un domaine de sécurité, connectez-vous à la console de gestion et suivre la procédure suivante :

Procédure 11.3. Configurez l'autorisation pour un domaine de sécurité

1. **Ouvrir l'affichage détaillé du domaine de sécurité.**
   a. Cliquez sur l'étiquette **`Configuration`** qui se trouve en haut de la console de gestion.
   b. Sélectionnez le profil à modifier à partir du menu déroulant **`Profile`** en haut et à gauche.
   c. Étendre l'item de menu **`Security`**, puis sélectionnez **`Security Domains`**.
   d. Cliquez sur le lien **`View`** pour obtenir le domaine de sécurité que vous voulez modifier.
2. **Naviguez dans la configuration du sous-système d'autorisation.**
   Sélectionnez l'étiquette **`Authorisation`** en haut de l'écran.

   La zone de configuration est divisée en deux: **`Policies`** et **`Details`**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune d'elle peut inclure plusieurs attributs ou options.
3. **Ajoutez une police.**
   Cliquez sur **`Add`** pour ajouter un module de police d'authentification JAAS. Remplissez les informations pour votre module.

   Le **`Code`** est le nom de classe de votre module. Les **`Flags`** controlent la façon dont le module est lié à d'autres modules de polices d'authentification du même domaine de sécurité.

   Explication des indicateurs

   La spécification Java Enterprise Edition 6 fournit l'explication suivante sur les marqueurs des modules de sécurité. La liste suivante provient de http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html#Appendix Consultez ce document pour obtenir des informations plus détaillées.

Marqueur	Détails
requis	Le LoginModule est requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule.

Marqueur	Détails
pré-requis	Le LoginModule est requis pour la réussite. En cas de succès, l'autorisation continuera vers le bas de la liste LoginModule. En cas d'échec, le contrôle est renvoyé immédiatement à l'application (l'autorisation ne continue pas son chemin le long de la liste LoginModule).
suffisant	Le LoginModule n'est pas nécessaire pour aboutir. S'il ne réussit pas, le contrôle retourne immédiatement à l'application (l'autorisation ne se déroule pas vers le bas de la liste LoginModule). S'il échoue, l'authentification se poursuit vers le bas de la liste LoginModule.
option	Le LoginModule n'est pas requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule.

4. **Modifier les paramètres d'authorisation**
   Une fois que vous aurez ajouté votre module, vous pourrez modifier son **`Code`** ou ses **`Flags`** (indicateurs) en cliquant **`Edit`** dans la section **`Details`** de l'écran. Veillez à ce que l'onglet **`Attributes`** soit bien sélectionné.

5. **Option : ajouter ou supprimer un module**
   Pour ajouter des options à votre module, cliquez sur leurs entrées dans la liste **`Policies`**, et sélectionnez l'onglet **`Module Options`** dans la section **`Details`** qui se trouve en bas de la page. Cliquez sur **`Add`**, et fournir la clé et la valeur de l'option. Utilisez le bouton **`Remove`** pour supprimer l'option.

Résultat

Votre module de police de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Rapporter un bogue

11.6.7. Security Auditing

Security Auditing se réfère au déclenchement d'événements, comme écrire un blog en réponse à un événement qui a lieu dans le sous-système de sécurité. Les mécanismes de sécurité sont configurés dans le cadre du domaine de sécurité, avec les informations d'authentification, d'autorisation ou de mappage de sécurité.

Auditing utilise *des modules de fournisseur*. Vous pouvez en utiliser un existant ou bien créer le vôtre.

Rapporter un bogue

11.6.8. Configurer Security Auditing

Pour configurer les paramètres de Security Auditing, connectez-vous à la console de gestion et suivre la procédure suivante :

Procédure 11.4. Configurez Security Auditing pour un domaine de sécurité

1. **Ouvrir l'affichage détaillé du domaine de sécurité.**

   a. Cliquez sur **Configuration** qui se trouve en haut de l'écran.

   b. Dans un domaine géré, sélectionnez le profil à modifier à partir du menu déroulant **Profile** en haut et à gauche.

   c. Étendre le menu **Security**, puis sélectionnez **Security Domains**.

   d. Cliquez sur **View** pour obtenir le domaine de sécurité que vous voulez modifier.

2. **Naviguez dans la configuration du sous-système Auditing.**
   Sélectionnez l'onglet **Audit** qui se trouve en haut et à droite.

   La zone de configuration est divisée en deux : **Provider Modules** et **Details**. Le module de fournisseur est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune peut inclure plusieurs attributs et options.

3. **Ajoutez un module de fournisseur.**
   Cliquez sur **Add**. Remplir la section **Code** avec le nom de classe du module du fournisseur.

4. **Vérifiez que le module fonctionne**
   Le but d'un module d'auditing est d'offrir un moyen de surveiller les événements dans le sous-système de sécurité. Ce monitoring peut être réalisé sous la forme d'écriture dans un fichier de journalisation, des notifications email ou autre mécanisme d'audit mesurable.

   Par exemple, JBoss EAP 6 inclut le module **LogAuditProvider** par défaut. S'il est activé par les étapes décrites ci-dessus, ce module d'audit écrira des notifications de sécurité dans un fichier **audit.log** qui se situe dans le sous-dossier **log** du répertoire ***EAP_HOME***.

   Pour vérifier si les étapes ci-dessus fonctionnent dans le contexte **LogAuditProvider**, procédez à une action qui risque de déclencher une notification, puis vérifiez le fichier de journalisation de l'audit.

   Pour obtenir une liste complète des Modules Security Auditing Provider, voir : Section 12.4, « Modules de fournisseurs d'auditing de sécurité inclus »

5. **Option : ajouter, éditer ou supprimer un module**
   Pour ajouter des options à votre module, cliquez sur leurs entrées dans la liste **Modules**, et sélectionnez l'onglet **Module Options** dans la section **Details** qui se trouve en bas de la page. Cliquez sur **Add**, et fournir la clé et la valeur de l'option.

   Pour modifier une option déjà existante, cliquez sur **Remove** pour la supprimer, et cliquez sur **Add** pour l'ajouter à nouveau avec les options qui conviennent.

Résultat

Votre module d'audit de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Rapporter un bogue

11.6.9. Audit Log

Si le module **LogAuditProvider** est activé, JBoss EAP 6 maintient un journal d'audit qui enregistre l'authentification et l'autorisation dans les appllications et les modules de login. Le fichier nommé **audit.log** par défaut est stocké dans le sous-répertoire **log** du répertoire *EAP_HOME*. Ce comportement est déterminé par la configuration des gestionnaires de journaux du sous-système de journalisation. La sortie du module **LogAuditProvider** peut être envoyée à un serveur **syslog** également ou à la place d'un fichier, en utilisant le gestionnaire de journaux **syslog**.

Par défaut, le module **LogAuditProvider** vont uniquement dans un fichier **audit.log** cumulatif. Pour implémenter un gestionnaire de rotation périodique des fichiers nommée **AUDIT**, saisir l'interface CLI suivante.

```
/subsystem=logging/periodic-rotating-file-handler=AUDIT/:add(suffix=.yyyy-
MM-dd,formatter=%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n,level=TRACE,file=
{"relative-to" => "jboss.server.log.dir","path" => "audit.log"})
```

Voir également :

- Section 14.1.12, « Gestionnaires de journaux »

Rapporter un bogue

11.6.10. Security Mapping

Le mappage de sécurité vous permet de combiner l'authentification et l'autorisation d'informations après que l'authentification ou l'autorisation aient eu lieu, mais avant que l'information ait été passée à votre application. Un exemple qui l'illustre est l'utilisation d'un certificat X509 pour l'authentification, puis la conversion du principal du certificat en nom logique que votre application puisse afficher.

Vous pouvez mapper vos principaux (authentification), rôles (autorisation), ou identifiants (attributs qui ne sont ni principaux, ni rôles).

Le mappage de rôles est utilisé pour ajouter, remplacer ou supprimer des rôles du sujet après l'authentification.

Le mappage du principal est utilisé pour modifier un principal après l'authentification.

Le mappage d'attributs est utilisé pour convertir des attributs d'un système externe à utiliser par votre application, et vice versa.

Rapporter un bogue

11.6.11. Configurer le mappage de sécurité dans un domaine de sécurité

Pour configurer les paramètres du mappage de sécurité, vous connecter à la console de gestion et suivre la procédure suivante :

Procédure 11.5. Configurer le mappage de sécurité pour un domaine de sécurité

1. **Ouvrir l'affichage détaillé du domaine de sécurité.**
   a. Cliquez sur l'étiquette **Configuration** qui se trouve en haut de la console de gestion.
   b. Dans un domaine géré, sélectionnez le profil à partir du menu déroulant **Profile** en haut et à gauche.

c. Étendre l'item de menu **Security**, puis sélectionnez **Security Domains**.

d. Cliquez sur **View** pour obtenir le domaine de sécurité que vous voulez modifier.

2. **Naviguez dans la configuration du sous-système de mappage.**
Sélectionnez l'étiquette **Mapping** en haut de l'écran.

   La zone de configuration est divisée en deux : **Modules** et **Details**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices de mapping, et chacune peut inclure plusieurs attributs et options.

3. **Ajouter un module de mappage de sécurité.**
Cliquez sur **Ajouter**.

   Remplissez les informations de module. Le **Code** est le nom de classe du module. Le champ **Type** se rapporte au type de mapping effectué par ce module. Les valeurs autorisées sont les suivantes : principal, role, attribut ou identifiant.

4. **Modifier un module de mappage de sécurité**
Une fois que vous aurez ajouté votre module, vous pourrez modifer son **Code** ou son **Type**.

   a. Sélectionnez l'onglet **Attributes**.

   b. Cliquez sur **Edit** dans le champ **Details** de cet écran.

5. **Option : ajouter, éditer ou supprimer un module**
Pour ajouter des options à votre module, cliquez sur leurs entrées dans la liste **Modules**, et sélectionnez l'onglet **Module Options** dans la section **Details** qui se trouve en bas de la page. Cliquez sur **Add**, et fournir la clé et la valeur de l'option.

   Pour modifier une option déjà existante, cliquez sur **Remove** pour la supprimer, et cliquez sur add pour l'ajouter à nouveau avec la nouvelle valeur.

   Utilisez **Remove** pour supprimer une option.

Résultat

Votre module de mappage de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Rapporter un bogue

11.6.12. Utiliser un domaine de sécurité dans votre application

Aperçu

Pour utiliser un domaine de sécurité dans votre application, vous devez tout d'abord définir le domaine dans le fichier de configuration du serveur, puis vous devez l'activer pour une application dans le descripteur de déploiement de l'application. Ensuite, vous devez ajouter les annotations requises à l'EJB qui les utilise. Cette rubrique décrit les étapes requises pour utiliser un domaine de sécurité dans votre application.

AVERTISSEMENT

Si une application fait partie d'un domaine de sécurité qui utilise un cache d'authentification, les authentifications utilisateur de cette application seront rendues disponibles à d'autres applications dans ce domaine de sécurité.

Procédure 11.6. Configurez votre application pour qu'elle puisse utiliser un domaine de sécurité

1. **Définir le domaine de sécurité**
   Vous devez définir le domaine de sécurité dans le fichier de configuration du serveur, puis l'activer pour une application dans le fichier du descripteur de l'application.

   a. **Configurez le domaine de sécurité dans le fichier de configuration du serveur**
   Le domaine de sécurité est configuré dans le sous-système de **`sécurité`** du fichier de configuration du serveur. Si l'instance de JBoss EAP 6 s'exécute dans un domaine géré, il s'agira du fichier **`domain/configuration/domain.xml`**. Si l'instance de JBoss EAP 6 s'exécute comme un serveur autonome, ce sera le fichier **`standalone/configuration/standalone.xml`**.

   Les domaines de sécurité **`other`**, **`jboss-web-policy`**, et **`jboss-ejb-policy`** sont fournis par défaut dans JBoss EAP 6. L'exemple XML suivant a été copié à partir du sous-système de **`sécurité`** dans le fichier de configuration du serveur.

   L'attribut **`cach-type`** d'un domaine de sécurité spécifie un cache pour pouvoir effectuer des contrôles d'authentification plus rapides. Les valeur autorisées sont les valeurs par **`défaut`** en cas de simple mappe comme cache ou **`infinispan`** pour un cache Infinispan.

```
<subsystem xmlns="urn:jboss:domain:security:1.2">
    <security-domains>
        <security-domain name="other" cache-type="default">
            <authentication>
                <login-module code="Remoting" flag="optional">
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
                <login-module code="RealmDirect"
flag="required">
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
            </authentication>
        </security-domain>
        <security-domain name="jboss-web-policy" cache-
type="default">
            <authorization>
                <policy-module code="Delegating"
flag="required"/>
            </authorization>
        </security-domain>
        <security-domain name="jboss-ejb-policy" cache-
type="default">
```

```
            <authorization>
                <policy-module code="Delegating"
flag="required"/>
            </authorization>
        </security-domain>
    </security-domains>
</subsystem>
```

Vous pouvez configurer des domaines de sécurité supplémentaires selon les besoins par la console de gestion ou par l'interface CLI.

b. **Activer le domaine de sécurité dans le fichier de descripteur de l'application.**
Le domaine de sécurité est spécifié dans l'élément enfant **`<security-domain>`** de l'élément **`<jboss-web>`** du fichier **`WEB-INF/jboss-web.xml`** de l'application. L'exemple suivant configure un domaine de sécurité nommé **`my-domain`**.

```
<jboss-web>
    <security-domain>my-domain</security-domain>
</jboss-web>
```

Il s'agit d'une des configurations que vous pouvez indiquer dans le descripteur **`WEB-INF/jboss-web.xml`**.

2. **Ajoutez l'annotation requise à l'EJB.**
Vous pouvez configurer la sécurité dans EJB par les annotations **`@SecurityDomain`** et **`@RolesAllowed`**. L'exemple de code EJB suivant limite l'accès au domaine de sécurité **`other`** aux utilisateurs ayant pour rôle **`guest`** (invité).

```
package example.ejb3;

import java.security.Principal;

import javax.annotation.Resource;
import javax.annotation.security.RolesAllowed;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;

import org.jboss.ejb3.annotation.SecurityDomain;

/**
 * Simple secured EJB using EJB security annotations
 * Allow access to "other" security domain by users in a "guest"
role.
 */
@Stateless
@RolesAllowed({ "guest" })
@SecurityDomain("other")
public class SecuredEJB {

   // Inject the Session Context
   @Resource
   private SessionContext ctx;

   /**
    * Secured EJB method using security annotations
```

```
     */
    public String getSecurityInfo() {
       // Session context injected using the resource annotation
       Principal principal = ctx.getCallerPrincipal();
       return principal.toString();
    }
}
```

Pour obtenir des exemples de code supplémentaires, voir **`ejb-security`** Quickstart dans le package JBoss EAP 6 Quickstarts disponible à partir du portail clients de Red Hat.

Rapporter un bogue

11.6.13. Java Authorization Contract for Containers (JACC)

11.6.13.1. Java Authorization Contract for Containers (JACC)

Java Authorization Contract for Containers (JACC) est une norme qui définit un contrat entre les conteneurs et les fournisseurs de services d'autorisation, et qui se traduit par l'implémentation de fournisseurs à utiliser par des conteneurs. Il a été défini dans JSR-115, qui se trouve sur le site Web de Java Community Process http://jcp.org/en/jsr/detail?id=115. A fait partie de la spécification Java Enterprise Edition (Java EE) depuis Java EE version 1.3.

JBoss EAP 6 implémente un support pour JACC dans la fonctionnalité de sécurité du sous-système de sécurité.

Rapporter un bogue

11.6.13.2. Configurer la sécurité JACC (Java Authorization Contract for Containers)

Pour configurer JACC (Java Authorization Contract for Containers), il convient de configurer votre domaine de sécurité avec le module qui convient, puis de modifier votre fichier **`jboss-web.xml`** pour y inclure les paramètres qu'il faut.

Ajouter JACC Support au domaine de sécurité

Pour ajouter JACC au domaine de sécurité, ajouter la police d'autorisation **`JACC`** à la pile d'autorisations du domaine de sécurité, avec l'indicateur **`requis`**. Voici un exemple de domaine de sécurité avec support JACC. Cependant, le domaine de sécurité est configuré dans la console de gestion ou dans l'interface CLI, plutôt que directement dans le code XML.

```
<security-domain name="jacc" cache-type="default">
    <authentication>
        <login-module code="UsersRoles" flag="required">
        </login-module>
    </authentication>
    <authorization>
        <policy-module code="JACC" flag="required"/>
    </authorization>
</security-domain>
```

Configurer une application web qui utilise JACC

Le fichier **`jboss-web.xml`** se trouve dans **`META-INF/`** ou dans le répertoire **`WEB-INF/`** de votre déploiement, et contient des ajouts ou remplacements de configuration spécifique JBoss pour le

conteneur web. Pour utiliser votre domaine de sécurité activé-JACC, vous devrez inclure l'élément **<security-domain>**, et aussi définir l'élément **<use-jboss-authorization>** sur **true**. L'application suivante est configurée correctement pour pouvoir utiliser le domaine de sécurité JACC ci-dessus.

```
<jboss-web>
    <security-domain>jacc</security-domain>
    <use-jboss-authorization>true</use-jboss-authorization>
</jboss-web>
```

Configurer une application EJB pour utiliser JACC

La façon de configurer les EJB à utiliser un domaine de sécurité et JACC diffère selon des application Web. Pour un EJB, vous pouvez déclarer des *method permissions* sur une méthode ou sur un groupe de méthodes, dans le descripteur **ejb-jar.xml**. Dans l'élément **<ejb-jar>**, chaque élément **<method-permission>** dépendant contient des informations sur les roles JACC. Voir l'exemple de configuration pour plus d'informations. La classe **EJBMethodPermission** fait partie de Java Enterprise Edition 6 API, et est documentée dans http://docs.oracle.com/javaee/6/api/javax/security/jacc/EJBMethodPermission.html.

Exemple 11.4. Exemple de permissions de méthode JACC dans un EJB

```
<ejb-jar>
  <assembly-descriptor>
    <method-permission>
      <description>The employee and temp-employee roles may access any
method of the EmployeeService bean </description>
      <role-name>employee</role-name>
      <role-name>temp-employee</role-name>
      <method>
        <ejb-name>EmployeeService</ejb-name>
        <method-name>*</method-name>
      </method>
    </method-permission>
  </assembly-descriptor>
</ejb-jar>
```

Vous pouvez également contraindre les mécanismes d'authentification et d'autorisation d'un EJB à l'aide d'un domaine de sécurité, comme vous pouvez le faire pour une application web. Les domaines de sécurité sont déclarés dans le descripteur **jboss-ejb3.xml** qui se trouve dans l'élément enfant **<security>**. En plus du domaine de sécurité, vous pouvez également spécifier le*run-as principal*, qui change le principal que l'EJB exécute.

Exemple 11.5. Exemple de déclaration de domaine de sécurité dans un EJB

```
<security>
  <ejb-name>*</ejb-name>
  <security-domain>myDomain</security-domain>
  <run-as-principal>myPrincipal</run-as-principal>
</security>
```

Rapporter un bogue

11.6.14. Java Authentication SPI for Containers (JASPI)

11.6.14.1. Sécurité Java Authentication SPI pour Conteneurs (JASPI)

Java Application SPI pour Conteneurs (JASPI or JASPIC) est une interface enfichable pour applications JSR-196 du Java Community Process. Consulter http://www.jcp.org/en/jsr/detail?id=196 pour obtenir des informations sur la spécification.

Rapporter un bogue

11.6.14.2. Configuration de la sécurité Java Authentication SPI pour conteneurs (JASPI)

Pour s'authentifier auprès d'un fournisseur JASPI, ajouter un élément **<authentication-jaspi>** à votre domaine de sécurité. La configuration est similaire à celle d'un module d'authentification standard, mais les éléments de module de login sont inclus dans l'élément **<login-module-stack>**. La structure de configuration est la suivante :

Exemple 11.6. Structure de l'élément `authentication-jaspi`

```
<authentication-jaspi>
 <login-module-stack name="...">
   <login-module code="..." flag="...">
     <module-option name="..." value="..."/>
   </login-module>
 </login-module-stack>
 <auth-module code="..." login-module-stack-ref="...">
   <module-option name="..." value="..."/>
 </auth-module>
</authentication-jaspi>
```

Le module de connexion est lui-même configuré de la même façon que le module d'authentification standard.

Comme la console de gestion basée web n'expose pas la configuration des modules d'authentification JASPI, vous devez stopper la plateforme JBoss EAP 6 complètement avant d'ajouter la configuration directement dans le fichier ***EAP_HOME*/domain/configuration/domain.xml** ou dans le fichier ***EAP_HOME*/standalone/configuration/standalone.xml**.

Rapporter un bogue

11.7. SÉCURISATION D'IIOP

11.7.1. JBoss IIOP

IIOP (Internet Inter-ORB Protocol) est le protocole de communication utilisé par les clients CORBA pour appeler les fonctions à distance sur un serveur CORBA. JBoss IIOP prend en charge les accès CORBA/IIOP des beans enterprise déployés sur un serveur JBoss EAP, tels qu'ils sont définis par la spécification EJB. Cela rend les méthodes de beans enterprise disponibles aux clients RMI et IIOP écrites en Java et aux clients CORBA écrites en Java, C++ ou autres langages. Le moteur IIOP peut être remplacé par un autre, dans la mesure où il est pleinement conforme à CORBA. Le moteur par défaut est JacORB, une implémentation libre du standard CORBA.

Rapporter un bogue

11.7.2. IOR

Un IOR (Interoperable Object Reference) est utilisé dans un système CORBA pour identifier des objets. Il se compose de plusieurs parties :

- Type d'objet.
- Nom d'hôte du serveur.
- Numéro de port du serveur.
- Clé d'objet qui identifie l'objet.

Le nom d'hôte et le numéro de port qui sont utilisés pour communiquer avec le serveur et la clé d'objet utilisée par le serveur pour identifier l'objet.

Rapporter un bogue

11.7.3. Paramètres de sécurité IOR

Conditions préalables

- Activer les paramètres de configuration IOR par la commande de Management CLI suivante.

```
/subsystem=jacorb/ior-settings=default:add
```

- Veillez à ce que le sous-système JacORB soit activé. Ainsi, il peut être activé dans le profil **`full`**, mais pas dans le profil par défaut (**`web`**). Pour obtenir des informations sur la façon d'activer le sous-système JacORB, voir Section 21.4.2, « Configurer l'ORB pour les transactions JTS ».

Le iorSASContextType indique les attributs à utiliser pour configurer les paramètres du service IOR Secure Attribute Service. Ces paramètres ne peuvent être définis qu'à l'aide du CLI.

Tableau 11.1. iorSASContextType

Paramètre	Description	Valeurs valides
`caller-propagation`	Indique si l'appelant doit être ou non propagé dans le contexte SAS	**`none`**, **`supported`**

```
/subsystem=jacorb/ior-settings=default/setting=sas-context:add
/subsystem=jacorb/ior-settings=default/setting=sas-context:write-
attribute(name=caller-propagation, value=NONE|SUPPORTED)
```

Le iorSASContextType indique les attributs à utiliser pour configurer les paramètres du service IOR Secure Attribute Service. Chacun de ces paramètres est optionnel.

Tableau 11.2. iorASContextType

Paramètre	Description	Type	Valeurs valides
auth-method	Méthode d'authentification.	String	**none**, **username_password**
realm	Nom de domaine du service d'authentification.	String	Valeur par défaut : **Default**
required	Indique si une authentification est requise.	Booléen	**true**, **false**

```
/subsystem=jacorb/ior-settings=default/setting=as-context:add
/subsystem=jacorb/ior-settings=default/setting=as-context:write-
attribute(name=ATTRIBUTE, value=VALUE)
```

Le iorTransportconfigType spécifie les attributs qui sont utilisés pour définir les paramètres de transport IOR.

Tableau 11.3. iorTransportconfigType

Paramètre	Description	Valeurs valides
integrity	Indique si le transport requiert ou non une protection d'intégrité du transport.	**none**, **supported**, ou **required**.
confidentiality	Indique si le transport requiert ou non une protection pour la confidentialité du transport.	**none**, **supported**, ou **required**.
trust-in-target	Indique si le transport requiert ou non une confiance de la cible.	**none**, **supported**
trust-in-client	Indique si le transport requiert ou non une confiance du client.	**none**, **supported**, ou **required**.
detect-replay	Indique si le transport requiert ou non un replay de détection du transport.	**none**, **supported**, ou **required**.
detect-misordering	Indique si le transport requiert ou non une détection du misordering du transport.	**none**, **supported**, ou **required**.

```
/subsystem=jacorb/ior-settings=default/setting=transport-config:add
/subsystem=jacorb/ior-settings=default/setting=transport-config:write-
attribute(name=ATTRIBUTE, value=VALUE)
```

Rapporter un bogue

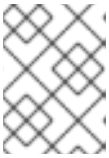
11.8. SÉCURITÉ DANS L'INTERFACE DE GESTION

11.8.1. Configuration de la sécurité utilisateur par défaut

Introduction

Toutes les interfaces de gestion de JBoss EAP 6 sont sécurisées par défaut. Cette sécurité existe sous deux formes :

- Les interfaces locales sont sécurisées par un contrat SASL entre des clients locaux et le serveur auquel ils se connectent. Ce mécanisme de sécurité repose sur la capacité du client à accéder au système de fichiers local. C'est parce que l'accès au système de fichiers local permettait au client d'ajouter un utilisateur ou encore de modifier la configuration pour déjouer les autres mécanismes de sécurité. Cela est conforme au principe selon lequel si l'accès physique au système de fichiers est possible, les autres mécanismes de sécurité sont alors superflus. Le mécanisme passe par quatre étapes :

  NOTE

  L'accès HTTP est considéré comme éloigné, même si vous vous connectez à l'hôte local par HTTP.

  1. Le client envoie un message au serveur incluant une requête pour authentifier le mécanisme SASL local.
  2. Le serveur génère un token unique, qu'il écrit dans un fichier unique, et envoie un message au client avec le chemin d'accès fichier complet.
  3. Le client lit le token dans le fichier et l'envoie au serveur, pour vérifier qu'il ait bien un accès local au système de fichiers.
  4. Le serveur vérifie le token et efface le fichier.

- Les clients distants, y compris HTTP, utilisent une sécurité basée domaine. Le domaine par défaut, qui comprend les autorisations pour configurer l'instance JBoss EAP 6 à distance en utilisant les interfaces de gestion, est **`ManagementRealm`**. Un script est fourni pour vous permettre d'ajouter des utilisateurs à ce domaine (ou à des domaines que vous créez). Pour plus d'informations sur la façon d'ajouter des utilisateurs, voir le chapitre *Getting Started* de *JBoss EAP 6 Installation Guide*. Pour chaque utilisateur, le nom d'utilisateur et un mot de passe haché sont stockés dans un fichier.

  Domaine géré

  ***`EAP_HOME`*`/domain/configuration/mgmt-users.properties`**

  Serveur autonome

  ***`EAP_HOME`*`/standalone/configuration/mgmt-users.properties`**

  Même si les contenus de **`mgmt-users.properties`** sont masqués, le fichier doit toujours être considéré comme un fichier sensible. Il est recommandé qu'il soit défini sur le mode de fichier **`600`**, qui ne donne aucun accès autre que l'accès en lecture et écriture au propriétaire du fichier.

Rapporter un bogue

11.8.2. Aperçu général de la configuration de l'interface de gestion avancée

La configuration de l'interface de gestion ***EAP_HOME*/domain/configuration/host.xml** ou ***EAP_HOME*/standalone/configuration/standalone.xml** contrôle l'interface réseau à laquelle se relie le processus contrôleur hôte, les types d'interfaces de gestion qui sont disponibles à tous, et quel type de système d'authentification est utilisé pour authentifier les utilisateurs sur chaque interface. Cette rubrique explique comment configurer les interfaces de gestion en fonction de votre environnement.

Le sous-système de gestion est formé d'un élément **<management>** avec plusieurs attributs configurables, et les trois éléments enfants configurables suivants. Les domaines de sécurité et les connexions sortantes sont tout d'abord définis, puis appliqués aux interfaces de gestion en tant qu'attributs.

- <security-realms>
- <outbound-connections>
- <management-interfaces>
- <audit-log>

NOTE

Voir la section *Management Interface Audit Logging* du guide *Administration and Configuration Guide* pour obtenir plus d'informations sur la journalisation d'auditing.

Domaines de sécurité

Le domaine de sécurité est chargé de l'authentification et de permettre aux utilisateurs autorisés d'administrer JBoss EAP 6 via l'API de gestion, l'interface CLI ou la console de gestion basée-web.

Il existe deux domaines de sécurité basés fichier différents, qui existent dans l'installation par défaut : **ManagementRealm** et **ApplicationRealm**. Chacun de ces domaines de sécurité utilise un fichier **-users.properties** pour stocker les utilisateurs et les mots de passe de hachage, et **-roles.properties** pour stocker les correspondances entre les utilisateurs et les rôles. Un support est également inclus pour le domaine de sécurité activé-LDAP.

NOTE

Les domaines de sécurité peuvent également être utilisés pour vos propres applications. Les domaines de sécurité dont il s'agit sont particuliers aux interfaces de gestion.

Les connexions de sortie

Certains domaines de sécurité se connectent à des interfaces externes, comme un serveur LDAP. Une connexion sortante définit la manière d'établir cette connexion. Un type de connexion prédéfinie, la connexion **ldap-connection**, définit tous les attributs obligatoires et facultatifs pour se connecter au serveur LDAP et vérifier les informations d'identification.

Pour obtenir plus d'informations sur la façon de configurer l'authentification LDAP, voir Section 11.8.4, « Utiliser LDAP pour vous authentifier auprès des interfaces de gestion ».

Interfaces de gestion

Une interface de gestion comprend des propriétés sur la façon de connecter et configurer JBoss EAP. Ces informations comprennent l'interface réseau nommée, le port, le domaine de sécurité et d'autres informations configurables sur l'interface. Deux interfaces sont incluses dans une installation par défaut :

- **`http-interface`** est la configuration de la console de gestion basée-web.
- **`native-interface`** est la configuration de l'interface CLI en ligne de commande et l'API de gestion Rest-like.

Chacun des quatre principaux éléments configurables du sous-système de gestion de l'hôte sont étroitement liés. Un domaine de sécurité fait référence à une connexion sortante et une interface de gestion à un domaine de sécurité.

On peut trouver des informations associées dans Section 11.10.1, « Sécuriser les interfaces de gestion ».

Rapporter un bogue

11.8.3. LDAP

Lightweight Directory Access Protocol (LDAP) est un protocole pour le stockage et la distribution d'informations en provenance de répertoires à travers un réseau. Ces informations de répertoires incluent des informations sur les utilisateurs, les périphériques physiques, les rôles d'accès et de restrictions et autres informations.

Certaines de implémentations communes de LDAP incluent OpenLDAP, Microsoft Active Directory, IBM Tivoli Directory Server, Oracle Internet Directory, et autres.

La plateforme JBoss EAP 6 comprend plusieurs modules d'authentification et d'autorisation qui vous permettent d'utiliser un serveur LDAP comme autorité d'authentification et d'autorisation pour vos applications Web et EJB.

Rapporter un bogue

11.8.4. Utiliser LDAP pour vous authentifier auprès des interfaces de gestion

Pour utiliser un serveur de répertoire LDAP comme source d'authentification pour la console de gestion, l'interface CLI ou l'API de gestion, vous devez effectuer les procédures suivantes :

1. Créer une connexion sortante au serveur LDAP.
2. Créer un domaine de sécurité activé-LDAP.
3. Référencer le nouveau domaine de sécurité dans l'interface de gestion.

Créer une connexion sortante au serveur LDAP

La connexion sortante LDAP autorise les attributs suivants :

Tableau 11.4. Attributs d'une connexion sortante LDAP

Attribut	Requis	Description

Attribut	Requis	Description
`url`	oui	L'adresse URL du serveur de répertoire.
`search-dn`	non	Le nom distinctif (DN) de l'utilisateur autorisé à effectuer des recherches.
`search-credentials`	non	Le mot de passe de l'utilisateur autorisé à effectuer des recherches.
`initial-context-factory`	non	L'usine de contexte initiale à utiliser quand on établit une connexion. Valeur par défaut **`com.sun.jndi.ldap.LdapCtxFactory`**.
`security-realm`	non	Domaine de sécurité à référencer pour obtenir un **`SSLContext`** configuré pour établir la connexion.

Exemple 11.7. Ajouter une connexion sortante LDAP

Cet exemple ajoute une connexion sortante par le jeu de propriétés suivant :

- Recherche DN : **`cn=search,dc=acme,dc=com`**
- Recherche indentifiants : **`myPass`**
- URL : **`ldap://127.0.0.1:389`**

La première commande ajoute le domaine de sécurité.

```
/host=master/core-service=management/security-
realm=ldap_security_realm:add
```

La seconde commande ajoute la connexion LDAP.

```
/host=master/core-service=management/ldap-
connection=ldap_connection/:add(search-
credential=myPass,url=ldap://127.0.0.1:389,search-
dn="cn=search,dc=acme,dc=com")
```

Créer un domaine de sécurité activé-LDAP

Les interfaces de gestion peuvent s'authentifier sur le serveur LDAP au lieu des domaines de sécurité basés fichiers de propriétés et configurés par défaut. L'authentificateur LDAP fonctionne en établissant tout d'abord une connexion vers le serveur de répertoires distant. Il effectue ensuite une recherche en utilisant le nom d'utilisateur que l'utilisateur a transmis au système d'authentification, afin de trouver le

nom distinctif complet (DN) du dossier LDAP. Une nouvelle connexion est alors établie, utilisant le DN de l'utilisateur comme information d'identification et mot de passe fournis par l'utilisateur. Si cette authentification au serveur LDAP réussit, le DN est considéré comme valide.

Le domaine de sécurité LDAP utilise les attributs de configuration suivants :

connection

Le nom de la connexion défini dans ***`outbound-connections`*** à utiliser pour se connecter au répertoire LDAP.

advanced-filter

Le filtre complet utilisé pour rechercher un utilisateur basé sur l'ID d'utilisateur fourni. Le filtre doit contenir une variable suivant le format suivant : **`{0}`**. Sera plus tard remplacé par le nom d'utilisateur fourni par l'utilisateur.

base-dn

Le nom distinctif (DN) du contexte pour commencer à chercher l'utilisateur.

recursive

Indique si la recherche doit être récursive dans toute l'arborescence de répertoires LDAP, ou si l'on doit rechercher uniquement le contexte spécifié. La valeur par défaut est **`false`**.

user-dn

Attribut de l'utilisateur qui contient le nom distinctif (DN). Utilisé par la suite pour tester l'authentification. Valeur par défaut **dn**.

username-attribute

Le nom de l'attribut à rechercher pour l'utilisateur. Ce filtre effectue une recherche simple où le nom d'utilisateur entré par l'utilisateur correspond à l'attribut spécifié.

allow-empty-passwords

Cet attribut détermine si un mot de passe vide est accepté. La valeur par défaut pour cet attribut est **`false`**.

Soit `username-filter` ou `advanced-filter`, devra être spécifié.

L'attribut ***`advanced-filter`*** contient une recherche par filtre dans la syntaxe standard LDP, par exemple :

```
(&(sAMAccountName={0})(memberOf=cn=admin,cn=users,dc=acme,dc=com))
```

Exemple 11.8. XML représentant un domaine de sécurité activé-LDAP

Cet exemple utilise les paramètres suivants :

- connection - **`ldap_connection`**
- base-dn - **`cn=users,dc=acme,dc=com`**.
- username-filter - **`attribute="sambaAccountName"`**

```
<security-realm name="ldap_security_realm">
   <authentication>
      <ldap connection="ldap_connection" base-
dn="cn=users,dc=acme,dc=com">
         <username-filter attribute="sambaAccountName" />
      </ldap>
  </authentication>
</security-realm>
```

AVERTISSEMENT

Il est important de veiller à ne pas autoriser les mots de passe LDAP. À moins que vous désiriez les avoir dans votre environnement, ils représentent un problème de sécurité sérieux.

EAP 6.1 inclut un correctif pour CVE-2012-5629, qui définit l'option allowEmptyPasswords des modules de connexion LDAP à false si l'option n'est pas déjà configurée. Pour les versions plus anciennes, cette option devra être configurée manuellement.

Exemple 11.9. Ajout d'un domaine de sécurité LDAP

La commande ci-dessous ajoute une authentification LDAP au domaine de sécurité et définit ses attributs pour un master nommé par l'hôte dans le domaine.

```
/host=master/core-service=management/security-
realm=ldap_security_realm/authentication=ldap:add(base-
dn="DC=mycompany,DC=org", recursive=true, username-
attribute="MyAccountName", connection="ldap_connection")
```

Appliquer le nouveau domaine de sécurité à l'interface de gestion

Après avoir créé un domaine de sécurité, vous devez le référencer dans la configuration de votre interface de gestion. L'interface de gestion utilisera le domaine de sécurité pour l'authentification HTTP digest.

Exemple 11.10. Ajouter le domaine de sécurité à l'interface HTTP

Une fois que la configuration est en place, et que vous aurez démarré à nouveau le contrôleur hôte, la console de gestion basée-web utilisera LDAP pour authentifier ses utilisateurs.

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=security-
realm,value=ldap_security_realm)
```

Exemple 11.11. Ajouter le domaine de sécurité à l'interface native

Utiliser la commande suivante pour appliquer les mêmes paramètres à l'interface native :

```
/host=master/core-service=management/management-interface=native-
interface/:write-attribute(name=security-
realm,value=ldap_security_realm)
```

Rapporter un bogue

11.8.5. Désactiver l'interface de gestion HTTP

Dans un domaine géré, vous avez seulement besoin d'un accès à l'interface HTTP sur le contrôleur de domaine, plutôt que sur des serveurs membres de domaine. En outre, sur un serveur de production, vous pouvez finalement décider de désactiver la console de gestion basée-web.

NOTE

Les autres clients, tels que JBoss Operations Network, opèrent également par l'interface HTTP. Si vous souhaitez utiliser ces services ou tout simplement désactiver la gestion de la console elle-même, vous pouvez définir l'attribut **`console-enabled`** de l'interface HTTP à **`false`** au lieu de désactiver l'interface complètement.

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=console-enabled,value=false)
```

Pour désactiver l'interface HTTP, ce qui désactive également l'accès à la console de gestion basée-web, vous pouvez finalement supprimer l'interface HTTP.

La commande JBoss CLI suivante vous permettra de lire le contenu actuel de votre interface HTTP, si vous décidez de l'ajouter à nouveau.

Exemple 11.12. Lire la configuration de l'interface HTTP

```
/host=master/core-service=management/management-interface=http-
interface/:read-resource(recursive=true,proxies=false,include-
runtime=false,include-defaults=true)
{
    "outcome" => "success",
    "result" => {
        "console-enabled" => true,
        "interface" => "management",
        "port" => expression "${jboss.management.http.port:9990}",
        "secure-port" => undefined,
        "security-realm" => "ManagementRealm"
    }
}
```

Pour supprimer l'interface HTTP, lancez la commande suivante :

Exemple 11.13. Supprimer l'interface HTTP

```
/host=master/core-service=management/management-interface=http-
interface/:remove
```

Pour activer l'accès à nouveau, lancez la commande suivante pour recréer l'Interface HTTP avec les valeurs par défaut.

Exemple 11.14. Recréer l'interface HTTP

```
/host=master/core-service=management/management-interface=http-
interface:add(console-
enabled=true,interface=management,port="${jboss.management.http.port:999
0}",security-realm=ManagementRealm)
```

Rapporter un bogue

11.8.6. Supprimer l'authentification silencieuse du domaine de sécurité par défaut.

Résumé

L'installation par défaut de JBoss EAP 6 contient une méthode d'authentification silencieuse pour un utilisateur de Management CLI local. Cela donne à l'utilisateur local la possibilité d'accéder au Management CLI sans authentification par nom d'utilisateur ou mot de passe. Cette fonctionnalité est activée dans un but pratique et pour faciliter l'exécution des scripts de Management CLI sans exiger l'authentification des utilisateurs locaux. Cette méthode est considérée comme une fonctionnalité utile étant donné que l'accès à la configuration locale donne généralement aussi à l'utilisateur la possibilité d'ajouter ses propres informations d'utilisateur ou sinon la possibilité de désactiver les contrôles de sécurité.

L'avantage de l'authentification silencieuse des utilisateurs locaux peut être désactivée quand un plus grand contrôle de sécurité est exigé. Ceci peut être réalisé en supprimant l'élément **`local`** dans la section **`security-realm`** du fichier de configuration. S'applique à la fois à **`standalone.xml`** pour une instance de serveur autonome, ou à **`host.xml`** pour un domaine géré. Vous ne devez envisager de supprimer l'élément **`local`** que si vous comprenez l'impact que ceci peut avoir sur la configuration de votre serveur particulier.

La meilleure méthode pour désactiver l'authentification silencieuse est d'utiliser l'interface CLI, qui retire l'élément **`local`** visible dans l'exemple suivant.

Exemple 11.15. Exemple d'élément `local` dans `security-realm`

```
<security-realms>
    <security-realm name="ManagementRealm">
        <authentication>
            <local default-user="$local"/>
            <properties path="mgmt-users.properties" relative-
to="jboss.server.config.dir"/>
        </authentication>
    </security-realm>
    <security-realm name="ApplicationRealm">
        <authentication>
```

```
            <local default-user="$local" allowed-users="*"/>
            <properties path="application-users.properties" relative-
to="jboss.server.config.dir"/>
        </authentication>
        <authorization>
            <properties path="application-roles.properties" relative-
to="jboss.server.config.dir"/>
        </authorization>
    </security-realm>
</security-realms>
```

Conditions préalables9

- Section 2.1.1, « Démarrer JBoss EAP 6 »
- Section 3.5.2, « Lancement de l'interface CLI »

Procédure 11.7. Supprimer l'authentification silencieuse du domaine de sécurité par défaut.

- **Supprimer l'authentification silencieuse par l'interface CLI**
  Supprimer l'élément **`local`** du Domaine de gestion et du Domaine d'applications comme requis.

  a. Supprimer l'élément **`local`** du Domaine de gestion.

  - **Pour les serveurs autonomes**

    ```
    /core-service=management/security-
    realm=ManagementRealm/authentication=local:remove
    ```

  - **Pour les domaines gérés**

    ```
    /host=HOST_NAME/core-service=management/security-
    realm=ManagementRealm/authentication=local:remove
    ```

  b. Supprimer l'élément **`local`** du Domaine d'applications.

  - **Pour les serveurs autonomes**

    ```
    /core-service=management/security-
    realm=ApplicationRealm/authentication=local:remove
    ```

  - **Pour les domaines gérés**

    ```
    /host=HOST_NAME/core-service=management/security-
    realm=ApplicationRealm/authentication=local:remove
    ```

Résultat

L'authentification silencieuse est retirée du **`ManagementRealm`** et de l'**`ApplicationRealm`**.

Rapporter un bogue

11.8.7. Désactiver l'accès à distance du sous-système JMX

Une connectivité à distance JMX vous permet de déclencher JDK et les opérations de gestion d'applications. Afin de garantir une installation, désactivez cette fonction. Vous pouvez faire cela soit en supprimant la configuration de la connexion à distance, soit en supprimant le sous-système JMX entièrement. Les commandes du JBoss CLI référencent le profil par défaut dans une configuration de domaine géré. Pour modifier un profil différent, modifiez la partie de la commande **`/profil=default`**. Pour un serveur autonome, retirez complètement cette partie de la commande.

NOTE

Dans un domaine géré, le connecteur distant est retiré du sous-système JMX par défaut. Cette commande est fournie pour votre information, au cas où vous souhaiteriez l'ajouter en cours de développement.

Exemple 11.16. Supprimez le connecteur distant du sous-système JMX

```
/profile=default/subsystem=jmx/remoting-connector=jmx/:remove
```

Exemple 11.17. Supprimez le sous-système JMX

Exécutez cette commande pour chaque profil que vous utilisez, si vous utilisez un domaine géré.

```
/profile=default/subsystem=jmx/:remove
```

Rapporter un bogue

11.8.8. Configurer les domaines de sécurité pour les interfaces de gestion

Les interfaces de gestion utilisent des domaines de sécurité pour contrôler l'authentification et l'accès aux mécanismes de configuration de JBoss EAP 6. Cette section vous montre comment lire et configurer les domaines de sécurité.

Lire une configuration de domaine de sécurité

Cet exemple montre la configuration par défaut du domaine de sécurité du **`ManagementRealm`**. Il utilise un fichier **`mgmt-users.properties`** qui contient ses informations de configuration.

Exemple 11.18. ManagementRealm par défaut

```
/host=master/core-service=management/security-
realm=ManagementRealm/:read-
resource(recursive=true,proxies=false,include-runtime=false,include-
defaults=true)
{
    "outcome" => "success",
    "result" => {
        "authorization" => undefined,
        "server-identity" => undefined,
        "authentication" => {"properties" => {
            "path" => "mgmt-users.properties",
```

```
            "plain-text" => false,
            "relative-to" => "jboss.domain.config.dir"
        }}
    }
}
```

Rédaction d'un domaine de sécurité

Les commandes suivantes créent un nouveau domaine de sécurité nommé **`TestRealm`** et définissent le répertoire du fichier de propriétés qui conviennent.

Exemple 11.19. Créer un domaine de sécurité

```
/host=master/core-service=management/security-realm=TestRealm/:add
/host=master/core-service=management/security-
realm=TestRealm/authentication=properties/:add(path=TestUsers.properties
, relative-to=jboss.domain.config.dir)
```

Ajouter le domaine de sécurité à l'interface de gestion

Après avoir ajouté un domaine de sécurité, l'utiliser comme référence à l'interface de gestion.

Exemple 11.20. Ajouter un domaine de sécurité à une interface de gestion

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=security-realm,value=TestRealm)
```

Rapporter un bogue

11.9. ACTIVER LES INTERFACES DE GESTION PAR LE CONTRÔLE D'ACCÈS BASÉ RÔLE

11.9.1. Les RBAC (Role-Based Access Control)

Le contrôle d'accès basé sur rôle (RBAC) est un mécanisme permettant de spécifier un jeu de permissions pour la gestion par les utilisateurs. Il permet à plusieurs utilisateurs de partager la responsabilité de gérer des serveurs JBoss EAP 6.3 sans qu'aucun d'entre eux n'ait besoin d'un accès illimité. En fournissant « une séparation des fonctions » pour la gestion des utilisateurs, JBoss EAP 6.3 facilite la tâche à une organisation de répandre la responsabilité entre individus ou groupes, sans octroi de privilèges inutiles. Cela garantit la sécurité maximale de vos serveurs et de vos données, tout en offrant une souplesse de configuration, de déploiement et de gestion.

Les RBAC (Role-Based Access Control) dans JBoss EAP 6.3 fonctionnent par une combinaison de permissions et de contraintes associées à des rôles.

Il y a sept rôles prédéfinis associés à des autorisations fixes différentes. Les rôles prédéfinis sont : Surveillant (Monitor), Opérateur (Operator), Mainteneur (Maintainer), Déployeur (Deployer), Auditeur (Auditor), Administrateur (Administrator) et Superutilisateur (Superuser). A chaque utilisateur de gestion est attribué un ou plusieurs rôles, qui définissent ce que l'utilisateur est autorisé à faire pour la gestion du serveur.

Rapporter un bogue

11.9.2. Les RBAC (Role-Based Access Control) dans la console de gestion et le CLI

Lorsque le contrôle d'accès basé sur les rôles (RBAC) est activé, le rôle assigné à un utilisateur déterminera les ressources auxquelles il aura accès et les opérations qu'il pourra effectuer avec des attributs de ressources.

La console de gestion

Dans la console de gestion, certains contrôles et vues sont désactivés (en gris) ou invisibles selon les permissions du rôle assignées à l'utilisateur.

Si vous n'avez pas de permission de lecture pour un attribut de ressource, cet attribut apparaîtra en blanc sur la console. Ainsi, la plupart des rôles ne peuvent pas lire les champs Nom d'utilisateur ou Mot de passe des sources de données.

Si vous n'avez pas de permission d'écriture sur un attribut de ressource, cet attribut sera désactivé (grisé) dans le formulaire de modification de la ressource. Si vous n'avez pas de permission d'écriture sur la ressource, le bouton de modification de la ressource ne s'affichera pas.

Si un utiliisateur ne possède pas les permissions nécessaires pour accéder à une ressource ou à un attribut (c'est « non adressable » ce pour rôle), ce n'apparaîtra pas dans la console. L'accès contrôle système lui-même uniquement visible par quelques rôles par défaut en est un exemple.

l'interface CLI ou l'API

Les utilisateurs de l'outil **`jboss-cli.sh`** ou de l'API rencontreront un comportement légèrement différent dans l'API lorsque RBAC est activé.

Les ressources et les attributs qui ne peuvent être lus sont filtrés des résultats. Si les éléments filtrés sont adressables par le rôle, leurs noms seront répertoriés en tant que **`filtered-attributes`** dans la section **`response-headers`** du résultat. Si une ressource ou un attribut n'est pas adressable par le rôle, il n'apparaîtra pas.

Toute tentative d'accès à une ressource non adressable résultera par l'erreur **`resource not found`**.

Si un utilisateur tente d'écrire ou de lire une ressource qu'il peut adresser, mais qu'il n'a pas les permissions de lecture ou d'écriture qui conviennent, une erreur **`Permission Denied`** apparaîtra.

Rapporter un bogue

11.9.3. Schémas d'authentification supportés

Les RBAC (Role-Based Access Control) fonctionnent avec les fournisseurs d'authentification standards inclus dans JBoss EAP 6.3, comme : **`username/password`**, **`client certificate`**, et **`local user`**.

Username/Password

Les utilisateurs sont authentifiés par une combinaison de nom d'utilisation et de mot de passe qui sont ensuite vérifiés par le fichier **`mgmt-users.properties`**, ou via un serveur LDAP.

Certificat Client

Utilisation du Trust Store

Utilisateur local

`jboss-cli.sh` authentifie automatiquement en tant qu'utilisateur local si le serveur exécute sur la même machine. Par défaut, l'utilisateur local est un membre du groupe **`SuperUser`**.

Quel que soit le fournisseur utilisé, JBoss EAP est responsable de l'attribution des rôles aux utilisateurs. Toutefois, lors de l'authentification avec le fichier **`mgmt-users.properties`** ou par un serveur LDAP, ces systèmes peuvent fournir des informations de groupes d'utilisateurs. Cette information peut également servir à JBoss EAP pour attribuer des rôles aux utilisateurs.

Rapporter un bogue

11.9.4. Les rôles standard

JBoss EAP 6 fournit sept rôles prédéfinis par l'utilisateur : Monitor, Operator, Maintainer, Deployer, Auditor, Administrator, et SuperUser. Chacun de ces rôles possède un ensemble de permissions différentes conçues pour les cas d'utilisation spécifiques. Les rôles Monitor, Operator, Maintainer, Deployer, Auditor, Administrator, et SuperUser se superposent, avec chaque rôle possédant plus d'autorisations que le précédent. Les rôles Auditor et Deployer sont semblables aux rôles Monitor et Maintainer respectivement mais ont leurs propres permissions et restrictions supplémentaires.

Monitor

Les utilisateurs du rôle Monitor ont le plus petit nombre de permissions et ne peuvent lire que la configuration ou l'état du serveur en cours. Ce rôle est destiné aux utilisateurs qui ont besoin de vérifier et reporter la performance du serveur.

Les utilisateurs Monitor ne peuvent pas modifier la configuration du serveur, et ne peuvent pas accéder aux données ou opérations confidentielles.

Operator

Le rôle Operator étend le rôle Monitor en ajoutant la possibilité de modifier l'état d'exécution du serveur. Cela signifie que les opérateurs peuvent recharger et arrêter le serveur, ainsi que suspendre et reprendre les destinations JMS. Le rôle Operator est idéal pour les utilisateurs responsables des hôtes physiques ou virtuels du serveur d'applications, puisqu'ils peuvent s'assurer ainsi que les serveurs puissent être arrêtés, redémarrés ou corrigés si nécessaire.

Les utilisateurs Operator ne peuvent pas modifier la configuration du serveur ou accéder à des données ou opérations confidentielles.

Maintainer

Le rôle Maintainer a le droit de visualiser et modifier l'état d'exécution et toute la configuration à l'exception des données sensibles et des opérations. Le rôle du mainteneur est un rôle d'usage général qui n'a pas accès aux données sensibles et aux opérations. Le rôle Maintainer permet aux utilisateurs de bénéficier d'un accès presque total d'administration du serveur, sans donner à ces utilisateurs l'accès aux mots de passe ou autres informations sensibles.

Les utilisateurs Maintainer ne peuvent pas accéder à des données ou opérations sensibles.

Administrator

Le rôle Administrator a un accès illimité aux ressources et opérations sur le serveur sauf sur le système d'enregistrement d'audit. Le rôle d'administrateur a accès à des données sensibles et aux opérations. Ce rôle peut également configurer le système de contrôle d'accès. Le rôle Administrator n'est requis que lorsque vous manipulez des données sensibles ou configurez les utilisateurs et les rôles.

Les utilisateurs Administrator ne peuvent pas accéder au système de journalisation d'auditing et ne peuvent pas se changer eux-mêmes en rôles Auditor ou SuperUser.

SuperUser

Le rôle SuperUser a un accès illimité aux ressources et opérations sur le serveur, y compris au système d'enregistrement d'audit. Ce rôle est équivalent aux rôles d'utilisateurs administrateurs des versions plus anciennes de JBoss EAP 6 (6.0 et 6.1). Si RBAC est désactivé, tous les utilisateurs de management auront une permission équivalente à celle d'un rôle SuperUser.

Deployer

Le rôle Deployer a les mêmes permissions que Monitor, mais peut modifier la configuration et l'état des déploiements ou n'importe quel autre type de ressource activée en tant que ressource d'application.

Auditor

Le rôle Auditor possède toutes les permissions du rôle Monitor et peut également voir (mais pas modifier) les données sensibles. Il a aussi accès au système de journalisation de l'auditing. Le rôle Auditor est le seul rôle en dehors de SuperUser qui puisse accéder au système de journalisation de l'auditing.

Les utilisateurs Auditor ne peuvent pas modifier les données ou les ressources sensibles. Seul l'accès lecture est permis.

Rapporter un bogue

11.9.5. Les permissions de rôle

Ce que chaque rôle peut faire est déterminé par ses permissions. Chaque rôle ne possède que quelques permissions. Seul le rôle SuperUser possède toutes les permissions et le rôle Monitor n'en possède que très peu.

Chaque permission peut donner un accès lecture et/ou écriture à une seule catégorie de ressource.

Les catégories sont les suivantes : l'état d'exécution, la configuration du serveur, les données sensibles, audit log, et le système de contrôle d'accès.

Tableau 11.5, « Matrice de permissions de rôle » résume les permissions de chaque rôle.

Tableau 11.5. Matrice de permissions de rôle

	Monitor	Operator	Maintainer	Deployer	Auditor	Admin	SuperUser

Lecture Config et État	X	X	X	X	X	X	X
Lire Données sensibles [2]					X	X	X
Modifier Données sensibles [2]						X	X
Lire/Modifer Audit Log					X		X
Modifier État Runtime		X	X	X[1]		X	X
Modifier Config persistante			X	X[1]		X	X
Lire/Modifier Contrôle d'accès						X	X

[1] les permissions sont limitées aux ressources d'applications.

[2] les ressources considérées comme "sensibles" sont configurées par les contraintes de sensibilité.

Rapporter un bogue

11.9.6. Contraintes

Les contraintes sont des jeux de configuration de contrôle d'accès désignés correspondant à une liste de ressources. Le système RBAC utilise la combinaison des contraintes et des permissions de rôle pour déterminer si un utilisateur peut exécuter une action de gestion.

Les contraintes sont divisées en trois classifications : application, sensibilité et expression d'archivage.

Contraintes d'application

Les contraintes d'application définissent des ensembles de ressources et d'attributs accessibles aux utilisateurs du rôle chargé du déploiement (rôle Deployer). Par défaut, la seule contrainte d'application activée est la principale qui inclut des déploiements, et des superpositions de déploiement. Les contraintes d'application sont également incluses (mais pas activées par défaut) pour les sources de données, connexion, mail, messagerie, nommage, adaptateurs de ressources et sécurité. Ces contraintes permettent aux utilisateurs Deployer de non seulement de déployer des applications, mais également de configurer et de maintenir les ressources requises par ces applications.

La configuration de contraintes d'applications se trouve dans l'API de gestion à l'adresse suivante : **`/core-service=management/access=authorization/constraint=application-classification`**.

Contraintes de sensibilité

Les contraintes de confidentialité définissent des ressources considérées comme « sensibles ». Une ressource sensible est généralement de nature secrète, comme un mot de passe, ou une information pouvant avoir de graves répercussions sur le fonctionnement du serveur, comme le networking, la configuration de la JVM ou les propriétés système. Le système de contrôle d'accès lui-même est aussi considéré comme sensible.

Les seuls rôles autorisés à écrire dans les ressources sensibles sont Administrator (Administrateur) et SuperUser (Superutilisateur). Le rôle Auditor est seulement capable de lire les ressources confidentielles. Aucun autre rôle ne peut avoir accès.

La configuration de contraintes de sensibilité se trouve dans l'API de gestion à l'adresse suivante : **`/core-service=management/access=authorization/constraint=sensitivity-classification`**.

Contrainte d'expression d'archivage sécurisé

Une contrainte d'expression d'archivage sécurisé détermine si la lecture ou l'écriture d'expressions d'archivage sécurisé est considérée comme une opération sensible. Par défaut, les deux opérations sont sensibles.

La configuration de contraintes d'expression d'archivage sécurisé se trouve dans l'API de gestion à l'adresse suivante : **`/core-service=management/access=authorization/constraint=vault-expression`**.

Les contraintes ne peuvent pas être configurées dans la console de gestion actuellement.

Rapporter un bogue

11.9.7. JMX et RBAC (Role-Based Access Control)

RBAC s'applique à JMX de trois façons :

1. L'API de gestion de JBoss EAP 6 est exposé comme JMX Management Beans. Ces Management Beans sont appelés « core mbeans » et leur accès est contrôlé et filtré exactement de la même façon que l'API de gestion sous-jacent lui-même.

2. Le sous-système JMX est configuré avec des permissions d'écriture « sensibles ». Cela signifie que seuls les utilisateurs ayant pour rôle Administrateur et Superutilisateur peuvent apporter des modifications à ce sous-système. Les utilisateurs avec le rôle Auditeur peuvent aussi lire cette configuration du sous-système.

3. Par défaut, les beans de gestion enregistrés par des applications déployées et des services (non-core mbeans) sont accessibles par tous les utilisateurs de gestion, mais seuls les utilisateurs ayant un rôle Maintenance, Opérateur, Administrateur, Superutilisateur peuvent écrire dessus.

Rapporter un bogue

11.9.8. Configurer le RBAC (Role-Based Access Control)

11.9.8.1. Aperçu des tâches de configuration RBAC

Quand RBAC est activé, seuls les utilisateurs qui possèdent le rôle Administration ou SuperUtilisateur peuvent voir ou modifier le système de contrôle d'accès.

La console de gestion fournit une interface pour les tâches RBAC standard suivantes :

- Voir et configurer les rôles assignés (ou exclus) pour chaque utilisateur
- Voir et configurer les rôles assignés (ou exclus) pour chaque groupe
- Voir Appartenance Utilisateur et Groupe par Rôle.
- Configurer l'appartenance par défaut d'un rôle.
- Créer un scoped rôle

Le CLI donne accès au système de contrôle d'accès total. Cela signifie que tout ce qui peut être fait à partir de la console de gestion doit être fait à cet endroit, mais qu'un nombre de tâches supplémentaires peut être fait via le CLI à défaut du système de contrôle d'accès.

Les tâches supplémentaires suivantes peuvent être faites par le CLI :

- Activer et désactiver RBAC
- Modifier la police de combinaison de permissions
- Configurer les contraintes de sensibilité des ressources et des ressources d'applications

Rapporter un bogue

11.9.8.2. Activer le RBAC (Role-Based Access Control)

Le système de contrôle d'accès basé sur les rôles (RABC) est désactivé par défaut. Il est activé en changeant l'attribut provider en le faisant passer de **`simple`** à **`rbac`**. Cela peut être fait en utilisant l'outil **`jboss-cli.sh`**, ou en modifiant la configuration du serveur XML du fichier si le serveur est hors ligne. Quand RBAC est désactivé ou activé sur un serveur en cours d'exécution, la configuration du serveur doit être rechargée avant de prendre effet.

Une fois activé, ne peut être désactivé que par un utilisateur ayant les rôles Adminstrateur ou Superutilisateur. Par défaut, le **`jboss-cli.sh`** est exécuté par le rôle **`SuperUser`** s'il est exécuté dans la même machine que le serveur.

Procédure 11.8. Activer RBAC

- Pour activer RBAC avec **`jboss-cli.sh`** utiliser l'opération **`write-attribute`** de la ressource d'autorisation d'accès, pour définir la valeur de l'attribut provider à **`rbac`**

```
/core-service=management/access=authorization:write-
attribute(name=provider, value=rbac)
```

```
[standalone@localhost:9999 /] /core-
service=management/access=authorization:write-
attribute(name=provider, value=rbac)
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
```

```
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /] /:reload
{
    "outcome" => "success",
    "result" => undefined
}
```

Procédure 11.9. Désactiver RBAC

- Pour désactiver RBAC avec **`jboss-cli.sh`** utiliser l'opération **`write-attribute`** de la ressource d'autorisation d'accès pour définir la valeur de l'attribut de fournisseur à **`simple`**

```
/core-service=management/access=authorization:write-
attribute(name=provider, value=simple)
```

```
[standalone@localhost:9999 /] /core-
service=management/access=authorization:write-
attribute(name=provider, value=simple)
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /] /:reload
{
    "outcome" => "success",
    "result" => undefined
}
```

Si le serveur est hors ligne, la configuration XML peuvent être sur RBAC « activé » ou « désactivé ». Pour ce faire, modifiez l'attribut **`provider`** de l'élément de contrôle d'accès de l'élément **`access-control`**. Définissez la valeur **`rbac`** sur « activé » et **`simple`** sur « désactivé ».

```
<management>

        <access-control provider="rbac">
            <role-mapping>
                <role name="SuperUser">
                    <include>
                        <user name="$local"/>
                    </include>
                </role>
            </role-mapping>
        </access-control>

    </management>
```

Rapporter un bogue

11.9.8.3. Modifier la police de combinaison de permissions

La police de combinaison de permissions détermine comment les permissions sont définies quand un utilisateur possède plus d'un seul rôle. Peut être définie sur **`permissive`** ou **`rejecting`**. La valeur par défaut est **`permissive`**.

Quand elle est définie à **`permissive`**, si un rôle est assigné à l'utilisateur pour permettre une action, alors l'action sera autorisée.

Quand elle est définie sur **`rejecting`**, si plusieurs rôles sont assignés à un utilisateur, alors aucune action ne sera requise. Lorsque la police est définie à **`rejecting`**, chaque utilisateur se verra assigner un rôle unique. Les utilisateurs ayant des rôles multiples ne seront pas en mesure d'utiliser l'outil de gestion de console ou **`jboss-cli.sh`** lorsque la police est définie sur **`rejecting`**.

La police de combinaison d'autorisations est configurée en affectant à l'attribut **`permission-combination-policy`** soit **`permissive`** ou **`rejecting`**. Cela peut être fait par l'outil **`jboss-cli.sh`**, ou en modifiant la configuration du serveur XML du fichier si le serveur est hors ligne.

Procédure 11.10. Définir la police de combinaison de permissions

- Effectuer l'opération **`write-attribute`** de la ressource d'autorisation d'accès pour définir l'attribut **`permission-combination-policy`** au nom de la police requise.

  ```
  /core-service=management/access=authorization:write-
  attribute(name=permission-combination-policy, value=POLICYNAME)
  ```

  Les noms de police valide sont « rejecting » ou « permissive ».

  ```
  [standalone@localhost:9999 /] /core-
  service=management/access=authorization:write-
  attribute(name=permission-combination-policy, value=rejecting)
  {"outcome" => "success"}
  [standalone@localhost:9999 access=authorization]
  ```

Si le serveur est hors ligne, la configuration XML peut être modifiée pour changer la valeur de la politique de combinaison de permissions. Pour ce faire, modifiez l'attribut **`permission-combination-policy`** de l'élément de contrôle d'accès.

```
<access-control provider="rbac" permission-combination-policy="rejecting">
  <role-mapping>
    <role name="SuperUser">
      <include>
        <user name="$local"/>
      </include>
    </role>
  </role-mapping>
</access-control>
```

Rapporter un bogue

11.9.9. Gestion des rôles

11.9.9.1. Appartenance à un rôle

Lorsque le contrôle d'accès basé sur les rôles (RBAC) est activé, ce qu'un utilisateur de gestion est autorisé à faire est déterminé par les rôles qui affectent l'utilisateur. JBoss EAP 6.3 utilise un système d'inclusion et d'exclusion basé à la fois sur l'appartenance utilisateur ou groupe pour déterminer à quel rôle un utilisateur appartient.

On considère qu'un rôle est assigné à un utilisateur si :

1. L'utilisateur est :
    - listé comme utilisateur à inclure dans le rôle, ou
    - membre d'un groupe qui est listé pour être inclus dans le rôle.
2. L'utilisateur n'est pas :
    - listé comme utilisateur à exclure du rôle, ou
    - membre d'un groupe qui est listé pour être exclus du rôle.

Les exclusions prennent la priorité sur les inclusions.

Les configurations d'exclusion et d'inclusion des utilisateurs ou groupes peuvent être effectuées par la console de gestion ou bien par le CLI.

Seuls les utilisateurs qui possèdent les rôles Superutilisateur ou Administrateur peuvent effectuer cette configuration.

Rapporter un bogue

11.9.9.2. Configurer le rôle d'utilisateur 'Assignment' (attribution de rôles)

Les rôles d'utilisateur à inclure ou à exclure peuvent être configurés dans la console de gestion et par le **`jboss-cli.sh`**. Cette section explique uniquement comment utiliser la console de gestion à cet effet.

Seuls les utilisateurs ayant les rôles **`SuperUser`** ou **`Administrator`** peuvent effectuer cette configuration.

La configuration des rôles Utilisateur de la console de gestion suit les étapes suivantes :

1. Connectez-vous à la console de gestion.
2. Cliquez sur l'onglet **Administration**.
3. Déployez le menu **`Access Control`** et sélectionner **Role Assignment**.
4. Sélectionnez l'onglet **USERS**.

Procédure 11.11. Créez une nouvelle attribution de rôle pour l'utilisateur

1. Connexion à la console de gestion.
2. Naviguez vers l'onglet **Users** (Utilisateurs) de la section **`Role Assignment`** (Attribution de rôles).
3. Cliquez sur le bouton **Add** en haut et à droite de la liste d'utilisateur. Le dialogue **`Add User`** apparaîtra.

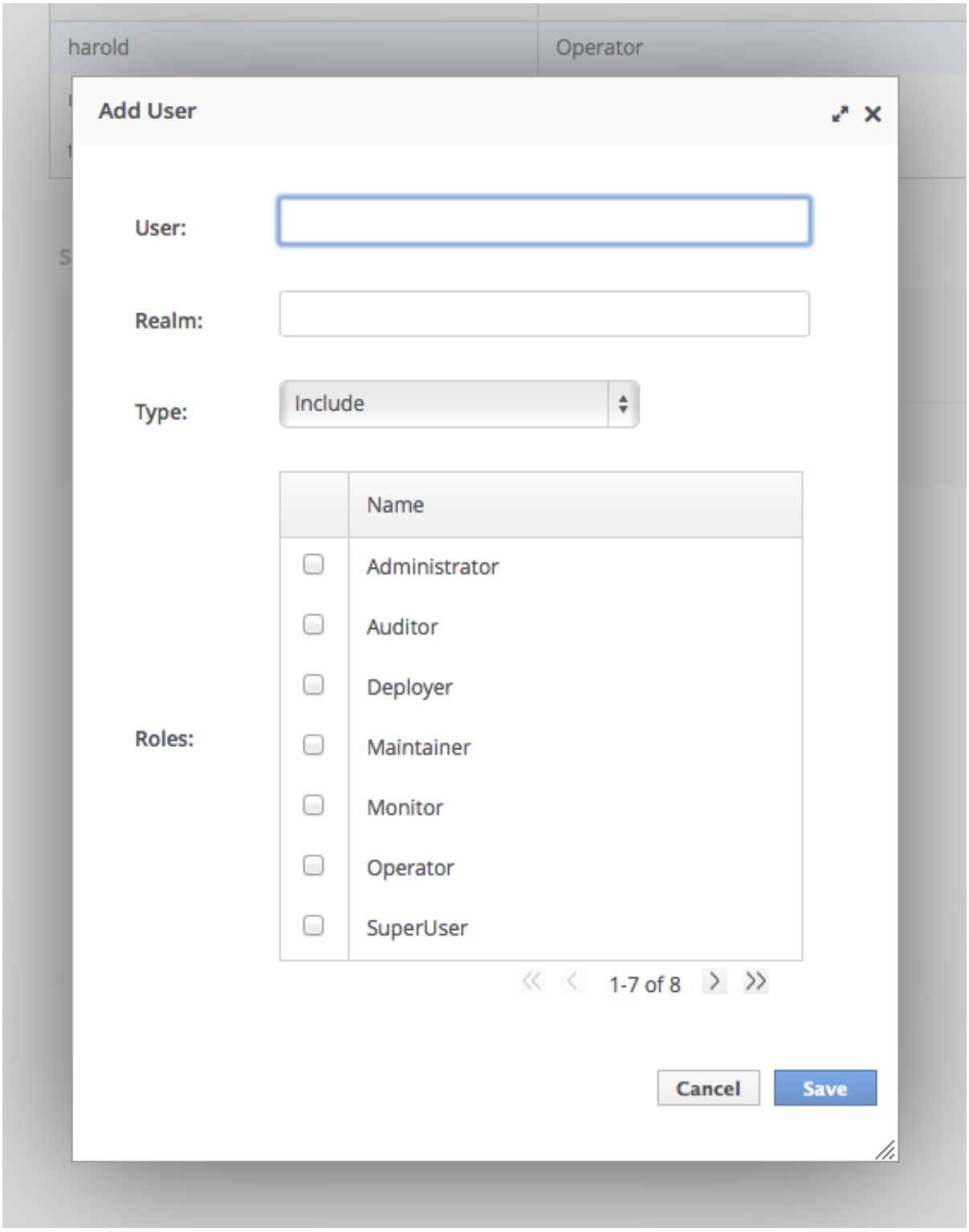


Figure 11.1. Boîte de dialogue 'Add User' (Ajouter Utilisateur)

4. Vous devez préciser un nom d'utilisateur, ou un domaine si possible.
5. Définissez le type de menu à inclure ou à exclure.
6. Cliquez la case des rôles à inclure ou à exclure. Vous pouvez utiliser la clé de contrôle (clé de commande OSX) pour cocher les divers items.
7. Cliquez sur le bouton **Save** pour terminer.

Si cela réussit, la boîte de dialogue **`Add User`** se ferme, et la liste des utilisateurs sera mise à jour pour refléter les modifications apportées. En cas d'échec, un message **`Failed to save role assignment`** (Impossible d'enregistrer l'attribution de rôle) s'affichera.

Procédure 11.12. Mise à jour d'une attribution de rôle pour un utilisateur

1. Connexion à la console de gestion.
2. Naviguez vers l'onglet **Users** (Utilisateurs) de la section **`Role Assignment`** (Attribution de rôles).
3. Sélectionnez un utilisateur dans la liste.
4. Cliquez sur le bouton **Edit**. Le panneau de sélection sera alors en mode Édit.

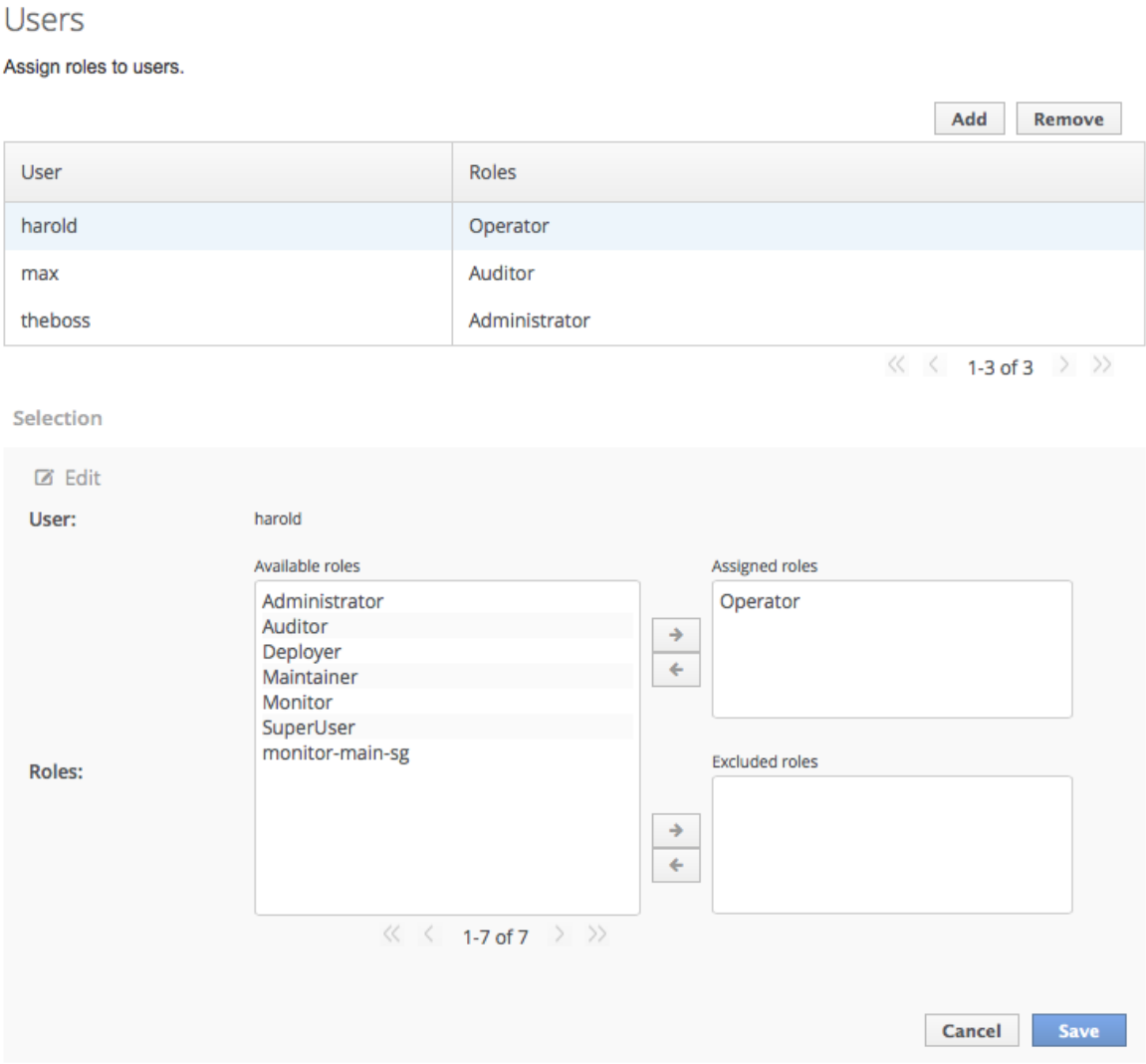


Figure 11.2. Sélectionnez « Edit View »

Vous pourrez ici ajouter ou supprimer les rôles assignés pour exclus pour l'utilisateur.

1. Pour ajouter un rôle assigné, sélectionnez le rôle requis de la liste de rôles disponibles sur la gauche et cliquez sur le bouton avec une flèche pointant sur la droite qui se trouve à côté de la liste de rôles assignés. Le rôle se déplacera alors depuis la liste de rôles disponibles vers la liste de rôles assignés.

2. Pour supprimer un rôle assigné, sélectionnez le rôle requis de la liste de rôles assignés sur la droite et cliquez sur le bouton avec une flèche pointant sur la gauche qui se trouve à côté de la liste de rôles assignés. Le rôle se déplacera alors depuis la liste de rôles assignés vers la liste de rôles disponibles.

3. Pour ajouter un rôle exclus, sélectionnez le rôle requis de la liste de rôles disponibles sur la gauche et cliquez sur le bouton avec une flèche pointant sur la droite qui se trouve à côté de la liste de rôles exclus. Le rôle se déplacera alors depuis la liste de rôles disponibles vers la liste de rôles exclus.

4. Pour supprimer un rôle exclus, sélectionnez le rôle requis de la liste de rôles assignés sur la droite et cliquer sur le bouton avec une flèche pointant sur la gauche qui se trouve à côté de la liste de rôles exclus. Le rôle se déplacera alors depuis la liste de rôles exclus vers la liste de rôles disponibles.

5. Cliquez sur le bouton **Save** pour terminer.

   Si cela réussit, la vue Édit se ferme, et la liste des utilisateurs sera mise à jour pour refléter les modifications apportées. En cas d'échec, le message **`Failed to save role assignment`** (Impossible d'enregistrer l'attribution de rôle) s'affichera.

Procédure 11.13. Suppression d'une attribution de rôle pour un utilisateur

1. Connexion à la console de gestion.

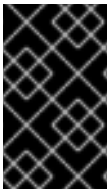
2. Naviguez vers l'onglet **Users** de la section d'attribution de rôles.

3. Sélectionnez l'utilisateur dans la liste.

4. Cliquez sur le bouton **`Remove`**. L'invite de confirmation **`Remove Role Assignment`** apparaîtra.

5. Cliquez sur **Confirm**.

   Si cela réussit, l'utilisateur n'apparaîtra plus sur la liste d'attributions de rôle d'utilisateur.



IMPORTANT

Suppression de l'utilisateur de la liste d'attribution de rôles ne retire pas l'utilisateur du système, ni ne garantit qu'aucun rôle ne sera assigné à l'utilisateur. Les rôles peuvent toujours être assignés sur la base de l'appartenance à un groupe.

Rapporter un bogue

11.9.9.3. Configurer l'attribution de rôle utilisateur avec jboss-cli.sh

Les rôles d'utilisateur à inclure ou à exclure peuvent être configurés dans la console de gestion et par le **`jboss-cli.sh`**. Cette section explique uniquement comment utiliser **`jboss-cli.sh`**.

La configuration du mappage des utilisateurs et des groupes en rôles se situe dans l'API de gestion à : **`/core-service=management/access=authorization`** en tant qu'éléments **`role-mapping`**.

Seuls les utilisateurs qui possèdent les rôles SuperUser ou Administrator peuvent faire cette configuration.

Procédure 11.14. Affichage de la configuration d'attribution de rôles

1. Utiliser l'opération :read-children-names pour obtenir une liste complète des rôles configurés :

```
/core-service=management/access=authorization:read-children-
names(child-type=role-mapping)
```

```
[standalone@localhost:9999 access=authorization] :read-children-
names(child-type=role-mapping)
{
    "outcome" => "success",
    "result" => [
        "ADMINISTRATOR",
        "DEPLOYER",
        "MAINTAINER",
        "MONITOR",
        "OPERATOR",
        "SuperUser"
    ]
}
```

2. Utiliser l'opération **`read-resource`** d'un mappage de rôle (role-mapping) pour obtenir toutes les informations sur un rôle particulier :

```
/core-service=management/access=authorization/role-
mapping=ROLENAME:read-resource(recursive=true)
```

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=ADMINISTRATOR:read-resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "include-all" => false,
        "exclude" => undefined,
        "include" => {
            "user-theboss" => {
                "name" => "theboss",
                "realm" => undefined,
                "type" => "USER"
            },
            "user-harold" => {
                "name" => "harold",
                "realm" => undefined,
                "type" => "USER"
            },
            "group-SysOps" => {
                "name" => "SysOps",
                "realm" => undefined,
                "type" => "GROUP"
            }
        }
    }
}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.15. Ajouter un nouveau rôle

Cette procédure montre comment ajouter un role-mapping pour un rôle. Cela doit être effectué avant que le rôle puisse être configuré.

- Utiliser l'opération **add** pour ajouter une nouvelle configuration de rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME:add
```

ROLENAME est le nom du rôle du nouveau mappage.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR:add
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.16. Ajouter un utilisateur comme inclus dans un rôle

Cette procédure montre comment ajouter un utilisateur dans la liste d'inclusions d'un rôle.

S'il n'y a pas de configuration de rôle, alors vous devrez commencer par la saisie du role-mapping.

- Utiliser l'opération **add** pour ajouter une entrée d'utilisateur dans la liste d'inclusions du rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/include=ALIAS:add(name=USERNAME, type=USER)
```

ROLENAME est le nom du rôle en cours de configuration.

ALIAS est un nom unique pour ce mappage. Red Hat conseille d'utiliser une convention de mappage pour tous les alias comme **user-*USERNAME***.

USERNAME est le nom de l'utilisateur entrain d'être ajouté à la liste des inclusions.

```
 [standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/include=user-max:add(name=max, type=USER)
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.17. Ajouter un utilisateur comme exclus dans un rôle

Cette procédure montre comment ajouter un utilisateur dans la liste d'exclusions d'un rôle.

S'il n'y a pas de configuration de rôle, alors vous devrez commencer par la saisie du role-mapping.

- Utiliser l'opération **add** pour ajouter une entrée d'utilisateur dans la liste d'exclusions du rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/exclude=ALIAS:add(name=USERNAME, type=USER)
```

ROLENAME est le nom du rôle en cours de configuration.

USERNAME est le nom de l'utilisateur ajouté à la liste des exclusions.

ALIAS est un nom unique pour ce mappage. Red Hat conseille d'utiliser une convention de mappage pour tous les alias comme **user-*USERNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/exclude=user-max:add(name=max, type=USER)
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.18. Configuration pour la suppression d'un rôle d'utilisateur d'inclusion

Cette procédure vous montre comment supprimer une entrée d'inclusion d'utilisateur d'un mappage de rôle.

- Utiliser l'opération **remove** pour supprimer la saisie.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/include=ALIAS:remove
```

ROLENAME est le nom du rôle en cours de configuration

ALIAS est un nom unique pour ce mappage. Red Hat conseille d'utiliser une convention de mappage pour tous les alias comme **user-*USERNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/include=user-max:remove
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Supprimer l'utilisateur de la liste d'inclusions ne permet pas de supprimer l'utilisateur du système, ni ne garantit que le rôle ne soit pas assigné à l'utilisateur. Le rôle peut être assigné sur la base de l'appartenance à un groupe.

Procédure 11.19. Configuration de la suppression d'un rôle d'utilisateur d'exclusion

Cette procédure vous montre comment supprimer une entrée d'exclusion d'utilisateur d'un mappage de rôle.

- Utiliser l'opération **remove** pour supprimer la saisie.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/exclude=ALIAS:remove
```

ROLENAME est le nom du rôle en cours de configuration.

ALIAS est un nom unique pour ce mappage. Red Hat conseille d'utiliser une convention de mappage pour tous les alias comme **user-*USERNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/exclude=user-max:remove
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Supprimer l'utilisateur de la liste d'exclusions ne permet pas de supprimer l'utilisateur du système, ni ne garantit que le rôle puisse être assigné à l'utilisateur. Les rôles peuvent être assignés sur la base de l'appartenance à un groupe.

Rapporter un bogue

11.9.9.4. Groupes Utilisateurs et Rôles

Les utilisateurs authentifiés à l'aide du fichier de **`mgmt-users.properties`** ou d'un serveur LDAP peuvent être membres des groupes d'utilisateurs. Un groupe d'utilisateurs est une étiquette arbitraire qui peut être assignée à un ou plusieurs utilisateurs.

Le système RBAC peut être configuré pour attribuer automatiquement des rôles aux utilisateurs selon les groupes d'utilisateurs auxquels ils appartiennent. Il peut également exclure des utilisateurs de rôles basés sur l'appartenance à un groupe.

Lorsque vous utilisez le fichier **`mgmt-users.properties`**, l'information groupe est stockée dans le fichier **`mgmt-groups.properties`**. Lorsque vous utilisez LDAP, l'information groupe est stockée dans le serveur LDAP et est maintenue par les responsables du serveur LDAP.

Rapporter un bogue

11.9.9.5. Configurer l'attribution de rôles de groupe

Des rôles peuvent être assignés à un utilisateur sur la base d'appartenance à un groupe d'utilisateurs.

Les groupes à inclure ou à exclure d'un rôle peuvent être configurés dans la console de gestion et par le **`jboss-cli.sh`**. Cette section explique uniquement comment utiliser la console de gestion.

Seuls les utilisateurs ayant les rôles **`SuperUser`** ou **`Administrator`** peuvent effectuer cette configuration.

La configuration des rôles de groupe de la console de gestion suit les étapes suivantes :

1. Connectez-vous à la console de gestion.
2. Cliquer sur l'onglet **Administration**.
3. Déployez le menu **Access Control** et sélectionner **Role Assignment**.
4. Sélectionner l'onglet **GROUPS**.

Procédure 11.20. Créer une nouvelle attribution de rôle pour un groupe

1. Connectez-vous à la console de gestion
2. Naviguer vers l'onglet **GROUPS** de la section **`Role Assignment`** (Attribution de rôles).
3. Cliquer sur le bouton **Add** en haut et à droite de la liste d'utilisateur. Le dialogue **`Add User`** (Ajouter Utilisateur) apparaîtra.

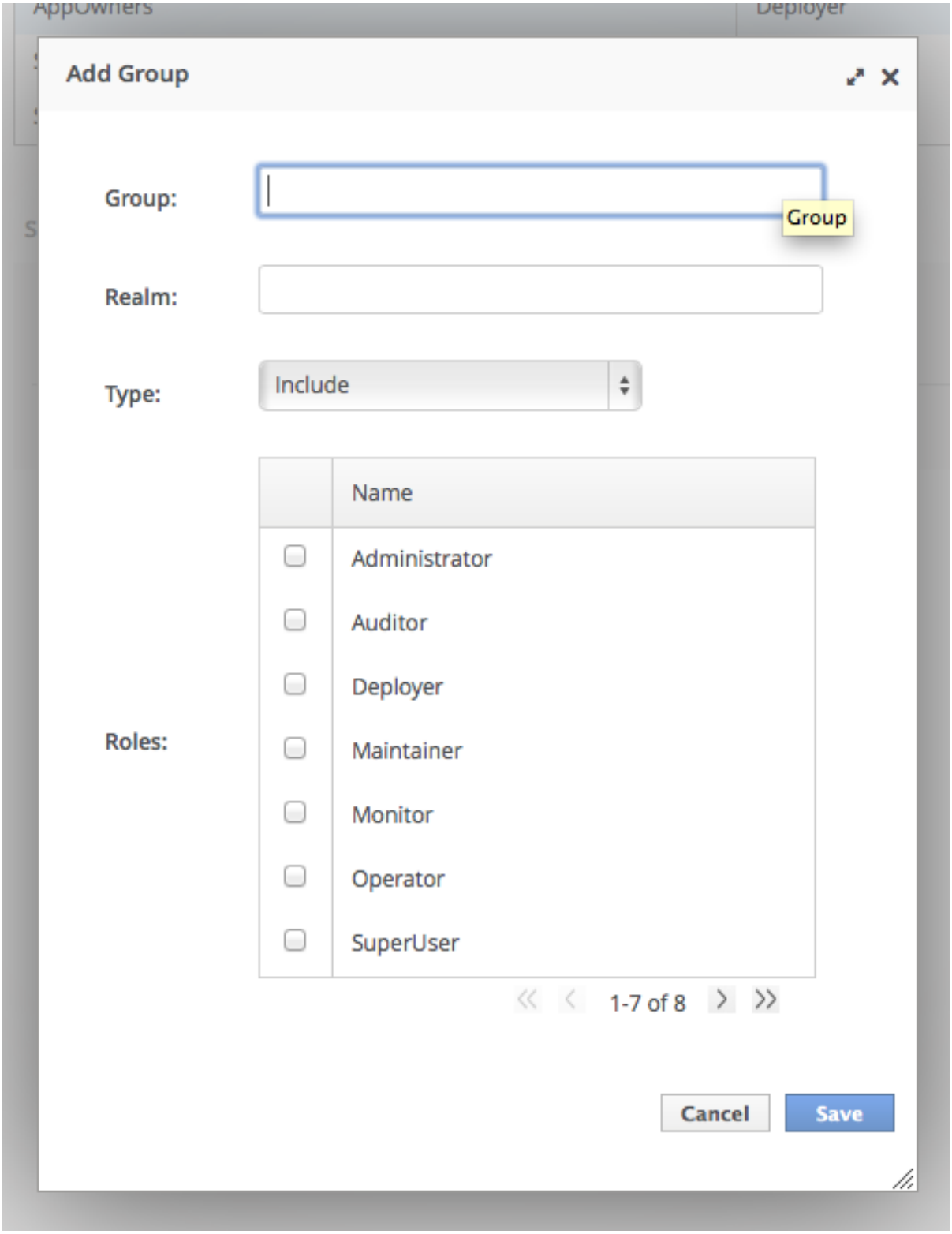


Figure 11.3. Ajouter le dialogue de groupe

4. Vous devez préciser un nom de groupe, et domaine si possible.
5. Définir le type de menu à inclure ou à exclure.
6. Cliquer la case des rôles à inclure ou à exclure. Vous pouvez utiliser la clé de contrôle (clé de commande OSX) pour cocher les divers items.
7. Cliquer sur le bouton **Save** pour terminer.

Si cela réussit, la boîte de dialogue d'ajout de groupe (**`Add Group`**) se ferme, et la liste des utilisateurs sera mise à jour pour refléter les modifications apportées. En cas d'échec, un message **`Impossible d'enregistrer l'attribution de rôle`** s'affichera.

Procédure 11.21. Mise à jour d'une attribution de rôle pour un groupe

1. Connexion à la console de gestion.
2. Naviguer vers l'onglet **GROUPS** de la section d'attribution de rôles.
3. Sélectionner un groupe dans la liste.
4. Cliquer sur le bouton Édit. La vue de sélection est alors en mode Édit.

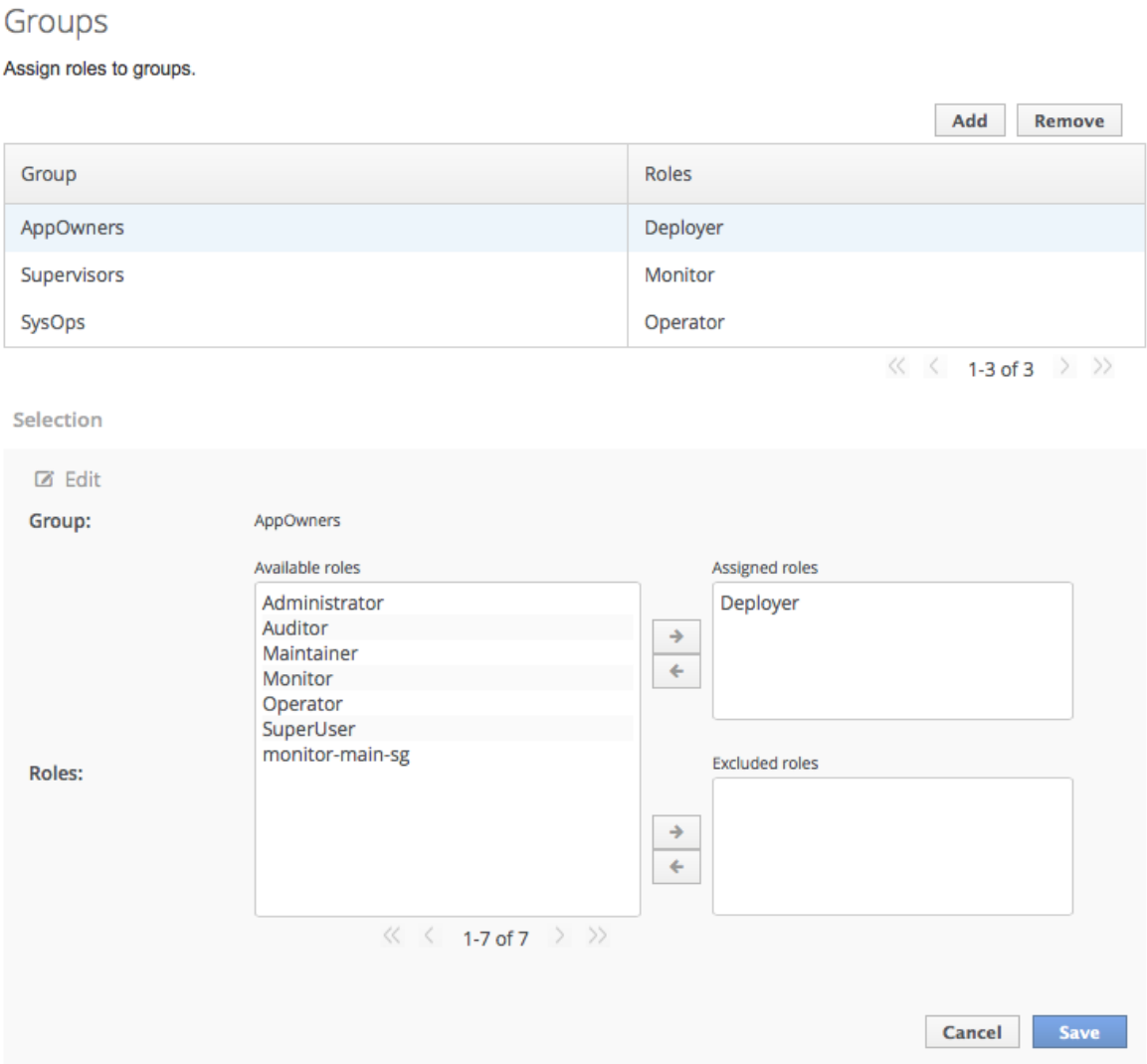


Figure 11.4. Sélection Vue Mode Édition

Vous pourrez ici ajouter ou supprimer les rôles assignés ou exclus du groupe :

- Pour ajouter un rôle assigné, sélectionner le rôle requis de la liste de rôles disponibles sur la gauche et cliquer sur le bouton avec une flèche pointant sur la droite qui se trouve à côté de la liste de rôles assignés. Le rôle se déplacera alors depuis la liste de rôles disponibles vers la liste de rôles assignés.
- Pour supprimer un rôle assigné, sélectionner le rôle requis de la liste de rôles assignés sur

la droite et cliquer sur le bouton avec une flèche pointant sur la gauche qui se trouve à côté de la liste de rôles assignés. Le rôle se déplacera alors depuis la liste de rôles assignés vers la liste de rôles disponibles.

- Pour ajouter un rôle exclus, sélectionner le rôle requis de la liste de rôles disponibles sur la gauche et cliquer sur le bouton avec une flèche pointant sur la droite qui se trouve à côté de la liste de rôles exclus. Le rôle se déplacera alors depuis la liste de rôles disponibles vers la liste de rôles exclus.
- Pour supprimer un rôle exclus, sélectionner le rôle requis de la liste de rôles assignés sur la droite et cliquer sur le bouton avec une flèche pointant sur la gauche qui se trouve à côté de la liste de rôles exclus. Le rôle se déplacera alors depuis la liste de rôles exclus vers la liste de rôles disponibles.

5. Cliquer sur le bouton **Save** pour terminer.

   Si cela réussit, la vue Édit se ferme, et la liste des groupes sera mise à jour pour refléter les modifications apportées. En cas d'échec, un message **Failed to save role assignment** (Impossible d'enregistrer l'attribution de rôle) s'affichera.

Procédure 11.22. Supprimer une attribution de rôle pour un groupe

1. Connexion à la console de gestion.
2. Naviguer vers l'onglet **GROUPS** de la section d'attribution de rôles (**Role Assignment**).
3. Sélectionner un groupe dans la liste.
4. Clique le bouton **`Remove`**. L'invite de confirmation **`Remove Role Assignment`** apparaîtra.
5. Cliquer sur **`Confirm`**.

   Si cela réussit, le rôle n'apparaîtra plus sur la liste d'attributions de rôle d'utilisateur.

   La suppression du groupe de la liste d'attribution de rôles ne retire pas l'utilisateur du système, ni ne garantit qu'aucun rôle ne sera assigné aux membres de ce groupe. Chaque membre du groupe peut encore avoir un rôle qui lui soit directement assigné.

Rapporter un bogue

11.9.9.6. Configurer l'attribution des rôles de groupe avec jboss-cli.sh

Les groupes à inclure ou à exclure d'un rôle peuvent être configurés dans la console de gestion et par le **`jboss-cli.sh`**. Cette section explique uniquement comment utiliser l'outil **`jboss-cli.sh`**.

La configuration du mappage des utilisateurs et des groupes en rôles se situe dans l'API de gestion à : **`/core-service=management/access=authorization`** en tant qu'éléments de mappage.

Seuls les utilisateurs ayant des rôles SuperUser ou Administrateur peuvent effectuer cette configuration.

Procédure 11.23. Affichage de la configuration d'attribution de rôles de groupe

1. Utiliser l'opération **`read-children-names`** pour obtenir une liste complète des rôles configurés :

```
/core-service=management/access=authorization:read-children-
names(child-type=role-mapping)
```

```
[standalone@localhost:9999 access=authorization] :read-children-
names(child-type=role-mapping)
{
    "outcome" => "success",
    "result" => [
        "ADMINISTRATOR",
        "DEPLOYER",
        "MAINTAINER",
        "MONITOR",
        "OPERATOR",
        "SuperUser"
    ]
}
```

2. Utiliser l'opération **`read-resource`** d'un mappage de rôle (role-mapping) pour obtenir toutes les informations sur un rôle particulier :

```
/core-service=management/access=authorization/role-
mapping=ROLENAME:read-resource(recursive=true)
```

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=ADMINISTRATOR:read-resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "include-all" => false,
        "exclude" => undefined,
        "include" => {
            "user-theboss" => {
                "name" => "theboss",
                "realm" => undefined,
                "type" => "USER"
            },
            "user-harold" => {
                "name" => "harold",
                "realm" => undefined,
                "type" => "USER"
            },
            "group-SysOps" => {
                "name" => "SysOps",
                "realm" => undefined,
                "type" => "GROUP"
            }
        }
    }
}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.24. Ajouter un nouveau rôle

Cette procédure montre comment ajouter un role-mapping pour un rôle. Cela doit être effectué avant que le rôle puisse être configuré.

- Utiliser l'opération **add** pour ajouter une nouvelle configuration de rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME:add
```

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR:add
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.25. Ajouter un groupe comme étant inclus dans un rôle

Cette procédure montre comment ajouter un groupe dans la liste d'inclusions d'un rôle.

S'il n'y a pas de configuration de rôle, alors vous devez commencer par la saisie du role-mapping.

- Utiliser l'opération **add** pour ajouter une entrée de groupe dans la liste d'inclusions du rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/include=ALIAS:add(name=GROUPNAME, type=GROUP)
```

ROLENAME est le nom du rôle en cours de configuration.

GROUPNAME est le nom du groupe entrain d'être ajouté à la liste des inclusions.

ALIAS est un nom unique pour ce mappage. Red Hat recommande d'utiliser une convention de mappage pour tous les alias comme **group-*GROUPNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/include=group-investigators:add(name=investigators,
type=GROUP)
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.26. Ajouter un groupe comme exclus dans un rôle

Cette procédure montre comment ajouter un groupe dans la liste d'exclusions d'un rôle.

S'il n'y a pas de configuration de rôle, alors vous devez commencer par la saisie du role-mapping.

- Utiliser l'opération **add** pour ajouter une entrée de groupe dans la liste d'exclusions du rôle.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/exclude=ALIAS:add(name=GROUPNAME, type=GROUP)
```

ROLENAME est le nom du rôle en cours de configuration

GROUPNAME est le nom du groupe entrain d'être ajouté à la liste des inclusions

ALIAS est un nom unique pour ce mappage. Red Hat recommande d'utiliser une convention de mappage pour tous les alias comme **group-*GROUPNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/exclude=group-supervisors:add(name=supervisors,
type=USER)
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Procédure 11.27. Configuration de la suppression d'un rôle de groupe

Cette procédure vous montre comment supprimer une entrée d'inclusion de groupe d'un mappage de groupe.

- Utiliser l'opération **remove** pour supprimer la saisie.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/include=ALIAS:remove
```

ROLENAME est le nom du rôle en cours de configuration

ALIAS est un nom unique pour ce mappage. Red Hat recommande d'utiliser une convention de mappage pour tous les alias comme **group-*GROUPNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/include=group-investigators:remove
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Supprimer le groupe de la liste d'inclusions ne permet pas de supprimer le groupe du système, ni ne garantit que le rôle ne sera pas assigné à des utilisateurs de ce groupe. Le rôle peut être assigné à des utilisateurs du groupe individuellement.

Procédure 11.28. Supprimer une entrée d'exclusion de groupe d'utilisateurs

Cette procédure montre comment supprimer une entrée d'exclusion d'un mappage de rôle.

- Utiliser l'opération **remove** pour supprimer la saisie.

```
/core-service=management/access=authorization/role-
mapping=ROLENAME/exclude=ALIAS:remove
```

ROLENAME est le nom du rôle en cours de configuration.

ALIAS est un nom unique pour ce mappage. Red Hat recommande d'utiliser une convention de mappage pour tous les alias comme **group-*GROUPNAME***.

```
[standalone@localhost:9999 access=authorization] ./role-
mapping=AUDITOR/exclude=group-supervisors:remove
{"outcome" => "success"}
[standalone@localhost:9999 access=authorization]
```

Supprimer le groupe de la liste d'exclusions ne permet pas de supprimer le groupe du système, ni ne garantit que le rôle ne va pas être assigné à des membres du groupe. Les rôles peuvent être exclus sur la base de l'appartenance à un groupe.

Rapporter un bogue

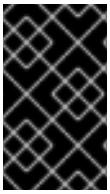
11.9.9.7. Autorisation et chargement de groupes avec LDAP

Dans l'annuaire LDAP, il est prévu qu'il y ait des entrées pour les comptes d'utilisateurs et les groupes, celles-ci sont ensuite vérifiées par l'utilisation d'attributs. Les attributs utilisés pour vérifier les deux peuvent consister en une référence pour vérifier l'entrée du compte utilisateur en fonction de l'entrée du groupe ou un attribut sur le groupe référençant les utilisateurs qui sont membres du groupe. Sur certains serveurs, les deux formes de renvoi existent.

Il est également fréquent qu'un utilisateur soit authentifié auprès du serveur à l'aide d'un simple nom utilisateur. Lorsqu'il recherche les informations d'appartenance de groupe, selon le serveur de répertoires utilisé, les recherches d'utilisation peuvent être effectuées à l'aide de ce simple nom ou peuvent être réalisées en utilisant le nom unique d'entrée utilisateur du répertoire.

L'étape d'authentification d'un utilisateur qui se connecte au serveur a toujours lieu en premier. C'est seulement une fois qu'il a été décidé que l'utilisateur est bien authentifié que le serveur se déplace pour charger un groupe d'utilisateurs. Tant que l'étape d'authentification et que l'étape d'autorisation utilisent une connexion au serveur LDAP, le domaine contient une optimisation qui consiste à ce que n'importe quelle connexion utilisée pour l'authentification soit réutilisée pour l'étape de chargement de groupe. Tel qu'il apparaît dans les étapes de configuration ci-dessous, il est possible de définir des règles au sein de la section autorisation pour convertir un nom d'utilisateur simple en nom unique d'utilisateur. C'est potentiellement dupliquer une recherche qui aurait eu lieu au cours de l'étape d'authentification, donc si une recherche de nom d'utilisateur à nom unique a déjà été effectuée, le résultat de cette recherche sera mis en cache et réutilisé sans répétition.

```
<authorization>
    <ldap connection="...">
       <username-to-dn> <!-- OPTIONAL -->
           <!-- Only one of the following. -->
           <username-is-dn />
           <username-filter base-dn="..." recursive="..." user-dn-
attribute="..." attribute="..." />
           <advanced-filter base-dn="..." recursive="..." user-dn-
attribute="..." filter="..." />
        </username-to-dn>
       <group-search group-name="..." iterative="..." group-dn-
attribute="..." group-name-attribute="..." >
           <!-- One of the following -->
           <group-to-principal base-dn="..." recursive="..." search-
by="...">
               <membership-filter principal-attribute="..." />
           </group-to-principal>
           <principal-to-group group-attribute="..." />
       </group-search>
    </ldap>
</authorization>
```



IMPORTANT

Certains de ces exemples indiquent des attributs qui utilisent des valeurs par défaut. Ces valeurs sont indiquées ici pour clarifier. Les attributs qui contiennent les valeurs par défaut sont supprimées de la configuration quand c'est persisté sur le serveur.

username-to-dn

Comme mentionné plus tôt, il peut parfois être utile de définir dans la configuration d'autorisation le mappage du nom de l'utilisateur authentifié au nom unique de leur entrée dans le répertoire LDAP. L'élément de **`username-to-dn`** indique comment elle est définie. Cet élément est uniquement nécessaire si ce qui suit est bien le cas :

- L'étape d'authentification n'a pas eu lieu avec LDAP.
- La recherche de groupe utilise le nom unique pendant la recherche.

1:1 username-to-dn

Il s'agit de la forme de configuration de base et elle est utilisée pour spécifier que le nom d'utilisateur saisi par l'utilisateur éloigné est le nom unique de l'utilisateur.

```
<username-to-dn>
   <username-is-dn />
</username-to-dn>
```

Comme cela définit un mappage 1:1, il n'y a pas de configuration supplémentaire possible.

username-filter

La prochaine option est très semblable à la simple option décrite ci-dessus dans l'étape d'authentification. Un attribut est spécifié et on recherche une correspondance avec le nom d'utilisateur fourni.

```
<username-to-dn>
    <username-filter base-dn="dc=people,dc=harold,dc=example,dc=com"
recursive="false" attribute="sn" user-dn-attribute="dn" />
</username-to-dn>
```

Les attributs qui peuvent être définis sont les suivants :

- **`base-dn`** : le nom distinctif du contexte pour commencer la recherche.
- **`recursive`** : indique si la recherche va s'étendre à des sous-contextes. La valeur par défaut est **`false`**.
- **`attribute`** : l'attribut de l'entrée de l'utilisateur à faire correspondre avec le nom d'utilisateur fourni. La valeur par défaut est **`uid`**.
- **`user-dn-attribute`** : l'attribut à lire pour obtenir les noms distinctifs d'utilisateurs. La valeur par défaut est **`dn`**.

advanced-filter

L'option finale est de spécifier un filtre avancé. Comme dans la section authentification, c'est l'opportunité d'utiliser un filtre personnalisé pour trouver le nom unique de l'utilisateur.

```
<username-to-dn>
    <advanced-filter base-dn="dc=people,dc=harold,dc=example,dc=com"
recursive="false" filter="sAMAccountName={0}" user-dn-attribute="dn" />
</username-to-dn>
```

Pour les attributs qui correspondent à ceux du *username-filter*, le sens et les valeurs par défaut sont les mêmes. Il y a un nouvel attribut :

- **`filter`** : filtre personnalisé utilisé pour chercher une entrée d'utilisateur quand le nom d'utilisateur est substitué dans l'espace réservé **`{0}`**

IMPORTANT

Le code XML doit rester valide une fois que le filtre est défini donc si des caractères spéciaux comme **&** sont utilisés, assurez-vous que la forme qui convient soit utilisée. Par exemple **`& amp ;`** pour le caractère **&**.

La recherche de groupe

Comme décrit ci-dessus, il existe deux styles différents qui puissent être utilisés lors de la recherche d'informations d'appartenance de groupe. Le premier style est quand l'entrée d'utilisateur contient un attribut qui référence les groupes dont l'utilisateur est membre. Le second style est quand le groupe contient un attribut référençant l'entrée des utilisateurs

Lorsqu'il y a un choix sur le style à utiliser, Red Hat recommande que la configuration d'entrée utilisateur référençant le groupe soit utilisée. C'est parce qu'avec cette méthode, l'information de groupe peut être chargée par la lecture des attributs des noms uniques connus sans avoir à effectuer les recherches. L'autre approche nécessite des recherches extensives pour identifier les groupes qui référencent l'utilisateur.

Avant de décrire la configuration, voici quelques exemples LDIF pour illustrer cela.

Exemple 11.21. Principal à Groupe - Exemple LDIF

Cet exemple illustre un cas où nous avons un utilisateur **`TestUserOne`**, qui est membre de **`GroupOne`**, **`GroupOne`** est alors à son tour membre de **`GroupFive`**. L'appartenance au groupe est démontrée par l'utilisation d'un attribut **`memberOf`** qui est défini sur le nom unique du groupe dont l'utilisateur est membre.

Ce n'est pas affiché ici, mais un utilisateur peut avoir plusieurs attributs **`memberOf`** définis, un pour chaque groupe dont l'utilisateur est un membre direct.

```
dn: uid=TestUserOne,ou=users,dc=principal-to-group,dc=example,dc=org
objectClass: extensibleObject
objectClass: top
objectClass: groupMember
objectClass: inetOrgPerson
objectClass: uidObject
objectClass: person
objectClass: organizationalPerson
```

```
cn: Test User One
sn: Test User One
uid: TestUserOne
distinguishedName: uid=TestUserOne,ou=users,dc=principal-to-
group,dc=example,dc=org
memberOf: uid=GroupOne,ou=groups,dc=principal-to-group,dc=example,dc=org
memberOf: uid=Slashy/Group,ou=groups,dc=principal-to-
group,dc=example,dc=org
userPassword::
e1NTSEF9WFpURzhLVjc4WVZBQUJNbEI3Ym96UVAva0RTNlFNWUpLOTdTMUE9PQ==

dn: uid=GroupOne,ou=groups,dc=principal-to-group,dc=example,dc=org
objectClass: extensibleObject
objectClass: top
objectClass: groupMember
objectClass: group
objectClass: uidObject
uid: GroupOne
distinguishedName: uid=GroupOne,ou=groups,dc=principal-to-
group,dc=example,dc=org
memberOf: uid=GroupFive,ou=subgroups,ou=groups,dc=principal-to-
group,dc=example,dc=org

dn: uid=GroupFive,ou=subgroups,ou=groups,dc=principal-to-
group,dc=example,dc=org
objectClass: extensibleObject
objectClass: top
objectClass: groupMember
objectClass: group
objectClass: uidObject
uid: GroupFive
distinguishedName: uid=GroupFive,ou=subgroups,ou=groups,dc=principal-to-
group,dc=example,dc=org
```

Exemple 11.22. Groupe à Principal - Exemple LDIF

Cet exemple montre le même utilisateur **`TestUserOne`** qui est un membre de **`GroupOne`**, qui est à son tour membre de **`GroupFive`** - cependant dans ce cas, c'est un attribut **`uniqueMember`** du groupe à l'utilisateur utilisé pour la référence croisée.

Encore une fois, l'attribut utilisé pour la mise en correspondance de l'appartenance à un groupe peut être répété, si vous regardez *GroupFive*, il y a également une référence à un autre utilisateur *TestUserFive* qui n'est pas visible ici.

```
dn: uid=TestUserOne,ou=users,dc=group-to-principal,dc=example,dc=org
objectClass: top
objectClass: inetOrgPerson
objectClass: uidObject
objectClass: person
objectClass: organizationalPerson
cn: Test User One
sn: Test User One
uid: TestUserOne
userPassword::
```

```
e1NTSEF9SjR0OTRDR1ltaHc1VVZQOEJvbXhUYjl1dkFVd1lQTmRLSEdzaWc9PQ==

dn: uid=GroupOne,ou=groups,dc=group-to-principal,dc=example,dc=org
objectClass: top
objectClass: groupOfUniqueNames
objectClass: uidObject
cn: Group One
uid: GroupOne
uniqueMember: uid=TestUserOne,ou=users,dc=group-to-
principal,dc=example,dc=org

dn: uid=GroupFive,ou=subgroups,ou=groups,dc=group-to-
principal,dc=example,dc=org
objectClass: top
objectClass: groupOfUniqueNames
objectClass: uidObject
cn: Group Five
uid: GroupFive
uniqueMember: uid=TestUserFive,ou=users,dc=group-to-
principal,dc=example,dc=org
uniqueMember: uid=GroupOne,ou=groups,dc=group-to-
principal,dc=example,dc=org
```

Recherche de groupe standard

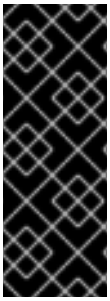
Avant de chercher des exemples pour les deux approches montrées ci-dessus, nous devons tout d'abord définir les attributs communs aux deux approches.

```
<group-search group-name="..." iterative="..." group-dn-attribute="..."
group-name-attribute="..." >
    ...
</group-search>
```

- **`group-name`** : cet attribut est utilisé pour indiquer le formulaire qui doit être utilisé pour le nom de groupe retourné comme liste de groupes dont l'utilisateur est membre, cela peut être sous la simple forme de nom du groupe ou le nom unique du groupe, si le nom unique est nécessaire, cet attribut peut être défini à **`DISTINGUISHED_NAME`**. Valeur par défaut **`SIMPLE`**.

- **`itérative`** : cet attribut est utilisé pour indiquer si, après avoir identifié les groupes qui appartiennent à un utilisateur, on doit rechercher également de manière itérative basée sur les groupes afin d'identifier quels groupes appartiennent à quels groupes. Si la recherche itérative est activée, nous continuons jusqu'à ce que nous rejoignions un groupe qui ne soit pas membre si aucun autre groupe ou cycle est détecté. Par défaut, **`false`**.

L'appartenance à un groupe cyclique n'est pas un problème. Il existe un registre de chaque recherche pour empêcher les groupes qui ont déjà été fouillés d'être recherchés à nouveau.



IMPORTANT

Pour une recherche itérative, les entrées de groupe doivent ressembler aux entrées utilisateur. La même approche qui permet d'identifier les groupes auxquels appartiennent un utilisateur est alors utilisée pour identifier les groupes auxquel le groupe appartient. Ce ne serait pas possible si, une fois que nous parlons d'appartenance inter groupes, le nom de l'attribut utilisé pour la référence croisée changeait ou si la direction de la référence changeait.

- **`group-dn-attribute`**: sur une entrée pour un groupe dont l'attribut est son nom unique. La valeur par défaut est **`dn`**.
- **`group-name-attribute`**: sur une entrée pour un groupe dont l'attribut est son simple nom. La valeur par défaut **`uid`**.

Exemple 11.23. Configuration d'exemple de Principal à Groupe

Basé sur l'exemple LDIF ci-dessus, voici un exemple de configuration chargeant itérativement un groupe d'utilisateurs où l'attribut utilisé pour référence est l'attribut **`memberOf`** de l'utilisateur.

```
<authorization>
    <ldap connection="LocalLdap">
        <username-to-dn>
            <username-filter base-dn="ou=users,dc=principal-to-
group,dc=example,dc=org" recursive="false" attribute="uid" user-dn-
attribute="dn" />
        </username-to-dn>
        <group-search group-name="SIMPLE" iterative="true" group-dn-
attribute="dn" group-name-attribute="uid">
            <principal-to-group group-attribute="memberOf" />
        </group-search>
    </ldap>
</authorization>
```

L'aspect le plus important de cette configuration est que l'élément **`principal-to-group`** a été ajouté avec un seul attribut.

- **`group-attribute`** : le nom de l'attribut sur l'entrée d'utilisateur qui correspond au nom unique du groupe qui appartient à l'utilisateur. La valeur par défaut **`memberOf`**.

Exemple 11.24. Configuration d'exemple de Groupe à Principal

Cet exemple vous montre une recherche interactive pour l'exemple groupe à principal LDIF montré ci-dessus.

```
<authorization>
      <ldap connection="LocalLdap">
          <username-to-dn>
              <username-filter base-dn="ou=users,dc=group-to-
principal,dc=example,dc=org" recursive="false" attribute="uid" user-dn-
attribute="dn" />
          </username-to-dn>
          <group-search group-name="SIMPLE" iterative="true" group-dn-
```

```
attribute="dn" group-name-attribute="uid">
                <group-to-principal base-dn="ou=groups,dc=group-to-
principal,dc=example,dc=org" recursive="true" search-
by="DISTINGUISHED_NAME">
                    <membership-filter principal-attribute="uniqueMember"
/>
                </group-to-principal>
            </group-search>
        </ldap>
    </authorization>
```

Un élément **`group-to-principal`** est ajouté ici. Cet élément est utilisé pour définir comment les recherches de groupes qui référencent l'entrée de l'utilisateur seront exécutées. Les attributs suivants sont définis :

- **`base-dn`** : le nom unique du contexte à utiliser pour commencer la recherche.
- **`recursive`** : indique si les sous-contextes peuvent également être recherchés. La valeur par défaut est **`false`**.
- **`search-by`** : la forme du nom de rôle utilisé dans les recherches. Valeurs valides **`SIMPLE`** et **`DISTINGUISHED_NAME`**. Valeur par défaut **`DISTINGUISHED_NAME`**.

Dans l'élément *group-to-principal*, il y a un élément *membership-filter* pour définir la correspondance.

- **`principal-attribute`** : le nom de l'attribut d'entrée de groupe qui référence l'entrée utilisateur. La valeur par défaut est **`member`**.

Rapporter un bogue

11.9.9.8. Scoped rôles

Les scoped rôles sont des rôles définis par l'utilisateur, qui donnent les permissions d'un des rôles standards, mais uniquement pour un ou plusieurs groupes ou hôtes spécifiés. Les scoped rôles permettent à la gestion d'utilisateurs d'octroyer des permissions limitées aux groupes de serveurs et aux hôtes requis.

Le scoped rôles peuvent être créés par des utilisateurs auxquels les rôles administrateur ou superutilisateur sont assignés.

Ils se résument par cinq caractéristiques :

1. Un nom unique.
2. Correspond aux rôles standards sur lesquels il est basé.
3. S'il s'applique aux groupes de serveurs ou aux hôtes
4. La liste des groupes de serveurs ou hôtes auxquels il est limité.
5. Si tous les utilisateurs sont inclus automatiquement. La valeur par défaut sera false.

Une fois créé, un scoped rôle peut être assigné à des utilisateurs ou à des groupes de la même façon que les rôles standards.

La création d'un scoped rôle ne vous laisse pas définir de nouvelles permissions. Les scoped rôles vous permettent uniquement d'appliquer les permissions d'un rôle existant dans une portée limitée. Par exemple, vous pouvez créer un scoped rôle basé sur le rôle Deployer, qui est limité à un groupe de serveurs uniques.

Il n'y a que deux scopes auxquels les rôles peuvent être limités, hôte et groupe de serveurs.

Rôles host-scoped

Un rôle non host-scoped limite les permissions de ce rôle à un ou plusieurs hôtes. Cela signifie que l'accès est donné aux arborescences de ressources **`/host=*/`** qui conviennent, mais les ressources spécifiques à d'autres hôtes sont cachées.

Rôles server-group-scoped

Un rôle server-group-scoped limite les permissions de ce rôle à un ou plusieurs groupes de serveurs. De plus, les permissions de rôle seront appliquées également au profil, groupe de liaison de socket, config du serveur et aux ressources de serveur associés avec les groupes de serveur spécifiés. Toutes les ressources secondaires à l'intérieur non liées de manière logique au groupe de serveurs ne seront pas visibles par l'utilisateur.

Les rôles sever-group-scoped et les hôtes ont les permissions du rôle Monitor pour le reste de la configuration du domaine partagé.

Rapporter un bogue

11.9.9.9. Création de scoped roles

Les scoped roles sont des rôles définis par l'utilisateur qui accordent les permissions d'un des rôles standard mais seulement pour un ou plusieurs groupes de serveurs ou d'hôtes. Cette rubrique montre comment créer des scoped roles.

Seuls les utilisateurs ayant les rôles **`SuperUser`** ou **`Administrator`** peuvent effectuer cette configuration.

La configuration de scoped roles de la console de gestion suit les étapes suivantes :

1. Connectez-vous à la console de gestion
2. Cliquez sur l'onglet **Administration**
3. Déployez le menu **Access Control** et sélectionnez **Role Assignment**.
4. Sélectionnez l'onglet **ROLES**, puis l'onglet **Scoped Roles** à l'intérieur.

La section **`Scoped Roles`** de la Console de gestion comprend deux parties principales, un tableau qui contient une liste de scoped roles configurés, et un panneau **`Selection`** qui affiche les détails du rôle actuellement sélectionné dans le tableau.

Les procédures suivantes vous montrent comment effectuer des tâches de configuration de scoped roles.

Procédure 11.29. Ajoutez un nouveau scoped role

1. Connectez-vous à la console de gestion

2. Naviguez dans la partie **Scoped Roles** de l'onglet **Roles**.
3. Cliquez sur le bouton **Add**. Le dialogue **`Add Scoped Role`** apparaîtra.
4. Indiquez les détails suivants :
   - **`Name`**, le nom distinctif du nouveau scoped role.
   - **`Base Role`**, le rôle sur lequel ce rôle basera sa permission.
   - **`Type`**, indique si ce rôle est limité à des hôtes ou à des groupes de serveurs.
   - **`Scope`**, la liste d'hôtes ou de groupes de serveurs auxquels le rôle est limité. Vous pouvez sélectionner plusieurs entrées.
   - **`Include All`**, indique si le rôle doit automatiquement inclure tous les utilisateurs. La valeur par défaut est non.
5. Cliquez sur le bouton de sauvegarde **Save**. La boîte de dialogue se fermera et le rôle nouvellement créé apparaîtra dans le tableau.

Procédure 11.30. Modifiez un scoped role.

1. Connectez-vous à la console de gestion
2. Naviguez dans la partie **Scoped Roles** de l'onglet **Roles**.
3. Cliquez sur le scoped role que vous souhaitez modifier dans le tableau. Les détails de ce rôle apparaissent dans le panneau **`Selection`** qui se trouve sous le tableau.
4. Cliquez sur **`Edit`** dans le panneau **`Selection`**. Le panneau **`Selection`** entre en mode d'édition.
5. Mettre à jour les informations que vous souhaitez modifier et cliquez sur le bouton de sauvegarde **`Save`**. Le panneau **`Selection`** retournera à son état précédent. Le panneau de sélection **`Selection`** et le tableau afficheront les informations nouvellement mises à jour.

Procédure 11.31. Affichez les membres de scoped role

1. Connectez-vous à la console de gestion
2. Naviguez dans la partie **`Scoped Roles`** de l'onglet **Roles**.
3. Cliquez sur le scoped role du tableau dont vous souhaitez voir les **Members**, puis cliquez sur le bouton **Members**. La boîte de dialogue **Members of role** apparaîtra. Elle nous affichera les utilisateurs et les groupes qui sont inclus ou exclus du rôle.
4. Cliquez sur le bouton **`Done`** (terminé) quand vous aurez fini de consulter cette information.

Procédure 11.32. Supprimez un scoped role.

IMPORTANT

Un **Scoped Role** ne peut pas être supprimé si des utilisateurs ou groupes y sont assignés. Supprimez les assignations de rôle pour commencer, puis le supprimer.

1. Connectez-vous à la console de gestion
2. Naviguez dans la partie **Scoped Roles** de l'onglet **Roles**.
3. Sélectionnez le scoped role à supprimer du tableau.
4. Cliquez sur le bouton **Remove**. Le dialogue **`Remove Scoped Role`** apparaîtra.
5. Cliquez sur le bouton **Confirm** (confirmer). La boîte de dialogue se fermera et le rôle sera supprimé.

Rapporter un bogue

11.9.10. Configurer les contraintes

11.9.10.1. Configurez les contraintes de sensibilité

Chaque contrainte de sensibilité définit un ensemble de ressources qui sont considérées comme « sensibles ». Une ressource sensible est généralement une ressource qui doit être tenue secrète, comme les mots de passe, ou une ressource qui aura de graves répercussions sur le serveur, comme la mise en réseau, la configuration de la JVM ou les propriétés système. Le système de contrôle d'accès lui-même est également considéré comme sensible. La sensibilité des ressources limite quels rôles sont capables de lire, écrire ou répondre à une ressource spécifique.

La configuration de contraintes de sensibilité se trouve dans l'API de gestion à l'adresse suivante : **`/core-service=management/access=authorization/constraint=sensitivity-classification`**.

Dans le modèle de gestion, chaque contrainte de ressource est identifiée en tant que **`classification`**. Les classifications sont ensuite regroupées en **`types`**. Il existe 39 catégories incluses qui sont regroupées en 13 types.

Pour configurer une contrainte de sensibilité, l'opération **`write-attribut`** permet de paramétrer les attributs **`configured-requires-read`**, **`configured-requires-write`**, ou **`configured-requires-addressable`**. Pour rendre ce type d'opération sensible, définir la valeur de l'attribut à **`true`**, sinon, pour le rendre non sensible, le définir à la valeur **`false`**. Par défaut, ces attributs ne sont pas définis et les valeurs **`default-requires-read`**, **`default-requires-write`**, et **`default-requires-addressable`** seront utilisées. Une fois que l'attribut est configuré, ce sera cette valeur qui sera utilisée à la place de la valeur par défaut. Impossible de modifier les valeurs par défaut.

Exemple 11.25. Comment rendre les propriétés système des opérations sensibles.

```
[domain@localhost:9999 /] cd /core-
service=management/access=authorization/constraint=sensitivity-
classification/type=core/classification=system-property
[domain@localhost:9999 classification=system-property] :write-
attribute(name=configured-requires-read, value=true)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" => {"master" => {
        "server-one" => {"response" => {"outcome" => "success"}},
        "server-two" => {"response" => {"outcome" => "success"}}
    }}}}
}
```

```
[domain@localhost:9999 classification=system-property] :read-resource
{
    "outcome" => "success",
    "result" => {
        "configured-requires-addressable" => undefined,
        "configured-requires-read" => true,
        "configured-requires-write" => undefined,
        "default-requires-addressable" => false,
        "default-requires-read" => false,
        "default-requires-write" => true,
        "applies-to" => {
            "/host=master/system-property=*" => undefined,
            "/host=master/core-service=platform-mbean/type=runtime" =>
undefined,
            "/server-group=*/system-property=*" => undefined,
            "/host=master/server-config=*/system-property=*" =>
undefined,
            "/host=master" => undefined,
            "/system-property=*" => undefined,
            "/" => undefined
        }
    }
}
[domain@localhost:9999 classification=system-property]
```

Les rôles qui seront en mesure d'effectuer ces opérations en fonction de la configuration de ces attributs sont résumés dans Tableau 11.6, « Résultats des configurations de contraintes de sensibilité ».

Tableau 11.6. Résultats des configurations de contraintes de sensibilité

Valeur	`requires-read`	`requires-write`	`requires-addressable`
`true`	L'opération lecture est sensible. Seuls les rôles **`Auditor`**, **`Administrator`**, et **`SuperUser`** peuvent lire.	L'opération écriture est sensible. Seuls les rôles **`Administrator`** et **`SuperUser`** peuvent écrire.	L'opération Adressage est sensible. Seuls les rôles d'**`Auditor`**, **`Administrator`**, et **`SuperUser`** peuvent lire.
`false`	L'opération lecture n'est pas sensible. Tous les utilisateurs de gestion peuvent lire.	L'opération écriture n'est pas sensible. Les rôles **`Maintainer`**, **`Administrator`**, et **`SuperUser`** peuvent écrire. Les rôles **`Deployers`** peuvent également écrire dans une ressource d'applications.	L'opération Adressage n'est pas sensible. Tous les utilisateurs de gestion peuvent adresser.

Rapporter un bogue

11.9.10.2. Configurer les contraintes de ressources d'application

Chaque contrainte de ressource d'application définit un ensemble de ressources, d'attributs et d'opérations généralement associées avec le déploiement d'applications et de services. Lorsqu'une contrainte de ressource d'application est activée, les utilisateurs gestionnaires du rôle Déployeur peuvent accéder aux ressources applicables.

La configuration de contraintes d'applications se trouve dans le modèle des gestion qui se trouve dans **`/core-service=management/access=authorization/constraint=application-classification/`**.

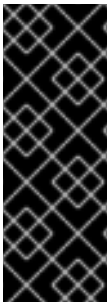
Dans le modèle de gestion, chaque contrainte de ressource d'application est identifiée comme une **`classification`**. Les classifications sont ensuite regroupées en **`types`**. Il existe 14 catégories incluses qui sont regroupées en 8 types. Chaque classification comporte un élément **`applies-to`** qui correspond à une liste de modèles de chemins d'accès de ressources s'appliquant à la configuration de la classification.

Par défaut, la seule classification de ressource d'application activée est **`core`**. Core inclut des déploiements, des superpositions de déploiement et des opérations de déploiement.

Afin d'activer une ressource d'application, l'opération **`write-attribute`** permet de définir l'attribut **`configured-application attribute`** de la classification sur **`true`**. Pour désactiver une ressource d'application, définir cet attribut sur **`false`**. Par défaut, ces attributs ne sont pas définis et la valeur **`default-application attribute`** sera utilisée. La valeur par défaut ne peut pas être modifiée.

Exemple 11.26. Activer la classification des ressources d'application logger-profile

```
[domain@localhost:9999 /] cd /core-
service=management/access=authorization/constraint=application-
classification/type=logging/classification=logging-profile
[domain@localhost:9999 classification=logging-profile] :write-
attribute(name=configured-application, value=true)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" => {"master" => {
        "server-one" => {"response" => {"outcome" => "success"}},
        "server-two" => {"response" => {"outcome" => "success"}}
    }}}}
}
[domain@localhost:9999 classification=logging-profile] :read-resource
{
    "outcome" => "success",
    "result" => {
        "configured-application" => true,
        "default-application" => false,
        "applies-to" => {"/profile=*/subsystem=logging/logging-
profile=*" => undefined}
    }
}
[domain@localhost:9999 classification=logging-profile]
```



IMPORTANT

Les contraintes de ressources d'application s'appliquent à toutes les ressources qui correspondent à sa configuration. Par exemple, il n'est pas possible d'accorder à un utilisateur ayant la permission **`Deployer`** d'accéder à une ressource de source de données mais pas à une autre. Si ce niveau de séparation est nécessaire, il est recommandé de configurer les ressources dans différents groupes de serveurs et de créer les rôles **`Deployer`** d'étendue différente pour chaque groupe.

Rapporter un bogue

11.9.10.3. Configuration de contraintes d'expressions d'archivage sécurisé

Par défaut, la lecture et l'écriture d'expressions d'archivage sécurisé sont des opérations sensibles. La configuration de contraintes d'expression d'archivage sécurisé permet de définir l'une, ou ces deux opérations à être insensibles. Modifier cette contrainte permet à un plus grand nombre de rôles de lire et d'écrire des expressions d'archivage sécurisé.

La contrainte d'expression de sécurité se trouve dans le modèle de gestion de **`/core-service=management/access=authorization/constraint=vault-expression`**.

Pour configurer la contrainte d'expression d'archivage sécurisé, utilisez l'opération **`write-attribute`** pour définir les attributs de **`configured-requires-write`** et **`configured-requires-read`** à **`true`** ou **`false`**. Par défaut, celles-ci ne sont pas définies et les valeurs **`default-requires-read`** et **`default-requires-write`** sont utilisées. Impossible de modifier les valeurs par défaut.

Exemple 11.27. Comment rendre l'écriture d'expressions d'archivage une opération non sensible

```
[domain@localhost:9999 /] cd /core-
service=management/access=authorization/constraint=vault-expression
[domain@localhost:9999 constraint=vault-expression] :write-
attribute(name=configured-requires-write, value=false)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"main-server-group" => {"host" => {"master" => {
        "server-one" => {"response" => {"outcome" => "success"}},
        "server-two" => {"response" => {"outcome" => "success"}}
    }}}}
}
[domain@localhost:9999 constraint=vault-expression] :read-resource
{
    "outcome" => "success",
    "result" => {
        "configured-requires-read" => undefined,
        "configured-requires-write" => false,
        "default-requires-read" => true,
        "default-requires-write" => true
    }
}
[domain@localhost:9999 constraint=vault-expression]
```

Les rôles qui seront en mesure de lire et d'écrire dans les expressions d'archivage sécurisé en fonction de cette configuration sont résumés dans Tableau 11.7, « Résultats des configurations de contraintes d'expressions d'archivage sécurisé ».

Tableau 11.7. Résultats des configurations de contraintes d'expressions d'archivage sécurisé

Valeur	`requires-read`	`requires-write`
`true`	L'opération lecture est sensible. Seuls les rôles d'**`Auditor`**, **`Administrator`**, et **`SuperUser`** peuvent lire.	L'opération écriture est sensible. Seuls les rôles **`Administrator`**, et **`SuperUser`** peuvent écrire.
`false`	L'opération lecture n'est pas sensible. Tous les utilisateurs de gestion peuvent lire.	L'opération d'écriture n'est pas sensible. Les rôles **`Monitor`**, **`Administrator`**, et **`SuperUser`** peuvent écrire. Les rôles **`Deployers`** peuvent également écrire si l'expression d'archivage sécurisé est dans une ressource d'application.

Rapporter un bogue

11.9.11. Références de contraintes

11.9.11.1. Références de contraintes de ressources d'application

Type: core

Classification: deployment-overlay

- default: true
- PATH: /deployment-overlay=*
- PATH: /deployment=*
- PATH: /

  Opération :

  upload-deployment-stream, full-replace-deployment, upload-deployment-url, upload-deployment-bytes

Type: datasources

Classification: datasource

- default: false
- PATH: /deployment=*/subdeployment=*/subsystem=datasources/data-source=*

- PATH: /subsystem=datasources/data-source=*
- PATH: /subsystem=datasources/data-source=ExampleDS
- PATH: /deployment=*/subsystem=datasources/data-source=*

Classification: jdbc-driver

- default: false
- PATH: /subsystem=datasources/jdbc-driver=*

Classification: xa-data-source

- default: false
- PATH: /subsystem=datasources/xa-data-source=*
- PATH: /deployment=*/subsystem=datasources/xa-data-source=*
- PATH: /deployment=*/subdeployment=*/subsystem=datasources/xa-data-source=*

Type: logging

Classification: logger

- default: false
- PATH: /subsystem=logging/logger=*
- PATH: /subsystem=logging/logging-profile=*/logger=*

Classification: logging-profile

- default: false
- PATH: /subsystem=logging/logging-profile=*

Type: mail

Classification: mail-session

- default: false
- PATH: /subsystem=mail/mail-session=*

Type: naming

Classification: binding

- default: false
- PATH: /subsystem=naming/binding=*

Type: resource-adapters

Classification: resource-adapters

- default: false
- PATH: /subsystem=resource-adapters/resource-adapter=*

Type: security

Classification: security-domain

- default: false
- PATH: /subsystem=security/security-domain=*

Rapporter un bogue

11.9.11.2. Références de contraintes de sensibilité

Type: core

Classification: access-control

- requires-addressable: true
- requires-read: true
- requires-write: true
- PATH: /core-service=management/access=authorization
- PATH: /subsystem=jmx ATTRIBUTE: non-core-mbean-sensitivity

Classification: credential

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=mail/mail-session=*/server=pop3 ATTRIBUTE: username , password
- PATH: /subsystem=mail/mail-session=*/server=imap ATTRIBUTE: username , password
- PATH: /subsystem=datasources/xa-data-source=* ATTRIBUTE: user-name, recovery-username, password, recovery-password
- PATH: /subsystem=mail/mail-session=*/custom=* ATTRIBUTE: username, password
- PATH: /subsystem=datasources/data-source=*" ATTRIBUTE: user-name, password
- PATH: /subsystem=remoting/remote-outbound-connection=*" ATTRIBUTE: username

- PATH: /subsystem=mail/mail-session=*/server=smtp ATTRIBUTE: username, password
- PATH: /subsystem=web/connector=*/configuration=ssl ATTRIBUTE: key-alias, password
- PATH: /subsystem=resource-adapters/resource-adapter=*/connection-definitions=*" ATTRIBUTE: recovery-username, recovery-password

Classification: domain-controller

- requires-addressable: false
- requires-read: false
- requires-write: true

Classification: domain-names

- requires-addressable: false
- requires-read: false
- requires-write: true

Classification: extensions

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /extension=*

Classification: jvm

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=platform-mbean/type=runtime ATTRIBUTE: input-arguments, boot-class-path, class-path, boot-class-path-supported, library-path

Classification: management-interfaces

- requires-addressable: false
- requires-read: false
- requires-write: true
- /core-service=management/management-interface=native-interface
- /core-service=management/management-interface=http-interface

Classification: module-loading

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=module-loading

Classification: patching

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=patching/addon=*
- PATH: /core-service=patching/layer=*"
- PATH: /core-service=patching

Classification: read-whole-config

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: / OPERATION: read-config-as-xml

Classification: security-domain

- requires-addressable: true
- requires-read: true
- requires-write: true
- PATH: /subsystem=security/security-domain=*

Classification: security-domain-ref

- requires-addressable: true
- requires-read: true
- requires-write: true
- PATH: /subsystem=datasources/xa-data-source=* ATTRIBUTE: security-domain
- PATH: /subsystem=datasources/data-source=* ATTRIBUTE: security-domain
- PATH: /subsystem=ejb3 ATTRIBUTE: default-security-domain

- PATH: /subsystem=resource-adapters/resource-adapter=*/connection-definitions=* ATTRIBUTE: security-domain, recovery-security-domain, security-application, security-domain-and-application

Classification: security-realm

- requires-addressable: true
- requires-read: true
- requires-write: true
- PATH: /core-service=management/security-realm=*

Classification: security-realm-ref

- requires-addressable: true
- requires-read: true
- requires-write: true
- PATH: /subsystem=remoting/connector=* ATTRIBUTE: security-realm
- PATH: /core-service=management/management-interface=native-interface ATTRIBUTE: security-realm
- PATH: /core-service=management/management-interface=http-interface ATTRIBUTE: security-realm
- PATH: /subsystem=remoting/remote-outbound-connection=* ATTRIBUTE: security-realm

Classification: security-vault

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=vault

Classification: service-container

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=service-container

Classification: snapshots

- requires-addressable: false

- requires-read: false
- requires-write: false
- PATH: / ATTRIBUTE: take-snapshot, list-snapshots, delete-snapshot

Classification: socket-binding-ref

- requires-addressable: false
- requires-read: false
- requires-write: false
- PATH: /subsystem=mail/mail-session=*/server=pop3 ATTRIBUTE: outbound-socket-binding-ref
- PATH: /subsystem=mail/mail-session=*/server=imap ATTRIBUTE: outbound-socket-binding-ref
- PATH: /subsystem=remoting/connector=* ATTRIBUTE: socket-binding
- PATH: /subsystem=web/connector=* ATTRIBUTE: socket-binding
- PATH: /subsystem=remoting/local-outbound-connection=* ATTRIBUTE: outbound-socket-binding-ref
- PATH: /socket-binding-group=*/local-destination-outbound-socket-binding=* ATTRIBUTE: socket-binding-ref
- PATH: /subsystem=remoting/remote-outbound-connection=* ATTRIBUTE: outbound-socket-binding-ref
- PATH: /subsystem=mail/mail-session=*/server=smtp ATTRIBUTE: outbound-socket-binding-ref
- PATH: /subsystem=transactions ATTRIBUTE: process-id-socket-binding, status-socket-binding, socket-binding

Classification: socket-config

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /interface=* OPERATION: resolve-internet-address
- PATH: /core-service=management/management-interface=native-interface ATTRIBUTE: port, interface, socket-binding
- PATH: /socket-binding-group=*
- PATH: /core-service=management/management-interface=http-interface ATTRIBUTE: port, secure-port, interface, secure-socket-binding, socket-binding

- PATH: / OPERATION: resolve-internet-address
- PATH: /subsystem=transactions ATTRIBUTE: process-id-socket-max-ports

Classification: system-property

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /core-service=platform-mbean/type=runtime ATTRIBUTE: system-properties
- PATH: /system-property=*
- PATH: / OPERATION: resolve-expression

Type: datasources

Classification: data-source-security

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=datasources/xa-data-source=* ATTRIBUTE: user-name, security-domain, password
- PATH: /subsystem=datasources/data-source=* ATTRIBUTE: user-name, security-domain, password

Type: jdr

Classification: jdr

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /subsystem=jdr OPERATION: generate-jdr-report

Type: jmx

Classification: jmx

- requires-addressable: false
- requires-read: false

- requires-write: true
- PATH: /subsystem=jmx

Type: mail

Classification: mail-server-security

- requires-addressable: false
- requires-read: false
- requires-write: true
- PATH: /subsystem=mail/mail-session=*/server=pop3 ATTRIBUTE: username, tls, ssl, password
- PATH: /subsystem=mail/mail-session=*/server=imap ATTRIBUTE: username, tls, ssl, password
- PATH: /subsystem=mail/mail-session=*/custom=* ATTRIBUTE: username, tls, ssl, password
- PATH: /subsystem=mail/mail-session=*/server=smtp ATTRIBUTE: username, tls, ssl, password

Type: naming

Classification: jndi-view

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=naming OPERATION: jndi-view

Classification: naming-binding

- requires-addressable: false
- requires-read: false
- requires-write: false
- PATH: /subsystem=naming/binding=*

Type: remoting

Classification: remoting-security

- requires-addressable: false
- requires-read: true

- requires-write: true
- PATH: /subsystem=remoting/connector=* ATTRIBUTE: authentication-provider, security-realm
- PATH: /subsystem=remoting/remote-outbound-connection=* ATTRIBUTE: username, security-realm
- PATH: /subsystem=remoting/connector=*/security=sasl

Type: resource-adapters

Classification: resource-adapter-security

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=resource-adapters/resource-adapter=*/connection-definitions=* ATTRIBUTE: security-domain, recovery-username, recovery-security-domain, security-application, security-domain-and-application, recovery-password

Type: security

Classification: misc-security

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=security ATTRIBUTE: deep-copy-subject-mode

Type: web

Classification: web-access-log

- requires-addressable: false
- requires-read: false
- requires-write: false
- PATH: /subsystem=web/virtual-server=*/configuration=access-log

Classification: web-connector

- requires-addressable: false
- requires-read: false

- requires-write: false
- PATH: /subsystem=web/connector=*

Classification: web-ssl

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=web/connector=*/configuration=ssl

Classification: web-sso

- requires-addressable: false
- requires-read: true
- requires-write: true
- PATH: /subsystem=web/virtual-server=*/configuration=sso

Classification: web-valve

- requires-addressable: false
- requires-read: false
- requires-write: false
- PATH: /subsystem=web/valve=*

Rapporter un bogue

11.10. SÉCURITÉ DE RÉSEAU

11.10.1. Sécuriser les interfaces de gestion

Résumé

Dans un environnement de test, il est de pratique courante de faire fonctionner JBoss EAP 6 sans couche de sécurité sur les interfaces de gestion, composées de la console de gestion, de l'interface CLI et d'autres implémentations de l'API. Cela permet des changements rapides de développement et de configuration.

De plus, un mode silencieux d'authentification est présent par défaut, permettant à un client local sur une machine hôte de se connecter au Management CLI sans exiger un nom d'utilisateur ou un mot de passe. Ce comportement est pratique pour les utilisateurs locaux et les scripts de l'interface CLI, mais il peut être désactivé si nécessaire. La procédure est décrite dans la rubrique Section 11.8.6, « Supprimer l'authentification silencieuse du domaine de sécurité par défaut. ».

Quand vous commencez à tester ou à préparer votre environnement pour aller en production, il est extrêmement important de sécuriser les interfaces de gestion au moins par l'une des méthodes suivantes :

- Section 11.10.2, « Indiquer l'interface de réseau que JBoss EAP 6 utilise »
- Section 11.10.4, « Configurer les pare-feux de réseau pour qu'ils fonctionnent avec JBoss EAP 6 »

Rapporter un bogue

11.10.2. Indiquer l'interface de réseau que JBoss EAP 6 utilise

Aperçu

En isolant les services afin qu'ils soient accessibles uniquement aux clients qui en ont besoin, vous augmentez la sécurité de votre réseau. JBoss EAP 6 comprend deux interfaces dans sa configuration par défaut, qui se lient à l'adresse IP **`127.0.0.1`** ou **`localhost`**, par défaut. Une des interfaces est appelée **`gestion`** et est utilisée par la console de gestion, le CLI et l'API. L'autre est appelée **`public`** et est utilisée pour déployer des applications. Ces interfaces ne sont pas spéciales ou importantes, mais sont fournies comme point de départ.

L'interface **`management`** utilise les ports **`9990`** et **`9999`** par défaut, et l'interface **`public`** utilise le port **`8080`**, ou le port **`8443`** avec HTTPS.

Vous pouvez changer l'adresse IP de l'interface de gestion, de l'interface publique ou bien les deux à la fois.

AVERTISSEMENT

Si vous exposez les interfaces de gestion à d'autres interfaces de réseau non accessibles par les hôtes distants, vérifiez les implications au niveau de la sécurité. Le plus souvent, il n'est pas conseillé de donner un accès à distance aux interfaces de gestion.

1. **Stopper le serveur JBoss EAP 6**
   Stoppez JBoss EAP 6 en envoyant une interruption de manière appropriée à votre système d'exploitation. Si vous exécutez JBoss EAP 6 comme une application de premier plan, il suffit d'appuyer sur **`Ctrl+C`**.

2. **Démarrer JBoss EAP 6 à nouveau, en spécifiant l'adresse de liaison.**
   Utilisez l'argument de ligne de commande **`-b`** pour démarrer JBoss EAP 6 sur une interface particulière.

   Exemple 11.28. Indiquez l'interface publique.

   ```
   EAP_HOME/bin/domain.sh -b 10.1.1.1
   ```

Exemple 11.29. Indiquez l'interface de gestion.

```
EAP_HOME/bin/domain.sh -bmanagement=10.1.1.1
```

Exemple 11.30. Indiquez des adresses différentes pour chaque interface.

```
EAP_HOME/bin/domain.sh -bmanagement=127.0.0.1 -b 10.1.1.1
```

Exemple 11.31. Liez l'interface publique à toutes les interfaces de réseau.

```
EAP_HOME/bin/domain.sh -b 0.0.0.0
```

Il est possible de modifier votre fichier de configuration XML directement, pour changer les adresses de liaison par défaut. Toutefois, si vous faites cela, vous ne serez plus en mesure d'utiliser l'argument de ligne de commande **-b** pour spécifier une adresse IP en cours d'exécution, donc ce n'est pas recommandé. Si vous décidez de le faire, n'oubliez pas de stopper JBoss EAP 6 complètement avant d'éditer le fichier XML.

Rapporter un bogue

11.10.3. Ports de réseau utilisés par JBoss EAP 6

Les ports utilisés par la configuration JBoss EAP 6 par défaut sont liés à plusieurs facteurs :

- Le fait que vos groupes de serveurs utilisent le groupe de liaisons de sockets par défaut, ou un groupe de liaisons de sockets personnalisé.
- Les exigences de vos déploiements individuels.

NOTE

Un décalage de port numérique peut être configuré pour atténuer les conflits de ports lorsque vous exécutez plusieurs serveurs sur un même serveur physique. Si votre serveur utilise un décalage de port numérique, ajoutez la valeur de décalage au numéro de port par défaut pour le groupe de liaisons de sockets de son groupe de serveurs. Par exemple, si le port HTTP du groupe de liaisons de socket est **8080** et si votre serveur utilise un décalage de port de **100**, son port HTTP sera **8180**.

À moins d'instruction particulière, les ports utilisent le protocole TCP.

Groupes de liaison de sockets par défaut

- **full-ha-sockets**
- **full-sockets**
- **ha-sockets**

- `standard-sockets`

Tableau 11.8. Référence aux groupes de liaisons de sockets par défaut

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
`ajp`	8009		Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges.	Oui	Oui	Oui	Oui
`http`	8080		Le port par défaut des applications déployées.	Oui	Oui	Oui	Oui
`https`	8443		Connexion cryptée-SSL entre les applications déployées et les clients.	Oui	Oui	Oui	Oui
`jacorb`	3528		Services CORBA pour les transactions JTS et autres services dépendants-ORB.	Oui	Oui	Non	Non
`jacorb-ssl`	3529		Services CORBA cryptés-SSL.	Oui	Oui	Non	Non
`jgroups-diagnostics`		7500	Multidiffusion. Utilisée pour découvrir des homologues dans les groupements HA. Non configurable par les interfaces de gestion.	Oui	Non	Oui	Non
`jgroups-mping`		45700	Multidiffusion. Utilisée pour découvrir l'appartenance de groupe d'origine dans un cluster HA.	Oui	Non	Oui	Non

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
jgroups-tcp	7600		Découverte d'homoplogues unicastes dans les groupements HA avec TCP.	Oui	Non	Oui	Non
jgroups-tcp-fd	57600		Utilisé pour la détection des échecs en TCP.	Oui	Non	Oui	Non
jgroups-udp	55200	45688	Découverte d'homologues unicastes dans les groupements HA avec UDP.	Oui	Non	Oui	Non
jgroups-udp-fd	54200		Utilisé pour la détection des échecs par UDP.	Oui	Non	Oui	Non
messaging	5445		Service JMS.	Oui	Oui	Non	Non
messaging-group			Référencé par la diffusion HornetQ JMS et les groupes Discovery	Oui	Oui	Non	Non
messaging-throughput	5455		Utilisé par JMS Remoting.	Oui	Oui	Non	Non
mod_cluster		23364	Port de multidiffusion pour la communication entre JBoss EAP 6 et l'équilibreur de charges HTTP.	Oui	Non	Oui	Non
osgi-http	8090		Utilisé par les composants internes qui utilisent le sous-système OSGi. Non configurable par les interfaces de gestion.	Oui	Oui	Oui	Oui

Nom	Port	Port multi-diffusion	Description	full-ha-sockets	full-sockets	ha-socket	standard-socket
remoting	4447		Utilisé pour l'invocation EJB.	Oui	Oui	Oui	Oui
txn-recovery-environment	4712		Gestionnaire de recouvrement des transactions JTA.	Oui	Oui	Oui	Oui
txn-status-manager	4713		Gestionnaire des transactions JTA / JTS.	Oui	Oui	Oui	Oui

Ports de gestion

En plus des groupes de liaisons de socket, chaque contrôleur hôte ouvre deux ports supplémentaires pour la gestion :

- **9990** - Le port de console de gestion web
- **9999** - Le port utilisé par la console de gestion et par l'API de gestion.

De plus, si HTTPS est activé pour la console de gestion, 9443 sera également ouvert en tant que valeur de port par défaut.

Rapporter un bogue

11.10.4. Configurer les pare-feux de réseau pour qu'ils fonctionnent avec JBoss EAP 6

Résumé

La plupart des environnements de production utilisent des pare-feux dans le cadre d'une stratégie globale de sécurité réseau. Si vous avez besoin que plusieurs instances de serveur puissent communiquer entre elles ou avec des services externes tels que des serveurs web ou des bases de données, votre pare-feu doit en tenir compte. Un pare-feu bien géré ouvre uniquement les ports qui sont nécessaires à l'opération et limite l'accès aux ports pour des adresses IP, des sous-réseaux et protocoles réseau spécifiques.

Une discussion plus complète sur les pare-feux est au delà du dessein de cette documentation.

Conditions préalables

- Déterminer les ports que vous souhaitez ouvrir.
- Vous devez avoir une bonne compréhension de vos logiciels de pare-feux. Cette procédure utilise la commande **system-config-firewall** de Red Hat Enterprise Linux 6. Microsoft

Windows Server inclut un pare-feu intégré, et plusieurs solutions de pare-feux de tierce partie existent pour chaque plate-forme. Sur un serveur Microsoft Windows, vous pouvez utiliser PowerShell pour configurer le pare-feu.

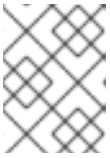
Hypothèses

Cette procédure configure un pare-feu dans un environnement qui comprend les hypothèses suivantes :

- Le système d'exploitation est Red Hat Enterprise Linux 6
- JBoss EAP 6 exécute sur l'hôte **`10.1.1.2`**. En option, le serveur peut avoir son propre pare-feu.
- Le serveur du pare-feu de réseau exécute sur l'hôte **`10.1.1.1`** sur l'interface **`eth0`**, et possède une interface externe **`eth1`**.
- Le trafic de réseau du port **`5445`** (port utilisé par JMS) devra être renvoyé sur JBoss EAP 6. Aucun autre trafic doit pouvoir transiter par le pare-feu du réseau.

Procédure 11.33. Gérer les pare-feux de réseau pour qu'ils soient opérationnels dans JBoss EAP 6

1. **Connectez-vous à la console de management.**
   Connectez-vous dans la console de gestion. Par défaut, elle exécute sur http://localhost:9990/console/.
2. **Déterminer les liaisons de socket utilisées par le groupe de liaisons de socket.**
   a. Cliquez sur l'étiquette **`Configuration`** qui se trouve en haut de la console de gestion.
   b. Étendre le menu **`General Configuration`**. Sélectionnez **`Socket Binding`** (liaison de sockets).
   c. L'écran **`Socket Binding Declarations`** apparaît. Au départ, le groupe **`standard-sockets`** apparaît. Choisissez un autre groupe en le sélectionnant à partir de la case de combo sur la droite.

   NOTE

   Si vous utilisez un serveur autonome, il ne possédera qu'un seul groupe de liaisons de socket.

   La liste de noms de sockets et des ports apparaît, avec huit valeurs par page. Vous pourrez naviguer entre les pages grâce à la flèche de navigation en dessous du tableau.
3. **Déterminer les ports que vous souhaitez ouvrir.**
   Suivant la fonction d'un port particulier, et suivant les besoins de votre environnement, certains ports devront sans doute être ouverts sur votre pare-feu.
4. **Configurer votre pare-feu pour rediriger le trafic réseau vers la plateforme JBoss EAP 6.**
   Procéder à ces étapes de configuration de votre pare-feu de réseau pour permettre au trafic de se diriger vers le port désiré.
   a. Connectez-vous au pare-feu de votre machine, et accéder à cette commande, en tant qu'utilisateur root.

b. Saisir la commande **system-config-firewall** pour lancer l'utilitaire de configuration du pare-feu. Un GUI ou Utilitaire de ligne de commande opérera selon la façon dont vous êtes connecté au système de pare-feu. Cette tâche assume que vous êtes connecté via SSH et que vous utilisez l'interface de ligne de commande.

c. Utiliser la clé **TAB** de votre clavier pour naviguer vers le bouton **Customize**, puis appuyer sur la clé **ENTER**. L'écran**Trusted Services** apparaîtra.

d. Ne changez aucune valeur, mais utilisez la clé **TAB** pour naviguer vers le bouton **Forward**, puis, appuyer sur **ENTER** pour aller vers le prochain écran. L'écran **Other Ports** apparaîtra.

e. Utiliser la clé **TAB** pour naviguer vers le bouton **<Add>**, puis appuyer sur la clé **ENTER**. L'écran **Port and Protocol** apparaîtra.

f. Saisir **5445** dans le champ **Port / Port Range**, puis utiliser la clé **TAB** pour vous rendre dans le champ **Protocol**, puis saisir **tcp**. Utiliser la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyer sur **ENTER**.

g. Utiliser la clé **TAB** pour naviguer vers le bouton **Forward** jusqu'à atteindre l'écran **Port Forwarding**.

h. Utiliser la clé **TAB** pour naviguer vers le bouton **<Add>**, puis appuyer sur la clé **ENTER**.

i. Remplir les valeurs suivantes pour définir la redirection de port à port **5445**.

- Interface source : **eth1**
- Protocole : **tcp**
- Port / Plage de port : **5445**
- Adresse IP de destination : **10.1.1.2**
- Port / Plage de port : **5445**

Utiliser la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyer sur la clé **ENTER**.

j. Utiliser la clé **TAB** pour naviguer vers le bouton **Close**, puis appuyer sur la clé **ENTER**.

k. Utiliser la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyer sur **ENTER**. Pour appliquer les changements, lire la notice d'avertissement, puis appuyer sur **Yes**.

5. **Configurer un pare-feu sur votre hôte de plateforme JBoss EAP 6.**
Certaines organisations choisissent de configurer un pare-feu sur le serveur JBoss EAP 6 lui-même et de fermer tous les ports qui ne sont pas utiles à son fonctionnement. Voir Section 11.10.3, « Ports de réseau utilisés par JBoss EAP 6 » pour déterminer quels ports ouvrir, puis fermer le reste. La configuration par défaut de Red Hat Enterprise Linux 6 ferme tous les ports sauf **22** (utilisé pour Secure Shell (SSH) et **5353** (utilisé pour la multi-diffusion DNS). Si vous configurez les ports, assurez-vous que vous avez un accès physique à votre serveur pour ne pas, par inadvertance, vous verrouiller vous-même.

Résultat

Votre pare-feu est configuré pour renvoyer le trafic vers votre serveur JBoss EAP 6 interne, de la façon dont vous l'avez spécifié dans la configuration de votre pare-feu. Si vous avez choisi d'activer un pare-

feu sur votre serveur JBoss Enterprise Application Platform 6, tous les ports seront fermés sauf ceux nécessaires à l'exécution de vos applications.

Procédure 11.34. Configurer le pare-feu dans Microsoft Windows avec PowerShell

1. Désactiver le pare-feu pour déterminer si le comportement de réseau actuel est lié à la configuration du pare-feu à des fins de débogage.

```
Start-Process "$psHome\powershell.exe" -Verb Runas -ArgumentList '-
command "NetSh Advfirewall set allprofiles state off"'
```

2. Autoriser les connexions UDP sur le port 23364. Par exemple :

```
Start-Process "$psHome\powershell.exe" -Verb Runas -ArgumentList '-
command "NetSh Advfirewall firewall add rule name="UDP Port 23364"
dir=in  action=allow protocol=UDP localport=23364"'
Start-Process "$psHome\powershell.exe" -Verb Runas -ArgumentList '-
command "NetSh Advfirewall firewall add rule name="UDP Port 23364"
dir=out action=allow protocol=UDP localport=23364"'
```

Procédure 11.35. Configurer le parefeu de Red Hat Enterprise Linux 7 afin d'activer mod_cluster advertising

- Pour activer mod_cluster advertising dans Red Hat Enterprise Linux 7, vous devrez activer le port UDP sur le parefeu comme suit :

```
firewall-cmd --permanent --zone=public --add-port=23364/udp
```

NOTE

224.0.1.105:23364 est l'adresse par défaut et le port de l'équilibreur de charges de mod_cluster advertising UDP multicast.

Rapporter un bogue

11.11. JAVA SECURITY MANAGER

11.11.1. Java Security Manager

Java Security Manager

Java Security Manager est une classe qui gère la limite extérieure de la sandbox Java Virtual Machine (JVM), contrôlant ainsi comment le code exécuté dans la JVM peut interagir avec les ressources extérieures à la machine virtuelle Java. Lorsque le gestionnaire de sécurité Java est activé, l'API Java vérifie l'authorisation avec le gestionnaire de sécurité avant d'exécuter une vaste gamme d'opérations potentiellement dangereuses.

Le Java Security Manager utilise une police de sécurité pour déterminer si une action sera permise ou refusée.

Rapporter un bogue

11.11.2. Exécuter JBoss EAP 6 dans le Java Security Manager

Pour spécifier une stratégie Java Security Manager, vous devez modifier les options Java transmises à l'instance de serveur ou de domaine lors du processus d'amorçage. Pour cette raison, vous ne pouvez passer les paramètres en option aux scripts **`domain.sh`** ou **`standalone.sh`**. La procédure suivante va vous guider à travers les étapes de configuration de votre instance pour exécuter au sein d'une stratégie Java Security Manager.

Pré-requis

- Avant de suivre cette procédure, vous devrez rédiger une stratégie de sécurité, en utilisant la commande **`policytool`** comprise dans votre Java Development Kit (JDK). Cette procédure assume que votre stratégie se trouve dans ***`EAP_HOME`*`/bin/server.policy`**. Sinon, vous pouvez écrire la stratégie de sécurité à l'aide d'un éditeur de texte et la sauvegarder comme ***`EAP_HOME`*`/bin/server.policy`**.
- Le domaine ou le serveur autonome doivent être tout à fait arrêtés avant d'éditer un fichier de configuration quelconque.

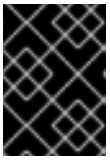
Procéder à la procédure suivante pour chaque hôte physique ou pour chaque instance de votre domaine, si vous avez des membres de domaine éparpillés dans des systèmes multiples.

Procédure 11.36. Configurer le gestionnaire de sécurité dans JBoss EAP 6

1. **Ouvrir le fichier de configuration.**
   Ouvrir le fichier de configuration pour le modifier. Ce fichier se trouve dans un de ces emplacements, suivant que vous utilisiez un domaine géré ou un serveur autonome. Il ne s'agit pas du fichier exécutable utilisé pour démarrer le serveur ou le domaine.
   - **Domaine géré**
     - Dans Linux : ***`EAP_HOME`*`/bin/domain.conf`**
     - Dans Windows : ***`EAP_HOME`*`\bin\domain.conf.bat`**
   - **Serveur autonome**
     - Dans Linux : ***`EAP_HOME`*`/bin/standalone.conf`**
     - Dans Windows : ***`EAP_HOME`*`\bin\standalone.conf.bat`**
2. **Ajouter les options Java au fichier.**
   Pour s'assurer que les options Java soient bien utilisées, ajouter les au bloc de code qui commence par :

   ```
   if [ "x$JAVA_OPTS" = "x" ]; then
   ```

   Vous pouvez modifier la valeur de **`-Djava.security.policy`** pour indiquer l'emplacement exact de votre police de sécurité. Cela ne doit prendre qu'une seule ligne, sans saut de ligne. Utiliser **`==`** quand vous définissez la propriété **`-Djava.security.policy`** indique que le gestionnaire de sécurité utilisera *uniquement* le fichier de police spécifié. Utiliser **`=`** indique que le gestionnaire de sécurité utilisera la police spécifiée *en combinaison à* la police définie dans la section **`policy.url`** de ***`JAVA_HOME`*`/lib/security/java.security`**.



IMPORTANT

Les versions de JBoss Enterprise Application Platform à partir de 6.2.2 requièrent que la propriété système **`jboss.modules.policy-permissions`** soit sur *true*.

Exemple 11.32. domain.conf

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
Djava.security.policy==$PWD/server.policy -
Djboss.home.dir=/path/to/EAP_HOME -Djboss.modules.policy-
permissions=true"
```

Exemple 11.33. domain.conf.bat

```
set "JAVA_OPTS=%JAVA_OPTS% -Djava.security.manager -
Djava.security.policy==\path\to\server.policy -
Djboss.home.dir=\path\to\EAP_HOME -Djboss.modules.policy-
permissions=true"
```

Exemple 11.34. standalone.conf

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
Djava.security.policy==$PWD/server.policy -
Djboss.home.dir=$JBOSS_HOME -Djboss.modules.policy-
permissions=true"
```

Exemple 11.35. standalone.conf.bat

```
set "JAVA_OPTS=%JAVA_OPTS% -Djava.security.manager -
Djava.security.policy==\path\to\server.policy -
Djboss.home.dir=%JBOSS_HOME% -Djboss.modules.policy-
permissions=true"
```

3. **Démarrer le domaine ou le serveur.**
   Démarrer le domaine ou le serveur en tant que normal.

Rapporter un bogue

11.11.3. Polices du Java Security Manager

Security Policy

Un jeu d'autorisations définies pour les différentes classes du code. Le Java Security Manager examine les requêtes en provenance des applications en fonction de la stratégie de sécurité. Si une action est autorisée par la stratégie, le Java Security

Manager permettra que l'action ait lieu. Si l'action n'est pas autorisée par la stratégie, le Java Security Manager refusera cette action. La stratégie de sécurité peut définir des autorisations basées sur l'emplacement du code ou sur les principaux du sujet.

Le Java Security Manager et la police de sécurité utilisés sont configurés à l'aide des options Java Virtual Machine **`java.security.manager`** et **`java.security.policy`**.

Informations de base

Une entrée de police de sécurité consiste en éléments de configuration suivants connectés au **`policytool`** :

CodeBase

L'emplacement de l'URL (à l'exclusion des informations sur l'hôte ou le domaine) d'où viennent les codes. Ce paramètre est en option.

SignedBy

L'alias est utilisé dans le fichier de clés pour référencer le signataire dont la clé privée a été utilisée pour signer le code. Cela peut être une valeur unique ou une liste séparée par des virgules. Ce paramètre est facultatif. Si omis, la présence ou l'absence de signature n'a aucun impact sur le gestionnaire de sécurité Java.

Principaux

Une liste de paires ***`principal_type`***/***`principal_name`***, qui doivent être présentes au sein de l'ensemble principal du thread en cours d'exécution. L'entrée de principaux est facultative. Si elle est omise, cela signifie que les principaux du thread en cours n'auront aucun impact sur le Java Security Manager.

Permissions

Une permission est l'accès qui est accordé au code. De nombreuses autorisations sont fournies dans le cadre de la spécification Java Enterprise Edition 6 (Java EE 6). Ce document couvre uniquement les autorisations supplémentaires qui sont fournies par JBoss EAP 6.

Rapporter un bogue

11.11.4. Écrire une stratégie pour le Java Security Manager

Introduction

Il y a une application nommée **`policytool`** dans la plupart des distributions JDK et JRE, ayant pour but la modification ou la création de polices de sécurité pour le Java Security Manager. Vous trouverez des informations sur **`policytool`** dans http://docs.oracle.com/javase/6/docs/technotes/tools/.

Procédure 11.37. Définir une stratégie pour le Java Security Manager

1. **Démarrez `policytool`.**
   Démarrez l'outil **`policytool`** d'une des façons suivantes.

   - **Red Hat Enterprise Linux**
     À partir de votre GUI ou invite de commande, exécutez **`/usr/bin/policytool`**.

   - **Microsoft Windows Server**

Exécutez **policytool.exe** à partir du menu de démarrage (Start) ou à partir de **bin** de votre installation Java. L'emplacement peut varier.

2. **Créer une stratégie.**
   Pour créer une stratégie, sélectionnez **Add Policy Entry**. Ajouter les paramètres dont vous aurez besoin, et cliquez sur **Done**.

3. **Modifier une stratégie existante**
   Sélectionnez la stratégie à partir d'une liste de stratégies existantes, et sélectionnez le bouton **Edit Policy Entry**. Modifiez les paramètres suivant les besoins.

4. **Supprimer une stratégie existante.**
   Sélectionnez la stratégie à partir d'une liste de stratégies existantes, et sélectionnez le bouton **Remove Policy Entry**.

Rapporter un bogue

11.11.5. Débogage des stratégies du gestionnaire de sécurité

Vous pouvez activer les informations de débogage pour vous aider à résoudre les problèmes liés aux stratégies de sécurité. L'option **java.security.debug** configure le niveau des informations liées à la sécurité qui ont été reportées. La commande **java -Djava.security.debug=help** vous produira de l'aide avec l'ensemble complet des options de débogage. Définir le niveau de débogage à **all** est utile quand on résout un échec lié à la sécurité dont la cause est inconnue, mais en général, cela produit trop d'informations. Une valeur par défaut utile et raisonnable est **access:failure**.

Procédure 11.38. Activer le débogage général

- **Cette procédure permettra un bon niveau de sécurité générale pour les informations de débogage liées à la sécurité.**
  Ajouter la ligne suivante au fichier de configuration du serveur.
  - Si l'instance de JBoss EAP 6 exécute sur un domaine géré, la ligne sera ajoutée au fichier **bin/domain.conf** de Linux ou au fichier **bin\domain.conf.bat** de Windows.
  - Si l'instance de JBoss EAP exécute sur un domaine autonome, la ligne sera ajoutée au fichier **bin/standalone.conf** de Linux ou au fichier **bin/standalone.conf.bat** de Windows.

Linux

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.debug=access:failure"
```

Windows

```
set "JAVA_OPTS=%JAVA_OPTS% -Djava.security.debug=access:failure"
```

Résultat

Un niveau général d'informations de débogage lié à la sécurité a été activé.

Rapporter un bogue

11.12. ENCODAGE SSL

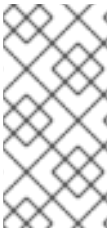
11.12.1. Implémentation du cryptage SSL pour le serveur de JBoss EAP 6.

Introduction

De nombreuses applications web requièrent une connexion cryptée-SSL entre les clients et le serveur, connue sous le nom de connexion **HTTPS**. Vous pouvez utiliser cette procédure pour activer **HTTPS** sur votre serveur ou groupe de serveurs.

Pré-requis

- Un ensemble de clés de cryptage SSL et un certificat de cryptage SSL. Vous pourrez vous les procurer par l'intermédiaire d'une autorité de signature de certificats. Pour générer les clés de cryptage par les utilitaires de Red Hat Enterprise Linux, voir Section 11.12.2, « Générer une clé de cryptage SSL et un certificat ».
- Informations utiles sur votre environnement et sur votre installation :
  - Le nom complet du répertoire où les fichiers de certificats sont stockés.
  - Le mot de passe de cryptage pour vos clés de cryptage.
- Exécuter l'interface CLI et le connecter à votre contrôleur de domaine ou à votre serveur autonome.



NOTE

Cette procédure utilise des commandes appropriées à la configuration de JBoss EAP 6, qui utilise un domaine géré. Si vous utilisez un domaine autonome, modifier les commandes de Management CLI en supprimant **`/profile=default`** du début d'une commande de Management CLI.

Procédure 11.39. Configurer le JBoss Web Server pour qu'il puisse utiliser HTTPS

1. **Ajoutez un nouveau connecteur HTTPS.**
   Exécutez la commande de Management CLI suivante, en changeant le profil comme il se doit. Cela va créer un nouveau connecteur sécurisé, nommé **HTTPS**, qui utilise le protocole **`https`**, la liaison de socket **`https`** (ayant comme valeur par défaut **8443**), et qui est définie pour être sécurisée.

   Exemple 11.36. Commande de Management CLI

   ```
   /profile=default/subsystem=web/connector=HTTPS/:add(socket-
   binding=https,scheme=https,protocol=HTTP/1.1,secure=true)
   ```

2. Éxécutez la commande de Management CLI suivante pour définir le protocole à **TLSv1**.

   Exemple 11.37. Commande de Management CLI

   ```
   /profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:
   write-attribute(name=protocol,value=TLSv1)
   ```

3. **Sélectionnez les suites de cryptage qui conviennent**

Il existe un certain nombre de primitives cryptographiques disponibles, utilisées comme blocs de base pour former des suites de chiffrement. Le premier tableau répertorie les primitives cryptographiques recommandées. Le deuxième énumère les primitives cryptographiques qui, bien qu'elles puissent être utilisées pour assurer la compatibilité avec les logiciels existants, ne sont pas considérées comme aussi sûres que celles qui sont recommandées.

AVERTISSEMENT

Red Hat recommande de faire une liste blanche sélective d'un ensemble d'algorithmes de chiffrage puissants à utiliser pour la `cipher-suite`. L'activation de systèmes de chiffrage faibles est un risque important de sécurité. Consultez la documentation du fournisseur de votre JDK avant de statuer sur les suites de chiffrement particulières à utiliser, car il peut y avoir des problèmes de compatibilité.

Tableau 11.9. Primitives cryptographiques recommandées

RSA avec clés 2048 bit et OAEP
AES-128 en mode CBC
SHA-256
HMAC-SHA-256
HMAC-SHA-1

Tableau 11.10. Autres primitives cryptographiques

RSA avec des clés qui dépassent 1024 et taille de remplissage héritée
AES-192
AES-256
3DES (triple DES, avec deux ou trois clés de 56 bit)
RC4 (fortement déconseillé)
SHA-1
HMAC-MD5

Pour obtenir une liste complète des paramètres que vous pouvez définir pour les propriétés SSL du connecteur, voir Section 11.12.3, « Référence de connecteur SSL ».

4. **Configurer le certificat de cryptage SSL et les clés.**
   Exécutez la commande CLI suivante pour configurer votre certificat SSL, en remplaçant vos propres valeurs par celles de l'exemple. Cet exemple suppose que le keystore est copié dans le répertoire de configuration du serveur, qui est ***EAP_HOME*/domain/configuration/** pour un domaine géré.

   Exemple 11.38. Commande de Management CLI

   ```
   /profile=default/subsystem=web/connector=HTTPS/ssl=configuration:a
   dd(name=https,certificate-key-
   file="${jboss.server.config.dir}/keystore.jks",password=SECRET,
   key-alias=KEY_ALIAS, cipher-suite=CIPHERS)
   ```

5. **Déployer une application.**
   Déployez une application dans un groupe de serveurs qui utilise le profil que vous avez configuré. Si vous utilisez un serveur autonome, déployer une application sur votre serveur. Les demandes HTTPS en sa direction utilisent la nouvelle connexion cryptée-SSL.

Rapporter un bogue

11.12.2. Générer une clé de cryptage SSL et un certificat

Pour utiliser une connexion chiffrée SSL HTTP (HTTPS), ainsi que d'autres types de communication cryptée-SSL, vous avez besoin d'un certificat de chiffrement signé. Vous pouvez acheter un certificat d'une autorité de certification (AC), ou vous pouvez utiliser un certificat auto-signé. Les certificats auto-signés ne sont pas considérés dignes de confiance par de nombreux tiers, mais conviennent à des fins de test internes.

Cette procédure vous permet de créer un certificat auto-signé lié à des utilitaires disponibles dans Red Hat Enterprise Linux.

Conditions préalables

- La commande **keytool** doit être disponible. Elle est fournie par le Java Development Kit. Dans Red Hat Enterprise Linux, OpenJDK installe cette commande à l'emplacement suivant **/usr/bin/keytool**.
- Comprendre la syntaxe et les paramètres de la commande **keytool**. Cette procédure utilise des instructions extrêmement génériques, car des discussions plus sophistiquées sur les spécificités des certificats SSL ou sur la commande **keytool** sont hors de portée de cette documentation.

Procédure 11.40. Générer une clé de cryptage SSL et un certificat

1. **Générer un keystore avec des clés privées et des clés publiques.**
   Exécuter la commande suivante pour générer un keystore nommé **server.keystore** ayant comme alias **jboss** dans votre répertoire actuel.

   ```
   keytool -genkeypair -alias jboss -keyalg RSA -keystore
   server.keystore -storepass mykeystorepass --dname
   "CN=jsmith,OU=Engineering,O=mycompany.com,L=Raleigh,S=NC,C=US"
   ```

   Le tableau suivant décrit les paramètres utilisés avec la commande « keytool ».

Paramètre	Description
-genkeypair	La commande **`keytool`** qui génère une paire de clés contenant une clé publique et une clé privée.
-alias	L'alias est pour le keystore. Cette valeur est arbitraire, mais l'alias **`jboss`** est la valeur par défaut utilisée par le serveur JBoss Web.
-keyalg	L'algorithme de création de paires de clés. Dans ce cas, c'est **RSA**.
-keystore	Le nom et l'emplacement du fichier keystore. L'emplacement par défaut est le répertoire en cours. Le nom que vous choisissez est arbitraire. Dans ce cas, il s'agit du fichier nommé **`server.keystore`**.
-storepass	Ce mot de passe est utilisé pour authentifier le keystore, et pour que la clé puisse être lue. Le mot de passe doit contenir au moins 6 caractères de long et doit être fourni quand on accède au keystore. Dans un tel cas, on utilise **`mykeystorepass`**. Si vous omettez ce paramètre, on vous demandera de le saisir quand vous exécuterez la commande.
-keypass	Il s'agit du mot de passe pour la clé.  **NOTE** À cause d'une limitation d'implémentation, il doit correspondre à celui du mot de passe du store.

Paramètre	Description
--dname	Une chaîne avec des guillemets qui décrit le nom distinct de la clé, comme par exemple : "CN=jsmith,OU=Engineering,O=mycompany.com,L=Raleigh,C=US". Cette chaîne est une compilation des composants suivants : • **CN** - Le nom commun ou le nom d'hôte. Si le nom d'hôte est "jsmith.mycompany.com", le **CN** sera "jsmith". • **OU** - L'unité organisationnelle, par exemple "Engineering" • **O** - Le nom de l'organisation, par exemple "mycompany.com". • **L** - La localité, par exemple "Raleigh" ou "London" • **S** - L'état ou la province, par exemple "NC". Ce paramètres est optionnel. • **C** - Les 2 lettres d'un code pays, par exemple "US" ou "UK".

Quand vous exécuterez la commande ci-dessus, on vous demandera les informations suivantes :

- Si vous n'utilisiez pas le paramètre ***-storepass*** sur la ligne de commande, on vous demandera de saisir le mot de passe du keystore. Saisir le nouveau mot de passe à la seconde invite.
- Si vous n'utilisiez pas le paramètre **-keypass** sur la ligne de commande, on vous demandera de saisir le mot de passe de la clé. Appuyez sur **Enter** pour le définir à la même valeur que celle du mot de passe du keystore.

Quand la commande s'achèvera, le fichier **server.keystore** contiendra la clé unique avec l'alias **jboss**.

2. **Vérifier la clé.**
   Vérifier que la clé fonctionne correctement en utilisant la commande suivante.

```
keytool -list -keystore server.keystore
```

   On vous demande le mot de passe du keystore. Les contenus du keystore sont affichés (dans ce cas, il s'agit d'une simple clé nommée **jboss**). Notez le type de la clé **jboss**, qui est **keyEntry**. Cela indique que le keystore contient à la fois une entrée publique et une entrée privée pour cette clé.

3. **Créer une demande de signature de certificat.**
   Exécutez la commande suivante pour générer une demande de signature de certificat en utilisant la clé publique du keystore que vous avez créée dans la 1ère étape.

```
keytool -certreq -keyalg RSA -alias jboss -keystore server.keystore
-file certreq.csr
```

On vous demandera le mot de passe pour pouvoir authentifier le keystore. La commande **keytool** crée alors une nouvelle demande de signature de certificat nommée **certreq.csr** dans le répertoire en cours d'utilisation.

4. **Tester la demande de signature de certificat nouvellement générée.**
Tester les contenus du certificat avec la commande suivante :

```
openssl req -in certreq.csr -noout -text
```

Les détails du certificat apparaissent.

5. **En option : soumettre votre demande de signature de certificat à une autorité de certification (AC).**
Une Autorité de Certification (AC) authentifie votre certificat pour qu'il soit considéré de confiance par des clients de tierce partie. L'AC vous a produit un certificat signé, et en option, vous a peut être fourni un ou plusieurs certificats intermédiaires.

6. **Option : exporter un certificat auto-signé du keystore.**
Si vous n'en avez besoin que dans un but de test ou en interne, vous pourrez utiliser un certificat auto-signé. Vous pourrez en exporter un, créé dans la première étape, en provenance du keystore, comme suit :

```
keytool -export -alias jboss -keystore server.keystore -file
server.crt
```

On vous demande un mot de passe pour pouvoir s'authentifier au keystore. Un certificat auto-signé, intitulé **server.crt**, a été créé dans le répertoire en cours d'utilisation.

7. **Importer le certificat signé avec tout certificat intermédiaire.**
Importer chaque certificat, dans l'ordre dans lequel l'AC vous le demande. Pour chaque AC que vous importez, remplacer **intermediate.ca** ou **server.crt** par le nom du fichier. Si vos certificats ne sont pas fournis dans des fichiers séparés, créer un fichier séparé pour chaque certificat, et coller leur contenu dans le fichier.

NOTE

Votre certificat signé et les clés de ce certificat sont des ressources de valeur. Soyez vigilant sur la façon dont vous les transportez entre les serveurs.

```
keytool -import -keystore server.keystore -alias intermediateCA -
file intermediate.ca
```

```
keytool -importcert -alias jboss -keystore server.keystore -file
server.crt
```

8. **Testez que vos certificats soient bien importés avec succès.**
Exécuter la commande suivante, et saisir le mot de passe de keystore quand on vous le demandera. Les contenus de votre keystore sont affichés, et les certificats font partie de la liste.

```
keytool -list -keystore server.keystore
```

Résultat

Votre certificat signé est maintenant inclus dans votre keystore et est prêt à l'utilisation pour crypter les connexions SSL, y compris les communications au serveur web HTTPS.

Rapporter un bogue

11.12.3. Référence de connecteur SSL

Les connecteurs JBoss Web peuvent inclure les attributs de configuration SSL suivants. Les commandes CLI fournies sont conçues pour un domaine géré à l'aide du profil **`par défaut`**. Changer le nom du profil à celui que vous souhaitez configurer, pour un domaine géré, ou omettre la portion **`/profile=default`** de la commande, pour un serveur autonome.

Tableau 11.11. Attributs de connecteurs SSL

Attribut	Description	Commande CLI
name	Le nom d'affichage du connecteur SSL	L'attribute **`name`** est en lecture seule.

Attribut	Description	Commande CLI

verify-client	Les valeurs possibles de **`verify-client`** diffèrent selon qu'il s'agisse d'un connecteur HTTP/HTTPS ou d'un connecteur natif APR qui est utilisé.
	
	Connecteur HTTP/HTTPS
	
	Les valeurs possibles sont **`true`**, **`false`**, ou **`want`**. Définir à **`true`** pour obtenir une chaîne de certificat valide de la part d'un client avant d'accepter une connexion. Définir à **`want`** si vous voulez que la pile SSL demande un certificat de client, mais ne pas l'échouer si celui-ci n'est pas présenté. Définir à **`false`** (la valeur par défaut) si vous ne souhaitez pas demander de chaîne de certificat, à moins que le client ne demande une ressource protégée par une contrainte de sécurité qui utilise l'authentification de **`CLIENT-CERT`**.
	
	Connecteur APR natif
	
	Les valeurs possibles sont **`optional`**, **`require`**, **`optionalNoCA`**, et **`none`** (ou tout autre string, ce qui aura le même effet que **`none`**). Ces valeurs déterminent si un certificat est optionnel, requis, ou optionnel sans Autorité de certification, ou encore, non repuis. La valeur par défaut est **`none`**, ce qui signifie que le client n'a pas la possibilité de soumettre un certificat.
		La commande du premier exemple utilise le connecteur HTTPS.
		
		`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=verify-client,value=want)`
		
		La commande du second exemple utilise le connecteur APR.
		
		`/profile=default/subsystem=web/connector=APR/ssl=configuration/:write-attribute(name=verify-client,value=require)`

verify-depth	Le nombre maximal d'émetteurs de certificats intermédiaires vérifiés avant de décider que les clients n'ont pas de certificat valide. La valeur par défaut est **`10`**.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=verify-depth,value=10)`

Attribut	Description	Commande CLI
certificate-key-file	Le chemin d'accès complet et nom de fichier du keystore où le certificat de serveur signé est stocké. Avec le cryptage JSSE, ce fichier de certificat sera le seul, tandis que OpenSSL utilise plusieurs fichiers. La valeur par défaut est le fichier **`.keystore`** dans le répertoire home de l'utilisateur exécutant JBoss EAP 6. Si votre **`keystoreType`** n'utilise pas de fichier, définissez le paramètre en chaîne vide.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-key-file,value=../domain/configuration/server.keystore)`
certificate-file	Si vous utilisez un cryptage OpenSSL, définir la valeur de ce paramètre au chemin d'accès du fichier qui contient le certificat de serveur.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-file,value=server.crt)`
password	Le mot de passe pour le trustore et le keystore à la fois. Dans l'exemple suivant, remplacer *PASSWORD* par votre propre mot de passe.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=password,value=PASSWORD)`
protocol	Version du protocole SSL à utiliser. Les valeurs prises en charge dépendent de l'implémentation sous-jacente (JSSE ou OpenSSL). Consulter Java SSE Documentation.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=protocol,value=ALL)`

Attribut	Description	Commande CLI
cipher-suite	Une liste des cryptages autorisés. Pour la syntaxe JSSE, cela devra corrrespondre à une liste séparée par des virgules. Pour la syntaxe OpenSSL, cela devra correspondre à une liste séparée par deux-points. La valeur par défaut est **`HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5`**. L'exemple ne mentionne que deux cryptages possibles, mais les exemples de la vie courante en utilisent d'autres.  **IMPORTANT** L'utilisation de cryptages faibles posent des risques de sécurité importants. Voir http://www.nist.gov/manuscript-publication-search.cfm?pub_id=915295 pour obtenir des informations NIST sur les suites de cryptage. Pour obtenir une liste de cryptage OpenSSL, consulter https://www.openssl.org/docs/apps/ciphers.html#CIPHER_STRINGS. Notez que ce qui suit n'est pas pris en charge : **`@SECLEVEL`**, **`SUITEB128`**, **`SUITEB128ONLY`**, **`SUITEB192`**.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=cipher-suite, value="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA")`
key-alias	L'alias utilisé pour le certificat de serveur dans le keystore. Dans l'exemple suivant, remplacer *KEY_ALIAS* par l'alias de votre certificat.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=key-alias,value=KEY_ALIAS)`

Attribut	Description	Commande CLI
truststore-type	Le type de truststore. Il existe différents types de keystores, dont **PKCS12** et le **JKS** standard de Java.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=truststore-type,value=jks)`
keystore-type	Le type de keystore. Il existe différents types de keystores, dont **PKCS12** et le **JKS** standard de Java.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=keystore-type,value=jks)`
ca-certificate-file	Le fichier contenant les certificats CA. Il s'agit du **`truststoreFile`**, dans le cas de JSSE, et il utilise le même mot de passe que le keystore. Le fichier **`ca-certificate-file`** est utilisé pour valider les certificats de clients.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-file,value=ca.crt)`
ca-certificate-password	Le mot de passe de certificat pour le **`ca-certificate-file`**. Dans l'exemple ci-dessous, remplacer *MASKED_PASSWORD* par votre propre mot de passe masqué.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=ca-certificate-password,value=`*`MASKED_PASSWORD`*`)`
ca-revocation-url	Un fichier ou URL qui contient la liste de révocations. Se réfère au **`crlFile`** pour JSSE ou au **`SSLCARevocationFile`** pour SSL.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=ca-revocation-url,value=ca.crl)`

Attribut	Description	Commande CLI
session-cache-size	La taille du cache SSLSession. Cet attribut s'applique uniquement aux connecteurs JSSE. La valeur par défaut est **0**, qui indique une taille cache illimitée.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=session-cache-size,value=100)`
session-timeout	Le nombre de secondes avant qu'une SSLSession n'expire. Cet attribut s'applique uniquement aux connecteurs JSSE. La valeur par défaut est **86400** secondes, ce qui correspond à 24 heures.	`/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=session-timeout,value=43200)`

Rapporter un bogue

11.13. L'ARCHIVAGE SÉCURISÉ DES MOTS DE PASSE POUR LES STRINGS DE NATURE CONFIDENTIELLE

11.13.1. Sécurisation des chaînes confidentielles de fichiers en texte clair

Les applications web et autres déploiements incluent souvent des fichiers en texte clair, comme des descripteurs de déploiement XML, qui comprennent des informations sensibles telles que les mots de passe et autres strings sensibles. JBoss EAP 6 inclut un mécanisme d'archivage sécurisé de mots de passe qui vous permet de crypter les strings sensibles et de les stocker dans un keystore chiffré. Le mécanisme d'archivage sécurisé parvient à décrypter les strings à utiliser dans les domaines de sécurité, ou autres systèmes de vérification. Ceci fournit une couche supplémentaire de sécurité. Le mécanisme s'appuie sur les outils qui sont inclus dans toutes les implémentations de Java Development Kit (JDK) prises en charge.

AVERTISSEMENT

Des problèmes ont été décelés lors de l'utilisation de la fonctionnalité de sécurité d'archivage sécurisé avec JBoss EAP 6. Il a été constaté que le **`vault.keystore`** généré par le **`keytool`** n'est pas un fichier de clés valide lorsqu'il est utilisé avec un JDK IBM. Cela est dû au fait que les implémentations de keystore JCEKS diffèrent selon les vendeurs de Java.

Le problème se présente lorsqu'un fichier de clés généré par Oracle Java est utilisé dans une instance de JBoss EAP sur une installation Java d'IBM. Dans ces cas, le serveur ne démarre pas et lève l'exception suivante :

```
java.io.IOException:
com.sun.crypto.provider.SealedObjectForKeyProtector
```

À l'heure actuelle, la seule solution qui existe consiste à éviter toute tentative d'utilisation d'un fichier de clés généré avec un keytool Oracle dans un environnement utilisant une implémentation de Java d'IBM.

Rapporter un bogue

11.13.2. Créer un keystore Java pour stocker des strings sensibles

Pré-requis

- La commande **`keytool`** doit être disponible. Elle est fournie par le Java Runtime Environment (JRE). Chercher le chemin du fichier. Se trouve à l'emplacement suivant **`/usr/bin/keytool`** dans Red Hat Enterprise Linux.

Procédure 11.41. Installation du Java Keystore

1. **Créer un répertoire pour stocker votre keystore et autres informations cryptées.**
   Créez un répertoire qui contiendra votre keystore et autres informations pertinentes. Le reste de cette procédure assume que le répertoire est ***`EAP_HOME`*`/vault/`**. Comme le répertoire devra contenir des informations sensibles, il devra être accessible à un nombre restraint d'utilisateurs. Au minimum, le compte d'utilisateur sous lequel JBoss EAP exécute requiert un accès en lecture-écriture.

2. **Déterminer les paramètres à utiliser avec `keytool`.**
   Déterminer les paramètres suivants :

   alias

   L'alias est un identificateur unique pour l'archivage sécurisé ou autres données stockées dans le keystore. L'alias dans l'exemple de commande à la fin de cette procédure est **`vault`** (archivage sécurisé). Les alias sont insensibles à la casse.

   keyalg

L'algorithme à utiliser pour le cryptage. Dans cette procédure, l'exemple utilise **RSA**. Consultez la documentation de votre JRE et de votre système d'exploitation pour étudier vos possibilités.

keysize

La taille d'une clé de cryptage impacte sur la difficulté de décrypter au seul moyen de la force brutale. Dans cette procédure, l'exemple utilise **`1024`**. Pour plus d'informations sur les valeurs appropriées, voir la documentation distribuée avec **`keytool`**.

keystore

Le keystore est une base de données qui contient des informations chiffrées et des informations sur la façon de déchiffrer. Si vous ne spécifiez pas de keystore, le keystore par défaut à utiliser est un fichier appelé **`.keystore`** dans votre répertoire personnel. La première fois que vous ajoutez des données dans un keystore, il sera créé. L'exemple de cette procédure utilise le keystore **`vault.keystore`**.

La commande du **`keytool`** a plusieurs options. Consultez la documentation de votre JRE ou de votre système d'exploitation pour obtenir plus d'informations.

3. **Détermine les réponses aux questions que la commande `keystore` vous demandera.**
Le **`keystore`** a besoin des informations suivantes pour remplir l'entrée du keytore :

Mot de passe du keystore

Lorsque vous créez un keystore, vous devez définir un mot de passe. Pour pouvoir travailler dans keystore dans le futur, vous devez fournir le mot de passe. Créer un mot de passe dont vous vous souviendrez. Le keystore est sécurisé par son mot de passe et par la sécurité du système d'exploitation et du système de fichiers où il se trouve.

Mot de passe clé (en option)

En plus du mot de passe du keystore, vous pouvez indiquer un mot de passe pour chaque clé contenue. Pour utiliser une clé, le mot de passe doit être donné à chaque utilisation. Normalement, cette fonction n'est pas utilisée.

Prénom et nom de famille

Cela, ainsi que le reste de l'information dans la liste, aide à identifier la clé de façon unique et à la placer dans une hiérarchie par rapport aux autres clés. Il ne doit pas nécessairement correspondre à un nom, mais doit être composé de deux mots et doit être unique à une clé. L'exemple dans cette procédure utilise **`Admninistrateur Comptabilité`**. En terme de répertoires, cela devient le *nom commun* du certificat.

Unité organisationnelle

Il s'agit d'un mot unique d'identification qui utilise le certificat. Il se peut que ce soit l'application ou l'unité commerciale. L'exemple de cette procédure utilise **`AccountingServices`**. Normalement, tous les keystores utilisés par un groupe ou une application utilisent la même unité organisationnelle.

Organisation

Il s'agit normalement d'une représentation de votre nom d'organisation en un seul mot. Demeure constant à travers tous les certificats qui sont utilisés par une organisation. Cet exemple utilise **`MyOrganization`**.

Ville ou municipalité

Votre ville.

État ou province

Votre état ou province, ou l'équivalent pour votre localité.

Pays

Le code pays en deux lettres.

Ces informations vont créer ensemble une hiérarchie de vos keystores et certificats, qui garantira qu'ils utilisent une structure de nommage consistante, et unique.

4. **Exécuter la commande `keytool`, en fournissant les informations que vous avez collectées.**

Exemple 11.39. Exemple d'entrée et de sortie de la commande `keystore`

```
$ keytool -genseckey -alias vault -storetype jceks -keyalg AES -
keysize 128 -storepass vault22 -keypass vault22 -validity 730 -
keystore EAP_HOME/vault/vault.keystore
Enter keystore password: vault22
Re-enter new password:vault22
What is your first and last name?
  [Unknown]:  Accounting Administrator
What is the name of your organizational unit?
  [Unknown]:  AccountingServices
What is the name of your organization?
  [Unknown]:  MyOrganization
What is the name of your City or Locality?
  [Unknown]:  Raleigh
What is the name of your State or Province?
  [Unknown]:  NC
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Accounting Administrator, OU=AccountingServices,
O=MyOrganization, L=Raleigh, ST=NC, C=US correct?
  [no]:  yes

Enter key password for <vault>
        (RETURN if same as keystore password):
```

Résultat

Un fichier nommé **`vault.keystore`** est créé dans le répertoire ***`EAP_HOME`*`/vault/`**. Il stocke une clé simple, nommée **`vault`**, qui sera utilisée pour stocker des strings cryptés, comme des mots de passe, pour la plateforme JBoss EAP 6.

Rapporter un bogue

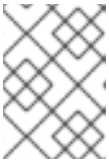
11.13.3. Masquer le mot de passe du keystore et initialiser le mot de passe de l'archivage de sécurité

Pré-requis

- Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles »

1. **Exécuter la commande `vault.sh`.**
   Exécuter ***`EAP_HOME`*`/bin/vault.sh`**. Démarrer une nouvelle session interactive en tapant **`0`**.

2. **Saisir le nom du répertoire où les fichiers cryptés seront stockés.**
   Ce répertoire doit être accessible à des utilisateurs limités uniquement. Au minimum, le compte d'utilisateur sous lequel JBoss EAP exécute requiert un accès lecture-écriture. Si vous avez suivi Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles », votre keystore se trouvera dans le répertoire nommé ***`EAP_HOME`*`/vault/`**.

   NOTE

   N'oubliez pas d'inclure la barre oblique finale dans le nom du répertoire. Soit `/` ou `\`, selon votre système d'exploitation.

3. **Saisir le nom de votre keystore.**
   Saisir le nom complet vers le fichier de keystore. Cet exemple utilise ***`EAP_HOME`*`/vault/vault.keystore`**.

4. **Crypter le mot de passe du keystore.**
   Les étapes suivantes vous servent à crypter le mot de passe du keystore, afin que vous puissiez l'utiliser dans les applications et les fichiers de configuration en toute sécurité.

   a. **Saisir le mot de passe du keystore.**
      Quand vous y serez invité, saisir le mot de passe du keystore.

   b. **Saisir une valeur salt.**
      Entrez une valeur salt de 8 caractères. La valeur salt, ainsi que le nombre d'itérations (ci-dessous), sont utilisés pour créer la valeur de hachage

   c. **Saisir le nombre d'itérations.**
      Saisir un nombre pour le nombre d'itérations.

   d. **Notez les informations de mot de passe masqué.**
      Le mot de passe masqué, salt, et le nombre d'itérations sont imprimés en sortie standard. Prenez-en note dans un endroit sûr. Un attaquant pourrait les utiliser pour déchiffrer le mot de passe.

   e. **Saisir un alias pour l'archivage de sécurité.**
      Quand on vous y invite, saisir un alias pour l'archivage de sécurité. Si vous suivez Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles » pour créer votre archivage de sécurité, l'alias sera **`vault`**.

5. **Sortir de la console interactive.**
   Saisir **2** pour sortir de la console interactive.

Résultat

Votre mot de passe de keystore est masqué afin de pouvoir être utilisé dans les fichiers de configuration et de déploiement. De plus, votre archivage de sécurité est complètement configuré et prêt à l'utilisation.

Rapporter un bogue

11.13.4. Configurer JBoss EAP pour qu'il utilise l'archivage sécurisé des mots de passe

Aperçu

Avant de masquer les mots de passe et d'autres attributs sensibles dans les fichiers de configuration, vous devez sensibiliser JBoss EAP 6 à l'archivage sécurisé des mots de passe qui les stocke et les déchiffre. Suivez cette procédure pour activer cette fonctionnalité.

Pré-requis

- Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles »
- Section 11.13.3, « Masquer le mot de passe du keystore et initialiser le mot de passe de l'archivage de sécurité »

Procédure 11.42. Assigner un mot de passe d'archivage sécurisé.

1. **Déterminer les valeurs qui conviennent pour la commande.**
   Déterminer les valeurs pour les paramètres suivants, qui sont déterminés par les commandes utilisées pour créer le keystore lui-même. Pour obtenir des informations sur la façon de créer un keystore, voir les sujets suivants : Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles » et Section 11.13.3, « Masquer le mot de passe du keystore et initialiser le mot de passe de l'archivage de sécurité ».

Paramètre	Description
KEYSTORE_URL	Le chemin d'accès ou URI du fichier keystore, qui s'appelle normalement `vault.keystore`
KEYSTORE_PASSWORD	Le mot de passe utilisé pour accéder au keystore. Cette valeur devrait être masquée.
KEYSTORE_ALIAS	Le nom du keystore.
SALT	Le salt utilisé pour crypter et décrypter les valeurs de keystore.
ITERATION_COUNT	Le nombre de fois que l'algorithme de chiffrement est exécuté.
ENC_FILE_DIR	Le chemin d'accès au répertoire à partir duquel les commandes de keystore sont exécutées. Normalement, le répertoire contient les mots de passe sécurisés.
hôte (domaine géré uniquement)	Le nom de l'hôte que vous configurez

2. **Utilisez l'interface CLI pour activer les mots de passe sécurisés.**
   Exécutez une des commandes suivantes, selon que vous utilisez un domaine géré ou une configuration de serveur autonome. Substituez les valeurs de la commande par celles de la première étape de cette procédure.

NOTE

Si vous utilisez Microsoft Windows Server, remplacez chaque caractère `\` du chemin d'accès du répertoire par un caractère `\` supplémentaire dans la commande CLI. Par exemple, **`C:\\data\\vault\\vault.keystore`**. C'est parce qu'un simple caractère `\` est utilisé comme caractère d'échappement.

- **Domaine géré**

```
/host=YOUR_HOST/core-service=vault:add(vault-options=
[("KEYSTORE_URL" => "PATH_TO_KEYSTORE"), ("KEYSTORE_PASSWORD" =>
"MASKED_PASSWORD"), ("KEYSTORE_ALIAS" => "ALIAS"), ("SALT" =>
"SALT"),("ITERATION_COUNT" => "ITERATION_COUNT"), ("ENC_FILE_DIR"
=> "ENC_FILE_DIR")])
```

- **Serveur autonome**

```
/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"PATH_TO_KEYSTORE"), ("KEYSTORE_PASSWORD" => "MASKED_PASSWORD"),
("KEYSTORE_ALIAS" => "ALIAS"), ("SALT" => "SALT"),
("ITERATION_COUNT" => "ITERATION_COUNT"), ("ENC_FILE_DIR" =>
"ENC_FILE_DIR")])
```

Ce qui suit est un exemple de la commande avec des valeurs hypothétiques :

```
/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"/home/user/vault/vault.keystore"), ("KEYSTORE_PASSWORD" => "MASK-
3y28rCZlcKR"), ("KEYSTORE_ALIAS" => "vault"), ("SALT" =>
"12438567"),("ITERATION_COUNT" => "50"), ("ENC_FILE_DIR" =>
"/home/user/vault/")])
```

Résultat

JBoss EAP 6 est configuré pour décrypter les strings masqués par l'intermédiaire de l'archivage sécurisé de mots de passe. Pour ajouter des strings à l'archivage sécurisé et les utiliser dans votre configuration, voir la section suivante : Section 11.13.6, « Stocker et résoudre les strings sensibles cryptés du keystore Java. ».

Rapporter un bogue

11.13.5. Configurer JBoss EAP pour qu'il utilise une implémentation d'archivage sécurisé personnalisée

Résumé

Vous pouvez utiliser votre propre implémentation de **`SecurityVault`** pour masquer les mots de passe, et les autres attributs sensibles dans les fichiers de configuration.

Procédure 11.43. Utilisez une implémentation sécurisée de l'archivage des mots de passe

1. Créez une classe qui implémente l'interface **`SecurityVault`**.
2. Créez un module contenant la classe de l'étape précédente, et spécifiez une dépendance sur **`org.picketbox`** où l'interface est **`SecurityVault`**.

3. Activez l'archivage de mots de passe sécurisée dans la configuration du serveur JBoss EAP en ajoutant l'élément d'archivage à l'aide des attributs suivants :

   code

   La nom complet de la classe qui implémente **`SecurityVault`**.

   module

   Le nom du module qui contient la classe personnalisée.

   Optionnellement, vous pouvez utiliser les paramètres **`vault-options`** pour initialiser la classe personnalisée d'un archivage de mots de passe. Par exemple :

```
/core-
service=vault:add(code="custom.vault.implementation.CustomSecurityVa
ult", module="custom.vault.module", vault-options=[("KEYSTORE_URL"
=> "PATH_TO_KEYSTORE"), ("KEYSTORE_PASSWORD" => "MASKED_PASSWORD"),
("KEYSTORE_ALIAS" => "ALIAS"), ("SALT" => "SALT"),("ITERATION_COUNT"
=> "ITERATION_COUNT"), ("ENC_FILE_DIR" => "ENC_FILE_DIR")])
```

Résultat

JBoss EAP 6 est configuré pour décrypter les chaînes masquées par l'intermédiaire d'une implémentation personnalisée de l'archivage des mots de passe.

Rapporter un bogue

11.13.6. Stocker et résoudre les strings sensibles cryptés du keystore Java.

Résumé

En comptant les mots de passe et les autres strings sensibles, les fichiers de configuration en texte brut ne sont pas sécurisés. JBoss EAP 6 inclut la capacité à stocker et à utiliser les valeurs masquées dans les fichiers de configuration, et à utiliser ces valeurs masquées dans les fichiers de configuration.

Pré-requis

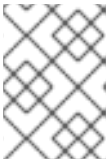
- Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles »
- Section 11.13.3, « Masquer le mot de passe du keystore et initialiser le mot de passe de l'archivage de sécurité »
- Section 11.13.4, « Configurer JBoss EAP pour qu'il utilise l'archivage sécurisé des mots de passe »
- L'application ***`EAP_HOME`*`/bin/vault.sh`** doit pouvoir être accessible via l'interface de ligne de commande.

Procédure 11.44. Installation du Java keystore

1. **Exécutez la commande `vault.sh`.**
   Exécutez ***`EAP_HOME`*`/bin/vault.sh`**. Démarrez une nouvelle session interactive en tapant **`0`**.

2. **Saisir le nom du répertoire où les fichiers cryptés seront stockés.**
   Si vous suivez Section 11.13.2, « Créer un keystore Java pour stocker des strings sensibles »,

votre keystore sera dans un répertoire nommé **vault/** de votre répertoire de base. Dans la plupart des cas, il est logique de stocker toutes vos informations cryptées au même endroit dans le keystore. Cet exemple utilise le répertoire **/home/*USER*/vault/**.



NOTE

N'oubliez pas d'inclure la barre oblique finale dans le nom du répertoire. Soit **/** ou **** selon votre système d'exploitation.

3. **Saisissez le nom de votre keystore.**
   Saisissez le nom complet vers le fichier de keystore. Cet exemple utilise **/home/*USER*/vault/vault.keystore**.

4. **Saisissez le mot de passe du keystore, le nom de l'archivage sécurisé, salt, et le nombre d'itérations.**
   Quand vous y êtes invité, saisissez le mot de passe du keystore, le nom de l'archivage sécurisé, salt, et le nombre d'itérations.

5. **Sélectionnez l'option de stockage d'un mot de passe.**
   Sélectionnez l'option **0** de stockage d'un mot de passe ou autre string sensible.

6. **Saisissez la valeur.**
   Une fois que vous y êtes invité, saisissez la valeur à deux reprises. Si les valeurs ne correspondent pas, vous serez invité à essayer à nouveau.

7. **Saisissez le bloc d'archivage sécurisé.**
   Saisissez le bloc d'archivage sécurisé, qui correspond à un conteneur pour les attributs ayant trait à la même ressource. Un exemple de nom d'attribut pourrait être **ds_ExampleDS**. Cela fera partie de la référence à la chaîne cryptée, dans votre source de données ou autre définition de service.

8. **Saissez le nom de l'attribut.**
   Saisissez le nom de l'attribut que vous stockez. Exemple de nom d'attribut **password**.

   Résultat

   Un message comme celui qui suit montre que l'attribut a été sauvegardé.

   ```
   La valeur d'attribut sécurisé est stockée dans l'archivage sécurisé.
   ```

9. **Notez les informations pour ce string crypté.**
   Un message s'affiche sur la sortie standard, montrant le bloc d'archivage sécurisé, le nom de l'attribut, la clé partagée et des conseils sur l'utilisation du string dans votre configuration. Prendre note de ces informations dans un emplacement sécurisé. Voici un exemple de sortie.

   ```
   ************************************************
   Vault Block:ds_ExampleDS
   Attribute Name:password
   Configuration should be done as follows:
   VAULT::ds_ExampleDS::password::1
   ************************************************
   ```

10. **Utilisez le string crypté dans votre configuration.**

Utilisez le string de l'étape de configuration précédente, à la place du string en texte brut. Une source de données utilisant le mot de passe crypté ci-dessus, est indiquée ci-dessous.

```
...
  <subsystem xmlns="urn:jboss:domain:datasources:1.0">
    <datasources>
      <datasource jndi-name="java:jboss/datasources/ExampleDS"
enabled="true" use-java-context="true" pool-name="H2DS">
        <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1</connection-url>
        <driver>h2</driver>
        <pool></pool>
        <security>
          <user-name>sa</user-name>
          <password>${VAULT::ds_ExampleDS::password::1}</password>
        </security>
      </datasource>
      <drivers>
         <driver name="h2" module="com.h2database.h2">
            <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-
datasource-class>
         </driver>
      </drivers>
    </datasources>
  </subsystem>
...
```

Vous pouvez utiliser un string crypté n'importe où dans votre fichier de configuration autonome ou de domaine pour lequel les expressions sont autorisées.

NOTE

Pour vérifier si les expressions sont utilisées dans un sous-système particulier, exécutez la commande CLI suivante sur ce sous-système :

```
/host=master/core-service=management/security-
realm=TestRealm:read-resource-description(recursive=true)
```

À partir du résultat de cette commande, cherchez la valeur du paramètre **`expressions-allowed`**. Si 'true', vous pourrez utiliser des expressions dans la configuration de ce sous-système particulier.

Une fois que vous aurez mis votre string dans le keystore, utilisez la syntaxe suivante pour remplacer tout string en texte claire par un texte crypté.

```
${VAULT::VAULT_BLOCK::ATTRIBUTE_NAME::ENCRYPTED_VALUE}
```

Voici un exemple de valeur réelle, où le bloc d'archivage sécurisé est **`ds_ExampleDS`** et l'attribut est **`password`**.

```
<password>${VAULT::ds_ExampleDS::password::1}</password>
```

Rapporter un bogue

11.13.7. Stocker et résoudre des strings sensibles de vos applications

Aperçu

Les éléments de configuration de la plate-forme JBoss EAP 6 prennent en charge la capacité à régler les chaînes cryptées en fonction des valeurs stockées dans Java Keystore, via le mécanisme d'archivage sécurisé. Vous pouvez ajouter le support de cette fonctionnalité à vos propres applications.

Tout d'abord, ajoutez le mot de passe dans votre archivage sécurisé. En second lieu, remplacez le mot de passe de texte clair par celui qui est stocké dans l'archivage sécurisé. Vous pouvez utiliser cette méthode pour obscurcir les strings sensibles de votre application.

Pré-requis

Avant d'effectuer cette procédure, assurez-vous que le répertoire utilisé pour stocker vos fichiers dans l'archivage sécurisé existe bien. Qu'importe où vous les placez, tant que l'utilisateur qui exécute JBoss EAP 6 dispose de l'autorisation de lire et écrire des fichiers. Cet exemple situe le répertoire **vault/** dans le répertoire **/home/*USER*/vault/**. L'archivage sécurisé lui-même correspond à un fichier nommé **vault.keystore** qui se trouve dans le répertoire **vault/**.

Exemple 11.40. Ajout de la chaîne de mot de passe dans l'archivage sécurisé

Ajoutez la chaîne de mot de passe dans l'archivage sécurisé par la commande ***EAP_HOME*/bin/vault.sh**. La série de commandes et de réponses est incluse dans la session suivante. Les valeurs saisies par l'utilisateur apparaîtront clairement. Certaines sorties seront supprimées pour le formatage. Dans Microsoft Windows, le nom de la commande est **vault.bat**. Notez que dans Microsoft Windows, les chemins d'accès au fichier utilisent le caractère \ comme séparateur de répertoire, et non pas le caractère /.

```
[user@host bin]$ ./vault.sh
*********************************
****  JBoss Vault ********
*********************************
Please enter a Digit::   0: Start Interactive Session  1: Remove
Interactive Session  2: Exit
0
Starting an interactive session
Enter directory to store encrypted files:/home/user/vault/
Enter Keystore URL:/home/user/vault/vault.keystore
Enter Keystore password: ...
Enter Keystore password again: ...
Values match
Enter 8 character salt:12345678
Enter iteration count as a number (Eg: 44):25

Enter Keystore Alias:vault
Vault is initialized and ready for use
Handshake with Vault complete
Please enter a Digit::   0: Store a password  1: Check whether password
exists  2: Exit
0
Task:  Store a password
Please enter attribute value: sa
Please enter attribute value again: sa
Values match
Enter Vault Block:DS
Enter Attribute Name:thePass
```

```
Secured attribute value has been stored in vault.

Please make note of the following:
********************************************
Vault Block:DS
Attribute Name:thePass
Configuration should be done as follows:
VAULT::DS::thePass::1
********************************************

Please enter a Digit::   0: Store a password  1: Check whether password
exists  2: Exit
2
```

La chaîne qui sera ajoutée au code Java est la dernière valeur de sortie, la ligne commençant par **VAULT**.

Le servlet suivant utilise la chaîne voûtée au lieu d'un mot de passe de texte clair. La version en texte clair est commentée afin que vous puissiez voir la différence.

Exemple 11.41. Servlet qui utilise un mot de passe d'archivage sécurisé.

```
package vaulterror.web;

import java.io.IOException;
import java.io.Writer;

import javax.annotation.Resource;
import javax.annotation.sql.DataSourceDefinition;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;


/*@DataSourceDefinition(
        name = "java:jboss/datasources/LoginDS",
        user = "sa",
        password = "sa",
        className = "org.h2.jdbcx.JdbcDataSource",
        url = "jdbc:h2:tcp://localhost/mem:test"
)*/
@DataSourceDefinition(
        name = "java:jboss/datasources/LoginDS",
        user = "sa",
        password = "VAULT::DS::thePass::1",
        className = "org.h2.jdbcx.JdbcDataSource",
        url = "jdbc:h2:tcp://localhost/mem:test"
)
@WebServlet(name = "MyTestServlet", urlPatterns = { "/my/" },
loadOnStartup = 1)
public class MyTestServlet  extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;


    @Resource(lookup = "java:jboss/datasources/LoginDS")
    private DataSource ds;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        Writer writer = resp.getWriter();
        writer.write((ds != null) + "");
    }
}
```

Votre servlet est maintenant capable de résoudre la chaîne d'archivage sécurisé.

Rapporter un bogue

11.14. ENCODAGE SE CONFORMANT À FIPS 140-2

11.14.1. Conformité FIPS 140-2

FIPS (Federal Information Processing Standard) 140-2 (FIPS 140-2) est un standard de sécurité informatique gouvernemental US pour l'accréditation des modules informatiques cryptographiques. Le standard FIPS 140-2 est souvent un pré-requis pour les systèmes informatiques des agences gouvernementales et pour le secteur commercial privé.

JBoss EAP 6 utilise le chiffrement de modules externes et peut être configuré pour pouvoir utiliser un module de cryptage compatible FIPS 140-2.

Rapporter un bogue

11.14.2. Mots de passe conformes à FIPS 140-2

Un mot de passe conforme à FIPS doit avoir les caractéristiques suivantes :

1. Une longueur minimale de sept (7) caractères.
2. Il doit inclure des caractères d'au moins trois (3) des classes suivantes :
    - chiffres ASCII,
    - minuscules ASCII,
    - majuscules ASCII,
    - non alphanumériques ASCII, et
    - non-ASCII.

Si le premier caractère du mot de passe est en lettre majuscule ASCII, il ne comptera pas comme une lettre majuscule ASCII pour la restriction 2.

Si le dernier caractère du mot de passe est un chiffre ASCII, il ne comptera pas comme chiffre ASCII pour la restriction 2.

Rapporter un bogue

11.14.3. Activer la Cryptography FIPS 140-2 pour SSL dans Red Hat Enterprise Linux 6

Cette tâche décrit comment configurer le conteneur web (JBoss Web) de JBoss EAP 6 pour que la cryptographie soit conforme à FIPS 140-2 pour SSL. Cette tâche ne couvre que les étapes spécifiques à Red Hat Enterprise Linux 6.

Cette tâche utilise la bibliothèque Mozilla NSS en mode FIPS pour cette fonctionnalité.

Pré-requis

- Red Hat Enterprise Linux 6 doit déjà être configuré pour être configuré en conformité avec FIPS 140-2. Voir https://access.redhat.com/knowledge/solutions/137833.

Procédure 11.45. Voir Conformité Cryptographie FIPS 140-2 pour SSL

1. **Créez la base de données**
   Créez la base de données NSS dans un répertoire qui appartienne à l'utilisateur **`jboss`**.

   ```
   $ mkdir -p  /usr/share/jboss-as/nssdb
   $ chown jboss /usr/share/jboss-as/nssdb
   $ modutil -create -dbdir /usr/share/jboss-as/nssdb
   ```

2. **Créez un fichier de configuration NSS**
   Créez un nouveau fichier texte ayant comme nom **`nss_pkcsll_fips.cfg`** dans le répertoire **`/usr/share/jboss-as`** avec le contenu suivant :

   ```
   name = nss-fips
   nssLibraryDirectory=/usr/lib64
   nssSecmodDirectory=/usr/share/jboss-as/nssdb
   nssModule = fips
   ```

   Le fichier de configuration NSS doit spécifier :

   - un nom,
   - le répertoire où se trouve la bibliothèque, et
   - le répertoire où la base de données NSS a été créée selon l'étape 1.

   Si vous n'êtes pas sur une version 64bit de Red Hat Enterprise Linux 6, alors définissez **`nssLibraryDirectory`** à **`/usr/lib`** à la place de **`/usr/lib64`**.

3. **Activez le fournisseur SunPKCS11**
   Modifiez le fichier de configuration **`java.security`** de votre JRE (**`$JAVA_HOME/jre/lib/security/java.security`**) et ajoutez la ligne suivante :

   ```
   security.provider.1=sun.security.pkcs11.SunPKCS11  /usr/share/jboss-
   as/nss_pkcsll_fips.cfg
   ```

Notez que le fichier de configuration spécifié sur cette ligne est le fichier créé à l'étape 2.

Toute autre ligne **`security.provider.X`** de ce fichier devra posséder la valeur *X* +1 pour que la priorité soit donnée à ce fournisseur.

4. **Activez le mode FIPS pour la bibliothèque NSS**
   Exécutez la commande **`modutil`** comme indiqué pour activer le mode FIPS :

   ```
   modutil -fips true -dbdir /usr/share/jboss-as/nssdb
   ```

   Notez que le répertoire indiqué ici est le répertoire créé à l'étape 1.

   Vous aurez sans doute une erreur de bibliothèque à ce niveau, ce qui vous oblige à régénérer les signatures de bibliothèques sur certains des objets NSS partagés.

5. **Modifiez le mot de passe du token FIPS**
   Définissez le mot de passe du token FIPS par la commande suivante. Notez que le nom du token doit correspondre à **`NSS FIPS 140-2 Certificate DB`**.

   ```
   modutil -changepw "NSS FIPS 140-2 Certificate DB" -dbdir
   /usr/share/jboss-as/nssdb
   ```

   Le mot de passe utilisé pour le token FIPS doit être un mot de passe conforme FIPS.

6. **Créez le certificat grâce aux outils NSS**
   Saisissez la commande suivante pour créer un certificat par les outils NSS.

   ```
   certutil -S -k rsa -n jbossweb  -t "u,u,u" -x -s "CN=localhost,
   OU=MYOU, O=MYORG, L=MYCITY, ST=MYSTATE, C=MY" -d /usr/share/jboss-
   as/nssdb
   ```

7. **Configurez le connecteur HTTPS pour utiliser le keystore PKCS11**
   Ajoutez un connecteur HTTPS par la commande suivante dans JBoss CLI :

   ```
   /subsystem=web/connector=https/:add(socket-
   binding=https,scheme=https,protocol=HTTP/1.1,secure=true)
   ```

   Puis, ajoutez la configuration SSL par la commande suivante, en remplaçant PASSWORD par le mot de passe conforme FIPS de l'étape 5.

   ```
   /subsystem=web/connector=https/ssl=configuration:add(name=https,pass
   word=PASSWORD,keystore-type=PKCS11,
   cipher-
   suite="SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_S
   HA,
   TLS_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
   TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,
   TLS_DHE_DSS_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
   TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC
   _SHA,
   TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CB
   C_SHA,
   TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CB
   C_SHA,
   ```

```
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
,
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SH
A,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SH
A,
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_anon_WITH_AES_128_CBC_S
HA,
TLS_ECDH_anon_WITH_AES_256_CBC_SHA")
```

8. **Vérifier**
   Vérifiez que la JVM puisse lire la clé privée du keystore PKCS11 en exécutant la commande suivante :

```
keytool -list -storetype pkcs11
```

Exemple 11.42. Configuration XML du connecteur HTTPS avec conformité FIPS 140-2

```
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-
binding="https" secure="true">
  <ssl name="https" password="****"
      cipher-
suite="SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
         TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
         TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,

TLS_DHE_DSS_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,

TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
,

TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SH
A,

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SH
A,

TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,

TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,

TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_anon_WITH_AES_128_CBC_SHA,
         TLS_ECDH_anon_WITH_AES_256_CBC_SHA"
      keystore-type="PKCS11"/>
</connector>
```

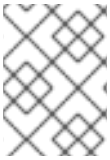
Notez que l'attribut **`cipher-suite`** a des sauts de ligne insérés pour faciliter la lecture.

Rapporter un bogue

11.14.4. Activer la cryptographie FIPS 140-2 dans Apache HTTP Server

Vous pouvez activer la cryptographie FIPS 140-2 dans le serveur Apache HTTP en insérant la directive ***SSLFIPS on*** dans le fichier de configuration du serveur Apache HTTP : **httpd.conf** ou **ssl.conf**. Cette directive est en dehors de la section de configuration du **VirtualHost**.

La directive ***SSLFIPS on*** active l'indicateur FIPS_mode de la bibliothèque SSL. Ce mode s'applique à toutes les opérations de la bibliothèque SSL. Après avoir ajouté cette directive, vous devrez démarrer à nouveau le serveur Apache HTTP pour que les changements deviennent actifs.



NOTE

Pour activer FIPS, vous devez avoir un OpenSSL qui supporte l'indicateur **FIPS_mode** installé sur votre système.

Rapporter un bogue

CHAPITRE 12. RÉFÉRENCE À L'ADMINISTRATION DE LA SÉCURITÉ

12.1. MODULES D'AUTHENTIFICATION INCLUS

Les modules d'authentification suivants sont inclus dans JBoss EAP 6. Certains d'entre eux gèrent l'autorisation ainsi que l'authentification. Ceux-ci incluent généralement le mot **`Role`** dans le nom de **`Code`**

Quand vous configurez ces modules, utiliser la valeur **`Code`** ou le nom complet (package) pour vous référer au module.

Modules d'authentification

Tableau 12.1. `RealmDirect`

Code	**`RealmDirect`**
Class	**`org.jboss.as.security.RealmDirectLoginModule`**

Description	Une implémentation du module de connexion en interface directe avec le domaine de la sécurité. Ce module de connexion permet à toutes les interactions avec le backing store d'être déléguées au domaine, supprimant ainsi la nécessité de définitions en duplicata ou synchronisées. Utilisé pour les appels d'accès distant et l'interface de gestion.

Tableau 12.2. Options du module `RealmDirect`

Option	Type	Par défaut	Description
`realm`	chaîne	**`ApplicationRealm`**	Nom de domaine souhaité.

Tableau 12.3. `Client`

Code	**`Client`**
Class	**`org.jboss.security.ClientLoginModule`**
Description	Ce module de connexion est conçu pour établir l'identité de l'appelant et les informations d'identification lorsque JBoss EAP agit en tant que client. Il ne doit jamais servir sous forme de domaine de sécurité pour l'authentification serveur.

Tableau 12.4. Options de module `Client`

Option	Type	Par défaut	Description
`multi-threaded`	**`true`** ou **`false`**	**`false`**	Définir la valeur sur true si chaque thread possède son propre stockage d'informations d'identification et principal. La valeur false pour indiquer que tous les threads de la machine virtuelle partagent la même identité et informations d'identification.

Option	Type	Par défaut	Description
password-stacking	**useFirstPass** ou **false**	**false**	Définir la valeur sur **useFirstPass** pour indiquer que ce module de connexion doive rechercher les informations stockées dans le **LoginContext** à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec celui-ci.
restore-login-identity	**true** ou **false**	**false**	Définir sur true si l'identité et les informations d'identification du début le la méthode **login()** doivent être restaurées une fois que la méthode **logout()** est invoquée.

Tableau 12.5. Remoting

Code	**Remoting**
Class	**org.jboss.as.security.remoting.RemotingLoginModule**
Description	Le module de connexion est utilisé pour vérifier si la demande actuellement authentifiée est une demande reçue via une connexion d'accès distant et, si c'est le cas, l'identité qui aura été créée pendant le processus d'authentification à accès distant sera utilisée et associée à la demande en cours. Si la demande n'est pas parvenue via une connexion d'accès distant, ce module ne fera rien et permettra à la connexion JAAS de continuer vers le module suivant.

Tableau 12.6. Options de Modules à accès distant

Option	Type	Par défaut	Description

Option	Type	Par défaut	Description
password-stacking	**useFirstPass** ou **false**	**false**	La valeur de **useFirstPass** indique si ce module de connexion doit tout d'abord rechercher les informations stockées dans le **LoginContext** à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec celui-ci.
principalClass	Un nom de classe complet.	aucun	Une classe d'implémentation de **Principal** contenant un constructeur qui prend les arguments du String comme nom de principal.
unauthenticatedIdentity	Un nom de principal.	aucun	Définit le nom principal devant être affecté aux demandes qui ne contiennent aucune information d'authentification. Cela peut permettre à des servlets non protégés d'appeler des méthodes sur EJB ne nécessitant pas un rôle spécifique. Un tel principal ne possédera aucun rôle associé et ne pourra accéder qu'aux méthodes EJB ou EJB non sécurisées associées à la contrainte de **unchecked permission**.

Tableau 12.7. `Certificate`

Code	**Certificate**
Class	**org.jboss.security.auth.spi.BaseCertLoginModule**
Description	Ce module de connexion est conçu pour authentifier les utilisateurs basés sur des **X509 Certificates**. Un cas d'utilisation pour ceci est l'authentification **CLIENT-CERT** d'une application web.

Tableau 12.8. Options de module `Certificate`

Option	Type	Par défaut	Description
`securityDomain`	chaîne	aucun	Nom du domaine de sécurité qui possède la configuration JSSE du truststore qui contient les certificats approuvés.
`verifier`	class	aucun	Le nom de classe du **`org.jboss.security.auth.certs.X509CertificateVerifier`** à utiliser pour vérifier le certificat de connexion.

Tableau 12.9. `CertificateRoles`

Code	**`CertificateRoles`**
Class	**`org.jboss.security.auth.spi.CertRolesLoginModule`**
Description	Ce module de connexion étend le module de connexion de certificat pour ajouter des fonctions de mappage de rôle à partir d'un fichier de propriétés. Il prend les mêmes options que le module de connexion du certificat et ajoute les options suivantes.

Tableau 12.10. Options de module `CertificateRoles`

Option	Type	Par défaut	Description

Option	Type	Par défaut	Description
rolesProperties	chaîne	**roles.properties**	Le nom de la ressource ou du fichier contenant les rôles à attribuer à chaque utilisateur. Le fichier de propriétés de rôle doit être dans sous format **username=role1,role2**, où le nom d'utilisateur est le nom unique du certificat, sans signe égal **=** ou espace blanc. L'exemple suivant est dans le bon format : `CN\=unit-tests-client,\ OU\=Red\ Hat\ Inc.,\ O\=Red\ Hat\ Inc.,\ ST\=North\ Carolina,\ C\=US`
defaultRolesProperties	chaîne	**defaultRoles.properties**	Nom de la ressource ou du fichier où se replier si le fichier **rolesProperties** est introuvable.
roleGroupSeparator	Un seul caractère.	**.** (un seul point)	Le caractère à utiliser comme séparateur de groupe rôle dans le fichier de **rolesProperties**

Tableau 12.11. Database

Code	**Database**
Class	**org.jboss.security.auth.spi.DatabaseServerLoginModule**
Description	Un module de connexion basé-JDBC supportant le mappage de rôle et l'authentification. Basé sur deux tables logiques, avec les définitions suivantes. • **Principals: PrincipalID (text), Password (text)** • **Roles: PrincipalID (text), Role (text), RoleGroup (text)**

Tableau 12.12. `Database` Module Options

Option	Type	Par défaut	Description
`digestCallback`	Un nom de classe complet	aucun	Le nom de classe de l'implémentation **`DigestCallback`** qui inclut le contenu pre/post digest comme les valeurs salt pour hacher le mot de passe donné. Utilisé uniquement si l'**`hashStoreAlgorithm`** est spécifié.
`dsJndiName`	Ressource JNDI	aucun	Le nom de la ressource JNDI qui store les informations d'authentification. Cette option est requise.
`hashAlgorithm`	String	Utiliser des mots de passe en texte brut	L'algorithme digest du message utilisé pour hâcher les mots de passe. Les algorithmes pris en charge dépendent du fournisseur de sécurité Java, mais **`MD5`**, **`SHA-1`**, et **`SHA-256`** sont pris en charge.
`hashCharset`	String	Le codage par défaut de la plateforme	Le nom du charset (jeu de caractères)/codification à utiliser pour convertir la chaîne de mot de passe en tableau d'octets. Inclut tous les noms pris en charge par Java.
`hashEncoding`	String	Base64	Indique le format de code de chaîne à utiliser.
`ignorePasswordCase`	booléen	false	Signe qui indique si la comparaison de mot de passe doit ignorer le cas.
`inputValidator`	Un nom de classe complet	aucun	L'instance de l'implémentation de InputValidator utilisée pour valider le nom d'utilisateur et le mot de passe fournis par le client.
`principalsQuery`	énoncé SQL préparé	**`select Password from Principals where PrincipalID=?`**	La requête SQL préparée pour obtenir les informations sur le principal.
`rolesQuery`	énoncé SQL préparé	aucun	Enoncé SQL préparé en vue d'obtenir les informations concernant les rôles. Il doit être équivalent à **`select Role, RoleGroup from Roles where PrincipalID=?`**, avec Role comme nom de rôle et la valeur de la colonne RoleGroup doit toujours être **`Roles`** avec un **`R`** majuscule ou **`CallerPrincipal`**.

Option	Type	Par défaut	Description
storeDigestCallback	Un nom de classe complet	aucun	Le nom de classe de l'implémentation **DigestCallback** qui inclut le contenu pre/post digest comme les valeurs salt pour hacher le mot de passe du store/attendu. Utilisé uniquement si **hashStorePassword** ou **hashUserPassword** est sur **true** et si **hashAlgorithm** est spécifié.
suspendResume	booléen	true	Indique si une transaction JTA existante doit être suspendue pendant les opérations de base de données.
throwValidatorError	booléen	false	Indique si les erreurs de validation soivent être exposées ou non aux clients
transactionManagerJndiName	Ressource JNDI	java:/TransactionManager	Le nom JNDI de la source du gestionnaire de transactions utilisé par le module de connexion.

Tableau 12.13. DatabaseCertificate

Code	**DatabaseCertificate**
Class	**org.jboss.security.auth.spi.DatabaseCertLoginModule**
Description	Ce module de connexion étend le module de connexion de certificat pour ajouter des fonctions de mappage de rôle d'une table de bases de données. Il possède les mêmes options mais aussi ces options supplémentaires :

Tableau 12.14. Options de module DatabaseCertificate

Option	Type	Par défaut	Description
dsJndiName	Ressource JNDI		Le nom de la ressource JNDI qui store les informations d'authentification. Cette option est requise.

Option	Type	Par défaut	Description
rolesQuery	énoncé SQL préparé	**select Role,RoleGroup from Roles where PrincipalID=?**	Enoncé SQL préparé en vue d'exécuter pour mapper les rôles. Doit être équivalent à **select Role, RoleGroup from Roles where PrincipalID=?**, avec Role comme nom de rôle et la valeur de la colonne RoleGroup doit toujours être **Roles** avec un **R** majuscule ou **CallerPrincipal**.
suspendResume	**true** ou **false**	**true**	Indique si une transaction JTA existante doit être suspendue pendant les opérations de base de données.

Tableau 12.15. `Identity`

Code	**Identity**
Class	**org.jboss.security.auth.spi.IdentityLoginModule**
Description	Associe le principal spécifié dans les options de module avec n'importe quel sujet authentifié dans le module. Le type de classe de principal utilisée est **org.jboss.security.SimplePrincipal**. Si aucune option de principal n'est spécifiée, on utilisera un principal ayant pour nom **guest**.

Tableau 12.16. Options de module `Identity`

Option	Type	Par défaut	Description
principal	String	**guest**	Le nom à utiliser pour le principal.
roles	liste de strings séparée par des virgules	aucun	Une liste de rôles séparés par des virgules qui seront assignés au sujet.

Tableau 12.17. `Ldap`

Code	**Ldap**

Class	**`org.jboss.security.auth.spi.LdapLoginModule`**
Description	Authentifie sur un serveur LDAP, lorsque le nom d'utilisateur et le mot de passe sont stockés dans un serveur LDAP qui est accessible à l'aide d'un fournisseur LDAP JNDI. Bon nombre des options ne sont pas requises, car elles sont déterminées par le fournisseur LDAP ou l'environnement.

Tableau 12.18. Options de module `Ldap`

Option	Type	Par défaut	Description
`java.naming.factory.initial`	class name	**`com.sun.jndi.ldap.LdapCtxFactory`**	Nom de classe de l'implémentation **`InitialContextFactory`**.
`java.naming.provider.url`	**`ldap://`** URL	aucun	URL pour le serveur LDAP
`java.naming.security.authentication`	**`none`**, **`simple`**, ou le nom d'un mécanisme SASL	**`simple`**	Le niveau de sécurité à utiliser pour la liaison avec le serveur LDAP.
`java.naming.security.protocol`	protocole de transport	Si non spécifié, déterminé par le fournisseur.	Le protocole de transport à utiliser pour l'accès sécurisé, comme SSL.
`java.naming.security.principal`	chaîne	aucun	Le nom du principal permettant d'authentifier l'appelant vers le service. Il est construit à partir des autres propriétés décrites ci-dessous.
`java.naming.security.credentials`	Type d'information d'identification	aucun	Le type d'information d'identification utilisée par le schéma d'authentification. Exemples : mot de passe haché, mot de passe de texte clair, clé ou certificat. Si cette propriété n'est pas spécifiée, le comportement est déterminé par le fournisseur de service.

Option	Type	Par défaut	Description
principalDNPrefix	chaîne		Préfixe ajouté au nom d'utilisateur pour former le DN de l'utilisateur. Vous pouvez demander à l'utilisateur un nom d'utilisateur et créer le nom de domaine DN complet en utilisant **principalDNPrefix** et **principalDNSuffix**.
principalDNSuffix	chaîne		Suffixe ajouté au nom d'utilisateur pour former le DN de l'utilisateur. Vous pouvez demander à l'utilisateur un nom d'utilisateur et créer le nom de domaine DN complet en utilisant **principalDNPrefix** et **principalDNSuffix**.
useObjectCredential	**true** ou **false**	false	Si les informations d'identification doivent être obtenues sous forme d'objet opaque à l'aide du type de Callback **org.jboss.security.auth.callback.ObjectCallback** plutôt que comme mot de passe **char[]** utilisant un JAAS PasswordCallback. Cela permet de passer les informations d'identification **char[]** au serveur LDAP.
rolesCtxDN	DN complet	aucun	Le DN complet pour le contexte à chercher pour les rôles d'utilisateur.

Option	Type	Par défaut	Description
userRolesCtxDNAttributeName	attribut	aucun	L'attribut dans l'objet utilisateur qui contient le nom unique DN pour que le contexte recherche des rôles d'utilisateur. Cela diffère de **rolesCtxDN** car le contexte de recherche de rôles d'utilisateur peut être unique pour chaque utilisateur.
roleAttributeID	attribut	**roles**	Nom de l'attribut qui contient les rôles d'utilisateur.
roleAttributeIsDN	**true** ou **false**	**false**	Indique si le **roleAttributeID** contient le nom de domaine complet d'un objet de rôle. Si false, le nom de rôle est tiré de la valeur de l'attribut **roleNameAttributeId** du nom de contexte. Certains schémas de répertoire, tel que Active Directory, requièrent que cet attribut soit défini à **true**.
roleNameAttributeID	attribut	**name**	Nom de l'attribut du contexte de **roleCtxDN** qui contient le nom de rôle. Si la propriété **roleAttributeIsDN** est définie sur **true**, cette propriété sera utilisée pour rechercher l'attribut de nom de l'objet rôle.
uidAttributeID	attribut	**uid**	Nom de l'attribut qui contient le **UserRolesAttributeDN** correspondant à l'ID d'utilisateur. Utilisé pour localiser les rôles d'utilisateur.

Option	Type	Par défaut	Description
matchOnUserDN	**true** ou **false**	**false**	Si la recherche de rôles d'utilisateur doit correspondre au DN de l'utilisateur entièrement distinct ou au nom d'utilisateur uniquement. Si **true**, l'userDN complet sera utilisé comme valeur de correspondance. Si **false**, seul le nom d'utilisateur sera utilisé comme valeur de correspondance pour l'attribut **UserRolesAttributeDN**.
allownEmptyPasswords	**true** ou **false**	**false**	Indique si on doit autoriser les mots de passe vides. La plupart des serveurs LDAP traitent les mots de passe vides comme des tentatives de connexion anonymes. Pour rejeter les mots de passe vides, définir à **false**.

Tableau 12.19. LdapExtended

Code	**LdapExtended**
Class	**org.jboss.security.auth.spi.LdapExtLoginModule**

Description	Une autre implémentation de module de connexion LDAP qui utilise les recherches pour localiser l'utilisateur de liaisons et les rôles associés. La requête de rôles suit récursivement les noms de domaines DN pour naviguer dans une structure de rôles hiérarchique. Il utilise les mêmes options **`java.naming`** que le module Ldap, et utilise les options suivantes à la place des autres options du module Ldap. L'authentification a lieu en 2 étapes : 1. Une première liaison au serveur LDAP est faite par les options **`bindCredential`**. **`bindDN`** est l'utilisateur ayant la possibilité de rechercher les arborescences baseCtxDN et **`rolesCtxDN`** pour l'utilisateur et les rôles. Le DN pour authentifier l'utilisateur est interrogé en utilisant le filtre spécifié par l'attribut **`baseFilter`**. 2. Le DN de l'utilisateur qui en résulte est authentifié par la liaison au serveur LDAP en utilisant le DN de l'utilisateur comme environnement de **`InitialLdapContext`** **`Context.SECURITY_PRINCIPAL`**. La propriété **`Context.SECURITY_CREDENTIALS`** est définie avec le mot de passe String obtenu par le gestionnaire de rappel.

Tableau 12.20. Options de module `LdapExtended`

Option	Type	Par défaut	Description
`baseCtxDN`	DN complet	aucun	Le DN fixe de contexte de niveau supérieur pour commencer la recherche utilisateur.
`bindCredential`	string, parfois crypté	aucun	Consulter le *JBoss EAP Security Guide* pour obtenir plus d'informations.
`bindDN`	DN complet	aucun	Le DN utilisé pour la liaison au serveur LDAP pour les requêtes d'utilisateurs et de rôles. Ce nom unique a besoin d'autorisations de lecture et de recherche pour les valeurs **`baseCtxDN`** et **`rolesCtxDN`**.

Option	Type	Par défaut	Description
baseFilter	String de filtre LDAP	aucun	Un filtre de recherche permettant de localiser le contexte de l'utilisateur à authentifier. L'entrée username ou **userDN** obtenue à partir du rappel de module de connexion est substitué dans le filtre à chaque fois qu'une expression **{0}** est utilisée. Un exemple de filtre de recherche est **(uid={0})**.
rolesCtxDN	DN complet	aucun	Le DN fixe du contexte pour rechercher des rôles d'utilisateur. Ce n'est pas le DN où les rôles se trouvent, mais le DN où les objets contenant les rôles d'utilisateur se trouvent. Par exemple, dans un serveur Active Directory de Microsoft, c'est le DN où le compte d'utilisateur se trouve.
roleFilter	String de filtre LDAP	aucun	Un filtre de recherche permettant de localiser les rôles associés à l'utilisateur authentifié. L'entrée user ou **userDN** obtenue à partir du rappel de module de connexion est substitué dans le filtre à chaque fois qu'une expression **{0}** est utilisée. L'utilisateurDN **userDN** authentifié sera substitué dans le filtre à chaque fois qu'un **{1}** est utilisé. Un exemple de filtre de recherche qui correspond au nom d'utilisateur d'entrée serait **(member={0})**. Une alternative correspondant au **userDN** authentifié serait **(member={1})**.

Option	Type	Par défaut	Description
roleAttributeIsDN	**true** ou **false**	**false**	Indique si le **roleAttributeID** contient le nom de domaine complet d'un objet de rôle. Si false, le nom de rôle est tiré de la valeur de l'attribut **roleNameAttributeId** du nom de contexte. Certains schémas de répertoire, tel que Active Directory, requièrent que cet attribut soit défini à **true**.
defaultRole	Un nom de rôle.	aucun	Un rôle inclus pour tous les utilisateurs authentifiés
parseRoleNameFromDN	**true** ou **false**	**false**	Un indicateur qui signale si le DN retourné par une requête contient le roleNameAttributeID. Si la valeur est **true**, le DN contient le roleNameATtributeID. Si la valeur est **false**, le DN ne contient pas le roleNameAttributeID. Cet indicateur peut améliorer les performances des requêtes LDAP.
parseUsername	**true** ou **false**	**false**	Un indicateur qui signale si le DN doit être vérifié niveau nom d'utilisateur. Si la valeur est **true**, le DN est vérifié niveau nom d'utilisateur. Si la valeur est **false**, le DN n'est pas vérifié au niveau nom d'utilisateur. Cette option peut à la fois être utilisée pour usernameBeginString et usernameEndString.
usernameBeginString	chaîne	aucun	Définit le string qui doit être supprimé au début du DN pour révéler le nom d'utilisateur. Cette option est utilisée avec **usernameEndString**.

Option	Type	Par défaut	Description
usernameEndString	chaîne	aucun	Définit le string qui doit être supprimé à la fin du DN pour révéler le nom d'utilisateur. Cette option est utilisée avec **userBeginString**.
roleNameAttributeID	attribut	**name**	Nom de l'attribut du contexte de **roleCtxDN** qui contient le nom de rôle. Si la propriété **roleAttributeIsDN** est définie sur **true**, cette propriété sera utilisée pour rechercher l'attribut de nom de l'objet rôle.
distinguishedNameAttribute	attribut	**distinguishedName**	Le nom de l'attribut dans l'entrée utilisateur qui contient le nom unique de l'utilisateur. Ceci peut être nécessaire si le DN de l'utilisateur lui-même contient des caractères spéciaux (barre oblique inverse par exemple) qui empêchent le mappage de l'utilisateur. Si l'attribut n'existe pas, le DN de l'entrée sera utilisé.
roleRecursion	entier relatif	**0**	Le nombre de niveaux de récursivité de la recherche de rôle dans un contexte de correspondance donné. Désactiver la récursivité en attribuant le paramètre **0**.
searchTimeLimit	entier relatif	**10000** (10 seconds)	Timeout en millisecondes pour les recherches utilisateur/rôle.
searchScope	Un parmi : **OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE**	**SUBTREE_SCOPE**	Étendue à utiliser.

Option	Type	Par défaut	Description
allownEmptyPasswords	**true** ou **false**	**false**	Indique si on doit autoriser les mots de passe vides. La plupart des serveurs LDAP traitent les mots de passe vides comme des tentatives de connexion anonymes. Pour rejeter les mots de passe vides, définir à false.
referralUserAttributeIDToCheck	attribut	aucun	Si vous n'utilisez pas de renvois, cette option peut être ignorée. Lorsque vous utilisez des références, cette option indique le nom d'attribut qui contient les utilisateurs définis pour un certain rôle (par exemple, **member**), si l'objet rôle est à l'intérieur du référentiel. Les utilisateurs sont vérifiés par le contenu de cet attribut de nom. Si cette option n'est pas définie, le contrôle échouera à chaque fois, alors les objets role ne peuvent pas être stockés dans un arbre de référence.

Tableau 12.21. `RoleMapping`

Code	**RoleMapping**
Class	**org.jboss.security.auth.spi.RoleMappingLoginModule**
Description	Mappe un rôle qui est le résultat final du processus d'authentification de manière déclarative. Ce module doit être marqué comme étant **optionnel** quand vous y ajoutez le domaine de sécurité.

Tableau 12.22. Options de module `RoleMapping`

Option	Type	Par défaut	Description

Option	Type	Par défaut	Description
rolesProperties	Le chemin d'accès complet et le nom d'une ressource ou d'un fichier de propriétés	**none**	Nom de la ressource ou du fichier de propriétés qui mappe les rôles aux rôles de remplacement. Le format est **original_role=role1,role2,role3**
replaceRole	**true** ou **false**	**false**	Indique si on doit ajouter les rôles en cours, ou les remplacer par les rôles mappés. Sont remplacés si définis sur **true**.

NOTE

L'option de module **rolesProperties** est requise pour le RoleMapping.

Tableau 12.23. RunAs

Code	**RunAs**
Class	**org.jboss.security.auth.spi.RunAsLoginModule**
Description	Un module d'assistance qui pousse un rôle **run as** dans la pile pour la durée de la phase d'authentification de la connexion, et qui extrait le rôle **run as** de la pile soit dans la phase commit ou abort. Ce module de connexion fournit un rôle pour les autres modules de connexion qui doivent accéder aux ressources sécurisées afin d'effectuer leur authentification, par exemple un module de connexion qui accède à un EJB sécurisé. **RunAsLoginModule** doit être configuré avant que les modules de connexion qui ont besoin d'un rôle **run as** soient mis en place.

Tableau 12.24. Options RunAs

Option	Type	Par défaut	Description
roleName	nom de rôle	**nobody**	Le nom de rôle à utiliser comme rôle **run as** pendant la phase de connexion.

Option	Type	Par défaut	Description
principalName	nom de principal	**nobody**	Le nom de principal à utiliser comme principal **run as** pendant la phase de connexion. S'il n'est pas spécifié, on utilisera la valeur par défaut **nobody**.
principalClass	Un nom de classe complet.	aucun	Une classe d'implémentation de **Principal** contenant un constructeur qui prend les arguments du String comme nom de principal.

Tableau 12.25. Simple

Code	**Simple**
Class	**org.jboss.security.auth.spi.SimpleServerLoginModule**
Description	Module d'installation rapide de sécurité pour les tests. Implémente ce simple algorithme : • Si le mot de passe est null, authentifie l'utilisateur et assigne une identité **guest** et un rôle **guest**. • Sinon, quand le mot de passe est égal à l'utilisateur, assigne une identité égale au nom d'utilisateur et aux deux rôles **admin** et **guest**. • Sinon, l'authentification échoue.

Options de module Simple

Le module **Simple** n'a pas d'options.

Tableau 12.26. ConfiguredIdentity

Code	**ConfiguredIdentity**
Class	**org.picketbox.datasource.security.ConfiguredIdentityLoginModule**
Description	Associe le principal spécifié dans les options du module avec n'importe quel sujet authentifié dans le module. Le type de classe du Principal classe est **org.jboss.security.SimplePrincipal**.

Tableau 12.27. Options de module `ConfiguredIdentity`

Option	Type	Par défaut	Description
`username`	chaîne	aucun	Le nom d'utilisateur pour l'authentification.
`password`	string encodé	""	Le mot de passe à utiliser pour l'authentification. Pour crypter le mot de passe, utilisez le module directement dans la ligne de commande. `java org.picketbox.datasource.security.SecureIdentityLoginModule` *`password_to_encrypt`* Coller le résultat de cette commande dans le champ d'option de module. La valeur par défaut correspond à une chaîne vide.
`principal`	Nom d'un principal	**`none`**	Le principal qui sera associé avec n'importe quel sujet authentifié dans le module.

Tableau 12.28. `SecureIdentity`

Code	**`SecureIdentity`**
Class	**`org.picketbox.datasource.security.SecureIdentityLoginModule`**
Description	Ce module est fourni à des fins d'héritage. Il permet de crypter un mot de passe et ensuite d'utiliser le mot de passe crypté avec un principal statique. Si votre application utilise **`SecureIdentity`**, envisager plutôt d'utiliser un mécanisme d'archivage sécurisé de mot de passe.

Tableau 12.29. Options de module `SecureIdentity`

Option	Type	Par défaut	Description
username	chaîne	aucun	Le nom d'utilisateur pour l'authentification.
password	string encodé	""	Le mot de passe à utiliser pour l'authentification. Pour crypter le mot de passe, utilisez le module directement dans la ligne de commande. `java org.picketbox.datasource.security.SecureIdentityLoginModule` *`password_to_encrypt`* Coller le résultat de cette commande dans le champ d'option de module. La valeur par défaut correspond à une chaîne vide.
managedConnectionFactoryName	Ressource JCA	aucun	Le nom de la fabrique de connexions JCA à utiliser pour votre source de données.

Tableau 12.30. `PropertiesUsers`

Code	**PropertiesUsers**
Class	**org.jboss.security.auth.spi.PropertiesUsersLoginModule**
Description	Utilise un fichier de propriétés pour stocker les noms d'utilisateur et les mots de passe pour l'authentification. Aucune autorisation (correspondance de rôle) n'est fournie. Ce module convient seulement pour les tests.

Tableau 12.31. `SimpleUsers`

Code	**SimpleUsers**
Class	**org.jboss.security.auth.spi.SimpleUsersLoginModule**

Description	Ce module de connexion stocke le nom d'utilisateur et le mot de passe en texte clair avec ***module-option***. le ***name*** de ***module-option*** et les attributs de ***value*** se réfèrent à un nom d'utlisateur et à un mot de passe. Il est inclus pour les essais uniquement et n'est pas approprié pour un environnement de production.

Tableau 12.32. `LdapUsers`

Code	**`LdapUsers`**
Class	**`org.jboss.security.auth.spi.LdapUsersLoginModule`**
Description	Le module **`LdapUsers`** est remplacé par les modules **`ExtendedLDAP`** et **`AdvancedLdap`**.

Tableau 12.33. `Kerberos`

Code	**`Kerberos`**
Class	**`com.sun.security.auth.module.Krb5LoginModule`**
Description	Effectue l'authentification de connexion Kerberos avec GSSAPI. Ce module fait partie de la structure de sécurité de l'API fourni par Sun Microsystems. Vous trouverez des informations à ce sujet dans http://docs.oracle.com/javase/7/docs/jre/api/security/jaas/spec/com/sun/security/auth/module/Krb5LoginModule.html Ce module devra être mis en correspondance avec un autre module qui gère les mappages d'authentification et de rôles.

Tableau 12.34. Options de module `Kerberos`

Option	Type	Par défaut	Description
`storekey`	**`true`** ou **`false`**	false	Indique si on doit ajouter **`KerberosKey`** dans les informations d'authentification privées du sujet.
`doNotPrompt`	**`true`** ou **`false`**	false	Si défini sur **`true`**, l'utilisateur n'aura pas besoin de mot de passe.

Option	Type	Par défaut	Description
useTicketCache	Valeur booléenne de **true** ou **false** .	false	Si défini à **true**, le TGT sera obtenu à partir du cache du ticket. Si défini sur **false**, le cache du ticket ne sera pas utilisé.
ticketcache	Un fichier ou une ressource qui représente un cache de ticket Kerberos.	La valeur par défaut dépend du système d'exploitation que vous utilisez. • Red Hat Enterprise Linux / Solaris: **/tmp/krb5cc_*uid***, utilisant la valeur UID du système d'exploitation. • Microsoft Windows Server: utilise l'API LSA (Local Security Authority) pour trouver le ticketcache.	Emplacement du ticketcache.
useKeyTab	**true** ou **false**	false	Indique si on doit obtenir la clé du principal à partir d'un keytab.
keytab	Un fichier ou une ressource représentant un onglet de clé Kerberos.	L'emplacement du fichier de configuration Kerberos du système d'exploitation, ou **/home/*user*/krb5.keytab**	Emplacement du fichier keytab.
principal	chaîne	aucun	Le nom du principal. Cela peut être un simple nom d'utilisateur ou un nom de service tel que **host/testserver.acme.com**. Utiliser cela au lieu d'obtenir le principal d'un fichier keytab, ou lorsque le fichier keytab contient plus d'un principal.

Option	Type	Par défaut	Description
useFirstPass	**true** ou **false**	false	Indique si on doit extraire le nom d'utilisateur et le mot de passe de l'état partagé du module à l'aide de **javax.security.auth.login.name** et de **javax.security.auth.login.password** comme clés. Si l'authentification échoue, il n'y aura pas de nouvelle tentative.
tryFirstPass	**true** ou **false**	false	Identique à **useFirstPass**, mais si l'authentification échoue, le module utilise le **CallbackHandler** pour récupérer un nouveau nom d'utilisateur et mot de passe. Si la seconde authentification échoue, l'échec sera signalé à l'application appelante.
storePass	**true** ou **false**	false	Indique si on doit stocker le nom d'utilisateur et le mot de passe de l'état partagé du module. N'a pas lieu si les clés existent déjà dans l'état partagé, ou si l'authentification a échoué.
clearPass	**true** ou **false**	false	Définir à **true** pour supprimer le nom de l'utilisateur et le mot de passe de l'état partagé une fois que les deux phases d'authentification sont terminées.

Tableau 12.35. SPNEGO

Code	**SPNEGO**
Class	**org.jboss.security.negotiation.spnego.SPNEGOLoginModule**

Description	Effectue l'authentification de connexion SPNEGO vers un serveur Microsoft Active Directory ou autre environnement qui supporte SPNEGO. SPNEGO peut également transporter les informations d'identification de Kerberos. Ce module a besoin de fonctionner en parallèle à un autre module qui gère l'authentification et le mappage des rôles.

Tableau 12.36. Options de module `SPNEGO`

Option	Type	Par défaut	Description
`serverSecurityDomain`	**`string`**	**`null`**.	Définit le domaine utilisé pour extraire l'identité du service de serveur par le biais du module de connexion kerberos. Cette propriété doit être définie.
`removeRealmFromPrincipal`	**`boolean`**	**`false`**	Indique si le domaine Kerberos doit être retiré du principal avant de continuer.
`usernamePasswordDomain`	**`string`**	**`null`**	Indique un autre domaine de sécurité de la configuration qui doit être utilisée comme journalisation d'échec quand Kerberos échoue.

Tableau 12.37. `AdvancedLdap`

Code	**`AdvancedLdap`**
Class	**`org.jboss.security.negotiation.AdvancedLdapLoginModule`**
Description	Module qui fournit davantage de fonctionnalité, comme SASL et l'utilisation d'un domaine de sécurité JAAS.

Tableau 12.38. Options de module `AdvancedLdap`

Option	Type	Par défaut	Description
`bindAuthentication`	chaîne	aucun	Le type d'authentification SASL à utiliser pour se lier au serveur de répertoires.

Option	Type	Par défaut	Description
java.naming.provider.url	**string**	aucun	L'URI du serveur de répertoires.
baseCtxDN	DN complet	aucun	Le nom distinctif à utiliser comme base pour les recherches.
baseFilter	Chaîne représentant un filtre de recherche LDAP	aucun	Le filtre à utiliser pour réduire les résultats des recherches.
roleAttributeID	Valeur de chaîne représentant un attribut LDAP	aucun	L'attribut LDAP qui contient les noms des rôles d'autorisation.
roleAttributeIsDN	**true** ou **false**	**false**	Indique si l'attribut de rôle est un Nom Distinctif (DN).
roleNameAttributeID	Chaîne représentant un attribut LDAP	aucun	L'attribut contenu dans **RoleAttributeId** qui contient lui-même l'attribut de rôle.
recurseRoles	**true** ou **false**	**false**	Indique si on doit chercher récursivement des rôles dans **RoleAttributeId**.
referralUserAttributeIDToCheck	attribut	aucun	Si vous n'utilisez pas de renvois, cette option peut être ignorée. Lorsque vous utilisez des références, cette option indique le nom d'attribut qui contient les utilisateurs définis pour un certain rôle (par exemple, **member**), si l'objet rôle est à l'intérieur du référentiel. Les utilisateurs sont vérifiés par le contenu de cet attribut de nom. Si cette option n'est pas définie, le contrôle échouera à chaque fois, alors les objets role ne peuvent pas être stockés dans un arbre de référence.

Tableau 12.39. AdvancedADLdap

Code	**AdvancedADLdap**
Class	**org.jboss.security.negotiation.AdvancedADLoginModule**
Description	Ce module étend le module de connexion **AdvancedLdap**, et ajoute des paramètres supplémentaires utiles au répertoire Microsoft Active Directory.

Tableau 12.40. UsersRoles

Code	**UsersRoles**
Class	**org.jboss.security.auth.spi.UsersRolesLoginModul**
Description	Un module de connexion supportant des utilisateurs multiples et des rôles utilisateur stockés dans deux fichiers de propriétés différents.

Tableau 12.41. Options de module UsersRoles

Option	Type	Par défaut	Description
usersProperties	Chemin d'accès à un fichier ou à une ressource.	**users.properties**	Fichier ou ressource qui contient les mappages d'utilisateur aux mots de passe. Le format du fichier est ***username=password***
rolesProperties	Chemin d'accès à un fichier ou à une ressource.	**roles.properties**	Fichier ou ressource qui contient les mappages d'utilisateur aux rôles. Le format du fichier est **username=role1,role2,role3**
password-stacking	**useFirstPass** ou **false**	**false**	La valeur de **useFirstPass** indique si ce module de connexion doit tout d'abord rechercher les informations stockées dans le **LoginContext** à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec celui-ci.

Option	Type	Par défaut	Description
hashAlgorithm	Chaîne représentant un algorithme de hachage de mot de passe.	**none**	Le nom de l'algorithme **java.security.MessageDigest** à utiliser pour hacher le mot de passe. Il n'y a pas de valeur par défaut pour cette option, donc vous devez la définir explicitement pour permettre le hachage. Quand **hashAlgorithm** est spécifié, le mot de passe en texte clair obtenu de **CallbackHandler** sera haché avant d'être passé à **UsernamePasswordLoginModule.validatePassword** comme argument **inputPassword**. Le mot de passe stocké dans le fichier **users.properties** doit être haché de façon similaire.
hashEncoding	**base64** ou **hex**	**base64**	Le format du string du mot de passe haché, si hashAlgorithm est défini également.
hashCharset	chaîne	Le codage par défaut défini dans l'environnement runtime du conteneur.	Le codage utilisé pour convertir le mot de passe de texte en clair en un tableau d'octets.
unauthenticatedIdentity	nom de principal	aucun	Définit le nom principal affecté aux demandes qui ne contiennent aucune information d'authentification. Cela peut permettre à des servlets non protégés d'appeler des méthodes sur EJB ne nécessitant pas un rôle spécifique. Un tel principal ne possédera aucun rôle associé et ne pourra accéder qu'aux méthodes EJB ou EJB non sécurisées associées à la contrainte de **permission non contrôlée**.

Modules d'authentification personnalisés

Les modules d'authentification sont des implémentations de **`javax.security.auth.spi.LoginModule`**. Reportez-vous à la documentation de l'API pour plus d'informations sur la création d'un module d'authentification personnalisé.

Rapporter un bogue

12.2. MODULES D'AUTORISATION INCLUS

Les modules suivants procurent des services d'autorisation.

Code	Class
DenyAll	org.jboss.security.authorization.modules.AllDenyAuthorizationModule
PermitAll	org.jboss.security.authorization.modules.AllPermitAuthorizationModule
Delegating	org.jboss.security.authorization.modules.DelegatingAuthorizationModule
Web	org.jboss.security.authorization.modules.web.WebAuthorizationModule
JACC	org.jboss.security.authorization.modules.JACCAuthorizationModule
XACML	org.jboss.security.authorization.modules.XACMLAuthorizationModule

AllDenyAuthorizationModule

Module d'autorisation simple qui refuse toujours une demande d'autorisation. Aucune option de configuration disponible.

AllPermitAuthorizationModule

Module d'autorisation simple qui accepte toujours une demande d'autorisation. Aucune option de configuration disponible.

DelegatingAuthorizationModule

Module d'autorisation par défaut qui délègue le processus de décision aux délégués configurés.

WebAuthorizationModule

Module d'autorisation web par défaut appartenant à la logique d'autorisation tomcat par défaut (permit all/tout permis).

JACCAuthorizationModule

Ce module applique les sémantiques JACC en utilisant deux délégués (WebJACCPolicyModuleDelegate pour les requêtes d'autorisation de conteneur web et EJBJACCPolicyModuleDelegate pour les requêtes de conteneurs EJB). Aucune option de configuration n'est disponible.

XACMLAuthorizationModule

Ce module applique l'autorisation XACML grâce à deux délégués pour les conteneurs EJB et web (WebXACMLPolicyModuleDelegate et EJBXACMLPolicyModuleDelegate). Ce module crée un objet PDP issu des politiques enregistrées et évalue la requête web ou EJB en fonction.

AbstractAuthorizationModule

Module d'autorisation de base remplacé et fournissant une fonction de délégation à d'autres modules d'autorisation.

Rapporter un bogue

12.3. MODULES DE SÉCURITÉ INCLUS

Les rôles de mappage de sécurité suivants sont fournis dans JBoss EAP 6.

Code	Class
PropertiesRoles	**`org.jboss.security.mapping.providers.role.PropertiesRolesMappingProvider`**
SimpleRoles	**`org.jboss.security.mapping.providers.role.SimpleRolesMappingProvider`**
DeploymentRoles	**`org.jboss.security.mapping.providers.DeploymentRolesMappingProvider`**
DatabaseRoles	**`org.jboss.security.mapping.providers.role.DatabaseRolesMappingProvider`**
LdapRoles	**`org.jboss.security.mapping.providers.role.LdapRolesMappingProvider`**
LdapAttributes	**`org.jboss.security.mapping.providers.attribute.LdapAttributeMappingProvider`**

DeploymentRolesMappingProvider

Un module de mappage de rôles qui prend en considération un principal pour des mappages de rôles qui peuvent être effectués dans les descripteurs de déploiement **`jboss-web.xml`** et **`jboss-app.xml`**.

Exemple 12.1. Exemple

```
<jboss-web>
...
  <security-role>
      <role-name>Support</role-name>
```

```
        <principal-name>Mark</principal-name>
        <principal-name>Tom</principal-name>
    </security-role>
...
</jboss-web>
```

org.jboss.security.mapping.providers.DeploymentRoleToRolesMappingProvider

Un module de mappage de rôles à rôle qui prend en considération un principal pour des mappages de rôles qui peuvent être effectués dans les descripteurs de déploiement **`jboss-web.xml`** et **`jboss-app.xml`**. Dans ce cas, le nom du principal désigne un rôle qui mappe d'autres rôles.

Exemple 12.2. Exemple

```
<jboss-web>
...
    <security-role>
        <role-name>Employee</role-name>
        <principal-name>Support</principal-name>
        <principal-name>Sales</principal-name>
    </security-role>
...
</jboss-web>
```

Ce qui signifie que chaque principal ayant pour rôle Support ou Sales aura également le rôle Employee assigné.

org.jboss.security.mapping.providers.OptionsRoleMappingProvider

Le fournisseur de mappage de rôles qui récupère les rôles à partir des options, puis qui les ajoute au Groupe passé. Prend le mappage de style de propriétés du nom de rôle (clé) dans une liste de rôles (valeurs) séparés par des virgules.

org.jboss.security.mapping.providers.principal.SimplePrincipalMappingProvider

Un fournisseur de mappage de principal qui prend un SimplePrincipal et qui le convertit en SimplePrincipal avec un nom de principal différent.

DatabaseRolesMappingProvider

Un MappingProvider qui lit les rôles à partir d'une base de données.

Options :

- **`dsJndiName`** : nom JNDI de la source de données utilisée pour mapper les rôles vers l'utilisateur.
- **`rolesQuery`** : cette option doit correspondre à un énoncé préparé équivalent à "select RoleName from Roles where User=?" ? est remplacé par le nom du principal en cours.
- **`suspendResume`**: Booléen - Suspend, puis termine la transaction associée au thread en cours tout en faisant une recherche de rôles.
- **`transactionManagerJndiName`**: nom JNDI du Gestionnaire de transactions (la valeur par défaut est java:/TransactionManager)

LdapRolesMappingProvider

Un fournisseur de mappage qui assigne les rôles à un utilisateur à l'aide d'un serveur LDAP pour chercher les rôles.

Options :

- **`bindDN`**: le DN utilisé pour faire la liaison avec le serveur LDAP pour les recherches de rôles et l'utilisateur. Ce DN a besoin de permissions lecture et recherche dans les valeurs de baseCtxDN et de rolesCtxDN.
- **`bindCredential`**: le mot de passe de bindDN. Peut être encodé si le jaasSecurityDomain est spécifié.
- **`rolesCtxDN`**: le DN fixe du contexte pour rechercher des rôles d'utilisateur. Ce n'est pas le DN où les rôles se trouvent, mais le DN où les objets contenant les rôles d'utilisateur se trouvent. Par exemple, dans un serveur Active Directory de Microsoft, c'est le DN où le compte d'utilisateur se trouve.
- **`roleAttributeID`**: l'attribut LDAP qui contient les noms des rôles d'autorisation.
- **`roleAttributeIsDN`** : indique si le **`roleAttributeID`** contient ou non le nom de domaine complet d'un objet de rôle. Si false, le nom de rôle est tiré de la valeur de l'attribut **`roleNameAttributeId`** du nom de contexte. Certains schémas de répertoire, tel que Active Directory, requièrent que cet attribut soit défini à **`true`**.
- **`roleNameAttributeID`** : nom de l'attribut qui se trouve dans le contexte **`roleCtxDN`** contenant le nom de rôle. Si la propriété **`roleAttributeIsDN`** est définie à **`true`**, cette propriété sera utilisée pour trouver l'attribut du nom de l'objet du rôle.
- **`parseRoleNameFromDN`**: un indicateur qui signale si le DN retourné par une requête contient le roleNameAttributeID. Si la valeur est définie à **`true`**, one cherchera le roleNameATtributeID du DN. Si la valeur est **`false`**, on ne cherchera pas le roleNameAttributeID du DN. Cet indicateur peut améliorer les performances des requêtes LDAP.
- **`roleFilter`** : un filtre de recherche permettant de localiser les rôles associés à l'utilisateur authentifié. L'entrée user ou **`userDN`** obtenue à partir du rappel de module de connexion est substitué dans le filtre à chaque fois qu'une expression **`{0}`** est utilisée. L'utilisateurDN **`userDN`** authentifié sera substitué dans le filtre à chaque fois qu'un **`{1}`** est utilisé. Un exemple de filtre de recherche qui correspond au nom d'utilisateur d'entrée serait **`(member={0})`**. Voici une alternative correspondant au **`userDN`** authentifié : **`(member={1})`**
- **`roleRecursion`** : le nombre de niveaux de récursivité de la recherche de rôle dans un contexte de correspondance donné. Désactiver la récursivité en attribuant le paramètre **`0`**.
- **`searchTimeLimit`**: la valeur d'expiration en millisecondes pour les recherches utilisateur/rôle. La valeur par défaut est 10000 (10 secondes).
- **`searchScope`**: l'étendue de recherche à utiliser.

PropertiesRolesMappingProvider

Un MappingProvider qui lit des rôles à partir d'un fichier de propriétés dans le format suivant : username=role1,role2,...

Options :

- **`rolesProperties`**: nom de fichier formaté de propriétés. L'expansion des variables de JBoss peuvent être utilisées sous la forme **`${jboss.variable}`**.

SimpleRolesMappingProvider

Simple MappingProvider qui lit des rôles à partir de la mappe d'options. Le nom d'attribut de l'option est le nom du principal à qui assigner des rôles et la valeur de l'attribut correspond aux noms de rôles séparés par des virgules à assigner au principal.

Exemple 12.3. Exemple

```
<module-option name="JavaDuke" value="JBossAdmin,Admin"/>
<module-option name="joe" value="Users"/>
```

org.jboss.security.mapping.providers.attribute.DefaultAttributeMappingProvider

Vérifie le module et cherche le nom de principal dans le contexte de mappage pour créer une adresse email d'attribut à partir d'une option de module nommée principalName + ".email" et la donner au principal donné.

LdapAttributeMappingProvider

Attributs de mappage de LDAP vers le sujet. Les options incluent les options que votre fournisseur LDAP JNDI prend en charge.

Exemple 12.4. Exemples de noms de propriété standard :

```
Context.INITIAL_CONTEXT_FACTORY = "java.naming.factory.initial"
Context.SECURITY_PROTOCOL = "java.naming.security.protocol"
Context.PROVIDER_URL = "java.naming.provider.url"
Context.SECURITY_AUTHENTICATION = "java.naming.security.authentication"
```

Options :

- **`bindDN`**: le DN utilisé pour faire la liaison avec le serveur LDAP pour les recherches de rôles et l'utilisateur. Ce DN a besoin de permissions lecture et recherche dans les valeurs de baseCtxDN et de rolesCtxDN.
- **`bindCredential`**: le mot de passe de bindDN. Peut être encodé si le jaasSecurityDomain est spécifié.
- **`baseCtxDN`** : le DN fixe de contexte pour commencer la recherche utilisateur
- **`baseFilter`**: un filtre de recherche permettant de localiser le contexte de l'utilisateur à authentifier. L'entrée username ou **`userDN`** obtenue à partir du rappel de module de connexion est substituée dans le filtre à chaque fois qu'une expression **`{0}`** est utilisée. Ce comportement de substitution vient de la méthode __**`DirContext.search(Name, String, Object[], SearchControls cons)`**__. **`(uid={0})`** est un exemple commun de filtre de recherche.
- **`searchTimeLimit`**: la valeur d'expiration en millisecondes pour les recherches utilisateur/rôle. La valeur par défaut est 10000 (10 secondes).

- **`attributeList`**: une liste d'attributs séparée par des virgules pour l'utilisateur. Exemple : mail,cn,sn,employeeType,employeeNumber.
- **`jaasSecurityDomain`** : Le JaasSecurityDomain à utiliser pour décrypter le **`java.naming.security.principal`**. La forme cryptée du mot de passe retourné par la méthode **`JaasSecurityDomain#encrypt64(byte[])`**. Vous pouvez également utiliser la classe **`org.jboss.security.plugins.PBEUtils`** pour générer la forme cryptée.

Rapporter un bogue

12.4. MODULES DE FOURNISSEURS D'AUDITING DE SÉCURITÉ INCLUS

JBoss EAP 6 contient un fournisseur d'auditing de sécurité.

Code	Class
LogAuditProvider	org.jboss.security.audit.providers.LogAuditProvider

Rapporter un bogue

CHAPITRE 13. CONFIGURATION DE SOUS-SYSTÈME

13.1. APERÇU DE LA CONFIGURATION DU SOUS-SYSTÈME

Introduction

JBoss EAP 6 utilise une configuration simplifiée, avec un fichier de configuration par domaine ou par serveur autonome. En mode de domaine, un fichier distinct existe pour chaque contrôleur hôte également. Les modifications apportées à la configuration persistent automatiquement, donc le XML ne doit pas être édité manuellement. La configuration est scannée et automatiquement remplacée par l'API de gestion. La ligne de commande de l'interface CLI et la console de gestion basée-web permettent de configurer chaque aspect de JBoss EAP 6.

JBoss EAP 6 repose sur le concept de chargement de classes modulaire. Chaque API ou service fourni par la plateforme est implémenté comme un module, qui est chargé et déchargé à la demande. La plupart des modules incluent un élément configurable appelé un sous-système. Les informations de configuration du sous-système sont stockées dans le fichier de configuration unifiée ***EAP_HOME*/domain/configuration/domain.xml** pour un domaine géré ou ***EAP_HOME*/standalone/configuration/standalone.xml** pour un serveur autonome. La plupart des sous-systèmes incluent les détails de configuration configurés par l'intermédiaire de descripteurs de déploiements dans les versions précédentes de JBoss EAP.

Schémas de configuration du sous-système

Chaque configuration de sous-système est définie dans un schéma XML. Les schémas de configuration se trouvent dans le répertoire ***EAP_HOME*/docs/schema/** de votre installation.

Les sous-systèmes suivants sont connus comme *sous-systèmes simples*, parce qu'ils n'ont pas d'attributs ou d'éléments configurables. Ils sont généralement répertoriés en haut du fichier de configuration.

Sous-systèmes simples

- **ee**– l'implémentation Java EE 6 API
- **ejb**– le sous-système d'Enterprise JavaBeans (EJB)
- **jaxrs**– l'API JAX-RS API, fourni par RESTeasy.
- **sar**– le sous-système qui supporte Service Archives.
- **threads**– le sous-système qui supporte le traitement des threads.
- **weld**– l'API Contexts and Dependency Injection, fourni par Weld.

Rapporter un bogue

CHAPITRE 14. LE SOUS-SYSTÈME DE JOURNALISATION

14.1. INTRODUCTION

14.1.1. Logging (Journalisation)

JBoss EAP 6 fournit des fonctionnalités de journalisation hautement configurables pour son propre usage et pour utilisation par des applications déployées. Le sous-système d'enregistrement est basé sur JBoss LogManager et prend en charge plusieurs frameworks de journalisation d'applications de tierce partie en plus du JBoss Logging.

Le sous-système de journalisation est configuré à l'aide d'un système de catégories de journaux et de gestionnaires de journaux. Les catégories de journalisation définissent quels messages capturer et les gestionnaires de journaux définissent comment procéder avec ces messages (écriture sur disque, envoyer à la console, etc.).

Le profils de journalisation permettent à des configurations de journalisation possédant un nom unique d'être créées et assignées à des applications indépendamment de toute autre configuration de journalisation. La configuration des profils de journalisation est presque identique pour le sous-système de journalisation principal.

Rapporter un bogue

14.1.2. Frameworks de journalisations d'applications pris en charge par JBoss LogManager

JBoss LogManager prend en charge les frameworks de journalisation suivants :

- JBoss Logging - inclus avec JBoss EAP 6
- Apache Commons Logging - http://commons.apache.org/logging/
- Simple Logging Facade Java (SLF4J) - http://www.slf4j.org/
- Apache log4j - http://logging.apache.org/log4j/1.2/
- Java SE Logging (java.util.logging) - http://download.oracle.com/javase/6/docs/api/java/util/logging/package-summary.html

Rapporter un bogue

14.1.3. Configuration du journal d'amorçage

La journalisation d'amorçage enregistre des événements qui ont lieu quand le serveur "est amorcé" ou démarre.

Si le fichier **`logging.properties`** est disponible quand le serveur démarre, ces propriétés de configuration seront utilisées pour enregistrer des événements qui ont eu lieu avant que le sous-système de journalisation ne soit initialisé. Ensuite, le sous-système de journalisation s'occupe des enregistremetns d'événements.

Quand vous modifiez le sous-système **`logging`** par le CLI, ou en éditant manuellement le fichier de configuration du serveur, il met à jour le fichier **`logging.properties`**.

Si le fichier **`logging.properties`** est manquant dans l'installation, tous les messages de journalisation qui apparaissent normalement au boot time avant que le sous-système de journalisation ne soit activé, sont perdus. Une fois que le sous-système est initialisé, les messages apparaitront à nouveau sur le journal.

AVERTISSEMENT

Il est recommandé de ne pas modifier le fichier **`logging.properties`** directement à moins que vous n'ayiez un sérieux problème à démarrer le serveur et que vous ayiez besoin d'enregistrements supplémentaires de l'hôte ou du contrôleur de processus.

Rapporter un bogue

14.1.4. Journalisation de Garbage Collection

La journalisation de Garbage collection consigne toutes les activités de collecte de la poubelle dans les fichiers journaux en texte clair. Ces fichiers journaux peuvent être utiles à des fins de diagnostique. À partir de JBoss EAP 6.3, la journalisation de la Garbage collection est activée par défaut en mode **`standalone`** (autonome) sur toutes les configurations prises en charge *sauf* JDK d'IBM.

La journalisation correspond à la sortie du fichier ***`$EAP_HOME`*`/standalone/log/gc.log.`*`digit`**. La rotation de la journalisation est activée, avec un nombre de fichiers de journalisation ne devant pas dépasser 5, et une taille de 3 MiB maximum par fichier.

Rapporter un bogue

14.1.5. Dépendances d'API de journalisation implicites

Le sous-système de journalisation de JBoss EAP 6 possède un attribut **`add-logging-api-dependencies`** qui contrôle la possibilité pour le conteneur d'ajouter des dépendances d'API de journalisation implicites aux déploiements. Par défaut, cet attribut est défini à **`true`**, ce qui signifie que toutes les dépendances d'API de journalisation implicites doivent être ajoutées aux déploiements. Si défini sur **`false`**, les dépendances d'API de journalisation se seront pas ajoutées.

L'attribut **`add-logging-api-dependencies`** peut être configuré par l'interface CLI. Par exemple :

```
/subsystem=logging:write-attribute(name=add-logging-api-dependencies,
value=false)
```

Rapporter un bogue

14.1.6. Emplacements de fichiers de journalisation par défaut

Il s'agit des fichiers de journalisation qui ont été créés pour les configurations de journalisation par défaut. La configuration par défaut écrit des fichiers de journalisation du serveur à l'aide de gestionnaires de journaux périodiques.

Tableau 14.1. Fichier de journalisation par défaut d'un serveur autonome

Fichier journal	Description
***EAP_HOME*/standalone/log/server.log**	Journal du serveur. Contient les messages de journalisation de serveur, dont les messages de démarrage de serveur.
***EAP_HOME*/standalone/log/gc.log**	Journalisation de Garbage collection. Contient des informations sur tous les nettoyagess de mémoire.

Tableau 14.2. Fichiers de journalisation par défaut d'un domaine géré

Fichier journal	Description
***EAP_HOME*/domain/log/host-controller.log**	Journal d'amorçage du contrôleur hôte. Contient les messages de journalisation liés au démarrage du contrôleur hôte.
***EAP_HOME*/domain/log/process-controller.log**	Journal d'amorçage du contrôleur de processus. Contient les messages de journalisation liés au démarrage du contrôleur de processus.
***EAP_HOME*/domain/servers/*SERVERNAME*/log/server.log**	Le journal du serveur pour le serveur nommé. Contient les messages de journalisation de ce serveur, dont les messages de démarrage de serveur.

Rapporter un bogue

14.1.7. Filtre les expressions de journalisation

NOTE

Un **filter-spec** spécifié pour le root logger n'est *pas* hérité par d'autres handlers. À la place, un **filter-spec** doit être spécifié par handler.

Tableau 14.3. Filtre les expressions de journalisation

Type de filtre **expression**	Description	Paramètres
Accepter **accept**	Accepter tous les messages de journalisation	**accept**
Refuser **deny**	Refuser tous les messages de journalisation	**deny**

Type de filtre `expression`	Description	Paramètres
Not **`not[filter expression]`**	Renvoie la valeur inversée de l'expression du filtre	Prend une expression de filtre unique comme paramètre **`not(match("JBAS"))`**
Tout **`all[filter expression]`**	Renvoie la valeur concaténée à partir de multiples expressions de filtre.	Prends plusieurs expressions de filtre séparées par des virgules **`all(match("JBAS"),match("WELD"))`**
Tous **`any[filter expression]`**	Renvoie une valeur à partir de multiples expressions de filtre.	Prends plusieurs expressions de filtre séparées par des virgules **`any(match("JBAS"),match("WELD"))`**
Changement de niveau **`levelChange[level]`**	Modifie l'enregistrement de journalisation avec le niveau indiqué	Prend un niveau basé chaîne unique comme argument **`levelChange("WARN")`**
Niveaux **`levels[levels]`**	Filtre les messages de journalisation avec un niveau apparaissant sur la liste des niveaux	Prend plusieurs niveaux basés chaînes séparées par des virgules en tant qu'arguments **`levels("DEBUG","INFO","WARN","ERROR")`**

Type de filtre `expression`	Description	Paramètres
Gamme de niveaux `levelRange[minLevel,maxLevel]`	Filtre les messages de journalisation dans la gamme de niveaux spécifiée.	L'expression de filtre utilise **[** pour indiquer un niveau inclusif minimum et **]** pour indiquer un niveau inclusif maximum. Sinon, vous pouvez utiliser **(** ou **)** respectivement pour indiquer une exclusivité. Le premier argument de l'expression est le niveau minimum autorisé, le deuxième argument est le niveau maximum autorisé. Voici des exemples ci-dessous. • **`levelRange("DEBUG","ERROR")`** Le niveau minimum doit être supérieur à **DEBUG** et le niveau maximum doit être inférieur à **ERROR**. • **`levelRange["DEBUG","ERROR")`** Le niveau minimum doit être supérieur ou égal à **DEBUG** et le niveau maximum doit être inférieur à **ERROR**. • **`levelRange["INFO","ERROR"]`** Le niveau minimum doit être supérieur ou égal à **INFO** et le niveau maximum doit être inférieur ou égal à **ERROR**.
Match (**`match["pattern"]`**)	Filtre basé sur une expression régulière. Le message non formaté est utilisé à l'encontre du modèle spécifié dans l'expression.	Prend une expression régulière comme argument **`match("JBAS\d+")`**
Substitute (**`substitute["pattern","replacement value"]`**)	Filtre qui remplace la première correspondance au modèle par une valeur de remplacement	Le premier argument de l'expression est le modèle, le deuxième est le texte de remplacement. **`substitute("JBAS","EAP")`**

Type de filtre `expression`	Description	Paramètres
Remplacer tout (**`substituteAll["pattern","replacement value"]`**)	Un filtre qui remplace toutes les correspondances du modèle avec la valeur de remplacement.	Le premier argument de l'expression est le modèle, le deuxième est le texte de remplacement. **`substituteAll("JBAS","EAP")`**

Rapporter un bogue

14.1.8. Niveaux de journalisation

Les niveaux de journalisation sont des ensembles ordonnés de valeurs énumérées qui indiquent la nature et la sévérité d'un message de journalisation. Le niveau d'un message de journalisation donné est indiqué par le développeur par des méthodes qui conviennent dans un framework de journalisation particulier pour envoyer le message.

JBoss EAP 6 accepte tous les niveaux de journalisation utilisés par les frameworks de journalisation de l'application prise en charge. Les six niveaux de journalisation les plus utilisés sont (dans l'ordre croissant) : **`TRACE`**, **`DEBOG`**, **`INFO`**, **`ATTENTION`**, **`ERREUR`** et **`FATAL`**.

Les niveaux de journalisation sont utilisés par des catégories et gestionnaires de journalisation pour limiter les messages dont ils sont responsables. Chaque niveau de journalisation possède une valeur numérique qui indique son ordre par rapport à d'autres niveaux de journalisation. Les catégories et gestionnaires de journalisation correspondent à un certain niveau de journalisation, et ils traitent les messages de journalisation du même niveau ou d'un niveau supérieur uniquement. Par exemple, un gestionnaire de journalisation du niveau **`ATTENTION`** enregistrera uniquement les messages des niveaux **`ATTENTION`**, **`ERREUR`** et **`FATAL`**.

Rapporter un bogue

14.1.9. Niveaux de journalisation pris en charge

Tableau 14.4. Niveaux de journalisation pris en charge

Niveau de journalisation	Valeur	Description
FINESSE MAX	300	-
PLUS FIN	400	-
TRACE	400	Utilisé pour des messages qui fournissent des informations détaillées sur l'état d'exécution d'une application. Les messages de journalisation **TRACE** sont habituellement capturés lors du débogage d'une application uniquement.

Niveau de journalisation	Valeur	Description
DEBOG	500	Utilisé pour des messages qui indiquent des demandes individuelles de progrès ou des activités d'une application. Les messages de journalisation **DEBUG** ne sont habituellement capturés que lors du débogage d'une application.
FINESSE	500	-
CONFIG	700	-
INFO	800	Utilisé pour des messages qui indiquent la progression globale de l'application. Souvent utilisé pour le démarrage de l'application, la fermeture et autres événements majeurs de cycle de vie.
AVERTISSEMENT	900	Utilisé pour indiquer une situation qui n'est pas en erreur, mais n'est pas considérée comme idéale. Peut indiquer des circonstances qui peuvent entraîner des erreurs dans le futur.
ATTENTION	900	-
ERREUR	1000	Utiliser pour indiquer une erreur qui s'est produite et qui puisse empêcher l'activité actuelle ou la demande de se remplir, mais qui n'empêchera pas l'application d'exécuter.
SÉVÈRE	1000	-
FATAL	1100	Utiliser pour indiquer les événements qui pourraient entraîner des défaillances de services critiques ou la fermeture de l'application, ou qui pourraient entraîner la fermeture de la plateforme JBoss EAP 6.

Rapporter un bogue

14.1.10. Catégories de journalisation

Les catégories de journalisation définissent les messages de journalisation à acquérir et un ou plusieurs gestionnaires de journalisation qui traitent les messages.

Les messages de journalisation à capturer sont définis par leur package Java d'origine et leur niveau de journalisation. Les messages de classes de ce package et de niveau de journalisation ou niveau inférieur sont capturés par la catégorie de journalisation et envoyés aux gestionnaires de journal spécifiés.

Les catégories ont la possibilité d'utiliser les gestionnaires de journalisation du root logger au lieu de leurs propres gestionnaires.

Rapporter un bogue

14.1.11. Root Logger

Le root logger capture tous les messages de journalisation qui sont envoyés au serveur (à un niveau indiqué) et qui ne sont pas capturés par une catégorie de journalisation particulière. Ces messages sont alors envoyés à un ou à plusieurs gestionnaires de journalisation.

Par défaut, le root logger est configuré pour utiliser une console et un gestionnaire de journalisation périodique. Le gestionnaire de journalisation périodique est configuré pour écrire sur le ficher **`server.log`**. On prénomme parfois ce fichier : journal du serveur (server log).

Rapporter un bogue

14.1.12. Gestionnaires de journaux

Les gestionnaires de journaux définissent la façon dont les messages de journalisation sont enregistrés dans JBoss EAP. Il existe six types de gestionnaires de journalisation configurables : **`Console`**, **`File`**, **`Periodic`**, **`Size`**, **`Async`** et **`Custom`**.

Rapporter un bogue

14.1.13. Types de gestionnaires de journalisation

Console

Les gestionnaires de journaux de console écrivent des messages de journalisation soit dans le système d'exploitation hôte (stdout) ou dans le flux d'erreurs standard (stderr). Ces messages sont affichés lorsque JBoss EAP 6 est exécuté à partir d'une invite de ligne de commande. Les messages d'un gestionnaire de journal de console ne sont pas enregistrés à moins que le système d'exploitation ne soit spécifiquement configuré pour capturer stdout ou stderr.

Fichier

Les gestionnaires de journaux de fichiers sont les gestionnaires de journalisation les plus simples, qui écrivent les messages de journalisation dans un fichier spécifique.

Périodique

Les gestionnaires de journaux périodiques écrivent des messages de journalisation dans un fichier nommé jusqu'à ce qu'une certaine durée se soit écoulée. Une fois que cette période a expiré, le fichier est nommé à nouveau en rajoutant l'horodatage et le gestionnaire continue d'écrire dans un fichier de journalisation nouvellement créé avec le nom d'origine.

Taille

Les gestionnaires de journaux de taille écrivent les messages de journalisation dans un fichier jusqu'à ce que le fichier atteigne une taille spécifiée. Lorsque le fichier atteint une taille donnée, il est renommé avec un préfixe numérique et le gestionnaire continue d'écrire dans un fichier journal récemment créé avec le nom d'origine. Chaque gestionnaire de journaux de taille doit spécifier le nombre maximal de fichiers contenus de cette façon.

Async

Les gestionnaires de journaux async sont des gestionnaires de journaux wrapper qui fournissent un comportement asynchrone pour un ou plusieurs autres gestionnaires de journaux. Ils sont utiles pour les gestionnaires de journaux qui pourraient avoir une latence élevée ou autres problèmes de performances comme l'écriture d'un fichier journal à un système de fichiers réseau.

Personnalisé

Les gestionnaires d'informations personnalisées vous permettent de configurer de nouveaux types de gestionnaires de journaux mis en place. Un gestionnaire personnalisé doit être implémenté comme classe Java qui s'étend **`java.util.logging.Handler`** et doit être contenu dans un module.

syslog

Les gestionnaires de syslog peuvent être utilisés pour envoyer des messages à un serveur de journalisation à distance. Cela permet à plusieurs applications d'envoyer leurs messages de journalisation au même serveur, où ils peuvent être analysés en même temps.

Rapporter un bogue

14.1.14. Log Formatters

Un formateur de journalisation (log formatter) est une propriété de configuration d'un gestionnaire de journalisation qui détermine l'apparence des messages de journalisation. Il s'agit d'un string qui utilise une syntaxe basée sur la classe **`java.util.Formatter`**.

Ainsi, le string du formateur de journalisation de la configuration par défaut, **`%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n`**, crée des messages de journalisation qui ressemblent à ceci :

```
15:53:26,546 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015951:
Admin console listening on http://127.0.0.1:9990
```

Rapporter un bogue

14.1.15. Syntaxe de Formateur de journaux

Tableau 14.5. Syntaxe de Formateur de journaux

Symbole	Description
%c	La catégorie de l'événement de journalisation
%p	Le niveau de saisie de la journalisation (info/déboggage/etc)
%P	Le niveau localisé de la saisie de journalisation
%d	Les date/heure (yyyy-MM-dd HH:mm:ss,SSS form)
%r	L'heure relative (en millisecondes depuis l'initialisation de la journalisation)
%z	Le réseau horaire
%k	Une clé de ressource de journalisation (utilisée pour la localisation de messages de journalisation)
%m	Le message de journalisation (avec trace d'exception)
%s	Le simple message de journalisation (sans trace d'exception)

Symbole	Description
%e	Suivi de la pile d'exceptions (sans informations sur les modules étendus)
%E	Suivi de la pile d'exceptions (avec informations sur les modules étendus)
%t	Le nom du thread en cours
%n	Un caractère de nouvelle ligne
%C	La classe du code appelant la méthode de journalisation (lente)
%F	Le nom de fichier de la classe appelant la méthode de journalisation (lente)
%l	L'emplacement d'origine du code appelant la méthode de journalisation (lente)
%L	Le numéro de ligne du code appelant la méthode de journalisation (lente)
%M	La méthode du code appelant la méthode de journalisation (lente)
%x	Contexte de diagnostique intégré
%X	Contexte de diagnostique du message
%%	Un pourcentage (caractère d'échappement)

Rapporter un bogue

14.2. CONFIGURER LA JOURNALISATION PAR LA CONSOLE DE GESTION

La console de gestion fournit une interface graphique utilisateur pour la configuration du root logger, des log handlers et des catégories de journalisation. Vous pourrez trouver la configuration du logger dans la console de gestion en suivant les étapes suivantes :

Vous pourrez accéder à cette configuration en suivant les étapes suivantes :

1. Connectez-vous à la console de gestion
2. Naviguez dans la configuration du sous-système de logging. Cette étape varie suivant qu'il s'agisse de serveurs qui s'exécutent sur les serveurs autonomes ou des serveurs exécutant dans un domaine géré.
    - **Serveur autonome**
      Cliquer sur **Configuration**, étendre **Core** dans le menu **Subsystems**, et cliquer sur **Logging**.
    - **Domaine géré**

Cliquer sur **Configuration**, sélectionner le profil que vous souhaitez modifier dans le menu déroulant. Étendre **Core** dans le menu **Subsystems**, et cliquer sur **Logging**.

Les tâches à effectuer pour configurer le root logger sont les suivantes :

- Modifier le niveau de journalisation.
- Ajouter et supprimer des log handlers.

Les tâches à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter et supprimer les catégories de journalisation.
- Modifier les propriétés de catégories.
- Ajouter et supprimer les log handlers d'une catégorie.

Les tâches principales à effectuer pour configurer les log handlers sont les suivantes :

- Ajouter de nouveaux handlers.
- Configurer les handlers.

Les six log handlers (y compris le handler personnalisé) peuvent être configurés dans la console de gestion.

Rapporter un bogue

14.3. CONFIGURATION DE LOGGING DANS LE CLI

Conditions préalables

L'interface CLI doit exécuter et est connectée à l'instance JBoss EAP qui convient. Pour plus d'informations, voir Section 3.5.2, « Lancement de l'interface CLI »

Rapporter un bogue

14.3.1. Configurer le root logger par le CLI

La configuration du root logger peut s'afficher ou être modifiée par le CLI.

Les tâches principales à effectuer pour configurer le root logger sont les suivantes :

- Ajouter des log handler au root logger.
- Afficher la configuration du root logger.
- Modifier le niveau de journalisation.
- Supprimer des log handler du root logger.

IMPORTANT

Pour la configuration du root logger dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

Ajouter un log handler au root logger

Utiliser l'opération **`add-handler`** avec la syntaxe suivante *HANDLER* dans le nom du gestionnaire de journalisation (Log Hangler) à ajouter.

```
/subsystem=logging/root-logger=ROOT:add-handler(name="HANDLER")
```

Le log handler doit déjà avoir été créé avant de l'ajouter au root handler.

Exemple 14.1. Opération root logger add-handler

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:add-handler(name="FILE")
{"outcome" => "success"}
```

Afficher le contenu de la configuration du root logger

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/root-logger=ROOT:read-resource
```

Exemple 14.2. Opération root logger «read-resource»

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:read-resource
{
   "outcome" => "success",
   "result" => {
        "filter" => undefined,
        "filter-spec" => undefined,
        "handlers" => [
            "CONSOLE",
            "FILE"
      ],
      "level" => "INFO"
   }
}
```

Définir le niveau de journalisation du root logger

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante avec *LEVEL* indiquant les niveaux de journalisation pris en charge.

```
/subsystem=logging/root-logger=ROOT:write-attribute(name="level",
value="LEVEL")
```

Exemple 14.3. L'opération «write-attribute» du root logger pour définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:write-attribute(name="level", value="DEBUG")
{"outcome" => "success"}
```

Supprimer un log handler du root logger

Utiliser **`remove-handler`** avec la syntaxe suivante, avec *HANDLER* comme nom du gestionnaire de journalisation à supprimer.

```
/subsystem=logging/root-logger=ROOT:remove-handler(name="HANDLER")
```

Exemple 14.4. Supprimer un log handler

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:remove-handler(name="FILE")
{"outcome" => "success"}
```

Rapporter un bogue

14.3.2. Configurer une Catégorie dans l'interface CLI

Les catégories de journalisation peuvent être ajoutées, supprimées et modifiées dans le CLI.

Les tâches principales à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter une nouvelle catégorie de journalisation :
- Afficher la configuration d'une catégorie de journalisation.
- Définir un niveau de journalisation.
- Ajouter des log handlers à une catégorie de journalisation.
- Supprimer des log handlers d'une catégorie de journalisation.
- Supprimer une catégorie de journalisation.

IMPORTANT

Pour la configuration d'une catégorie de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

Ajouter une catégorie de journalisation

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *CATEGORY* par la catégorie à ajouter.

```
/subsystem=logging/logger=CATEGORY:add
```

Exemple 14.5. Ajouter une nouvelle catégorie de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:add
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de catégorie de journalisation

Utiliser l'opération **`read-resource`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie.

```
/subsystem=logging/logger=CATEGORY:read-resource
```

Exemple 14.6. Opération de lecture de ressource de la catégorie de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=org.apache.tomcat.util.modeler:read-
resource
{
    "outcome" => "success",
    "result" => {
        "category" => "org.apache.tomcat.util.modeler",
        "filter" => undefined,
        "filter-spec" => undefined,
        "handlers" => undefined,
        "level" => "WARN",
        "use-parent-handlers" => true
    }
}
[standalone@localhost:9999 /]
```

Définir le niveau de journalisation

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie de journalisation et *LEVEL* par le niveau de journalisation à définir.

```
/subsystem=logging/logger=CATEGORY:write-attribute(name="level",
value="LEVEL")
```

Exemple 14.7. Définir un niveau de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la Catégorie de journalisation pour utiliser le log handler du root logger.

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie de journalisation. Remplacer *BOOLEAN* par true pour cette catégorie de

journalisation pour utiliser les handlers du root logger. Le remplacer par false s'il doit utiliser ses propres handlers.

```
/subsystem=logging/logger=CATEGORY:write-attribute(name="use-parent-
handlers", value="BOOLEAN")
```

Exemple 14.8. Configurer use-parent-handlers

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="use-parent-handlers", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Ajouter un log handler à une catégorie de journalisation.

Utiliser l'opération **`add-handler`** avec la syntaxe suivante. Remplacer *CATEGORY* par la catégorie à ajouter et *HANDLER* par le nom du handler à ajouter.

```
/subsystem=logging/logger=CATEGORY:add-handler(name="HANDLER")
```

Le log handler doit déjà avoir été créé avant de l'ajouter au Root Handler.

Exemple 14.9. Ajouter un log handler

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:add-
handler(name="AccountsNFSAsync")
{"outcome" => "success"}
```

Supprimer un log handler d'une catégorie de journalisation

Utiliser l'opération **`remove-handler`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie à *HANDLER* par le nom du log handler à supprimer.

```
/subsystem=logging/logger=CATEGORY:remove-handler(name="HANDLER")
```

Exemple 14.10. Supprimer un log handler

```
[standalone@localhost:9999 /] /subsystem=logging/logger=jacorb:remove-
handler(name="AccountsNFSAsync")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer une catégorie

Utiliser l'opération **`remove`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie à supprimer.

```
/subsystem=logging/logger=CATEGORY:remove
```

Exemple 14.11. Supprimer une catégorie de journalisation

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

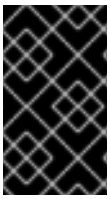
Rapporter un bogue

14.3.3. Configurer un log handler de console dans le CLI

Les log handlers de console peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler de console :

- Ajouter un nouveau log handler de console.
- Afficher la configuration d'un log handler de console.
- Définir le niveau de journalisation du handler.
- Définir la cible de la sortie du handler.
- Définir la codification utilisée pour la sortie du handler.
- Définir le formateur utilisé pour la sortie du handler.
- Définir si le handler utilise autoflush ou non.
- Supprimer un handler de journalisation de console.



IMPORTANT

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

Ajouter un log handler de console

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log haandler de journalisation de la console à ajouter.

```
/subsystem=logging/console-handler=HANDLER:add
```

Exemple 14.12. Ajouter un log handler de console

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:add
{"outcome" => "success"}
```

Afficher une configuration de log handler de journalisation de la console

Utiliser l'opération **`read-resource`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console.

```
/subsystem=logging/console-handler=HANDLER:read-resource
```

Exemple 14.13. Afficher une configuration de log handler de journalisation de la console

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=CONSOLE:read-resource
{
    "outcome" => "success",
    "result" => {
        "autoflush" => true,
        "enabled" => true,
        "encoding" => undefined,
        "filter" => undefined,
        "filter-spec" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => "INFO",
        "name" => "CONSOLE",
        "named-formatter" => "COLOR-PATTERN",
        "target" => "System.out"
    }
}
```

Définir le niveau de journalisation

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console et *LEVEL* par le niveau de journalisation à définir.

```
/subsystem=logging/console-handler=HANDLER:write-attribute(name="level",
value="INFO")
```

Exemple 14.14. Définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="level", value="TRACE")
{"outcome" => "success"}
```

Définir la cible

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console et *TARGET* par **`System.err`** ou **`System.out`** pour le flux Erreurs système ou le flux Standard out.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="target", value="TARGET")
```

Exemple 14.15. Définir la cible

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="target",
value="System.err")
{"outcome" => "success"}
```

Définir le codage

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console et *ENCODING* par le nom de codification du caractère qui convient.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Exemple 14.16. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Définir le formateur

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console et *FORMAT* par le string de formateur requis.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Exemple 14.17. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
```

Définir auto flush

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console et *BOOLEAN* par **`true`** si le handler doit écrire sa sortie immédiatement.

```
/subsystem=logging/console-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Exemple 14.18. Définir auto flush

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
```

Supprimer un log handler de console

Utiliser l'opération **`remove`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de la console à supprimer.

```
/subsystem=logging/console-handler=HANDLER:remove
```

Exemple 14.19. Supprimer un log handler de console

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:remove
{"outcome" => "success"}
```

Rapporter un bogue

14.3.4. Configurer un log handler de fichiers dans le CLI

Les log handlers de fichiers peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler de fichiers :

- Ajouter un nouveau log handler de fichiers.
- Afficher la configuration d'un log handler de fichiers
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Supprimer un log handler de fichiers.

IMPORTANT

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

Ajouter un log handler de fichiers

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

```
 /subsystem=logging/file-handler=HANDLER:add(file={"path"=>"PATH",
"relative-to"=>"DIR"})
```

Exemple 14.20. Ajouter un log handler de fichiers

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:add(file={"path"=>"accounts.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de log handler de fichiers

Utiliser l'opération **read-resource** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers.

```
 /subsystem=logging/file-handler=HANDLER:read-resource
```

Exemple 14.21. Utiliser l'opération read-resource

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "accounts.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined
    }
}
```

Définir le niveau de journalisation

Utiliser l'opération **write-attribute** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

```
 /subsystem=logging/file-handler=HANDLER:write-attribute(name="level",
value="LOG_LEVEL_VALUE")
```

Exemple 14.22. Changer le niveau de journalisation

```
/subsystem=logging/file-handler=accounts_log:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le comportement d'ajout

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers. Remplacer *BOOLÉEN* par false si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

```
 /subsystem=logging/file-handler=HANDLER:write-attribute(name="append",
value="BOOLEAN")
```

Exemple 14.23. Changer la propriété d'ajout

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Définir auto flush

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *BOOLEAN* par **`true`** si le handler doit écrire sa sortie immédiatement.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Exemple 14.24. Changer la propriété autoflush

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="autoflush", value="false")
{
    "outcome" => "success",
    "response-headers" => {"process-state" => "reload-required"}
}
[standalone@localhost:9999 /]
```

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Définir le codage

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *ENCODING* par le nom de codification du caractère qui convient.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Exemple 14.25. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

Changer le fichier dans lequel le log handler écrit.

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *PATH* par le nom du fichier du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

```
 /subsystem=logging/file-handler=HANDLER:write-attribute(name="file",
value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Exemple 14.26. Changer le fichier dans lequel le log handler écrit.

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="file", value=
{"path"=>"accounts-debug.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le formateur

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers et *FORMAT* par le string de formateur requis.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Exemple 14.27. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts-log:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler de fichiers.

Utiliser l'opération **remove** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du log handler de fichiers à supprimer.

```
/subsystem=logging/file-handler=HANDLER:remove
```

Exemple 14.28. Supprimer un log handler de fichiers.

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Un log handler ne peut être supprimé que s'il ne peut pas être référencé par une catégorie de journalisation ou par un log handler async.

Rapporter un bogue

14.3.5. Configurer un log handler périodique dans le CLI

Les log handlers périodiques peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un log handler périodique :

- Ajouter un nouveau log handler périodique.
- Afficher la configuration d'un log handler périodique
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir le suffixe pour les journaux en rotation
- Supprimer un log handler périodique

Chacune de ces actions sont décrites ci-dessous.

IMPORTANT

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **/subsystem=logging/logging-profile=NAME/** au lieu de **/subsystem=logging/**.

Ajouter un nouveau log handler périodique en rotation.

Utiliser l'opération **add** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:add(file=
{"path"=>"PATH", "relative-to"=>"DIR"}, suffix="SUFFIX")
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin. Remplacer *SUFFIX* par le suffixe de rotation de fichiers à utiliser.

Exemple 14.29. Créer un nouvel handler

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:add(file={"path"=>"daily-debug.log",
"relative-to"=>"jboss.server.log.dir"}, suffix=".yyyy.MM.dd")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Afficher une configuration de log handler de fichiers en rotation périodique

Utiliser l'opération **read-resource** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 14.30. Utiliser l'opération read-resource

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "daily-debug.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined
    }
}
```

Définir le niveau de journalisation

Utiliser l'opération **write-attribute** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="level". value="LOG_LEVEL_VALUE")
```

Remplacer *HANDLER* par le nom du log handler périodique et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 14.31. Définir le niveau de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="level",
value="DEBUG")
{"outcome" => "success"}
```

Définir le comportement d'ajout

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler périodique. Remplacer *BOOLÉEN* par **`false`** si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 14.32. Définir le comportement d'ajout

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

Définir auto flush

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler périodique et *BOOLEAN* par **`true`** si le handler doit écrire sa sortie immédiatement.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 14.33. Définir le comportement auto flush

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="autoflush",
value="false")
{
    "outcome" => "success",
    "response-headers" => {"process-state" => "reload-required"}
}
```

Définir le codage

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du log handler périodique et *ENCODING* par le nom de codification du caractère qui convient.

Exemple 14.34. Définir le codage

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="encoding", value="utf-
8")
{"outcome" => "success"}
```

Changer le fichier dans lequel le log handler écrit.

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="file", value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du log handler périodique. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 14.35. Changer le fichier dans lequel le log handler écrit.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="file", value=
{"path"=>"daily-debug.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
```

Définir le formateur

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Remplacer *HANDLER* par le nom du log handler périodique et *FORMAT* par le string de formateur à définir.

Exemple 14.36. Définir le formateur

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le suffixe pour les journaux en rotation

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="suffix", value="SUFFIX")
```

Remplacer *HANDLER* par le nom du log handler et *SUFFIX* par le string de suffixe à définir.

Exemple 14.37.

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="suffix", value=".yyyy-
MM-dd-HH")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler périodique

Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler périodique.

Exemple 14.38. Supprimer un log handler périodique

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

14.3.6. Configurer un log handler de taille dans le CLI

Les log handlers de taille de fichiers en rotation peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches qui vous reviendront pour configurer un log handler de taille de fichiers en rotations :

- Ajouter un nouveau log handler
- Afficher la configuration du log handler
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le log handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir la taille maximum de chaque fichier de journalisation
- Définir le nombre maximum de journaux de sauvegarde à conserver
- Définit l'option de démarrage de la rotation à l'amorçage pour le log handler de taille de fichiers
- Supprimer un log handler.

Chacune de ces tâches sont décrites ci-dessous.

IMPORTANT

Pour la configuration d'un log handler dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

Ajouter un nouveau log handler

Utiliser l'opération **add** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:add(file=
{"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 14.39. Ajouter un nouveau log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:add(file={"path"=>"accounts_trace.log",
"relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
```

Afficher la configuration du log handler

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 14.40. Afficher la configuration du log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "accounts_trace.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined,
        "max-backup-index" => 1,
        "rotate-size" => "2m"
    }
}
[standalone@localhost:9999 /]
```

Définir le niveau de journalisation du handler

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:write-
attributel(name="level", value="LOG_LEVEL_VALUE")
```

Remplacer *HANDLER* par le nom du log handler et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 14.41. Définir le niveau de journalisation du handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="level", value="TRACE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le comportement d'ajout du handler.

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *BOOLÉEN* par **`false`** si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 14.42. Définir le comportement d'ajout du handler.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

Définir si le handler utilise autoflush ou non

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler et *BOOLEAN* par **`true`** si le handler doit écrire sa sortie immédiatement.

Exemple 14.43. Définir si le handler utilise autoflush ou non

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la codification utilisée pour la sortie du handler

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du log handler et *ENCODING* par le nom de codification du caractère qui convient.

Exemple 14.44. Définir la codification utilisée pour la sortie du handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}]
```

Indiquer le fichier dans lequel le log handler écrit

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="file", value={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du log handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

Exemple 14.45. Indiquer le fichier dans lequel le log handler écrit

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="file", value=
{"path"=>"accounts_trace.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
```

Définir le formateur utilisé pour la sortie du handler.

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="formatter", value="FORMATTER")
```

Remplacer *HANDLER* par le nom du log handler et *FORMAT* par le string de formateur à définir.

Exemple 14.46. Définir le formateur utilisé pour la sortie du handler.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p (%c) [%t] %s%E%n")
{"outcome" => "success"}
```

Définir la taille maximum de chaque fichier de journalisation

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="rotate-size", value="SIZE")
```

Remplacer *HANDLER* par le nom du log handler et *SIZE* par la taille de fichier maximum.

Exemple 14.47. Définir la taille maximum de chaque fichier de journalisation

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="rotate-size",
value="50m")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir le nombre maximum de journaux de sauvegarde à conserver

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="max-backup-index", value="NUMBER")
```

Remplacer *HANDLER* par le nom du log handler et *NUMBER* par le nombre de fichiers de journalisation à conserver.

Exemple 14.48. Définir le nombre maximum de journaux de sauvegarde à conserver

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="max-backup-index",
value="5")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définit l'option de rotation au démarrage du `size-rotating-file-handler`

Cette option n'est disponible que pour le gestionnaire de fichiers **`size-rotating-file-handler`**. Une valeur par défaut de **`false`** indique qu'un nouveau fichier de journalisation n'est pas créé au redémarrage du serveur.

Pour changer cela, utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="rotate-on-boot", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du log handler **`size-rotating-file-handler`**. Remplacer *BOOLEAN* par **`true`** si un nouveau fichier de journalisation **`size-rotating-file-handler`** doit être créé au redémarrage.

Exemple 14.49. Indique qu'il faut créer un nouveau fichier de journalisation `size-rotating-file-handler` lors du redémarrage du serveur.

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="rotate-on-boot",
value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un log handler

Utiliser l'opération **remove** avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 14.50. Supprimer un log handler

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:remove
{"outcome" => "success"}
```

Rapporter un bogue

14.3.7. Configurer un log handler async dans le CLI

Les log handlers async peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Les tâches que vous allez effectuer pour configurer un log handler async sont les suivantes :

- Ajouter un nouveau log handler async
- Afficher la configuration d'un log handler async
- Modifier le niveau de journalisation
- Définir la longueur de la file d'attente
- Définir l'action de dépassement
- Ajouter les sous-handlers
- Supprimer les sous-handlers
- Supprimer un sous-handler async

Chacune de ces tâches sont décrites ci-dessous.

IMPORTANT

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **/subsystem=logging/logging-profile=NAME/** au lieu de **/subsystem=logging/**.

Ajouter un nouveau log handler async

Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:add(queue-length="LENGTH")
```

Remplacer *HANDLER* par le nom du log handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

Exemple 14.51.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:add(queue-length="10")
{"outcome" => "success"}
```

Afficher la configuration d'un log handler async

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 14.52.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:read-resource
{
    "outcome" => "success",
    "result" => {
        "encoding" => undefined,
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined,
        "overflow-action" => "BLOCK",
        "queue-length" => "50",
        "subhandlers" => undefined
    }
}
[standalone@localhost:9999 /]
```

Modifier le niveau de journalisation

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:write-attribute(name="level",
value="LOG_LEVEL_VALUE")
```

Remplacer *HANDLER* par le nom du log handler et *LOG_LEVEL_VALUE* par le niveau de journalisation à définir.

Exemple 14.53.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="level", value="INFO")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Définir la longueur de la file d'attente

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:write-attribute(name="queue-
length", value="LENGTH")
```

Remplacer *HANDLER* par le nom du log handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

JBoss EAP 6 doit démarrer à nouveau pour prendre effet.

Exemple 14.54.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="queue-length", value="150")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

Définir l'action de dépassement

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:write-
attribute(name="overflow-action", value="ACTION")
```

Remplacer *HANDLER* par le nom du log handler et *ACTION* par *DISCARD* ou *BLOCK*.

Exemple 14.55.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="overflow-action",
value="DISCARD")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Ajouter les sous-handlers

Utiliser l'opération **`add-handler`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:add-handler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du log handler et *SUBHANDLER* par le nom du log handler qui doit être ajouté comme sous-handler de ce handler asynch.

Exemple 14.56.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:add-handler(name="NFS_FILE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer les sous-handlers

Utiliser l'opération **`remove-handler`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:remove-
handler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du log handler et *SUBHANDLER* par le nom du log handler qui doit être supprimé.

Exemple 14.57.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:remove-handler(name="NFS_FILE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Supprimer un sous-handler async

Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du log handler.

Exemple 14.58.

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

14.3.8. Configurer un gestionnaire syslog

Le logmanager de JBoss EAP 6 contient désormais un gestionnaire syslog. Les gestionnaires syslog peuvent être utilisés pour envoyer des messages à un serveur de journalisation à distance qui supporte **`Syslog`** protocol (RFC-3164 or RFC-5424). Cela permet à plusieurs applications d'envoyer leurs messages de journalisation au même serveur, où ils peuvent être analysés en même temps. Ce sujet traite de la création et de la configuration d'un gestionnaire avec le management CLI, ainsi que des options de configuration disponibles.

- Accès et permissions correctes pour le Management CLI.

Procédure 14.1. Ajouter un gestionnaire syslog

- Exécuter la commande suivante pour ajouter un gestionnaire syslog :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:add
```

Procédure 14.2. Configurer un gestionnaire syslog

- Exécuter la commande suivante pour configurer un attribut de gestionnaire syslog :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:write-
attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

Procédure 14.3. Supprimer un gestionnaire syslog

- Exécuter la commande suivante pour supprimer un gestionnaire syslog existant :

```
/subsystem=logging/syslog-handler=HANDLER_NAME:remove
```

Tableau 14.6. Attributs de configuration de syslog-handler

Attribut	Description	Valeur par défaut
Important	Le port que le serveur syslog écoute.	514
app-name	Le nom de l'application utilisé lors du formatage du message dans le format RFC5424.	vide
enabled	Si défini sur true, le gestionnaire sera activé et fonctionnera normalement. Si défini sur false, le gestionnaire sera ignoré lors du traitement des messages de journalisation.	true
level	Le niveau de journalisation indiquant quels niveaux de message seront journalisés. Les niveaux de message inférieurs seront supprimés.	TOUTES LES VERSIONS
aménagement	Tel que défini par RFC-5424 et RFC-3164	niveau utilisateur
adresse serveur	L'adresse du serveur syslog	localhost
hostname	Le nom de l'hôte à partir duquel les messages sont envoyés.	vide

Attribut	Description	Valeur par défaut
syslog-format	Formate le message de journalisation selon la spécification RFC.	RFC5424

Rapporter un bogue

14.3.9. Configurer un log handler personnalisé dans le CLI

Résumé

En plus de la syntaxe de formatage de journalisation spécifiée dans Section 14.1.15, « Syntaxe de Formateur de journaux », un module de formatage de journal personnalisé peut être créé pour une utilisation avec n'importe quel gestionnaire de journal. Cet exemple de procédure vous le montre en créant un formateur XML pour un gestionnaire de journal de console.

Conditions préalables

- Accéder au Management CLI dans le serveur JBoss EAP 6.
- Un log handler précédemment configuré. Cet exemple de procédure utilise un log handler de console.

Procédure 14.4. Configurer un formateur XML personnalisé de log handler

1. Créer un formateur personnalisé.

   Dans cet exemple, la commande suivante crée un formateur personnalisé intitulé **`XML_FORMATTER`** qui utilise la classe **`java.util.logging.XMLFormatter`**.

   ```
   [standalone@localhost:9999 /] /subsystem=logging/custom-
   formatter=XML_FORMATTER:add(class=java.util.logging.XMLFormatter,
   module=org.jboss.logmanager)
   ```

2. Enregistrer un formateur personnalisé pour le log handler que vous souhaitez utiliser avec.

   Dans cet exemple, le formateur de l'étape suivante est ajouté au log handler de la console.

   ```
   [standalone@localhost:9999 /] /subsystem=logging/console-
   handler=HANDLER:write-attribute(name=named-formatter,
   value=XML_FORMATTER)
   ```

3. Démarrer à nouveau le serveur JBoss EAP 6 pour que le changement puisse prendre effet :

   ```
   [standalone@localhost:9999 /] shutdown --restart=true
   ```

Résultat

Le formateur XML personnalisé est ajouté au log handler de la console. La sortie du log de la console sera formaté en XML, comme suit :

```
<record>
```

```
  <date>2014-03-11T13:02:53</date>
  <millis>1394539373833</millis>
  <sequence>116</sequence>
  <logger>org.jboss.as</logger>
  <level>INFO</level>
  <class>org.jboss.as.server.BootstrapListener</class>
  <method>logAdminConsole</method>
  <thread>282</thread>
  <message>JBAS015951: Admin console listening on http://%s:%d</message>
  <param>127.0.0.1</param>
  <param>9990</param>
</record>
```

Rapporter un bogue

14.4. LA JOURNALISATION PAR DÉPLOIEMENT

14.4.1. La journalisation par déploiement

La journalisation par déploiement permet à un développeur de configurer à l'avance la configuration de journalisation de ses applications. Lorsque l'application est déployée, la journalisation commence selon la configuration définie. Les fichiers journaux créés par le biais de cette configuration contiennent uniquement des informations sur le comportement de l'application.

Cette approche a des avantages et des inconvénients pour l'utilisation de la journalisation dans l'ensemble du système. Un des avantages est que l'administrateur de l'instance JBoss EAP n'a pas besoin de configurer la journalisation. Un inconvénient est que la configuration de journalisation par déploiement est en lecture seule au démarrage et donc ne peut donc pas être modifiée pendant l'exécution.

Rapporter un bogue

14.4.2. Désactivation de la journalisation par déploiement

Procédure 14.5. Désactivation de la journalisation par déploiement

- **Il existe deux méthodes pour désactiver la journalisation par déploiement. Il y en a une qui fonctionne pour toutes les versions de JBoss EAP 6, et une autre qui ne fonctionne que sur JBoss EAP 6.3 ou version supérieure.**
  - **JBoss EAP 6 (toutes les versions)**
    Ajouter la propriété système :

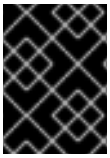
    ```
    org.jboss.as.logging.per-deployment=false
    ```

  - **JBoss EAP 6.3 (ou version supérieure)**
    Exclure le sous-système de journalisation via le fichier **`jboss-deployment-structure.xml`**. Pour obtenir des informations sur la façon de procéder, voir *Exclude a Subsystem from a Deployment* dans le *Development Guide*.

Rapporter un bogue

14.5. PROFILS DE JOURNALISATION

14.5.1. Profils de journalisation



IMPORTANT

Les profils de journalisation ne sont disponibles que dans les versions 6.1.0 ou versions supérieures. Ils ne peuvent pas être configurés par la console de gestion.

Les profils de journalisation sont des ensembles de configuration de journalisation indépendants qui peuvent être assignés à des applications déployées. Un profil de journalisation peut définir les handlers, les catégories et un root logger à la manière du sous-système de journalisation standard, mais ne peut pas recommander la configuration à d'autres profils ou au sous-système de journalisation principal. La conception des profils de journalisation émule le sous-système de journalisation au niveau de l'aisance de configuration.

Les profils de journalisation permettent aux administrateurs de créer des configurations de journalisation spécifiques à une ou à plusieurs applications sans affecter une autre configuration de journalisation. Comme chaque profil est défini dans une configuration serveur, cela implique que la configuration de journalisation peut être modifiée sans exiger que les applications affectées ne soient redéployées.

Chaque profil de journalisation peut avoir la configuration suivante :

- Un nom unique. Ceci est requis.
- N'importe quel nombre de log handlers.
- N'importe quelle catégorie log.
- Un root logger maximum.

Une application peut spécifier un profil de journalisation à utiliser dans son fichier **`MANIFEST.MF`**, en utilisant l'attribut ***`logging-profile`***.

Rapporter un bogue

14.5.2. Créer un nouveau profil de journalisation par le CLI

On peut créer un nouveau profil de journalisation par la commande CLI suivante, en remplaçant *NAME* par le nom de profil qui convient :

```
/subsystem=logging/logging-profile=NAME:add
```

Cela va créer un nouveau profil vide auquel les handlers, catégories et root logger peuvent être ajoutés.

Rapporter un bogue

14.5.3. Créer un profil de journalisation par le CLI

Un profil de journalisation peut être configuré par les log handlers, catégories et root logger en utilisant exactement la même syntaxe que lorsque l'on utilise le sous-système de journalisation principal.

Il existe uniquement deux différences entre configurer le sous-système de journalisation principal et le profil de journalisation :

1. Le chemin de configuration root est **`/subsystem=logging/logging-profile=NAME`**

2. Un profil de journalisation ne peut pas contenir d'autres profils de journalisation.

Référez vous à la tâche de gestion de journalisation qui convient :

- Section 14.3.1, « Configurer le root logger par le CLI »
- Section 14.3.2, « Configurer une Catégorie dans l'interface CLI »
- Section 14.3.3, « Configurer un log handler de console dans le CLI »
- Section 14.3.4, « Configurer un log handler de fichiers dans le CLI »
- Section 14.3.5, « Configurer un log handler périodique dans le CLI »
- Section 14.3.6, « Configurer un log handler de taille dans le CLI »
- Section 14.3.7, « Configurer un log handler async dans le CLI »

Exemple 14.59. Créer et configurer un profil de journalisation

Créer un profil de journalisation et ajouter une catégorie et un log handler de fichiers.

1. Créer le profil :

```
/subsystem=logging/logging-profile=accounts-app-profile:add
```

2. Créer un gestionnaire de fichiers

```
/subsystem=logging/logging-profile=accounts-app-profile/file-
handler=ejb-trace-file:add(file={path=>"ejb-trace.log", "relative-
to"=>"jboss.server.log.dir"})
```

```
/subsystem=logging/logging-profile=accounts-app-profile/file-
handler=ejb-trace-file:write-attribute(name="level",
value="DEBUG")
```

3. Créer une catégorie de logger

```
/subsystem=logging/logging-profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
```

4. Assigner un gestionnaire de fichiers à une catégorie

```
/subsystem=logging/logging-profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add-handler(name="ejb-
trace-file")
```

Rapporter un bogue

14.5.4. Spécifier un profil de journalisation dans une application

Une application spécifie le profil de journalisation à utiliser dans son fichier **`MANIFEST.MF`**.

Conditions préalables :

1. Vous devez connaître le nom du profil de journalisation qui a été défini sur le serveur pour cette application. Demandez à votre administrateur de systèmes le nom du profil à utiliser.

Procédure 14.6. Ajouter une configuration de profil de journalisation à une application

- **Modifier `MANIFEST.MF`**
  Si votre application ne possède pas de fichier **`MANIFEST.MF`** : le créer avec le contenu suivant, en remplaçant *NAME* par le nom de profil qui convient.

  ```
  Manifest-Version: 1.0
  Logging-Profile: NAME
  ```

  Si votre application contient déjà un fichier **`MANIFEST.MF`** : ajouter la ligne suivante, en remplaçant *NAME* par le nom du profil qui convient.

  ```
  Logging-Profile: NAME
  ```

NOTE

Si vous utilisez Maven et **`maven-war-plugin`**, vous pourrez mettre votre fichier MANIFEST.MF dans **`src/main/resources/META-INF/`** et ajouter la configuration suivante à votre fichier **`pom.xml`**.

```
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <configuration>
    <archive>
      <manifestFile>src/main/resources/META-
INF/MANIFEST.MF</manifestFile>
    </archive>
  </configuration>
</plugin>
```

Quand l'application sera déployée, elle utilisera la configuration qui se trouve dans le profil de journalisation spécifié pour ses messages de journalisation.

Rapporter un bogue

14.5.5. Exemple de configuration de profil de journalisation

Cet exemple montre la configuration du profil de journalisation et l'application qui en fait usage. Cela comprend la session CLI affichée, la configuration XML qui est générée et le fichier **`MANIFEST.MF`** de l'application.

L'exemple de profil de journalisation a les caractéristiques suivantes :

- Le nom est **`accounts-app-profile`**.
- La catégorie de journalisation est **`com.company.accounts.ejbs`**.
- Le niveau de journalisations est **`TRACE`**.

- Le log handler est un gestionnaire de fichiers qui utilise **`ejb-trace.log`**.

Exemple 14.60. Session CLI

```
localhost:bin user$ ./jboss-cli.sh -c
[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile:add
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:add(file=
{path=>"ejb-trace.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/logger=com.company.accounts.ejbs:add-
handler(name="ejb-trace-file")
{"outcome" => "success"}

[standalone@localhost:9999 /]
```

Exemple 14.61. Configuration XML

```
<logging-profiles>
   <logging-profile name="accounts-app-profile">
      <file-handler name="ejb-trace-file">
         <level name="DEBUG"/>
         <file relative-to="jboss.server.log.dir" path="ejb-
trace.log"/>
      </file-handler>
      <logger category="com.company.accounts.ejbs">
         <level name="TRACE"/>
         <handlers>
            <handler name="ejb-trace-file"/>
         </handlers>
      </logger>
   </logging-profile>
</logging-profiles>
```

Exemple 14.62. Fichier de l'application MANIFEST.MF

```
Manifest-Version: 1.0
```

```
Logging-Profile: accounts-app-profile
```

Rapporter un bogue

14.6. PROPRIÉTÉS DE LA CONFIGURATION DE JOURNALISATION

14.6.1. Propriétés root logger

Tableau 14.7. Propriétés root logger

Property	Datatype	Description
level	String	Le niveau maximum de messages log que le root logger souhaite enregistrer.
handlers	String[]	Une liste des log handlers utilisés par le root logger.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui exclut les entrées de journalisation qui *ne correspondent pas* à un modèle : **`not(match("JBAS.*"))`**

NOTE

Un **`filter-spec`** spécifié pour le root logger n'est *pas* hérité par les autres gestionnaires. Au lieu de cela, un **`filter-spec`** devra être spécifié pour chaque handler.

Rapporter un bogue

14.6.2. Propriétés de catégorie de journalisation

Tableau 14.8. Propriétés de catégorie de journalisation

Property	Datatype	Description
level	String	Le niveau maximum de messages log que le root logger souhaite enregistrer.
handlers	String[]	Une liste des log handlers utilisés par le root logger.
use-parent-handlers	Booléen	Si défini sur true, cette catégorie utilisera les log handlers du root logger en plus des handlers assignés.
catégorie	String	La catégorie de journalisation à partir de laquelle les messages de journalisation seront capturés.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**

Rapporter un bogue

14.6.3. Propriétés de log handlers de console

Tableau 14.9. Propriétés de log handlers de console

Property	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
target	String	Le flux de sortie du système vers lequel la sortie du log handler se dirige. Peut correspondre à System.err ou System.out pour le flux d'erreurs système ou standard out respectivement.
autoflush	Boolean	Si défini sur true, les messages de journalisation seront envoyés vers la cible des handlers immédiatement après la réception.
name	String	L'identifiant unique de ce log handler.
enabled	Boolean	Si défini sur **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**

Rapporter un bogue

14.6.4. Propriétés de log handlers de fichiers

Tableau 14.10. Propriétés de log handlers de fichiers

Propriété	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.

Propriété	Datatype	Description
append	Boolean	Si la valeur est true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à **`append`** nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.
autoflush	Boolean	Si défini sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les changements à **`autoflush`** nécessitent un redémarrage de serveur pour pouvoir prendre effet.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**.
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès de fichier JBoss EAP 6 peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet.
enabled	Boolean	Si défini sur **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**

Rapporter un bogue

14.6.5. Propriétés de log handlers périodiques

Tableau 14.11. Propriétés de log handlers périodiques

Propriété	Datatype	Description
append	Boolean	Si la valeur est true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications apportées à « append » (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.

Propriété	Datatype	Description
autoflush	Boolean	Si définis sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les changements à « autoflush » nécessitent un redémarrage de serveur pour pouvoir prendre effet.
encoding	String	Définir la codification utilisée pour la sortie.
formatter	String	Le formateur de journalisation utilisé par ce log handler.
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**.
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet.
suffix	String	Cette chaîne est ajoutée au nom de fichier des journaux en rotation et sert à déterminer la fréquence de rotation. Le format du suffixe est un point (.) suivi d'une date de chaîne qui est analysable par la classe **`Ijava.text.SimpleDateFormat`**. Le journal est mis en rotation sur la base de la plus petite unité de temps définie par le suffixe. Par exemple, le suffixe **`.yyyy-MM-dd`** se traduira par rotation quotidienne du log ou journal. Refer to http://docs.oracle.com/javase/6/docs/api/index.html?java/text/SimpleDateFormat.html
enabled	Boolean	Si défini sur **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**

Rapporter un bogue

14.6.6. Propriétés de log handlers de taille

Tableau 14.12. Propriétés de log handlers de taille

Propriété	Datatype	Description
append	Boolean	Si la valeur est true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications apportées à « append » (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte.
autoflush	Boolean	Si défini sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les modifications apportées à « append » (ajout) nécessitent un redémarrage de serveur pour pouvoir prendre effet.
encoding	String	Définir la codification utilisée pour la sortie.
formateur	String	Le formateur de journalisation utilisé par ce log handler.
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
file	Object	L'objet qui représente le fichier dans lequel la sortie de ce log handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**.
relative-to	String	C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur.
path	String	C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrits les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet.
rotate-size	Integer	La taille maximale que le fichier journal peut atteindre avant qu'il soit mis en rotation. Un seul caractère ajouté au nombre indique les unités de taille : **`b`** pour bytes, **`k`** pour kilobytes, **`m`** pour megabytes, **`g`** pour gigabytes. Par ex. **`50m`** pour 50 megabytes.
max-backup-index	Integer	Le nombre maximum de journaux en rotation conservés. Quand ce nombre est atteint, le journal le plus ancien est utilisé à nouveau.
enabled	Boolean	Si défini sur **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation.

Propriété	Datatype	Description
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**
rotate-on-boot	Boolean	Si la valeur est **`true`**, un nouveau fichier de journalisation sera créé au redémarrage du serveur. La valeur par défaut est **`false`**.

Rapporter un bogue

14.6.7. Propriétés de log handlers async

Tableau 14.13. Propriétés de log handlers async

Propriété	Datatype	Description
level	String	Le niveau maximum de messages de journalisation que le log handler enregistre.
name	String	L'identifiant unique de ce log handler.
queue-length	Integer	Nombre maximal de messages de journalisation gardés par le handler en attendant que les sub-handlers répondent.
overflow-action	String	La façon dont ce handler répond quand sa file d'attente est dépassée. Peut être défini sur **`BLOCK`** ou **`DISCARD`**. **`BLOCK`** fait patienter l'application de journalisation jusqu'à ce qu'il y ait suffisamment d'espace disponible dans la file d'attente. C'est le même comportement qu'avec un handler non-async. **`DISCARD`** permet à l'application de journalisation de continuer, mais le message de journalisation sera effacé.
subhandlers	String[]	Il s'agit de la liste de log handlers à laquelle cet handler async passe ses messages log.
enabled	Boolean	Si défini sur **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation.
filter-spec	String	Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`**

Rapporter un bogue

14.7. EXEMPLE DE CONFIGURATION XML DE LOGGING

14.7.1. Échantillon de Configuration XML pour root logger

```
<root-logger>
  <level name="INFO"/>
  <handlers>
    <handler name="CONSOLE"/>
      <handler name="FILE"/>
  </handlers>
</root-logger>
```

Rapporter un bogue

14.7.2. Échantillon de Configuration XML pour une catégorie de journalisation

```
<logger category="com.company.accounts.rec">
  <handlers>
    <handler name="accounts-rec"/>
  </handlers>
</logger>
```

Rapporter un bogue

14.7.3. Échantillon de configuration XML pour un log handler de console

```
<console-handler name="CONSOLE">
  <level name="INFO"/>
  <formatter>
    <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
  </formatter>
</console-handler>
```

Rapporter un bogue

14.7.4. Échantillon de configuration XML pour un gestionnaire de journalisation de fichiers

```
<file-handler name="accounts-rec-trail" autoflush="true">
    <level name="INFO"/>
    <file relative-to="jboss.server.log.dir" path="accounts-rec-
trail.log"/>
    <append value="true"/>
</file-handler>
```

Rapporter un bogue

14.7.5. Échantillon de configuration XML pour un log handler périodique

```
<periodic-rotating-file-handler name="FILE">
   <formatter>
      <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t)
%s%E%n"/>
   </formatter>
   <file relative-to="jboss.server.log.dir" path="server.log"/>
```

```
    <suffix value=".yyyy-MM-dd"/>
    <append value="true"/>
</periodic-rotating-file-handler>
```

Rapporter un bogue

14.7.6. Échantillon de configuration XML pour un log handler de taille

```
<size-rotating-file-handler name="accounts_debug" autoflush="false">
    <level name="DEBUG"/>
    <file relative-to="jboss.server.log.dir" path="accounts-debug.log"/>
    <rotate-size value="500k"/>
    <max-backup-index value="5"/>
    <append value="true"/>
</size-rotating-file-handler>
```

Rapporter un bogue

14.7.7. Échantillon de Configuration XML pour un Log Handler Async

```
<async-handler name="Async_NFS_handlers">
    <level name="INFO"/>
    <queue-length value="512"/>
    <overflow-action value="block"/>
    <subhandlers>
        <handler name="FILE"/>
        <handler name="accounts-record"/>
    </subhandlers>
</async-handler>
```

Rapporter un bogue

CHAPITRE 15. INFINISPAN

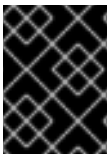
15.1. INFINISPAN

Infinispan est une plateforme de grille de données Java. Il fournit une interface cache compatible JSR-107 pour gérer les données mises en cache.

Les conteneurs cache d'infinispan sont utilisés dans JBoss Enterprise Application Platform 6 :

- **`web`** dans Web Session Clustering
- **`ejb`** dans Stateful Session Bean Clustering
- **`hibernate`** pour la mise en cache d'entité
- **`singleton`** pour la mise en cache de singleton

Chaque conteneur cache définit un "repl" et un "dist". Ces caches ne doivent pas être utilisés directement par les applications utilisateur.



IMPORTANT

Les utilisateurs peuvent ajouter les conteneurs cache et des caches, et les référencer via JNDI. Cependant, cela n'est pas pris en charge par JBoss EAP 6.

Pour obtenir plus d'informations sur la fonctionnalité Infinispan et ses options de configuration, consultez *Infinispan Documentation*.

Rapporter un bogue

15.2. MODES DE CLUSTERING

Le clustering peut être configuré de deux façons différentes dans JBoss EAP 6 en utilisant Infinispan. La méthode qui convient à votre application dépendra de vos besoins. Il y a des considérations à prendre quand vous choisissez entre la disponibilité, la fiabilité et l'évolubilité de chaque mode. Avant de choisir un mode de clusterisation, vous devez identifier ce qui est le plus important pour vous sur votre réseau, et équilibrer en fonction des besoins.

Mode répliqué

Le mode répliqué détecte et ajoute automatiquement de nouvelles instances sur le cluster. Les changements faits à ces instances seront répliqués à tous les noeuds du cluster. Le mode répliqué fonctionne normalement mieux dans les petits clusters à cause du montant d'informations qui aura été répliqué sur le réseau. Infinispan peut être configuré pour utiliser UDP

Mode de distribution

Le mode de distribution permet à Infinispan de mettre le cluster à l'échelle linéairement. Le mode de distribution utilise un algorithme de hachage uniforme pour déterminer où un nouveau nœud doit être placé dans un cluster. Le nombre de copies d'informations à conserver est configurable. C'est un compromis entre le nombre de copies conservées, la durabilité des données et la performance : plus le nombre de copies qui restent sera grand, plus cela aura d'impact sur les performances, mais il sera alors moins probable que vous perdiez des données en cas de panne de serveur. L'algorithme de hachage permet également de réduire le trafic réseau en détectant les entrées sans multidiffusion ou en stockant des métadonnées.

Réplication synchrone et asynchrone

La réplication peut être effectuée soit en mode synchrone ou soit en mode asynchrone et le mode choisi dépendra de vos besoins et de votre application. Avec la réplication synchrone, le thread qui gère la demande de l'utilisateur sera bloqué jusqu'à ce que la réplication ait réussi. Quand la réplication aura réussi, une réponse sera envoyée au client, et le thread sera libéré. La réplication synchrone aura un impact sur le trafic réseau parce qu'elle requiert une réponse de chaque nœud du cluster. Elle a l'avantage, cependant, de veiller à ce que toutes les modifications soient apportées à tous les nœuds du cluster.

La réplication asynchrone est effectuée en arrière-plan. Infinispan implémente une file d'attente de réplication, qui est utilisée par un thread d'arrière-plan pour effectuer la réplication. La réplication est déclenchée sur une base de temps, ou sur la taille de la file d'attente. Une file d'attente de réplication permet de meilleures performances parce qu'il n'y a aucune conversation menée entre les nœuds du cluster. Le compromis avec la réplication asynchrone, c'est qu'elle n'est pas aussi précise. Les tentatives de réplication qui ont échoué sont inscrites dans un journal, et ne sont pas notifiées en temps réel.

Rapporter un bogue

15.3. CONTENEURS DE CACHE

Conteneurs de cache

Un conteneur de cache est le référentiel de caches utilisés par un sous-système. Dans Infinispan, les conteneurs de cache par défaut sont définis dans les fichiers de configuration xml (standalone-ha.xml, standalone-full-ha.xml, domain.xml). Un cache est défini comme le cache par défaut, et ce sera le cache qui sera utilisé pour le clustering.

Exemple 15.1. Définitions de conteneur cache du fichier de configuration standalone-ha.xml

```
<subsystem xmlns="urn:jboss:domain:infinispan:1.5">
       <cache-container name="singleton" aliases="cluster ha-
partition" default-cache="default">
          <transport lock-timeout="60000"/>
          <replicated-cache name="default" mode="SYNC"
batching="true">
             <locking isolation="REPEATABLE_READ"/>
          </replicated-cache>
       </cache-container>
       <cache-container name="web" aliases="standard-session-cache"
default-cache="repl" module="org.jboss.as.clustering.web.infinispan">
        <transport lock-timeout="60000"/>
        <replicated-cache name="repl" mode="ASYNC" batching="true">
          <file-store/>
        </replicated-cache>
        <replicated-cache name="sso" mode="SYNC" batching="true"/>
          <distributed-cache name="dist" l1-lifespan="0"
mode="ASYNC" batching="true">
           <file-store/>
          </distributed-cache>
        </cache-container>
```

Notez le cache par défaut défini dans chaque conteneur cache. Dans cet exemple, dans le conteneur cache **web**, le cache **repl** est défini par défaut. Le cache **repl** sera donc utilisé pour le clustering des sessions web.

Les conteneurs de cache et les attributs de cache peuvent être configurés par la console de gestion ou les commandes CLI, mais il n'est pas conseillé de modifier les noms des conteneurs cache, ni des caches eux-mêmes.

Configuration des conteneurs cache

Les conteneurs cache d'Infinispan peuvent être configurés par le CLI ou par la console de gestion.

Procédure 15.1. Configurer les conteneurs cache Infinispan dans la console de gestion

1. Sélectionner l'onglet **Configuration** en haut de l'écran.
2. En mode de domaine uniquement, sélectionner **ha** ou **full-ha** à partir du menu déroulant en haut et à gauche.
3. Étendre le menu **Subsystems**, puis étendre le menu **Infinispan**. Sélectionner **Cache Containers**.
4. Sélectionner un conteneur cache à partir du tableau **Cache Containers**.
5. **Ajouter, supprimer ou définir le conteneur de caches par défaut**
   a. Pour créer un nouveau conteneur cache, cliquer sur **Add** dans le tableau **Cache Containers**.
   b. Pour supprimer le conteneur cache, sélectionner le conteneur cache dans le tableau **Cache Containers**. Cliquer sur **Remove** puis sur **OK** pour confirmer.
   c. Pour définir un conteneur cache comme défaut, cliquer sur **Set Default**, puis, saisir un nom de conteneur cache à partir de la liste de menu déroulant, puis sur **Save** pour confirmer.
6. Pour ajouter ou mettre à jour les attributs d'un conteneur de cache, sélectionner le conteneur cache dans le tableau **Cache Containers**. Sélectionnez en un à partir des onglets **Attributes**, **Transport** ou **Aliases** qui se trouvent dans la partie d'écran **Details**, puis cliquer sur **Edit**. Pour avoir des informations supplémentaires sur le contenu des onglets **Attributes**, **Transport** et **Aliases**, cliquer sur **Need Help?**.

Procédure 15.2. Configurer les conteneurs cache Infinispan dans l'interface CLI

1. Pour obtenir une liste des attributs configurables, saisir la commande CLI suivante :

```
/profile=profile name/subsystem=infinispan/cache-
container=container name:read-resource
```

2. Vous pouvez utiliser l'interface CLI pour ajouter, supprimer, et mettre à jour les conteneurs cache. Avant d'utiliser des commandes avec les conteneurs cache, assurez-vous d'avoir le profil qui convient dans la commande CLI.
   a. **Ajouter un conteneur cache**
   Pour ajouter un conteneur cache, basez votre commande sur l'exemple suivant :

```
/profile=profile-name/subsystem=infinispan/cache-
container="cache container name":add
```

b. **Supprimer un conteneur cache**
Pour supprimer un conteneur cache, basez votre commande sur l'exemple suivant :

```
/profile=profile-name/subsystem=infinispan/cache-
container="cache container name":remove
```

c. **Mettez à jour les attributs de conteneur de cache**
Utiliser l'opération write-attribute pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour statistics-enable à true.

```
/profile=profile name/subsystem=infinispan/cache-
container=cache container name:write-attribute(name=statistics-
enabled,value=true)
```

Rapporter un bogue

15.4. CACHE STORES

Un cache store est un stockage externe de données dans un cache. Les types de stores de données externes qui importent à JBoss EAP 6 sont basés sur des fichiers, JDBC, ou le store distant Infinispan/JDG.

Pour les cache stores basés sur des fichiers, chaque nœud du cluster a généralement son propre système de fichiers et donc, son propre cache store basé fichier. Il n'est pas conseillé de mettre un cache store basé fichiers sur un système de fichiers partagé (NFS, etc.), car ils n'implémentent pas de verrouillage de fichier approprié et peuvent provoquer une corruption des données.

Pour les cache stores basés JDBC, il est possible d'avoir une base de données SQL unique agissant comme cache store pour tous les nœuds du cluster. Cependant, le cache store doit être configuré comme partagé (avec l'attribut **`shared`** sur true sur les cache stores JDBC). Si les cache stores ne sont pas définis sur « shared », des blocages de base de données peuvent se produire, ainsi que d'autres problèmes pouvant avoir un impact sur les performances.

Rapporter un bogue

15.5. STATISTIQUES INFINISPAN

Les statistiques de temps d'exécution d'objets cache-container ou Innispan cache peuvent être activés à but de surveillance. La collecte des statistiques n'est pas activée par défaut pour des raisons de performance.

AVERTISSEMENT

Activer les statistiques Infinispan peut avoir un impact négatif sur la performance des sous-systèmes Infinispan. Les statistiques ne devraient être activés que si besoin est.

La collecte de statistiques peut être activée pour chaque conteneur-cache, cache, ou les deux. L'option de statistiques pour chaque cache remplace l'option de cache-conteneur. L'activation ou la désactivation de la collecte de statistiques d'un conteneur de cache entraînera tous les caches de ce conteneur à hériter la configuration, à moins qu'ils ne spécifient explicitement les leur. Si seulement un cache-conteneur est activé pour les statistiques, des statistiques utiles seront disponibles.

Rapporter un bogue

15.6. ACTIVER LA COLLECTE DES STATISTIQUES D'INFINISPAN

La collecte des statistiques ne peut pas être activée à partir du fichier de configuration de démarrage (par exemple, **`standalone.xml`**, **`standalone-ha.xml`**, **`domain.xml`**) ou l'interfacce CLI.

Rapporter un bogue

15.6.1. Activer la collecte des statistiques d'Infinispan dans un fichier de configuration de démarrage

Procédure 15.3. Activer les statistiques d'Infinispan dans un fichier de configuration de démarrage

- Ajouter l'attribut **`statistics-enabled`**=***`VALUE`*** dans la balise XML du **`cache-container`** ou du **`cache`** dans le sous-système d'Infinispan.

Exemple 15.2. Activer la collecte de statistiques pour un `cache`

```
<replicated-cache name="sso" mode="SYNC" batching="true" statistics-
enabled="true"/>
```

Exemple 15.3. Activer la collecte de statistiques pour un `cache-container`

```
<cache-container name="singleton" aliases="cluster ha-partition"
default-cache="default" statistics-enabled="true">
```

Rapporter un bogue

15.6.2. Active la collecte des statistiques d'Infinispan à partir de l'interface CLI

Procédure 15.4. Active la collecte des statistiques d'Infinispan à partir de l'interface CLI

Dans cette procédure :

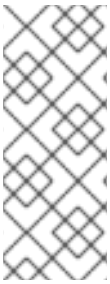
- ***CACHE_CONTAINER*** est le **cache-container** préféré (par exemple, **web**)
- ***CACHE_TYPE*** est le type de cache préféré (par exemple, **distributed-cache**)
- ***CACHE*** est le nom de cache (par exemple, **dist**)

1. Saisir la commande suivante :

```
/subsystem=infinispan/cache-
container=CACHE_CONTAINER/CACHE_TYPE=CACHE:write-
attribute(name=statistics-enabled,value=true)
```

2. Saisir la commande suivante pour recharger le serveur :

```
:reload
```



NOTE

Pour annuler la définition d'un attribut, saisir la commande suivante :

```
/subsystem=infinispan/cache-
container=CACHE_CONTAINER/CACHE_TYPE=CACHE:undefine-
attribute(name=statistics-enabled)
```

Rapporter un bogue

15.6.3. Vérifier que la collecte des statistiques d'Infinispan soit activée

Procédure 15.5. Vérifier que la collecte des statistiques d'Infinispan soit activée

Suivant que vous confirmiez que la collecte de statistiques est activée sur un **cache** ou sur un **cache-container**, utiliser une des commandes d'interface CLI suivantes :

- **Pour un cache**

```
/subsystem=infinispan/cache-
container=CACHE_CONTAINER/CACHE_TYPE=CACHE:read-
attribute(name=statistics-enabled)
```

- **Pour un cache-container**

```
/subsystem=infinispan/cache-container=CACHE_CONTAINER:read-
attribute(name=statistics-enabled)
```

Rapporter un bogue

15.7. JGROUPS

15.7.1. JGroups

JGroups est un outil de messagerie qui permet aux développeurs de créer des applications de messagerie fiables où la fiabilité du système est en cause. JGroups peut être utilisé pour créer des clusters dont les nœuds peuvent envoyer des messages de l'un à l'autre.

Le sous-système de JGroups fournit tous les mécanismes de communication sur la façon dont les serveurs d'un cluster communiquent entre eux. EAP est préconfiguré avec deux piles de JGroups.

- UDP - les nœuds du cluster utilisent la multidiffusion UDP (User Datagram Protocol) pour communiquer entre eux. UDP est généralement plus rapide mais moins fiable que TCP.
- TCP - les nœuds du cluster utilisent TCP (Transmission Control Protocol) pour communiquer entre eux. TCP a tendance à être plus lent que l'UDP, mais plus fiable pour délivrer des données vers sa destination.

Les piles préconfigurées peuvent être utilisées, ou vous pouvez définir votre propre pile pour répondre aux besoins spécifiques de votre système.

Rapporter un bogue

CHAPITRE 16. JVM

16.1. JVM

16.1.1. Paramètres de configuration de JVM

Les paramètres de configuration de machines virtuelles Java (JVM) varient selon les instances de domaine géré et les instances de serveur autonome. Dans un domaine géré, les paramètres de JVM sont déclarés dans les fichiers de configuration **`host.xml`** et **`domain.xml`**, et ils sont déterminés par les composants de contrôleur de domaine chargés de lancer et d'arrêter le processus du serveur. Dans une instance de serveur autonome, les processus de démarrage de serveur peuvent passer des paramètres de ligne de commande au démarrage. Ceux-ci peuvent être déclarés depuis la ligne de commande ou via l'écran **`System Properties`** dans la console de gestion.

Domaine géré

Une caractéristique importante du domaine géré est la possibilité de définir des paramètres de la JVM à plusieurs niveaux. Vous pouvez configurer les paramètres de JVM personnalisés au niveau de l'hôte, par groupe de serveurs, ou par instance de serveur. Les éléments enfants plus spécialisés remplacent la configuration parent, permettant la déclaration des configurations de serveur spécifique sans nécessiter d'exclusions au niveau groupe ou hôte. Cela permet également à la configuration parent d'être héritée par les autres niveaux jusqu'à ce que les paramètres soient déclarés dans les fichiers de configuration ou transmis pendant le runtime.

Exemple 16.1. Les paramètres de configuration JVM du fichier de configuration du domaine

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration **`domain.xml`**.

```
<server-groups>
       <server-group name="main-server-group" profile="default">
           <jvm name="default">
               <heap size="64m" max-size="512m"/>
           </jvm>
           <socket-binding-group ref="standard-sockets"/>
       </server-group>
       <server-group name="other-server-group" profile="default">
           <jvm name="default">
               <heap size="64m" max-size="512m"/>
           </jvm>
           <socket-binding-group ref="standard-sockets"/>
       </server-group>
</server-groups>
```

Dans cette instance, un groupe de serveurs appelé **`main-server-group`** déclare une taille de segment de 64 mégaoctets et une taille de segment maximale de 512 mégaoctets. Tout serveur qui appartient à ce groupe héritera de ces paramètres. Vous pouvez modifier ces paramètres pour le groupe dans son ensemble, par hôte ou serveur individuel.

Exemple 16.2. Les paramètres de configuration du domaine dans le fichier de configuration de l'hôte

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration **host.xml**.

```
<servers>
  <server name="server-one" group="main-server-group" auto-start="true">
    <jvm name="default"/>
  </server>
  <server name="server-two" group="main-server-group" auto-start="true">
    <jvm name="default">
      <heap size="64m" max-size="256m"/>
    </jvm>
    <socket-bindings port-offset="150"/>
  </server>
  <server name="server-three" group="other-server-group" auto-
start="false">
    <socket-bindings port-offset="250"/>
  </server>
</servers>
```

Dans ce cas, un serveur appelé **server-two** appartient au groupe de serveurs nommé **main-server-group**, qui hérite les paramètres du groupe de JVM **par défaut**. Dans l'exemple précédent, la taille du segment principal de **main-server-group** a été fixée à 512 méga-octets. En déclarant une taille de segment basse de 256 mégaoctets, **server-two** peut substituer les paramètres de **domain.xml** pour ajuster les performances comme vous le souhaitez.

Paramètres de configuration de serveur autonome en cours d'exécution

Les paramètres de JVM pour des instances de serveurs autonomes peuvent être déclarés pendant l'exécution en définissant la variable d'environnement **JAVA_OPTS** avant de démarrer le serveur. Un exemple de définition de la variable d'environnement **JAVA_OPTS** en ligne de commande Linux est :

```
[user@host bin]$ export JAVA_OPTS="-Xmx1024M"
```

La même configuration peut être utilisée dans un environnement Microsoft Windows, comme suit :

```
C:\> set JAVA_OPTS="Xmx1024M"
```

Alternativement, les paramètres de configuration JVM peuvent être ajoutés au fichier **standalone.conf** qui se trouve dans le dossier ***EAP_HOME*/bin** contenant des exemples d'options à passer à la JVM.

AVERTISSEMENT

En définissant la variable d'environnement JAVA_OPTS, vous redéfinissez les valeur par défaut de la variable d'environnement JAVA_OPTS. Cela peut compromettre ou résilier le démarrage d'EAP.

Rapporter un bogue

16.1.2. Afficher le statut JVM dans la console de gestion

Conditions préalables

- Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »
- Section 2.1.3, « Démarrez JBoss EAP 6 comme domaine géré »
- Section 3.4.2, « Se conncecter à la console de gestion »

Le statut de la Machine virtuelle Java (JVM) peut être affichée dans la console de gestion pour le serveur autonome ou un domaine géré. La console affiche l'utilisation de segments, leur non utilisation, et l'usage de threads du serveur. Malgré que les statistiques ne soient pas affichés en temps réel, vous pouvez actualiser l'affichage de la console pour donner un aperçu à jour des ressources de la machine virtuelle Java.

Le statut de la JVM affiche les valeurs suivantes.

Tableau 16.1. Attributs de Statut JVM

Type	Description
Max	Le montant de mémoire maximale pouvant être utilisé pour la gestion de la mémoire. La mémoire maximum disponible s'affiche sur la barre grise claire.
Utilisé	Le montant de mémoire utilisée. La mémoire utilisée s'affiche sur la barre grise claire.
Validé	Le montant de mémoire allouée à l'utilisation pour une machine virtuelle Java. La mémoire utilisée s'affiche sur la barre grise claire.
Init	Le montant de mémoire demandée au départ par la machine virtuelle Java au système d'exploitation en matière de gestion de mémoire. Le montant init s'affiche sur la barre grise claire.

Procédure 16.1. Afficher le statut JVM dans la console de gestion

1. - **Affichage du statut de la JVM pour une instance de serveur autonome**
   Sélectionner l'onglet **Runtime** en haut de l'écran. Étendre le menu **Status**, puis étendre le menu **Platform**. Sélectionner **JVM**.

   - **Afficher le statut de la JVM d'un domaine géré**
     Sélectionner l'onglet **Runtime** en haut de l'écran. Étendre le menu **Server Status**, puis étendre le menu **Platform**. Sélectionner **JVM**.

2. Le domaine géré peut rendre visibles toutes les instances de serveur dans le groupe de serveurs, mais ne vous permettra d'afficher qu'un seul serveur à la fois en sélectionnant dans le menu serveur. Pour afficher le statut des autres serveurs dans votre groupe de serveurs, cliquez sur **`Change Server`** en haut à gauche de l'écran pour sélectionner à partir de l'hôte et des serveurs affichés dans votre groupe. Cliquez sur le bouton **`Done`** pour terminer.

Résultat

Le statut des paramètres de configuration de la JVM de l'instance de serveur est affiché.

Rapporter un bogue

16.1.3. Configuration d'une JVM

Les balises <jvm></jvm> prennent en charge l'utilisation des options <jvm-options></jvm-options>, pouvant être utilisées pour ajouter des paramètres supplémentaires à la configuration de la JVM en utlisant l'option <option value=""/>.

Par exemple

<jvm name="default"> <heap size="1303m" max-size="1303m"/> <permgen max-size="256m"/> <jvm-options> <option value="-XX:+UseCompressedOops"/> </jvm-options> </jvm>

Configurer une JVM par le CLI

Pour configurer une JVM par le CLI, utiliser la syntaxe suivante :

```
# cd /server-group=main-server-group/jvm=default

  # :add-jvm-option(jvm-option="-XX:+UseCompressedOops")
   {
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => undefined
   }

  # :read-resource

  # Expected Result:

   [domain@localhost:9999 jvm=default] :read-resource
   {
    "outcome" => "success",
    "result" => {
     "agent-lib" => undefined,
     "agent-path" => undefined,
     "env-classpath-ignored" => undefined,
     "environment-variables" => undefined,
     "heap-size" => "1303m",
     "java-agent" => undefined,
     "java-home" => undefined,
     "jvm-options" => ["-XX:+UseCompressedOops"],
     "max-heap-size" => "1303m",
     "max-permgen-size" => "256m",
     "permgen-size" => undefined,
     "stack-size" => undefined,
     "type" => undefined
    }
   }
```

Supprimer l'entrée jvm-options

Pour supprimer l'entrée jvm-options, utiliser la syntaxe suivante :

```
 # cd /server-group=main-server-group/jvm=default

 # :remove-jvm-option(jvm-option="-XX:+UseCompressedOops")
```

```
 # Expected Result:

  [domain@localhost:9999 jvm=default] :remove-jvm-option(jvm-option="-
XX:+UseCompressedOops")
  {
   "outcome" => "success",
   "result" => undefined,
   "server-groups" => undefined
  }
```

Rapporter un bogue

CHAPITRE 17. SOUS-SYSTÈME WEB

17.1. CONFIGURER LE SOUS-SYSTÈME WEB

Vous pouvez configurer la plupart des aspects du sous-système web à l'aide de la console de gestion sur le web ou l'interface CLI de ligne de commande. Chaque paramètre est expliqué dans l'ordre dans lequel il apparaît dans la console de gestion, et les commandes de l'interface CLI sont également fournies.

Afficher le sous-système basé web par la console de gestion

Pour configurer le sous-système de web à l'aide de la console de gestion sur le web, cliquez sur l'onglet **`Configuration`** en haut à droite de l'écran. Développez le menu de **`Subsystems`**, puis le menu **`Web`**. Chaque partie configurable du sous-système web est montrée.

NOTE

Le composant **`mod_cluster`** n'est disponible que si votre profil est **`ha`** ou **`full-ha`**, dans un domaine géré, ou si vous démarrez votre serveur autonome avec le profil **`standalone-ha`** ou **`standalone-full-ha`**. La configuration **`mod_cluster`** est abordée dans Section 19.5.2, « Configurer le sous-système **`mod_cluster`** ».

Configurer le conteneur JSP, les connecteurs HTTP, et les serveurs HTTP virtuels

Pour configurer le conteneur JSP, les connecteurs HTTP et des serveurs virtuels HTTP, cliquer sur l'entrée de menu **`Servlet/HTTP`**. Cliquer sur le bouton **`Edit`** pour modifier une valeur. Cliquer sur le bouton **`Advanced`** pour afficher les options avancées. Les options sont expliquées ci-dessous. Les options pour les serveurs virtuels et les connecteurs HTTP figurent dans des tableaux distincts.

Tableau 17.1. Options de configuration Servlet/HTTP

Option	Description	Commande CLI
ID de l'instance	L'identificateur utilisé pour activer l'affinité de session dans les scénarios d'équilibrage de charge. L'identificateur doit être unique sur tous les serveurs du cluster JBoss EAP et est ajouté à l'identificateur de session généré. Cela permet au proxy frontal de transmettre la session spécifique à une même instance de JBoss EAP. L'ID d'instance n'est pas défini par défaut.	`/profile=full-ha/subsystem=web:write-attribute(name=instance-id,value=worker1)`
Disabled?	Si sur **`true`**, désactive le conteneur Java ServerPages (JSP). Valeur par défaut **`false`**. Utile si vous n'utilisez pas les pages JSP.	`/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=disabled,value=false)`

Option	Description	Commande CLI
Development?	Si sur **`true`**, active Development Mode, qui produit davantage d'informations verbeuses de débogage. Valeur par défaut **`false`**.	`/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=development,value=false)`
Keep Generated?	Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur **`true`**, garde les servlets générés. Défini sur **`true`** par défaut.	`/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=keep-generated,value=true)`
Check Interval?	Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Valeur en secondes qui détermine la fréquence des vérifications de mises à jour JSP par un processus en arrière-plan. La valeur par défaut est **`0`**.	`/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=check-interval,value=0)`
Display Source?	Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur **`true`**, le fragment de source JSP est affiché quand une erreur d'exécution a lieu. La valeur par défaut est **`true`**.	`/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=display-source-fragment,value=true)`

Les connecteurs AJP and HTTP utilisent **`mod_cluster`**, **`mod_jk`**, **`mod_proxy`**, **`ISAPI`**, et **`NSAPI`** pour l'équilibrage des charges et pour le clustering HA. Pour configurer un connecteur, sélectionner l'onglet **`Connectors`** et cliquer sur **`Add`**. Pour supprimer un connecteur, le sélectionner et cliquer sur **`Remove`**. Pour modifier un connecteur, le sélectionner et cliquer sur **`Edit`**.

Quand vous créez un nouveau connecteur par l'interface CLI, ses options sont définies aussitôt, comme dans la commande suivante :

Exemple 17.1. Créer un nouveau connecteur

```
/profile=full-ha/subsystem=web/connector=ajp/:add(socket-
```

```
binding=ajp,scheme=http,protocol=AJP/1.3,secure=false,name=ajp,max-post-size=2097152,enabled=true,enable-lookups=false,redirect-port=8433,max-save-post-size=4096)
```

Tableau 17.2. Options de connecteur

Option	Description	Commande CLI
Nom	Un nom unique de connecteur, à but d'affichage.	`/profile=full-ha/subsystem=web/connector=ajp/:read-attribute(name=name)`
Liaisons de sockets	La liaison de socket nommée à laquelle le connecteur doit se lier. La liaison de socket est un mappage entre un nom de socket et un port réseau. Les liaisons de socket sont configurées pour chaque serveur autonome, ou par l'intermédiaire de groupes de liaison de socket dans un domaine géré. Un groupe de liaisons de sockets est appliqué à un groupe de serveurs.	`/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=socket-binding,value=ajp)`
Schéma	Le schéma de connecteur web, comme HTTP ou HTTPS.	`/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=scheme,value=http)`
Protocole	Le protocole de connecteur web à utiliser, comme AJP ou HTTP.	`/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=protocol,value=AJP/1.3)`
Activé	Indique si le connecteur web connecté est activé.	`/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=enabled,value=true)`

Option	Description	Commande CLI
Redirect Port	Utilisé pour spécifier un numéro de port à utiliser en cas de redirection; les plus communs étant redirigés vers un connecteur «secure» (**`https`**) ou **`AJP`**	`/profile=full-ha/subsystem=web/connector=http:write-attribute(name=redirect-port,value=8443)`
Redirect Binding	La redirection de liaison est similaire pour rediriger le port en termes de comportement, sauf qu'il faut la spécification d'un nom de liaison de socket comme "valeur" au lieu d'un numéro de port. **`redirect-liaison`** fournit une plus grande flexibilité de configuration car il permet l'utilisation d'une liaison de socket pré-définie liaison (https, AJP etc.) vers le port spécifique de redirection. Il donne les mêmes résultats que l'option **`redirection-port`**	`/profile=full-ha/subsystem=web/connector=http:write-attribute(name=redirect-binding,value=https)`

Pour configurer des serveurs virtuels, cliquer sur l'onglet **`Virtual Servers`**. Utiliser le bouton **`Add`** pour ajouter un nouveau serveur virtuel. Pour modifier ou supprimer un serveur virtuel, le sélectionner et cliquer le bouton **`Edit`** ou **`Remove`**.

Quand vous ajoutez un nouveau serveur virtuel par l'interface CLI, toutes les options requises sont définies en même temps, comme par la commande suivante.

Exemple 17.2. Ajouter un nouveau serveur virtuel

```
/profile=full-ha/subsystem=web/virtual-server=default-host/:add(enable-welcome-root=true,default-web-module=ROOT.war,alias=["localhost","example.com"],name=default-host)
```

Tableau 17.3. Options de serveurs virtuels

Option	Description	Commande CLI
Nom	Nom unique de serveur virtuel, à but d'affichage.	`/profile=full-ha/subsystem=web/virtual-server=default-host/:read-attribute(name=name)`

Option	Description	Commande CLI
Alias	Une liste de noms d'hôtes qui doivent correspondre à ce serveur virtuel. Dans la console de gestion, utiliser un nom d'hôte par ligne.	`/profile=full-ha/subsystem=web/virtual-server=default-host/:write-attribute(name=alias,value=["localhost","example.com"])`
Module par défaut	Le module dont l'application web doit être déployée au nœud racine de ce serveur virtuel et qui sera affiché quand aucun répertoire n'est donné par la requête HTTP.	`/profile=full-ha/subsystem=web/virtual-server=default-host/:write-attribute(name=default-web-module,value=ROOT.war)`

Rapporter un bogue

17.2. REMPLACER L'APPLICATION WEB WELCOME PAR DÉFAUT

JBoss EAP 6 inclut l'application Welcome, qui s'affiche quand on ouvre l'URL du serveur sur le port 8080. Vous pouvez remplacer cette application par votre propre application web, en suivant la procédure suivante.

Procédure 17.1. Remplacer l'application web Welcome par défaut par votre propre application web

1. **Désactiver l'application Welcome**
   Utiliser le script de Management CLI ***EAP_HOME*/bin/jboss-cli.sh** pour exécuter la commande suivante. Vous aurez sans doute besoin de modifier un profil de domaine géré, ou retirer une portion de la commande **/profile=default** du serveur autonome.

   ```
   /profile=default/subsystem=web/virtual-server=default-host:write-attribute(name=enable-welcome-root,value=false)
   ```

2. **Configurer votre application web par le contexte root.**
   Afin de configurer votre application web, pour qu'elle utilise (/) comme adresse URL, modifier son fichier **jboss-web.xml**, qui se trouve dans le répertoire **META-INF/** ou **WEB-INF/**. Remplacer sa directive **<context-root>** par une autre qui ressemble à ce qui suit.

   ```
   <jboss-web>
       <context-root>/</context-root>
   </jboss-web>
   ```

-

3. **Déployer votre application.**
   Déployer votre application pour le groupe de serveurs ou le serveur que vous avez modifié lors de la première étape. L'application est maintenant disponible sur `http://SERVER_URL:PORT/`.

Rapporter un bogue

CHAPITRE 18. SOUS-SYSTÈME DE SERVICES WEB

18.1. CONFIGURER LES OPTIONS DE SERVICES WEB

Pour configurer les options de Services Web, cliquer sur **`Web Services`**. Les options sont expliquées dans le tableau ci-dessous.

Tableau 18.1. Options de configuration des Services Web

Option	Description	Commande CLI
Modifier l'adresse WSDL	Indique si l'adresse WSDL peut être modifiée par les applications. Valeur par défaut **`true`**.	`/profile=full-ha/subsystem=webservices/:write-attribute(name=modify-wsdl-address,value=true)`
Hôte WSDL	Le contrat WSDL d'un service Web JAX-WS inclut un élément <soap:address> qui pointe vers l'emplacement du point de terminaison. Si la valeur de <soap:address> est un URL valide, elle n'est pas remplacée à moins que **`modify-wsdl-adress`** soit défini à la valeur **`true`**. Si la valeur de <soap:address> n'est pas un URL valide, elle est remplacée en utilisant les valeurs **`wsdl-host`** et **`wsdl-port`** ou **`wsdl-secure-port`**. Si **`wsdl-host`** est défini sur **`jbossws.undefined.host`**, l'adresse hôte de l'auteur de la demande est utilisée lorsque <soap:address> est réécrite. Par défaut, **`${jboss.bind.address:127.0.0.1}`**, qui utilise **`127.0.0.1`** si aucune adresse de liaison est spécifiée lors du démarrage de JBoss EAP 6.	`/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-host,value=127.0.0.1)`
Port WSDL	Le port non-sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si défini sur **`0`** (défaut), le port sera identifié en demandant la liste des connecteurs installés.	`/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-port,value=80)`

Option	Description	Commande CLI
Port sécurisé WSDL	Le port sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si définie sur **`0`** (défaut), le port sera identifié en demandant la liste des connecteurs installés.	`/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-secure-port,value=443)`

NOTE

Vous aurez sans doute besoin de modifier le profil pour modifier un profil de domaine géré différent, ou supprimer la partie /profile=full-ha de la commande d'un serveur autonome.

Sous-système de Services Web

Pour activer la journalisation dans Apache CXF, configurer la propriété système suivante dans le fichier **`standalone/domain.xml`** :

```
<system-properties>
<property name="org.apache.cxf.logging.enabled" value="true"/>
</system-properties>
```

Rapporter un bogue

CHAPITRE 19. HTTP CLUSTERING ET ÉQUILIBRAGE DES CHARGES

19.1. INTRODUCTION

19.1.1. Clusters haute disponibilité (HA) et clusters d'équilibrage des charges

Le terme *Clustering* se réfère à l'utilisation de ressources multiples, comme des serveurs, comme s'ils constituaient une seule entité. Les deux principaux types de clustering sont *Load balancing (LB)* et *High-availability (HA)*. Dans un groupement LB, toutes les ressources exécutent en même temps, et une couche de gestion se charge de répartir la charge de travail entre eux.

Dans le clustering HA, une ressource exécute, et une autre est prête à prendre position quand la première est rendue disponible. Le but du clustering HA est de réduire les chances de panne niveau matériel, logiciels ou réseau.

JBoss Enterprise Application Platform supporte le clustering à plusieurs niveaux. Certains sous-systèmes que l'on puisse rendre disponibles sont les suivants :

- Les instances du serveur d'applications
- Les applications web, lorsqu'elles sont utilisées en conjonction avec le serveur interne JBoss Web, Apache HTTP, Microsoft IIS ou Oracle iPlanet Web Server.
- Les EJB (Enterprise JavaBeans) avec, ou sans état
- Mécanismes Single Sign On (SSO)
- Cache distribué
- Sessions HTTP
- Les services JMS et les MDB (Messages Driven Beans)

Le clustering est rendu disponible par deux sous-systémes : **`jgroups`** et **`modcluster`** dans JBoss EAP 6. Les profils **`ha`** et **`full-ha`** ont ces systèmes activés. Dans JBoss EAP 6, ces services démarrent et se ferment à la demande, mais ils ne démarreront que si une application configurée comme **`distributable`** est déployée sur les serveurs.

Infinispan est fourni comme fournisseur de cache dans JBoss EAP 6. Infinispan gére le clustering et la réplication des caches de JBoss EAP 6.

Rapporter un bogue

19.1.2. Composants pouvant bénéficier de la haute disponibilité (HA)

La haute disponibilité (HA) tombe dans un certain nombre de larges catégories de JBoss EAP 6.

Le conteneur

Plusieurs instances de JBoss EAP 6 (exécutant en tant que serveur autonome) ou les membres d'un groupe de serveurs (exécutant en tant que domaine géré) peuvent être configurés pour être hautement disponibles. Cela signifie que si une instance ou un membre est arrêté ou disparaît du groupement, sa charge de travail sera déplacée vers un père. La charge de travail peut être gérée de manière à fournir

une fonctionnalité d'équilibrage de la charge, afin que les serveurs ou les groupes de serveurs avec des ressources plus ou moins supérieures puissent prendre une part plus importante à la charge de travail, ou qu'une capacité supplémentaire puisse être ajoutée pendant les périodes de forte charge.

Le serveur web

Le serveur web lui-même peut être groupé pour HA, à l'aide d'un des mécanismes d'équilibrage de charge compatible. Le plus souple est le connecteur **`mod_cluster`**, qui est intégré dans le conteneur de JBoss EAP. Les autres possibilités incluent les connecteurs Apache **`mod_jk`** ou **`mod_proxy`**, ou les connecteurs ISAPI et NSAPI.

L'application

Les application déployées peuvent être rendues HA (Highly Available) à cause de la spécification Java Enterprise Edition 6 (Java EE 6). Les EJB de session avec ou sans état peuvent être clusterisés, de façon à ce que si le nœud impliqué dans le travail disparaît, un autre nœud prendra sa place, et dans le cas de beans de session avec état, préservera l'état.

Rapporter un bogue

19.1.3. Connecteurs HTTP - Aperçu général

JBoss EAP 6 a la possibilité d'utiliser des mécanismes d'équilibrage de charge et de haute disponibilité intégrés à des serveurs web externes, tels que Apache Web Server, IIS de Microsoft et Oracle iPlanet. JBoss EAP 6 communique avec le serveur web externe à l'aide d'un connecteur HTTP. Ces connecteurs HTTP sont configurés dans le sous-système de web de JBoss EAP 6.

Les serveurs web incluent des modules informatiques qui contrôlent la façon dont les requêtes HTTP sont routées vers les nœuds de worker de JBoss EAP 6. Chacun de ces modules varie dans la façon dont il fonctionne et comment il est configuré. Les modules sont configurés pour équilibrer les charges de travail entre plusieurs nœuds de serveur de JBoss EAP 6, pour déplacer des charges de travail vers d'autres serveurs dans le cas d'échec, ou les deux. Ces fonctions sont appelées *équilibrage de charge* et *Haute disponibilité (HA)*.

JBoss EAP 6 prend en charge un certain nombre de connecteurs HTTP. Celui que vous choisirez dépendra du serveur web que vous utilisez et de la fonctionnalité dont vous aurez besoin.

Le tableau ci-dessous liste les différences entre les différents connecteurs HTTP compatibles avec JBoss EAP 6. Pour obtenir les dernières informations sur les configurations prises en charge pour les connecteurs HTTP, voir https://access.redhat.com/site/articles/111663.

Tableau 19.1. Caractéristiques et contraintes des connecteurs HTTP

Connecteur	Web server	Systèmes d'exploitation pris en charge	Protocoles pris en charge	S'adapte au statut de déploiement	Prend en charge une sticky session

Connecteur	Web server	Systèmes d'exploitation pris en charge	Protocoles pris en charge	S'adapte au statut de déploiement	Prend en charge une sticky session
mod_cluster	httpd dans JBoss Enterprise Web Server, httpd fourni par un système d'exploitation (Red Hat Enterprise Linux, Hewlett-Packard HP-UX)	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris, Hewlett-Packard HP-UX	HTTP, HTTPS, AJP	Oui. Détecte le déploiement et l'annulation du déploiement d'applications et décide dynamiquement s'il faut diriger les demandes clients vers un serveur basé sur la question de savoir si l'application est déployée sur ce serveur.	Oui
mod_jk	httpd dans JBoss Enterprise Web Server, httpd fourni par un système d'exploitation (Red Hat Enterprise Linux, Hewlett-Packard HP-UX)	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris, Hewlett-Packard HP-UX	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui
mod_proxy	httpd dans JBoss Enterprise Web Server	Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris	HTTP, HTTPS, AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui
ISAPI	Microsoft IIS	Microsoft Windows Server	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui

Connecteur	Web server	Systèmes d'exploitation pris en charge	Protocoles pris en charge	S'adapte au statut de déploiement	Prend en charge une stick y session
NSAPI	Oracle iPlanet Web Server	Oracle Solaris	AJP	Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application.	Oui

Pour en savoir plus sur les connecteurs HTTP

- Section 19.5.1, « Le connecteur HTTP `mod_cluster` »
- Section 19.6.1, « Le connecteur Apache mod_jk HTTP »
- Section 19.7.1, « Le connecteur Apache mod_proxy HTTP »
- Section 19.8.1, « Internet Server API (ISAPI) HTTP Connector »
- Section 19.9.1, « Netscape Server API (NSAPI) HTTP Connector »

JBoss EAP 6 prend en charge les configurations disponibles ici : https://access.redhat.com/site/articles/111663.

Rapporter un bogue

19.1.4. Nœud de worker

Nœud de connecteur HTTP

Un *worker node*, connu sous le simple nom de *node*, est un serveur JBoss EAP 6 qui accepte des requêtes d'un ou plusieurs serveurs Web faisant face au client. JBoss EAP 6 peut accepter des requêtes de son propre serveur Web, tel quApache HTTP Server, Microsoft IIS, ou Oracle iPlanet Web Server.

Pour avoir un aperçu des connecteurs HTTP pris en charge par JBoss EAP 6 et sur la façon de les configurer, voir Section 19.1.3, « Connecteurs HTTP - Aperçu général ».

Noeud de cluster

Un nœud de cluster est un membre d'un groupement de serveurs. Un tel cluster peut être en équilibrage de charge, en haute disponibilité, ou les deux. Dans un cluster d'équilibrage de charge, un gestionnaire central distribue également la charge de travail parmi ses nœuds, par mesure d'égalité suivant la situation particulière. Dans un cluster de haute disponibilité (HA), certains nœuds travaillent activement, tandis que d'autres sont en attente d'intervenir si un des nœuds actifs quitte le cluster.

Rapporter un bogue

19.2. CONFIGURATION DE CONNECTEUR

19.2.1. Définir les pools de thread pour le connecteur HTTP dans JBoss EAP 6

Résumé

Les pools de threads de JBoss EAP 6 peuvent être partagés parmi les différents éléments à l'aide du modèle Executor. Ces pools peuvent être partagés non seulement par les différents connecteurs (HTTP), mais aussi par d'autres composants de JBoss EAP 6 qui prennent en charge le modèle Executor. Obtenir le pool de threads de connecteurs HTTP qui corresponde à vos exigences de performance web actuel est un travail délicat qui nécessite une étroite surveillance du pool de threads en cours et des exigences de charge web en cours ou anticipées. Dans cette tâche, vous allez apprendre à définir un pool de threads de connecteur HTTP en utilisant le modèle Executor. Vous apprendrez comment définir cela en utilisant à la fois l'interface en ligne de commande et en modifiant le fichier de configuration XML.

Procédure 19.1. Installation d'un pool de threads pour un connecteur HTTP

1. **Définir une usine de threads**
   Ouvrir votre fichier de configuration (**`standalone.xml`** si vous modifiez un serveur autonome ou **`domain.xml`** si vous modifiez une configuration basée domaine. Ce fichier se trouve dans le dossier ***`EAP_HOME`*`/standalone/configuration`** ou dans ***`EAP_HOME`*`/domain/configuration`**).

   Ajouter l'entrée de sous-système suivante, en modifiant les valeurs en fonction de vos besoins de serveur.

   ```
   <subsystem xmlns="urn:jboss:domain:threads:1.0">
       <thread-factory name="http-connector-factory" thread-name-
   pattern="HTTP-%t" priority="9" group-name="uq-thread-pool"/>
   </subsystem>
   ```

   Si vous préférez utiliser le CLI pour cette tâche, alors exécutez la commande suivante dans une invite de commande CLI :

   ```
   [standalone@localhost:9999 /] ./subsystem=threads/thread-
   factory=http-connector-factory:add(thread-name-pattern="HTTP-%t",
   priority="9", group-name="uq-thread-pool")
   ```

2. **Créer un exécuteur**
   Vous pouvez utiliser une des six classes d'exécuteur intégrées pour qu'elle agisse en tant qu'exécuteur pour cette usine :

   - **`unbounded-queue-thread-pool`** : Ce type de pool de threads accepte toujours les tâches. Si une valeur moindre que la valeur maximale de threads est en cours d'exécution, un nouveau thread démarre pour exécuter la tâche soumise. Sinon, la tâche sera placée dans une file d'attente FIFO illimitée qui sera exécutée lorsqu'un thread sera disponible.

     NOTE

     Le type d'exécuteur en single-thread fourni par **`Executors.singleThreadExecutor()`** est essentiellement un exécuteur de file d'attente illimitée avec une limite de thread d'un. Ce type d'exécuteur est déployé par l'élément **`unbounded-queue-thread-pool-executor`**.

   - **`bounded-queue-thread-pool`** : ce type d'exécuteur maintient une file d'attente de

longueur fixe et deux tailles de pool : une taille de base **`core`** et une taille maximale **`maximum`**. Lorsqu'une tâche est acceptée, si le nombre de pool de threads en cours d'exécution est inférieur à la taille de **`core`**, un nouveau thread est démarré pour exécuter la tâche. Si l'espace reste dans la file d'attente, la tâche est placée dans la file d'attente. Si le nombre de pool de threads en cours d'exécution est inférieur à la taille **`maximum`**, un nouveau thread est démarré pour exécuter la tâche. Si un blocage est activé sur l'exécuteur, le thread appelant se bloque jusqu'à ce que l'espace soit rendu disponible dans la file d'attente. La tâche est déléguée à l'exécuteur de la procédure de transfert si un exécuteur de la procédure de transfert est configuré. Dans le cas contraire, la tâche sera rejetée.

- **`blocking-bounded-queue-thread-pool`** : un exécuteur de pool de thread avec une file d'attente délimitée où les tâches de soubmission des threads peuvent bloquer. Un tel pool de threads a une taille de base et une taille maximum, ainsi qu'une longueur de la file d'attente spécifiée. Lorsqu'une tâche est soumise, si le nombre de threads en cours d'exécution est inférieur à la taille de base, un nouveau thread sera créé. Sinon, s'il y a de la place dans la file d'attente, la tâche sera mise en file d'attente. Dans le cas contraire, si le nombre de threads en cours d'exécution est inférieur à la taille maximale, un nouveau thread sera créé. Dans le cas contraire, l'appelant sera bloqué jusqu'à ce que de la place se libère dans la file d'attente.

- **`queueless-thread-pool`** : parfois, il faut un pool de threads simple pour exécuter des tâches dans des threads séparés, en réutilisant les threads ayant terminé leurs tâches sans aucune file d'attente intermédiaire. Ce type de pool est idéal pour le traitement des tâches qui sont longues, en utilisant sans doute des E/S bloquantes, puisque les tâches sont toujours démarrées immédiatement après l'acceptation plutôt que d'accepter une tâche et puis retarder son exécution avant que d'autres tâches en cours d'exécution soient terminées. Ce type d'exécuteur est déclaré à l'aide de l'élément **`queueless-thread-pool-executor`**.

- **`blocking-queueless-thread-pool`** : un exécuteur de pool de thread sans file d'attente où les tâches de submission de threads puissent bloquer. Lorsqu'une tâche est soumise, si le nombre de threads en cours d'exécution est inférieur à la taille maximum, un nouveau thread sera créé. Dans le cas contraire, l'appelant sera bloqué jusqu'à ce qu'un autre thread se termine et accepte le nouveau.

- **`scheduled-thread-pool`** : il s'agit d'un type spécial d'exécuteur dont le but est d'exécuter des tâches à des moments et intervalles précis, basés sur la classe **`java.util.concurrent.ScheduledThreadPoolExecutor`**. Ce type d'exécuteur est configuré dans l'élément **`scheduled-thread-pool-executor`** :

Dans cet exemple, nous utiliserons **`unbounded-queue-thread-pool`** comme exécuteur. Modifier les valeurs des paramètres ***`max-threads`*** et ***`keepalive-time`*** selon les besoins de votre serveur.

```
<unbounded-queue-thread-pool name="uq-thread-pool">
  <thread-factory name="http-connector-factory" />
  <max-threads count="10" />
  <keepalive-time time="30" unit="seconds" />
</unbounded-queue-thread-pool>
```

Ou bien, si vous préférez utiliser le CLI :

```
[standalone@localhost:9999 /] ./subsystem=threads/unbounded-queue-
thread-pool=uq-thread-pool:add(thread-factory="http-connector-
factory", keepalive-time={time=30, unit="seconds"}, max-threads=30)
```

3. **Forcez le connecteur web HTTP à utiliser ce pool de threads**
   Dans le même fichier de configurations, cherchez l'élément de connecteur HTTP dans le sous-système web et modifiez-le en utilisant le pool de threads défini dans les étapes suivantes.

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http" executor="uq-thread-pool" />
```

   Si vous préférez utiliser le CLI :

```
[standalone@localhost:9999 /] ./subsystem=web/connector=http:write-
attribute(name=executor, value="uq-thread-pool")
```

4. **Redémarrer le serveur**
   Redémarrer le serveur (autonome ou domaine) pour que les changements puissent prendre effet. Utiliser les commandes CLI suivantes pour confirmer si les changements des étapes ci-dessus ont eu lieu :

```
[standalone@localhost:9999 /] ./subsystem=threads:read-
resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "blocking-bounded-queue-thread-pool" => undefined,
        "blocking-queueless-thread-pool" => undefined,
        "bounded-queue-thread-pool" => undefined,
        "queueless-thread-pool" => undefined,
        "scheduled-thread-pool" => undefined,
        "thread-factory" => {"http-connector-factory" => {
            "group-name" => "uq-thread-pool",
            "name" => "http-connector-factory",
            "priority" => 9,
            "thread-name-pattern" => "HTTP-%t"
        }},
        "unbounded-queue-thread-pool" => {"uq-thread-pool" => {
            "keepalive-time" => {
                "time" => 30L,
                "unit" => "SECONDS"
            },
            "max-threads" => 30,
            "name" => "uq-thread-pool",
            "thread-factory" => "http-connector-factory"
        }}
    }
}
[standalone@localhost:9999 /] ./subsystem=web/connector=http:read-
resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
```

```
            "enable-lookups" => false,
            "enabled" => true,
            "executor" => "uq-thread-pool",
            "max-connections" => undefined,
            "max-post-size" => 2097152,
            "max-save-post-size" => 4096,
            "name" => "http",
            "protocol" => "HTTP/1.1",
            "proxy-name" => undefined,
            "proxy-port" => undefined,
            "redirect-port" => 443,
            "scheme" => "http",
            "secure" => false,
            "socket-binding" => "http",
            "ssl" => undefined,
            "virtual-server" => undefined
        }
    }
```

Résultat

Vous avez pu créer une usine de threads et un exécuteur, tout en modifiant votre connecteur HTTP pour qu'il utilise ce pool de threads.

Rapporter un bogue

19.3. CONFIGURATION DU SERVEUR WEB

19.3.1. Le serveur Apache HTTP Autonome

JBoss EAP 6 est testé et pris en charge avec le serveur Apache HTTP qui est inclus avec les versions Red Hat Enterprise Linux 6 certifiées. Le serveur Apache HTTP est également disponible pour les autres configurations, telles que Microsoft Windows Server. Cependant, comme le serveur Apache HTTP est un produit distinct, produit de la Fondation Apache, il était difficile auparavant d'être certain que la version de serveur Apache HTTP qu'un client utilisait était compatible avec JBoss EAP.

Le serveur Apache HTTP Autonome est maintenant disponible en tant que téléchargement séparé dans JBoss EAP 6. Ceci simplifie l'installation et la configuration dans les environnements autres que Red Hat Enterprise Linux, ou sur des systèmes qui déjà ont une configuration de serveur Apache HTTP et qui souhaitent utiliser une instance distincte pour les applications web. Vous pouvez télécharger ce serveur HTTP Apache en tant que téléchargement distinct dans le Portail de Services Client, qui se trouve dans la liste des déchargements de JBoss EAP 6 disponibles pour votre plate-forme d'installation.

Rapporter un bogue

19.3.2. Installer le serveur Apache HTTP inclus dans JBoss EAP 6

Conditions préalables

- Accès root-level ou admin.
- Une version prise en charge de Java a été installée.
- Les packages suivants ont été installés :

- krb5-workstation
- mod_auth_kerb
- elinks (requis pour la fonctionnalité apachectl)

- L'APR (Apache Portability Runtime) doit être installé. Sous Red Hat Enterprise Linux, installer le package **`apr-util-devel`**.

NOTE

Dans Red Hat Enterprise Linux 7, **`apr-util-ldap`** doit être installé pour que l'authentification LDAP fonctionne.

NOTE

Pour obtenir des informations sur l'installation du serveur Apache HTTP dans un environnement de serveur Microsoft Windows, voir le paragraphe *Configuring the Environment* qui se trouve dans la section *Installing Enterprise Web Server on Windows* du guide *JBoss Enterprise Web Server 2 Installation Guide*.

Procédure 19.2. Installer le serveur Apache HHTP

1. **Naviguer dans la liste des téléchargements de JBoss EAP de votre plateforme dans le portail clients de Red Hat.**
   Conenctez-vous au portail clients à l'adresse suivante https://access.redhat.com. Cliquer sur **`Downloads`**, puis **`Red Hat Enterprise Application Platform`** dans la liste de **`Product Downloads`**. Sélectionner la version JBoss EAP du menu déroulant **`Version`**.

2. **Sélectionner le binaire httpd de la liste.**
   Cherchez l'option **`Apache HTTP Server`** pour votre système d'exploitation et votre architecture. Cliquer sur le lien **`Download`**. Un fichier ZIP qui contient la distribution HTTP se télécharge dans votre ordinateur.

3. **Extraire le Zip dans le système où le binaire du serveur Apache HTTP exécutera.**
   Extraire le fichier Zip sur votre serveur préféré à un emplacement temporaire. Le fichier Zip contiendra le répertoire **`httpd`** sous le dossier *jboss-ews-version-number*. Copier le dossier **`httpd`** et le placer à l'intérieur du répertoire où vous avez installé JBoss EAP 6, couramment appelé *EAP_HOME*.

   Votre serveur Apache HTTP se trouve maintenant dans le répertoire ***`EAP_HOME`*`/httpd/`**. Vous pouvez maintenant utiliser cet emplacement pour *HTTPD_HOME*, comme dans les autres documentations JBoss EAP 6.

4. **Exécuter le script de post-installation et créer un utilisateur apache et des comptes de groupe.**
   Dans un émulateur de terminal, passer sur le compte utilisateur root, naviguer vers le répertoire **`EAP_HOME/httpd`** et exécuter la commande suivante.

   ```
   ./.postinstall
   ```

   Ensuite, vérifier si un utilisateur appelé **apache** existe sur le système en exécutant la commande suivante :

```
id apache
```

Si l'utilisateur n'existe pas, il devra être ajouté avec l'utilisateur approprié. Pour cela, exécuter la commande suivante :

```
/usr/sbin/groupadd -g 91 -r apache 2> /dev/null || :
/usr/sbin/useradd -c "Apache" -u 48 -g 91 -s /sbin/nologin -r apache
2>
/dev/null || :
```

Une fois complété, si l'utilisateur **apache** compte exécuter le service httpd, la propriété des répertoires HTTP devra être changée pour indiquer ceci :

```
chown -R apache:apache httpd
```

Pour vérifier que les commandes ci-dessus ont été exécutées avec succès, vérifier que l'utilisateur **apache** possède une permission d'exécution pour le chemin d'installation du serveur Apache HTTP.

```
ls -l
```

Le résultat doit correspondre à cela :

```
drwxrwxr-- 11 apache apache 4096 Feb 14 06:52 httpd
```

5. **Configurer le serveur Apache HHTP.**
   Passer au nouveau compte utilisateur en utilisant la commande suivante :

   ```
   sudo su apache
   ```

   Puis configurer le serveur Apache HTTP comme utilisateur **apache** pour répondre aux besoins de votre organisation. Vous pouvez utiliser la documentation disponible à partir de la Apache Foundation à l'adresse http://httpd.apache.org/ pour un guide général.

6. **Démarrez le serveur HTTP Apache.**
   Démarrer le serveur Apache HTTP par la commande suivante :

   ```
   EAP_HOME/httpd/sbin/apachectl start
   ```

7. **Stopper le serveur Apache HTTP.**
   Pour stopper le serveur Apache HTTP, lancer la commande suivante :

   ```
   EAP_HOME/httpd/sbin/apachectl stop
   ```

Rapporter un bogue

19.3.3. Installer le serveur Apache HTTP dans Red Hat Enterprise Linux (RHEL) 5, 6, et 7 (RPM)

Conditions préalables

- Accès root-level
- Une version de Java prise en charge
- La dernière version du package elinks installé (requise pour la fonctionnalité apachectl)
- Inscrivez-vous sur les canaux Red Hat Enterprise Linux (RHEL) (pour installer le serveur Apache HTTP à partir de canaux RHEL).
- Abonnez-vous au canal **jbappplatform-6-ARCH-server-VERS-rpm** Red Hat Network (pour installer la distribution spéciale EAP du serveur Apache HTTP).

Vous pouvez installer le serveur Apache HTTP en utilisant une des méthodes suivantes :

- À partir des canaux Red Hat Enterprise Linux (RHEL) : un abonnement actif aux canaux Red Hat Enterprise Linux (RHEL) est obligatoire pour installer le serveur Apache HTTP.
- À partir du canal **jbappplatform-6-ARCH-server-VERS-rpm** (JBoss EAP specific distribution) : JBoss EAP distribue sa propre version du serveur Apache HTTP. Un abonnnement actif au canal **jbappplatform-6-ARCH-server-VERS-rpm** est obligatoire pour installer la distribution spécifique de JBoss EAP du serveur Apache HTTP.

Procédure 19.3. Installer et configurer le serveur Apache HTTP dans Red Hat Enterprise Linux 5 et 6 (RPM)

1. **Installer `httpd`**
   Pour installer la version spécifique de JBoss EAP du package **`httpd`**, exécutez la commande suivante :

   ```
   yum install httpd
   ```

   Pour installer **`httpd`** explicitement à partir des canaux Red Hat Enterprise Linux (RHEL), exécutez la commande suivante :

   ```
   yum install httpd --disablerepo=jbappplatform-6-*
   ```

   NOTE

   Vous devez exécuter une des commandes ci-dessus uniquement pour installer **`httpd`** sur votre système.

2. **Définir le comportement de démarrage du service**
   Vous pouvez définir le comportement de démarrage du service **`httpd`** en ligne de commande ou par l'outil graphique de configuration du service. Exécutez la commande suivante pour définir le comportement :

   ```
   chkconfig httpd on
   ```

   Pour utilisser l'outil de configuration du service, exécutez la commande suivante et modifiez la configuration du service dans la fenêtre qui s'affiche :

   ```
   system-config-services
   ```

3. **Démarrer `httpd`**
   Démarrer le **`httpd`** par la commande suivante :

   ```
   service httpd start
   ```

4. **Stopper `httpd`**
   Stopper le **`httpd`** par la commande suivante :

   ```
   service httpd stop
   ```

Procédure 19.4. Installer et configurer le serveur Apache HTTP dans Red Hat Enterprise Linux 7 (RPM)

1. **Installer `httpd22`**
   Pour installer la version spécifique de JBoss EAP du package **`httpd22`**, exécutez la commande suivante :

   ```
   yum install httpd22
   ```

2. **Définir le comportement de démarrage du service**
   Exécutez la commande suivante pour démarrer le service **`httpd22`** au démarrage :

   ```
   systemctl enable httpd22.service
   ```

3. **Démarrer `httpd22`**
   Démarrer le **`httpd22`** par la commande suivante :

   ```
   systemctl start httpd22.service
   ```

4. **Stopper `httpd22`**
   Stopper le **`httpd22`** par la commande suivante :

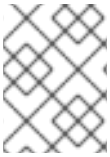
   ```
   systemctl stop httpd22.service
   ```

Rapporter un bogue

19.3.4. Configuration mod_cluster sur httpd

Résumé

mod_cluster est un équilibreur de charges basé httpd. Il utilise un réseau de communication pour envoyer des requêtes de httpd vers un groupe de noeuds de serveur d'application. Les dérivatifs suivants peuvent être définis pour configurer mod cluster sur httpd.



NOTE

Il n'est nul besoin d'utiliser les directives ProxyPass car mod_cluster configure automatiquement les URL qui doivent être envoyés à JBossWEB.

Tableau 19.2. Dérivatifs mod_cluster

Dérivatif	Description	Valeurs
CreateBalancers	Définit comment les équilibreurs de charge sont créés dans les hôtes virtuels httpd. Cela active les directives comme : `ProxyPass /balancer://mycluster1/` .	0: Créer tous les hôtes virtuels httpd 1: Ne pas créer d'équilibreurs (vous aurez besoin d'un ProxyPass ou d'un ProxyMatch au moins pour définir les noms des équilibreurs) 2: Ne créer que le serveur principal Par défaut: 2 **Si vous utilisez la valeur 1, n'oubliez pas de configurer l'équilibreur dans la directive ProxyPass, car la valeur par défaut correspond à une session sticky vide. De plus `nofailover=Off` et les valeurs reçues via message MCMP CONFIG sont ignorées.**
UseAlias	Vérifier que l'alias corresponde bien au nom du serveur.	0: Ignorer les alias 1: Vérifier les alias Par défaut : 0
LBstatusRecalTime	Intervalle d'équilibrage de charge (en secondes) de la logique pour recalculer le statut d'un nœud.	Par défaut : 5 secondes
WaitForRemove	Durée en secondes avant qu'un nœud supprimé soit oublié par httpd.	Par défaut : 10 secondes
ProxyPassMatch/ProxyPass	ProxyPassMatch et ProxyPass sont des directives mod_proxy qui, quand on utilise `!` (à la place de l'url de back-end), évitent un proxy inverse sur le chemin d'accès. Utilisé pour autoriser httpd à fournir des informations statiques comme des images. Ainsi, `ProxyPassMatch ^(/.*\.gif)$ !` L'exemple ci-dessus permet à httpd de servir les fichiers .gif directement.	

Un nœud en hot-standby dans la logique mod_cluster est le dernier nœud vers qui toutes les demandes seront dirigées si tous les autres nœuds sont hors d'opération. Ceci est similaire à la logique hot-standby dans mod_proxy.

Pour configurer un noeud en hot-standby, remplacer le dynamic-load-provider en mod_cluster subsystem avec un simple-load-provider ayant pour facteur 0, comme par exemple :

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.2">
    <mod-cluster-config advertise-socket="modcluster" connector="ajp">
-        <dynamic-load-provider>
-            <load-metric type="busyness"/>
-        </dynamic-load-provider>
+        <simple-load-provider factor="0"/>
    </mod-cluster-config>
</subsystem>
```

Dans la console mod_cluster-manager, le noeud s'affiche avec le statut OK et une charge: 0. Pour plus d'informations, voir la section *Apache mod_cluster-manager Application* dans le guide *JBoss Enterprise Application Platform Development Guide*.

Ainsi, s'il y a trois noeuds :

- Noeud A, Charge : 10
- Noeud B, Charge : 10
- Noeud C, Charge : 10

La charge sera équilibrée entre les noeuds A et B. Si les deux noeuds ne sont pas rendus disponibles, le noeud C prendra la charge.

mod_manager

Le contexte d'une directive mod_manager est l'hôte virtuel dans tous les cas, sauf contre indication. Le contexte **`server config`** implique que la directive doit se trouver en dehors d'une configuration de l'hôte virtuel. Si tel n'est pas le cas, un message erreur apparaîtra et httpd ne démarrera pas.

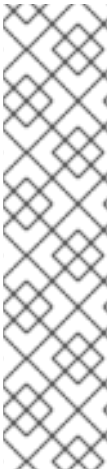
Tableau 19.3. Dérivatifs mod_manager

Dérivatif	Description	Valeurs
EnableMCPMReceive	Autorise l'hôte virtuel à recevoir MCPM des noeuds. Inclure EnableMCPMReceive dans la configuration httpd pour permettre au mod_cluster de fonctionner. Le sauvegarder dans l'hôte virtuel de configuration d'advertise.	

Dérivatif	Description	Valeurs
MemManagerFile	Le nom de base pour les noms que mod_manager utilise pour stocker la configuration, générer des clés pour mémoire partagée ou fichiers verrouillés. Ce doit être un nom de chemin d'accès absolu ; les répertoires seront créés si nécessaire. Il est recommandé que ces fichiers soient placés sur un lecteur local et non pas en NFS share. Context: config serveur	`$server_root/logs/`
Maxcontext	Le nombre maximum de contextes pris en charge par mod_cluster Context: config serveur	Par défaut: 100
Maxnode	Le nombre maximum de nœuds supportés par le mod_cluster. Context: config serveur	Par défaut: 20
Maxhost	Le nombre maximum d'hôtes (alias) supportés par mod_cluster. Inclut également le nombre maximum d'équilibreurs de charge. Context: config serveur	10
Maxsessionid	Le nombre d'ID de sessions actives stockés afin de procurer le nombre de sessions actives du gestionnaire mod_cluster-manager. Une session est inactive quand mod_cluster ne reçoit aucune information de la session pendant 5 minutes. Context: config serveur Ce champ est à but de démonstration et de débogage uniquement.	0: la logique n'est pas activée.
MaxMCMPMaxMessSize	Taille maximum des messages MCMP en provenance d'autres directives max	Calculé sur la base d'autres directives max. Min: 1024
ManagerBalancerName	Le nom que l'équilibreur des charges utilise quand JBoss AS/JBossWeb/Tomcat ne fournit pas de nom d'équilibreur.	`mycluster`

Dérivatif	Description	Valeurs
PersistSlots	Indique à mod_slotmem de persister les nœuds, les alias et les contextes dans des fichiers. Context: config serveur	Off
CheckNonce	Contrôle de la vérification de la valeur unique avec le gestionnaire mod_cluster=manager.	on/off Par défaut : on - Nonce checked
AllowDisplay	Contrôle des affichages supplémentaires sur la page principale du mod_cluster-manager.	on/off Par défaut : off - seule la version sera affichée
AllowCmd	Autorise les commandes qui utilisent l'URL mod_cluster-manager.	on/off Par défaut: on - Les commandes sont autorisées
ReduceDisplay	Réduit le montant d'informations affichées sur la page principale de mod_cluster-manager, afin qu'un plus grand nombre de noeuds puissent être affichés sur la page.	on/off Par défaut : off - les informations complètes s'affichent

SetHandler mod_cluster-manager	Affiche des informations sur le nœud que mod_cluster voit dans le cluster. L'information comprend des informations génériques et compte aussi le nombre de sessions actives.
	
	`<Location /mod_cluster-manager>`
	`SetHandler mod_cluster-manager`
	`Order deny,allow`
	`Allow from 127.0.0.1`
	`</Location>`
		on/off
		
		Par défaut : off



NOTE

Quand on accède à l'emplacement défini dans httpd.conf :

Transferred : Correspond aux données POST envoyées du serveur de back-end.

Connected : Correspond au nombre de requêtes traitées au moment où la page de statuts du mod_cluster a été demandée.

Num_sessions : Correspond au nombre de sessions que le rapport mod_cluster a reporté comme étant inactives (sur lesquelles il y a eu une requête au cours des 5 dernières minutes). Ce champ n'est pas présent quand Maxsessionid est égal à zéro. Ce champ est à but de démonstration et de débogage uniquement.

Rapporter un bogue

19.3.5. Utiliser un serveur web externe comme Web frontal pour les applications JBoss EAP 6.

Aperçu

Pour comprendre les raisons d'utiliser un service web externe comme le serveur web frontal, et pour connaître les avantages et inconvénients des différents connecteurs HTTP pris en charge par JBoss EAP 6, consulter Section 19.1.3, « Connecteurs HTTP - Aperçu général ». Dans certaines situations, vous pouvez utiliser le serveur Apache HTTP de votre système d'exploitation. Sinon, vous pouvez utiliser le serveur Apache HTTP qui est fourni par JBoss Enterprise Web Server.

Une fois que vous aurez décidé quel serveur web ou quel connecteur HTTP utiliser, voir une des procédures suivantes :

- Section 19.3.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 »
- Section 19.5.3, « Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip) »
- Section 19.6.3, « Installer le module jk_mod dans un serveur Apache HTTP (ZIP) »
- Section 19.8.3, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI »
- Section 19.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris »
- Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »

Rapporter un bogue

19.3.6. Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes

Aperçu

JBoss EAP 6 n'a pas besoin de savoir de quel proxy elle accepte les requêtes, uniquement le port et le protocole. Ce n'est pas le cas pour **`mod_cluster`**, qui est davantage lié à la configuration de JBoss EAP 6. En revanche, les tâches suivantes fonctionnent pour **`mod_jk`**, **`mod_proxy`**, **`ISAPI`**, et **`NSAPI`**. Substituer les protocoles et les ports que vous aurez besoin de configurer par ceux des exemples.

Pour configurer JBoss EAP 6 pour **mod_cluster**, consulter Section 19.5.6, « Configurer un nœud de worker de mod_cluster ».

Conditions préalables

- Vous devrez être connecté au Management CLI ou à la console de gestion pour effectuer cette tâche. Les étapes précises utilisent l'interface CLI, mais la même procédure de base est utilisée dans la console de gestion.
- Vous aurez besoin d'une liste des protocoles que vous devrez utiliser, que ce soit HTTP, HTTPS ou AJP.

Procédure 19.5. Modifier la configuration et ajouter les liaisons de socket

1. **Configurer les propriétés de système `jvmRoute`**
   Par défaut, le jvmRoute est sur la même valeur que le nom du serveur. Si vous avez besoin de le personnaliser, vous pouvez utiliser une commande semblable à la suivante. Remplacer ou supprimer la partie de la commande **/profile=ha**, en fonction du profil ou si vous utilisiez un serveur autonome. Remplacez la chaîne **CUSTOM_ROUTE_NAME** par votre nom jvmRoute personnalisé.

   ```
   [user@localhost:9999 /] /profile=ha/subsystem=web:write-
   attribute(name="instance-id",value="CUSTOM_ROUTE_NAME")
   ```

2. **Lister les connecteurs disponibles dans le sous-sytème.**

   NOTE

   Cette étape est nécessaire uniquement si vous n'utilisez pas la configuration **autonome-ha.xml** pour un serveur autonome, ou les profils **ha** ou **full-ha** d'un groupe de serveurs dans un domaine géré. Ces configurations ont déjà tous les connecteurs nécessaires.

   Pour qu'un service de serveur web externe puisse se connecter au serveur web de JBoss EAP 6, le sous-système web doit avoir un connecteur. Chaque protocole a besoin de son propre connecteur, lié à un groupe de sockets.

   Pour avoir la liste des connecteurs actuellement disponibles, lancer la commande suivante :

   ```
   [standalone@localhost:9999 /] /subsystem=web:read-children-
   names(child-type=connector)
   ```

   S'il n'y a aucune ligne sur le connecteur dont vous avez besoin (HTTP, HTTPS, AJP), vous devrez ajouter un connecteur.

3. **Lire la configuration d'un connecteur.**
   Pour voir les détails de configuration d'un connecteur, vous pourrez lire sa configuration. La commande suivante lit la configuration du connecteur AJP. Les autres connecteurs ont des sorties de configuration semblables.

   ```
   [standalone@localhost:9999 /] /subsystem=web/connector=ajp:read-
   resource(recursive=true)
   {
   ```

```
    "outcome" => "success",
    "result" => {
        "enable-lookups" => false,
        "enabled" => true,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "protocol" => "AJP/1.3",
        "redirect-port" => 8443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "ajp",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

4. **Ajouter les connecteurs utiles au sous-système web.**
   Pour ajouter un connecteur au sous-système web, il doit y avoir une liaison de sockets. La liaison de sockets est ajoutée au groupe de liaisons de sockets utilisées par votre serveur ou groupe de serveurs. Les étapes suivantes supposent que votre groupe de serveurs est **`server-group-one`** et que votre groupe de liaisons de sockets est **`standard-sockets`**.

   a. **Ajouter une liaison au groupe de liaisons de sockets.**
      Pour ajouter un groupe de liaison de sockets, lancer la commande suivante, en remplaçant le protocole et le port par ceux dont vous avez besoin.

      ```
      [standalone@localhost:9999 /] /socket-binding-group=standard-
      sockets/socket-binding=ajp:add(port=8009)
      ```

   b. **Ajouter la liaison de sockets au sous-système web.**
      Lancer la commande suivante pour ajouter un connecteur au sous-système web, en substituant le nom de liaison de socket et le protocole par ceux que vous avez besoin.

      ```
      [standalone@localhost:9999 /]
      /subsystem=web/connector=ajp:add(socket-binding=ajp,
      protocol="AJP/1.3", enabled=true, scheme="http")
      ```

Rapporter un bogue

19.4. CLUSTERING

19.4.1. Utiliser la communication TCP dans le sous-système de clusterisation

Par défaut, les nœuds de cluster surveillent leurs statuts respectifs avec le protocole UDP. Certains réseaux ne permettent que TCP. Dans ce cas, vous pouvez ajouter la pile de protocole **`TCPPING`** à votre configuration et l'utiliser comme mécanisme par défaut. Ces options de configuration sont disponibles en ligne de commandes par l'interface CLI.

Le sous-système **`mod_cluster`** utilise également la communication UDP par défaut, et vous pouvez opter pour TCP également.

Voir les deux procédures suivantes pour configurer les sous-systèmes mod_cluster et JGroups pour qu'ils puissent utiliser TCP pour la communication réseau :

- Section 19.4.2, « Configurer le sous-système JGroup pour une utilisation TCP »
- Section 19.4.3, « Désactiver les annonces dans le sous-système `mod_cluster`. »

Rapporter un bogue

19.4.2. Configurer le sous-système JGroup pour une utilisation TCP

Par défaut, le sous-système JGroups communique à l'aide de la multidiffusion UDP. Utilisez la procédure suivante pour configurer le sous-système JGroups pour qu'il puisse utiliser la monodiffusion TCP à la place.

Pour configurer le sous-système `mod_cluster` pour qu'il puisse utiliser TCP également, consulter Section 19.4.3, « Désactiver les annonces dans le sous-système `mod_cluster`. ».

1. **Modifier le script suivant selon votre environnement.**
   Copier le script suivant dans un éditeur de texte. Si vous utilisez un profil différent sur un domaine géré, changer le nom du profil. Si vous utilisez un serveur autonome, supprimer la portion `/profile=full-ha` des commandes. Modifier les propriétés figurant au bas de la commande comme suit. Chacune de ces propriétés est facultative.

   initial_hosts

   Une liste des hôtes considérés comme connus, séparés par des virgules, sera à votre disposition pour rechercher l'adhésion de départ.

   port_range

   Si vous le souhaitez, vous pouvez attribuer une plage de ports. Si vous affectez une plage de ports de 2, et que le port initial est 7600, alors TCPPING tentera de contacter chaque hôte sur les ports 7600-7601. Cette propriété est facultative.

   timeout

   Une valeur de timeout facultative, en millisecondes, pour les membres d'un cluster.

   num_initial_members

   Le nombre de nœuds avant qu'un cluster soit considéré comme complet. Cette propriété est facultative.

```
batch
##  If  tcp  is already added then you can remove it  ##
/profile=full-ha/subsystem=jgroups/stack=tcp:remove
/profile=full-ha/subsystem=jgroups/stack=tcp:add(transport={"type"
=>"TCP", "socket-binding" => "jgroups-tcp"})
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=TCPPING)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=MERGE2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=FD_SOCK,socket-binding=jgroups-tcp-fd)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=FD)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=VERIFY_SUSPECT)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
```

```
protocol(type=BARRIER)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.NAKACK)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=UNICAST2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.STABLE)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=pbcast.GMS)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=UFC)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-protocol(type=MFC)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=FRAG2)
/profile=full-ha/subsystem=jgroups/stack=tcp/:add-
protocol(type=RSVP)
/profile=full-ha/subsystem=jgroups:write-attribute(name=default-
stack,value=tcp)
run-batch
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=initial_hos
ts/:add(value="HostA[7600],HostB[7600]")
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=port_range/
:add(value=0)
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=timeout/:ad
d(value=3000)
/profile=full-
ha/subsystem=jgroups/stack=tcp/protocol=TCPPING/property=num_initial
_members/:add(value=3)
```

2. **Exécuter le script en mode de lot.**

AVERTISSEMENT

Les serveurs qui exécutent le profil devront être fermés avant de pouvoir exécuter le fichier de commandes.

Dans un émulateur de terminal, naviguez vers le répertoire contenant le script **`jboss-cli.sh`** et saisir la commande

```
./jboss-cli.sh -c --file=SCRIPT_NAME
```

où le nom de script *SCRIPT_NAME* correspond au nom et au chemin contenant le script.

Résultat

La pile **`TCPPING`** est maintenant disponible pour les sous-systèmes JGroups. Si elle est utilisée, le sous-système JGroups utilisera TCP pour toute la communication de réseau. Pour configurer le sous-système **`mod_cluster`** à utiliser TCP également, consulter Section 19.4.3, « Désactiver les annonces dans le sous-système **`mod_cluster`**. ».

Rapporter un bogue

19.4.3. Désactiver les annonces dans le sous-système `mod_cluster`.

Par défaut, l'équilibreur du sous-système **`mod_cluster`** utilise l'UDP multidiffusion pour annoncer sa présence aux workers d'arrière-plan. Si vous le souhaitez, vous pouvez désactiver les annonces. Utiliser la procédure suivante pour configurer ce comportement.

Procédure 19.6.

1. **Modifier la configuration httpd.**
   Modifier la configuration httpd pour désactiver « server advertising » et pour utiliser un proxy à la place. La liste de proxys est configurée sur le worker, et contient tous les serveurs HTTPS activés-**`mod_cluster`** avec lesquels le worker peut communiquer.

   La configuration de **`mod_cluster`** pour le serveur Web se trouve généralement dans le répertoire **`/etc/httpd/`** ou **`etc/httpd/`** au sein de l'installation httpd, s'il est installé dans un emplacement non standard. Reportez-vous à Section 19.5.3, « Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip) » et Section 19.5.5, « Configurer les propriétés de Server Advertisement de votre serveur web activé par votre mod_cluster » pour plus d'informations sur le fichier lui-même. Ouvrez le fichier contenant l'hôte virtuel qui écoute les requêtes MCPM (à l'aide de la directive **`EnableMCPMReceive`**) et désactiver le serveur d'annonces en remplaçant la directive **`ServerAdvertise`** comme suit.

   ```
   ServerAdvertise Off
   ```

2. **Désactiver les annonces dans le sous-système `mod_cluster` de JBoss EAP 6, et fournir une liste de proxys.**
   Vous pouvez désactiver les annonces du sous-système **`mod_cluster`** et fournir une liste de proxys, en utilisant la console de gestion basée web ou l'interface CLI de lignes de commande. La liste de proxys est utile car le sous-système **`mod_cluster`** ne sera pas en mesure de découvrir les proxys automatiquement si les annonces sont désactivées.

   - **console de gestion**
     Si vous utilisez un domaine géré, vous ne pourrez uniquement configurer que **`mod_cluster`** dans les profils où il est activé, tels que **`ha`** et **`full-ha`**.

     1. Connectez-vous à la console de gestion et sélectionner **`Configuration`** en haut de l'écran. Si vous utilisez un domaine géré, sélectionner le profil **`ha`** ou **`full-ha`** à partir du menu déroulant **`Profile`** en haut et à gauche.

     2. Étendre sur le menu **`Subsystems`**. Étendre **`Web`**, et sélectionner **`mod_cluster`**.

     3. Cliquer sur **`Edit`** sous l'onglet **`Advertising`** dans **`mod_cluster`**. Pour désactiver la publicité, retirer la marque de la case qui se situe à côté d'**`Advertise`**, et cliquer sur **`Save`**.

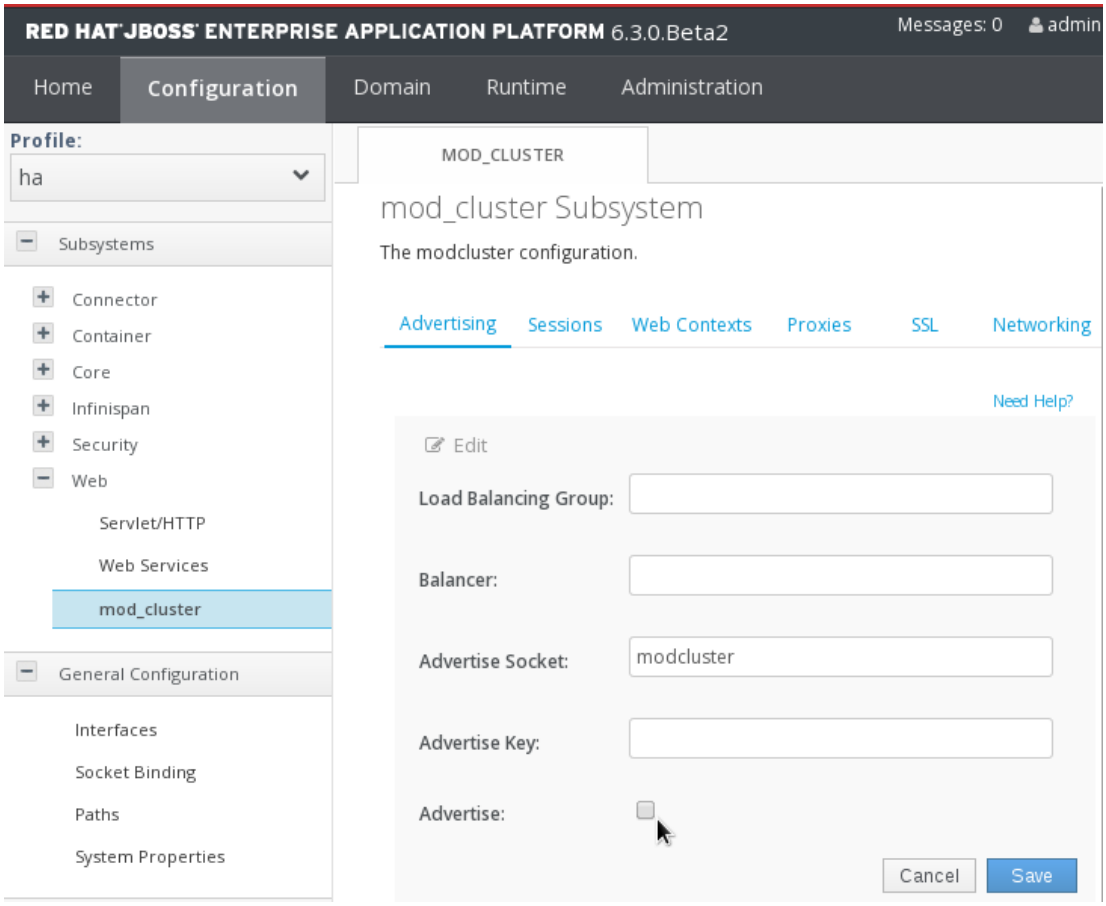


Figure 19.1. Écran de configuration de publicité `mod_cluster`

4. Cliquer sur l'onglet **`Proxies`**. Cliquer sur **`Edit`** et saisir une liste des serveurs proxy dans le champ **`Proxy List`**. La syntaxe qui convient est une liste séparée par des virgules de strings **`HOSTNAME:PORT`**, comme suit :

```
10.33.144.3:6666,10.33.144.1:6666
```

Cliquez sur le bouton **`Enregistrer`** pour terminer.

- **Management CLI**
  Les deux commandes de Management CLI suivantes créent la même configuration que les instructions de la console de gestion ci-dessus. Elles supposent que vous exécutez un domaine géré et que votre groupe de serveurs utilise le profil **`full-ha`**. Si vous utilisez un profil différent, modifier son nom dans les commandes. Si vous utilisez un serveur autonome à l'aide du profil **`standalone-ha`** profil, supprimer la portion **`/profile=full-ha`** des commandes.

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=advertise,value=false)

/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=proxy-
list,value="10.33.144.3:6666,10.33.144.1:6666")
```

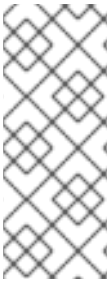
Résultat

L'équilibreur httpd n'annonce plus sa présence aux nœuds de worker et la multidiffusion UDP n'est plus utilisée.

Rapporter un bogue

19.4.4. Passez d'UDP à TCP dans HornetQ Clustering

L'exemple suivant utilise le fichier standalone-full-ha.xml par défaut fourni dans EAP 6.



NOTE

Si la sécurité est activée, vous devrez définir l'attribut cluster-password :

```
<cluster-
password>${jboss.messaging.cluster.password:ChangeMe>}</cluster
-password>
```

1. **Supprimer les broadcast-groups et les discovery-groups :**

```
<broadcast-groups>
    <broadcast-group name="bg-group1">
        <socket-binding>messaging-group</socket-binding>
        <broadcast-period>5000</broadcast-period>
        <connector-ref>netty</connector-ref>
    </broadcast-group>
</broadcast-groups>
<discovery-groups>
    <discovery-group name="dg-group1">
        <socket-binding>messaging-group</socket-binding>
        <refresh-timeout>10000</refresh-timeout>
    </discovery-group>
</discovery-groups
```

2. **En option, supprimer la liaison de socket de "messaging-group" :**

```
<socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
```

3. **Configurez les connecteur(s) Netty qui conviennent - un pour chacun des autres noeuds du cluster.**
   Ainsi, si le cluster a 3 noeuds, alors configurez 2 connecteurs Netty, etc. Si le cluster a 2 noeuds, alors configurez 1 connecteur Netty, etc. Voici un exemple de configuration d'un cluster 3-noeuds :

```
<netty-connector name="other-cluster-node1" socket-binding="other-
cluster-node1"/>
<netty-connector name="other-cluster-node2" socket-binding="other-
cluster-node2"/>
```

4. **Configurer les liaisons de sockets correspondantes.**

NOTE

La substitution de propriété système peut être utilisée soit pour "hôte", soit pour "port" selon les besoins.

```
<outbound-socket-binding name="other-cluster-node1">
    <remote-destination host="otherNodeHostName1" port="5445"/>
</outbound-socket-binding>
<outbound-socket-binding name="other-cluster-node2">
    <remote-destination host="otherNodeHostName2" port="5445"/>
</outbound-socket-binding>
```

5. **Configurer la connexion au cluster pour qu'elle utilise ces connecteurs à la place du discovery-group utilisé par défaut :**

```
<cluster-connection name="my-cluster">
    <address>jms</address>
    <connector-ref>netty</connector-ref>
    <static-connectors>
        <connector-ref>other-cluster-node1</connector-ref>
        <connector-ref>other-cluster-node2</connector-ref>
    </static-connectors>
</cluster-connection>
```

Ce processus devra être répété sur chaque noeud du groupement pour que chaque noeud ait des connecteurs sur chaque noeud du cluster.

NOTE

Ne pas configurer un noeud avec une connexion sur lui-même. Cela est considéré comme une erreur de configuration.

Rapporter un bogue

19.5. WEB, CONNECTEURS HTTP, ET HTTP CLUSTERING

19.5.1. Le connecteur HTTP `mod_cluster`

mod_cluster est le module qui permet l'équilibrage des charges dans le conteneur de JBoss Web. On l'appelle le *connector*. Le choix de connecteur dépend du conteneur web que vous choisissez d'utiliser avec JBoss EAP 6. Pour en savoir plus sur les autres connecteurs, voir ce qui suit :

- Section 19.6.1, « Le connecteur Apache mod_jk HTTP »
- Section 19.8.1, « Internet Server API (ISAPI) HTTP Connector »
- Section 19.9.1, « Netscape Server API (NSAPI) HTTP Connector »

Le mod_cluster possède plusieurs avantages.

- *mod_cluster Management Protocol (MCMP)* est un lien supplémentaire entre les nœuds de serveur d'applications et Apache Web Server (connu sous l'appellation "the web server"), utilisé par les nœuds de serveur d'applications pour transmettre des facteurs d'équilibrage des charges

côté serveur et des événements de cycle de vie de retour vers le serveur web via un ensemble personnalisé de méthodes HTTP.

- La configuration dynamique des serveurs web permettent à JBoss EAP 6 de s'adapter sur le champ, sans besoin de configuration supplémentaire.
- JBoss EAP 6 se charge des calculs de factorisation d'équilibre des charges, au lieu de s'en remettre au serveur web. Cela rend les métriques d'équilibrage des charges plus précis qu'avec les autres connecteurs.
- mod_cluster offre un contrôle précis du cycle de vie. Chaque serveur transmet tout événement de cycle de vie du contexte d'application web au serveur web, l'informant de démarrer ou d'arrêter les demandes de routage dans un contexte donné. Ceci empêche les utilisateurs finaux de voir les erreurs HTTP en raison des ressources non disponibles.
- Les transports AJP, HTTP ou HTTPS peuvent être utilisés.

Rapporter un bogue

19.5.2. Configurer le sous-système `mod_cluster`

Dans la console de gestion, les options de **`mod_cluster`** sont disponibles dans la zone de configuration du sous-système Web. Cliquer sur l'onglet **`Configuration`**. Si vous utilisez un domaine géré, sélectionnez le bon profil pour configurer dans la boîte de sélection de **`Profile`**. Par défaut, les profils **`ha`** et **`full-ha`** ont le sous-système **`mod_cluster`** activé. Si vous utilisez un serveur autonome, vous devez utiliser le profil **`standalone-ha`** ou **`standalone-full-ha`** pour démarrer le serveur. Étendrer le menu **`Web`** et choisissez **`mod_cluster`**. Les options sont expliquées dans les tableaux ci-dessous. La configuration générale est indiquée en premier, suivie de la configuration de sessions, les contextes web, proxy, SSL et réseautage. Chacune d'elles possède son propre onglet dans l'écran de configuration de **`mod_cluster`**

NOTE

La page de configuration **`mod_cluster`** n'est visible que pour les profils avec le sous-système HA Clustering subsystem activé. Ces profils sont **`ha`** et **`full-ha`** pour un domaine géré, ou **`standalone-ha`** et **`standalone-full-ha`** pour un serveur autonome.

Tableau 19.4. Options de configuration `mod_cluster`

Option	Description	CLI Command
Load Balancing Group	Si non nulles, les requêtes devront être envoyées vers un groupe d'équilibrage de charges sur l'équilibreur de charges. Laisser cet espace vide si vous ne souhaitez pas utiliser ces groupes d'équilibrage des charges.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=load-balancing-group,value=myGroup)`

Option	Description	CLI Command
Balancer	Le nom de l'équilibreur. Doit correspondre à la configuration du proxy httpd.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=balancer,value=myBalancer)`
Socket Advertise	Le nom de la liaison de sockets à utiliser pour les annonces de cluster.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-socket,value=modcluster)`
Advertise Security Key	Une chaîne contenant la clé de sécurité à annonce.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-security-key,value=myKey)`
Advertise	Indique si les annonces sont activées. Valeur par défaut **`true`**.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise,value=true)`

Tableau 19.5. Options de configuration de session `mod_cluster`

Option	Description	CLI Command

Option	Description	CLI Command
Sticky Session	Si vous souhaitez utiliser des sessions pour les demandes. Cela signifie qu'après que le client ait établi une connexion sur un nœud de cluster spécifique, leur transmission ultérieure est routée vers ce même nœud à moins qu'il ne soit plus disponible. La valeur par défaut est **`true`**, qui est le paramètre recommandé.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session,value=true)`
Sticky Session Force	Si sur **`true`**, la demande ne sera pas redirigée vers un nouveau nœud de cluster si son nœud initial n'est plus disponible mais échoue à la place. La valeur par défaut est **`false`**.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-force,value=false)`
Sticky Session Remove	Supprime les informations de la session après le basculement. Cela a comme valeur par défaut **`false`**.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-remove,value=false)`

Tableau 19.6. Options de configuration de contexte web `mod_cluster`

Option	Description	CLI Command
Auto Enable Contexts	Si on doit ajouter de nouveaux contextes à **`mod_cluster`** par défaut ou non. La valeur par défaut **`true`**. Si vous modifiez la valeur par défaut et que vous devez activer le contexte manuellement, l'application web peut activer son contexte à l'aide de la méthode MBean **`enable()`**, ou via le gestionnaire **`mod_cluster`**, une application web qui s'exécute sur le serveur proxy httpd, sur un hôte virtuel nommé ou le port qui est spécifié dans la configuration de cet httpd.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=auto-enable-contexts,value=true)`

Option	Description	CLI Command
Excluded Contexts	Une liste séparée par des virgules de contextes que **mod_cluster** doit ignorer. Si aucun hôte n'est indiqué, l'hôte est censé être **localhost**. **ROOT** indique le contexte racine de l'application web. La valeur par défaut est **ROOT,invoker,jbossws,juddi,console**.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=excluded-contexts,value="ROOT,invoker,jbossws,juddi,console")`

Tableau 19.7. Options de configuration proxy de mod_cluster

Option	Description	CLI Command
Proxy URL	Si défini, cette valeur sera ajoutée à l'URL des commandes MCMP.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-url,value=myhost)`
Proxy List	Une liste séparée par des virgules des adresses proxy httpd, dans le format **hostname:port**. Ceci indique la liste des serveurs proxy avec lesquels le processus de **mod_cluster** va tenter de communiquer au départ.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-list,value="127.0.0.1,127.0.0.2")`

Configurer la communication SSL pour mod_cluster

Par défaut, la communication **mod_cluster** a lieu sur un lien HTTP crypté. Si vous définissez le schéma du connecteur à **HTTPS** (voir Tableau 19.5, « Options de configuration de session **mod_cluster** »), les paramètres ci-dessous indiquent à **mod_cluster** où trouver les informations pour encoder la connexion.

Tableau 19.8. Options de configuration SSL de mod_cluster

Option	Description	CLI Command

Option	Description	CLI Command
ssl	Indique si on doit activer SSL. Valeur par défaut **`false`**.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ssl,value=true)`
Key Alias	Clé alias choisie quand le certificat est créé.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-alias,value=jboss)`
Key Store	L'emplacement où le keystore garde les certificats clients.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store,value=System.getProperty("user.home") + "/.keystore")`
Key Store Type	Le type de keystore	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store-type,value=JKS)`

Option	Description	CLI Command
Key Store Provider	Le fournisseur de keystore	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store-provider,value=IBMJCE)`
Password	Mot de passe choisi quand le certificat est créé.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=password,value=changeit)`
Trust Algorithm	L'algorithme de la fabrique de gestionnaire de confiance	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=trust-algorithm,value=PKIX)`
Cert File	L'emplacement du fichier de certificats.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-certificate-file,value=${user.home}/jboss.crt)`

Option	Description	CLI Command
CRL File	Fichier de la liste de révocation du certificat.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-crl-file,value=${user.home}/jboss.crl)`
Max Certificate Length	La longueur maximum du certificat contenue dans le trust store. Valeur par défaut **5**.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=trust-max-cert-length,value=5)`
Key File	L'emplacement du fichier clé du certificat.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=certificate-key-file,value=${user.home}/.keystore)`
Cipher Suite	La suite cipher d'encodage autorisée.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=cipher-suite,value=ALL)`

Option	Description	CLI Command
Certificate Encoding Algorithms	L'algorithme de la fabrique de gestionnaire de clés.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=encoding-algorithms,value=ALL)`
Revocation URL	L'URL de la liste de révocation de l'autorité de certificat	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-revocation-url,value=jboss.crl)`
Protocol	Les protocoles SSL activés.	`/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=protocol,value=SSLv3)`

Configurer les options de réseautage de `mod_cluster`

Les options de réseautage de **`mod_cluster`** contrôlent des comportements de timeout différents pour des types de services variés avec lesquels le service **`mod_cluster`** communique.

Tableau 19.9. Options de configuration de réseautage de `mod_cluster`

Option	Description	CLI Command

Option	Description	CLI Command
Node Timeout	Timeout (en secondes) des connexions de proxy vers un nœud. C'est que le temps pendant lequel **mod_cluster** attendra la réponse du back-end avant de retourner l'erreur. Cela correspond au délai d'attente de la documentation mod_proxy du worker. La valeur **-1** indique aucun délai d'attente. Notez que **mod_cluster** utilise toujours un cping/cpong avant d'adresser une demande et la valeur **connectiontimeout** utilisée par **mod_cluster** est la valeur de ping.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)`
Socket Timeout	Nombre de millisecondes durant lesquelles patienter avant d'obtenir une réponse d'un proxy httpd suite à des commandes MCMP avant le timeout, et indiquer erreur de proxy.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=socket-timeout,value=20)`
Stop Context Timeout	Durée, mesurée dans les unités spécifiées par stopContextTimeoutUnit, pendant laquelle attendre l'arrêt net d'un contexte (fin des demandes en attente pour un contexte distribuable; ou destruction/expiration des sessions actives pour un contexte non distribuable).	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=stop-context-timeout,value=10)`

Option	Description	CLI Command
Session Draining Strategy	Indique si on doit drainer les sessions avant de retirer le déploiement d'une application web. **DEFAULT** Sessions de drainage avant qu'une application web retire son déploiement si l'application web n'est pas distribuable. **ALWAYS** Toujours drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web distribuables. **NEVER** Ne pas drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web non distribuables.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=session-draining-strategy,value=DEFAULT)`
Max Attempts	Nombre de fois qu'un proxy httpd va tenter d'envoyer une requête donnée à un worker avant d'abandonner. La valeur minimale est **1**, ce qui signifie essayer une seule fois. La valeur par défaut du **module mod_proxy** est également 1, ce qui signifie qu'aucune nouvelle tentative ne se produit.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=max-attempts,value=1)`
Flush Packets	Indique si on doit activer le vidage des paquets dans le serveur web. Valeur par défaut **false**.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-packets,value=false)`

Option	Description	CLI Command
Flush Wait	Durée, en secondes, pendant laquelle on doit attendre le vidage des paquets dans le serveur web. **-1** est la valeur par défaut. A value of **-1** indique une attente indéfinie avant de vider les paquets.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-wait,value=-1)`
Ping	Durée, en secondes, pendant laquelle attendre une réponse au ping d'un noeud de cluster. Valeur par défaut **10** secondes.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ping,value=10)`
SMAX	Le nombre de soft connexions inactives maximales (le même que **smax** dans la documentation du module de worker mod_proxy). La valeur maximale dépend de la configuration de thread httpd et peut être **ThreadsPerChild** ou **1**.	`profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=smax,value=ThreadsPerChild)`
TTL	Time To live (en secondes) pour les connexions inactives au dessus de smax, la valeur par défaut est 60 Quand **nodeTimeout** n'est pas défini, le **Proxy** de la directive **ProxyTimeout** est utilisé. Si **ProxyTimeout** n'est pas défini, alors le **Timeout** sera utilisé. La valeur par défaut est de 300 secondes. **nodeTimeout**, **ProxyTimeout**, et **Timeout** sont définis au niveau socket.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ttl,value=-1)`
Node Timeout	La durée d'attente, en secondes, pour le traitement d'une requête par un processus de travail de serveur web externe. Par défaut, **-1**, ce qui signifie que mod_cluster attend indéfiniment la requête traitée par le worker httpd.	`/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)`

Options de configuration de load provider de `mod_cluster`

Les options de configuration de **`mod_cluster`** suivantes ne sont pas disponibles dans la console de gestion, et ne peuvent être uniquement être définies en utilisant l'interface CLI en ligne de commandes.

Un simple fournisseur de charge est utilisé si aucun processeur de charge dynamique n'est présent. Il donne à chaque membre du cluster un facteur de charge **`1`**, et répartit uniformément les travaux sans prendre en compte un algorithme d'équilibrage de charges. Pour l'ajouter, utilisez la commande CLI suivante :

```
/subsystem=modcluster/mod-cluster-config=configuration/simple-load-
provider:add
```

Un fournisseur de charge dynamique peut être configuré pour utiliser une variété d'algorithmes, en combinaison, pour déterminer quel nœud de cluster recevra la demande suivante. Vous pouvez créer un fournisseur de charge et le configurer pour qu'il soit adapté à votre environnement. et vous pourrez avoir plus d'un métrique de charge active en même temps, en les ajoutant par l'interface CLI. Le fournisseur de charge dynamique par défaut utilise **`busyness`** comme métrique de charge déterminant. Les options de fournisseur de charge dynamique et les métriques de charge possible se trouvent ci-dessous.

Tableau 19.10. Options de load provider dynamic de `mod_cluster`

Option	Description	CLI Command
Decay	Le facteur par lequel les métriques historiques se désintègrent de façon significative.	`/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/:write-attribute(name=decay,value=2)`
History	Le nombre d'enregistrements de métriques de charge historique à considérer pour déterminer la charge.	`/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/:write-attribute(name=history,value=9)`

Option	Description	CLI Command

Load Metric	Le métrique de charge par défaut incluse avec le fournisseur de charge dynamique dans JBoss EAP 6 est **busyness**, qui calcule la charge du nœud de worker en fonction de la quantité de threads dans le pool de threads devant servir les requêtes. Vous pouvez définir la capacité de ce métrique par lequel la charge réelle est divisée : charge calculée / capacité. Vous pouvez définir plusieurs paramètres de charge dans le fournisseur de la charge dynamique.	`/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=capacity,value=1.0)`
		`/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=type,value=busyness)`
		`/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=weight,value=1)`

Load Metric Algorithms

cpu

Le métrique de charge **cpu** utilise la charge CPU moyenne pour déterminer quel nœud de cluster reçoit la charge de travail suivante.

mem

Le métrique de charge **mem** utilise la mémoire native RAM comme facteur de charge. L'utilisation de ce métrique est déconseillée car elle fournit une valeur qui inclut les tampons et le cache. C'est donc toujours un chiffre très faible sur chaque système décent pourvu d'une bonne gestion de mémoire.

heap

Le métrique de charge **heap** utilise l'usage heap (de tas) pour déterminer quel cluster reçoit la charge de travail suivante.

sessions

Le métrique de charge de **session** utilise le nombre de sessions actives comme métrique.

requests

Le métrique de charge **requests** utilise le nombre de requêtes en provenance des clients pour déterminer quel nœud de cluster reçoit la charge de travail suivante. Par exemple, capacité 1000 signifie que 1000 requêtes/s est considéré comme une « pleine charge ».

send-traffic

Le métrique de charge **send-traffic** (trafic envoyé) utilise le volume de trafic envoyé à partir d'un nœud de worker vers les clients. Par ex. une capacité par défaut de 512 indique que le nœud doit être considéré en pleine charge, si le trafic sortant moyen est 512 KB/s ou supérieur.

receive-traffic

Le métrique de charge receive-traffic (réception de trafic) utilise le volume de trafic envoyé vers le nœud de worker en provenance des clients. Par ex. une capacité par défaut de 1024 indique que le nœud doit être considéré en pleine charge, si le trafic entrant moyen est 1024 KB/s ou supérieur.

busyness

Ce métrique représente le nombre de threads d'une pool de threads en train de répondre à des requêtes.

Exemple 19.1. Ajouter un métrique de charge

Pour ajouter un métrique de charge, utiliser la commande **add-metric**.

```
/subsystem=modcluster/mod-cluster-config=configuration/:add-
metric(type=sessions)
```

Exemple 19.2. Définir une valeur de métrique existant

Pour définir la valeur d'un métrique existant, utiliser la commande **write-attribute**.

```
/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-
provider=configuration/load-metric=cpu/:write-
attribute(name="weight",value="3")
```

Exemple 19.3. Modifier une valeur de métrique existant

Pour changer la valeur d'un métrique existant, utiliser la commande **write-attribute**.

```
/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-
provider=configuration/load-metric=cpu/:write-
attribute(name="type",value="busyness")
```

Exemple 19.4. Supprimer un métrique existant

Pour supprimer un métrique existant, utiliser la commande **`remove-metric`**.

```
/subsystem=modcluster/mod-cluster-config=configuration/:remove-
metric(type=sessions)
```

Rapporter un bogue

19.5.3. Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip)

Conditions préalables

- Pour cette tâche, vous devrez utiliser un serveur Apache HTTP installé dans Red Hat Enterprise Linux 6, ou JBoss Enterprise Web Server, ou encore un serveur Apache HTTP autonome comme composant de JBoss EAP 6 à télécharger séparément.
- Si vous avez besoin d'installer un serveur Apache HTTP dans Red Hat Enterprise Linux 6, utiliser les instructions dans *Red Hat Enterprise Linux 6 Deployment Guide*.
- Si vous avez besoin d'installer un serveur Apache HTTP autonome en tant que composant téléchargeable de JBoss EAP 6, consulter Section 19.3.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 ».
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*.
- Télécharger le package **`Webserver Connecter Natives`** correspondant à votre système d'exploitation et architecture depuis le portail client de Red Hat à https://access.redhat.com. Ce paquet contient les modules HTTPD mod_cluster binaires précompilés pour votre système d'exploitation. Après avoir extrait l'archive, les modules se trouvent dans le répertoire **`EAP_HOME/modules/system/layers/base/native/lib/httpd/modules`**.

  Le répertoire **`etc/`** contient quelques exemples de fichiers de configuration et le répertoire **`share/`** contient une documentation supplémentaire.
- Vous devez être connectés avec des privilèges administratifs (root).

NOTE

Si vous utilisez un système 64 bit, les modules binaires de serveur web de mod_cluster se trouveront ici : **`EAP_HOME/modules/system/layers/base/native/lib64/httpd/modules`**. Vous devrez utiliser ce chemin si vous devez accéder aux modules.

Procédure 19.7. Installer le Module mod cluster

1. **Déterminer l'emplacement de votre configuration de serveur Apache HTTP.**
   Votre emplacement de seveur Apache HTTP sera différent selon que vous utilisez Apache HTTP de Red Hat Enterprise Linux, le serveur Apache HTTP autonome inclus comme

composant séparé téléchargeable dans JBoss EAP 6 ou le serveur Apache HTTP disponible dans JBoss Enterprise Web Server. C'est l'une des trois options suivantes qui sera mentionnée au cours de cette tâche sous le nom *HTTPD_HOME*.

- Apache HTTP Server - **`/etc/httpd/`**
- JBoss EAP 6 HTTP Server - cet emplacement est choisi par vous-même sur la base des exigences de votre infrastructure.
- JBoss Enterprise Web Server Apache HTTP Server - ***`EWS_HOME`*`/httpd/`**

2. **Copier les modules dans le répertoire de modules d'Apache HTTP Server.**
Copier les quatre modules (les fichiers qui se terminent par **`.so`**) à partir du répertoire **`EAP_HOME/modules/system/layers/base/native/lib/httpd/modules`** de l'archive extraite Webserver Natives vers le répertoire ***`HTTPD_HOME`*`/modules/`**.

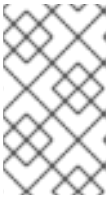
3. **Pour JBoss Enterprise Web Server, désactiver le module `mod_proxy_balancer`.**
Si vous utilisez JBoss Enterprise Web Server, le module **`mod_proxy_balancer`** sera activé par défaut. Il est incompatible avec mod cluster. Pour le désactiver, modifier ***`HTTPD_HOME`*`/conf/httpd.conf`** et décommenter la ligne suivante en mettant le symbole # (hachage) devant la ligne qui charge le module. La ligne apparaîtra sans le commentaire, puis avec, comme ci-dessous.

```
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

```
# LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

Sauvegarder et fermer le fichier.

4. **Configurer le module mod_cluster.**
L'archive Webserver Natives contient un échantillon de fichier **`mod_cluster.conf`** (**`EAP_HOME/modules/system/layers/base/native/etc/httpd/conf`**). Ce fichier peut être utilisé comme guide ou copié et modifié pour créer un fichier ***`HTTPD_HOME`*`/httpd/conf.d/`*`JBoss_HTTP`*`.conf`**.



NOTE

L'utilisation du nom ***`JBoss_HTTP`*`.conf`** est une convention arbitraire de ce document. le fichier de configuration sera chargé, indépendemment de son nom, s'il est enregistré dans le répertoire **`conf.d/`** avec l'extension **`.conf`**.

Ajouter l'entrée suivante à votre fichier de configuration :

```
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so
```

Cela oblige le serveur Apache HTTP à charger les modules dont **`mod_cluster`** a besoin automatiquement pour fonctionner.

5. **Créer un proxy de listener de serveur.**

Continuer à éditer ***HTTPD_HOME*/httpd/conf/JBoss_HTTP.conf** et ajouter la configuration minimale suivante, en remplaçant les valeurs en lettres majuscules par des valeurs adaptées à votre environnement.

```
Listen IP_ADDRESS:PORT
<VirtualHost IP_ADDRESS:PORT>
   <Location />
          Order deny,allow
          Deny from all
          Allow from *.MYDOMAIN.COM
   </Location>

   KeepAliveTimeout 60
   MaxKeepAliveRequests 0
   EnableMCPMReceive On

   ManagerBalancerName mycluster
   ServerAdvertise On

</VirtualHost>
```

Ces directives créent un nouveau serveur virtuel qui écoute sur le port **IP_ADDRESS:PORT**, permet des connexions de **MYDOMAIN.COM** et se présente comme un équilibreur de charge du nom **mycluster**. Ces directives sont traitées en détail dans la documentation pour le serveur Web Apache Server. Pour en savoir plus sur les directives **ServerAdvertise** et **EnableMCPMReceive** ou les implications des annonces de serveur, consulter Section 19.5.5, « Configurer les propriétés de Server Advertisement de votre serveur web activé par votre mod_cluster ».

Enregistrer le fichier et sortir.

6. **Démarrer à nouveau le serveur HTTP Apache.**
   La façon de redémarrer le serveur Apache HTTP dépend de savoir si vous utilisez le serveur Apache HTTP de Red Hat Enterprise Linux ou le serveur HTTP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

   - **Serveur Apache HTTP de Red Hat Enterprise Linux 6**
     Exécuter la commande suivante :

     ```
     [root@host]# service httpd restart
     ```

   - **JBoss Enterprise Web Server HTTP Server**
     JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du serveur Apache HTTP est différente pour chacun.

     - **Red Hat Enterprise Linux**
       Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son serveur Apache HTTP en tant que service. Pour redémarrer le serveur Apache HTTP, lancer les deux commandes suivantes :

       ```
       [root@host ~]# service httpd stop
       [root@host ~]# service httpd start
       ```

- **Microsoft Windows Server**
  Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

```
C:\> net stop httpd
C:\> net start httpd
```

Résultat

Le serveur Apache HTTP est maintenant configuré comme équilibreur de charges, et peut fonctionner avec le sous-système **`mod_cluster`** qui exécute sur JBoss EAP 6. Pour configurer JBoss EAP 6 pour qu'il soit au fait de mod_cluster, consulter Section 19.5.6, « Configurer un nœud de worker de mod_cluster ».

Rapporter un bogue

19.5.4. Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (RPM)

Conditions préalables

- Pour cette tâche, vous devrez utiliser le serveur Apache HTTP installé dans Red Hat Enterprise Linux 6, ou JBoss Enterprise Web Server, ou encore un serveur Apache HTTP autonome comme composant de JBoss EAP 6 à télécharger séparément.
- Si vous avez besoin d'installer un serveur Apache HTTP dans Red Hat Enterprise Linux 6, utiliser les instructions dans *Red Hat Enterprise Linux 6 Deployment Guide*.
- Si vous avez besoin d'installer un serveur Apache HTTP autonome en tant que composant téléchargeable de JBoss EAP 6, consulter Section 19.3.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 » .
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*.
- Vous devez être connectés avec des privilèges administratifs (root).
- Vous devez posséder un abonnement actif au canal RHN **`jbappplatform-6-`*`ARCH`*`-server-`*`VERS`*`-rpm`**.

La méthode d'installation RPM reste la même sur Red Hat Enterprise Linux 5 et 6 et ne requiert que de mineures modifications pour les utilisateurs de Red Hat Enterprise Linux 6 ayant installé un serveur Apache HTTP 2.2.15.

1. Instaler le package **`mod_cluster-native`** avec YUM :

```
yum install mod_cluster-native
```

2. **Apache HTTP Server 2.2.15**:
   - Si vous choisissez de rester sur un serveur HTTP 2.2.15, vous devez désactiver le module **`mod_proxy_balancer`** chargé par défaut en commentant la ligne **`LoadModule proxy_balancer_module`** dans le fichier httpd.conf.

     Vous pouvez modifier le fichier manuellement ou utiliser la commande suivante :

```
sed -i 's/^LoadModule proxy_balancer_module/#LoadModule
proxy_balancer_module/;s/$//' /etc/httpd/conf/httpd.conf
```

- Si vous choisissez d'effectuer une mise à niveau vers le serveur Apache HTTP 2.2.26, installer la dernière version par la commande suivante.

```
yum install httpd
```

3. Pour que le service du serveur Apache HTTP démarre automatiquent, saisissez la commande suivante :
   - Pour Red Hat Enterprise Linux 5 et 6 :

```
service httpd add
```

   - Pour Red Hat Enterprise Linux 7 :

```
systemctl enable httpd22.service
```

4. Démarrer l'équilibreur mod_cluster par la commande suivante :
   - Pour Red Hat Enterprise Linux 5 et 6 :

```
service httpd start
```

   - Pour Red Hat Enterprise Linux 7 :

```
systemctl start httpd22.service
```

Rapporter un bogue

19.5.5. Configurer les propriétés de Server Advertisement de votre serveur web activé par votre mod_cluster

Résumé

Pour obtenir des instructions sur la façon de configurer votre serveur web pour qu'il interagisse avec l'équilibreur de charges, consulter Section 19.5.3, « Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip) ». L'élément de configuration *server advertisement* requiert davantage d'explications.

Quand Server Advertisement est inactif, le serveur web envoie des messages qui contiennent l'adresse IP et le numéro de port spécifié dans l'hôte virtuel du mod_cluster. Pour configurer ces valeurs, consulter Section 19.5.3, « Installer le module mod cluster dans un serveur Apache HTTP ou dans JBoss Enterprise Web Server (Zip) ». Si UDP multicast n'est pas disponible sur votre réseau, ou si vous préférez configurer les worker nodes avec une liste statique de serveurs proxy, vous pouvez désactiver Server Advertisement et configurer les noeuds proxy manuellement. Voir Section 19.5.6, « Configurer un nœud de worker de mod_cluster » pour obtenir des informations sur la façon de configurer un worker node.

Vous devrez modifier le **`httpd.conf`** associé à votre instance de serveur Apache HTTP. Il s'agit le plus souvent de **`/etc/httpd/conf/httpd.conf`** dans Red Hat Enterprise Linux, ou peut-être du répertoire **`etc/`** de votre instance HTTP Apache autonome.

Procédure 19.8. Modifier le fichier httpd.conf et implémenter les changements

1. **Désactivez le paramètre `AdvertiseFrequency`, s'il existe.**
   Si vous voyez une ligne qui ressemble à ceci dans votre énoncé **`<VirtualHost>`**, décommenter cette ligne en ajoutant un signe # (hachage) avant le premier caractère. La valeur ne devra pas correspondre à **5**.

   ```
   AdvertiseFrequency 5
   ```

2. **Ajouter la directive qui permet de désactiver Server Advertisement.**
   Ajouter la directive suivante dans l'énoncé **`<VirtualHost>`** afin de désactiver Server Advertisement.

   ```
   ServerAdvertise Off
   ```

3. **Actionner la possibilité de recevoir des messages MCPM.**
   Ajouter la directive suivante pour permettre au serveur web de recevoir des messages MCPM de la part des worker nodes.

   ```
   EnableMCPMReceive On
   ```

4. **Redémarrer le serveur web.**
   Redémarrer le serveur web en lançant une des commandes suivantes, selon que vous utilisez Red Hat Enterprise Linux ou Microsoft Windows Server.

   - **Red Hat Enterprise Linux**

     ```
     [root@host ]# service httpd restart
     ```

   - **Microsoft Windows Server**

     ```
     C:\> net service http
     C:\> net service httpd start
     ```

Résultat

Le serveur web n'annonce plus l'adresse IP et le port de votre serveur proxy mod_cluster. Pour l'annoncer, vous devez configurer vos worker nodes pour qu'ils utilisent une adresse statique et un port pour communiquer avec le proxy. Consulter Section 19.5.6, « Configurer un nœud de worker de mod_cluster » pour plus de détails.

Rapporter un bogue

19.5.6. Configurer un nœud de worker de mod_cluster

Résumé

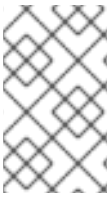
Un worker node de mod_cluster se compose d'un serveur JBoss EAP 6. Ce serveur peut faire partie d'un groupe de serveurs dans un domaine géré ou un serveur autonome. Un processus distinct s'exécute dans JBoss EAP 6, et gère tous les nœuds du cluster. C'est ce qu'on appelle le master. Pour plus d'informations conceptuelles sur les worker nodes, consulter Section 19.1.4, « Nœud de worker ». Pour une vue d'ensemble de l'équilibrage de la charge du serveur web, consulter Section 19.1.3, « Connecteurs HTTP - Aperçu général ».

Le maître n'est configuré qu'une fois, par l'intermédiaire du sous-système de **mod_cluster**. Pour configurer le sous-système **mod_cluster**, reportez-vous à *Configure the mod_cluster Subsystem* dans *Administration and Configuration Guide*. Chaque nœud de worker est configuré séparément, alors répétez cette procédure pour chaque nœud que vous souhaitez ajouter au cluster.

Si vous utilisez un domaine géré, chaque serveur de groupe de serveurs est un noeud de worker qui partage une configuration identique. Par conséquent, la configuration s'effectue sur un groupe de serveurs dans son entier. Dans un serveur autonome, la configuration s'effectue sur une seule instance de JBoss EAP 6. Les étapes de configuration sont sinon identiques.

Configuration d'un nœud de worker

- Si vous utilisez un serveur autonome, il devra être démarré par le profil **standalone-ha**.
- Si vous utilisez un domaine géré, votre groupe de serveurs devra utiliser le profil **ha** ou **full-ha**, et le groupe de liaisons de sockets **ha-sockets** ou **full-ha-sockets**. JBoss EAP 6 est fourni avec un groupe de serveurs à clusterisation activée, nommé **other-server-group** qui remplit ces prérequis.



NOTE

Quand vous avez des commandes d'interface CLI, celles-ci présument que vous utilisez un domaine géré. Si vous utilisez un serveur autonome, supprimez la portion **/profile=full-ha** des commandes.

Procédure 19.9. Configurez un nœud de worker

1. **Configurez les interfaces de réseau**
   Les interfaces de réseau ont toutes la valeur **127.0.0.1** par défaut. Chaque hôte physique, qui accueille un serveur autonome ou bien un ou plusieurs serveurs au sein d'un groupe de serveurs, a besoin d'interfaces configurées pour utiliser son adresse IP, que les autres serveurs peuvent apercevoir.

   Pour changer l'adresse IP d'un hôte de JBoss EAP 6, vous devrez le fermer et modifier son fichier de configuration directement. C'est parce que l'API de gestion qui actionne la console de gestion et le CLI dépendent d'une adresse de gestion stable.

   Suivez ces étapes pour changer l'adresse IP sur chaque serveur de votre cluster par votre adresse IP publique de master.

   a. Démarrez le serveur JBoss EAP en utilisant le profil décrit plus haut dans cette section.

   b. Lancez la Management CLI, avec la commande ***EAP_HOME*/bin/jboss-cli.sh** dans Linux ou bien la commande ***EAP_HOME*\bin\jboss-cli.bat** dans le serveur Microsoft Windows. Saisissez la commande **connect** pour connecter le contrôleur de domaine sur l'hôte local, ou **connect *IP_ADDRESS*** pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

   c. Modifier l'adresse IP externe des interfaces **management**, **public** et **unsecure** en saisissant les commandes suivantes. Veillez bien à remplacer **EXTERNAL_IP_ADDRESS** qui se trouve dans la commande par l'adresse IP externe de l'hôte.

      ```
      /interface=management:write-attribute(name=inet-
      address,value="${jboss.bind.address.management:EXTERNAL_IP_ADDRES
      S}"
      ```

```
/interface=public:write-attribute(name=inet-
address,value="${jboss.bind.address.public:EXTERNAL_IP_ADDRESS}"
/interface=unsecure:write-attribute(name=inet-
address,value="${jboss.bind.address.unsecure:EXTERNAL_IP_ADDRESS}
"
:reload
```

Vous devriez voir le résultat suivant pour chaque commande.

```
 "outcome" => "success"
```

d. Pour les hôtes qui participent à un domaine géré, mais qui ne sont pas master, vous devrez modifiez le nom d'hôte du **master** par un nom unique. Ce nom devra être unique (parmi les noms d'esclave) et sera utilisé par l'esclave pour s'authentifier au cluster, donc notez bien le nom que vous utilisez.

   i. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

   ```
   bin/domain.sh --host-config=HOST_SLAVE_XML_FILE_NAME
   ```

   Par exemple :

   ```
   bin/domain.sh --host-config=host-slave01.xml
   ```

   ii. Lancer l'interface CLI.

   iii. Utiliser la syntaxe suivante pour remplacer le nom d'hôte :

   ```
   /host=master:write-
   attribute(name="name",value=UNIQUE_HOST_SLAVE_NAME)
   ```

   Par exemple :

   ```
   /host=master:write-attribute(name="name",value="host-slave01")
   ```

   Vous devriez voir apparaître le résultat suivant.

   ```
    "outcome" => "success"
   ```

   Cela modifie le XML du fichier **host-slave01.xml** comme suit :

   ```
   <host name="host-slave01" xmlns="urn:jboss:domain:1.6">
   ```

e. Pour les hôtes nouvellement configurés qui ont besoin de rejoindre un domaine géré, cherchez l'élément **local** et ajoutez l'attribut **host** de l'élement **remote** qui pointe en direction du contrôleur de domaine. Cette étape ne s'applique pas à un serveur autonome.

   i. Démarrer l'hôte esclave JBoss EAP à l'aide de la syntaxe suivante :

   ```
   bin/domain.sh --host-config=HOST_SLAVE_XML_FILE_NAME
   ```

   Par exemple :

```
bin/domain.sh --host-config=host-slave01.xml
```

ii. Lancer l'interface CLI.

iii. Utilisez la syntaxe suivante en spécifiant le contrôleur de domaine :

```
/host=UNIQUE_HOST_SLAVE_NAME/:write-remote-domain-
controller(host=DOMAIN_CONTROLLER_IP_ADDRESS,port=${jboss.doma
in.master.port:9999},security-realm="ManagementRealm")
```

Par exemple :

```
/host=host-slave01/:write-remote-domain-
controller(host="192.168.1.200",port=${jboss.domain.master.por
t:9999},security-realm="ManagementRealm")
```

Vous devriez voir apparaître le résultat suivant.

```
 "outcome" => "success"
```

Cela modifie le XML du fichier **host-slave01.xml** comme suit :

```
<domain-controller>
    <remote host="192.168.1.200"
port="${jboss.domain.master.port:9999}" security-
realm="ManagementRealm"/>
</domain-controller>
```

2. **Configurez l'authentification pour chaque serveur esclave.**
Chaque serveur esclave a besoin d'un nom d'utilisateur et d'un mot de passe créé dans le **ManagementRealm** du contrôleur de domaine ou du master autonome. Sur le contrôleur de domaine ou sur le master autonome, exécutez la commande ***EAP_HOME*/bin/add-user.sh**. Ajoutez un utilisateur avec le même nom d'utilisateur comme esclave au **ManagementRealm**. Quand on vous demandera si cet utilisateur doit s'authentifier auprès d'une instance de JBoss AS externe, répondez **Oui**. Vous trouverez un exemple de l'entrée et de la sortie de la commande ci-dessous, pour un esclave appelé **slave1**, et un mot de passe **changeme**.

```
user:bin user$ ./add-user.sh

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : slave1
Password : changeme
Re-enter Password : changeme
About to add user 'slave1' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'slave1' to file '/home/user/jboss-eap-
6.0/standalone/configuration/mgmt-users.properties'
Added user 'slave1' to file '/home/user/jboss-eap-
```

```
6.0/domain/configuration/mgmt-users.properties'
Is this new user going to be used for one AS process to connect to
another AS process e.g. slave domain controller?
yes/no? yes
To represent the user add the following to the server-identities
definition <secret value="Y2hhbmdlbWU=" />
```

3. **Copiez l'élément codé-Base64 `<secret>` à partir de la sortie `add-user.sh`.**
   Si vous prévoyez de spécifier un mot de passe codé Base64 pour l'authentification, copiez l'élément **`<secret>`** à partir de la dernière ligne de la sortie **`add-user.sh`** car vous en aurez besoin à l'étape suivante.

4. **Modifiez le domaine de sécurité de l'hôte esclave pour la nouvelle authentification.**
   Vous pourrez spécifier la valeur secrète d'une des manières suivantes :

   - **Spécifier le mot de passe codé-Base64 dans le fichier de configuration du serveur par le CLI.**

     a. Lancez la Management CLI, avec la commande ***`EAP_HOME`*`/bin/jboss-cli.sh`** dans Linux ou bien la commande ***`EAP_HOME`*`\bin\jboss-cli.bat`** dans le serveur Microsoft Windows. Saisissez la commande **`connect`** pour connecter le contrôleur de domaine sur l'hôte local, ou **`connect` *`IP_ADDRESS`*** pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

     b. Spécifiez la valeur secrète en saisissant la commande suivante. Veillez bien à remplacer **`SECRET_VALUE`** par la valeur secrète retournée de la sortie **`add-user.sh`** lors de l'étape précédente.

     ```
     /core-service=management/security-
     realm=ManagementRealm/server-
     identity=secret:add(value="SECRET_VALUE")
     :reload
     ```

     Vous devriez voir le résultat suivant pour chaque commande.

     ```
      "outcome" => "success"
     ```

   - **Configurez l'hôte pour obtenir un mot de passe de l'archivage sécurisé.**

     a. Utilisez le script **`vault.sh`** pour générer un mot de passe masqué. Cela génèrera une chaîne comme la suivante :
     `VAULT::secret::password::ODVmYmJjNGMtZDU2ZC00YmNlLWE4ODMtZjQ1NWNmNDU4ZDc1TElORV9CUkVBS3ZhdWx0`.

     Vous pourrez trouver plus d'informations sur l'archivage sécurisé dans Password Vaults de la section Sensitive Strings de ce guide, ici : Section 11.13.1, « Sécurisation des chaînes confidentielles de fichiers en texte clair ».

     b. Lancez la Management CLI, avec la commande ***`EAP_HOME`*`/bin/jboss-cli.sh`** dans Linux ou bien la commande ***`EAP_HOME`*`\bin\jboss-cli.bat`** dans le serveur Microsoft Windows. Saisissez la commande **`connect`** pour connecter le contrôleur de domaine sur l'hôte local, ou **`connect` *`IP_ADDRESS`*** pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

c. Spécifiez la valeur secrète en saisissant la commande suivante. Veillez bien à remplacer **SECRET_VALUE** par le mot de passe masqué généré lors de l'étape précédente.

```
/core-service=management/security-
realm=ManagementRealm/server-
identity=secret:add(value="${VAULT::secret::password::SECRET_V
ALUE}")
:reload
```

Vous devriez voir le résultat suivant pour chaque commande.

```
 "outcome" => "success"
```

NOTE

Quand vous créez un mot de passe dans l'archivage sécurisé, celui-ci devra être spécifié en texte brut, et non pas codé Base64.

- **Spécifiez le mot de passe en tant que propriété système.**
  Les exemples suivants utilisent **`server.identity.password`** comme nom de propriété de système pour le mot de passe.

  a. Spécifiez la propriété système pour le mot de passe dans le fichier de configuration du serveur par le CLI.

     i. Lancez la Management CLI, avec la commande ***`EAP_HOME`*`/bin/jboss-cli.sh`** dans Linux ou bien la commande ***`EAP_HOME`*`\bin\jboss-cli.bat`** dans le serveur Microsoft Windows. Saisissez la commande **`connect`** pour connecter le contrôleur de domaine sur l'hôte local, ou **`connect`** ***`IP_ADDRESS`*** pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

     ii. Saisissez la commande suivante pour configurer l'identité secrète pour utiliser la propriété système.

```
/core-service=management/security-
realm=ManagementRealm/server-
identity=secret:add(value="${server.identity.password}")
:reload
```

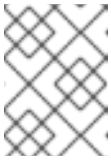
Vous devriez voir le résultat suivant pour chaque commande.

```
 "outcome" => "success"
```

  b. Quand vous spécifiez le mot de passe en tant que propriété système, vous pouvez configurer l'hôte d'une des manières suivantes :

     - Démarrez le serveur en saisissant le mot de passe en texte brut comme argument de ligne de commande, comme par exemple :

```
-Dserver.identity.password=changeme
```



NOTE

Le mot de passe doit être saisi en texte brut et sera visible par quiconque lance la commande **`ps -ef`**.

- Mettez le mot de passe dans un fichier de propriétés et passez l'URL du fichier de propriétés sous forme d'argument de ligne de commande.

  i. Ajoutez la paire clé/valeur à un fichier de propriétés. Par exemple :

  ```
  server.identity.password=changeme
  ```

  ii. Démarrez le serveur par les arguments de ligne de commande

  ```
  --properties=URL_TO_PROPERTIES_FILE
  ```

  .

5. **Redémarrez le serveur.**
   L'esclave va maintenant authentifier le master en utilisant son nom d'hôte comme nom d'utilisateur et le string codifié comme mot de passe.

Résultat

Votre serveur autonome, ou les serveurs au sein d'un groupe de serveurs d'un domaine géré, sont désormais configurés en tant que worker nodes du mod_cluster. Si vous déployez une application en cluster, ses sessions seront répliquées sur tous les nœuds de cluster de basculement, et seront en mesure d'accepter les demandes provenant d'un serveur web externe ou d'un équilibreur de charges. Chaque nœud du cluster détecte les autres nœuds à l'aide d'automatic discovery, par défaut. Pour configurer la détection automatique et les autres paramètres spécifiques du sous-système **`mod_cluster`**, reportez-vous à Section 19.5.2, « Configurer le sous-système **`mod_cluster`** ». Pour configurer le serveur Apache HTTP, reportez-vous à Section 19.3.5, « Utiliser un serveur web externe comme Web frontal pour les applications JBoss EAP 6. ».

Rapporter un bogue

19.5.7. Migration du trafic entre les clusters

Résumé

Après avoir créé un nouveau cluster à l'aide de JBoss EAP 6, vous souhaiterez sans doute migrer le trafic d'un ancien cluster vers un nouveau dans le cadre d'un processus de mise à niveau. Au cours de cette tâche, vous verrez la stratégie qui peut être utilisée pour migrer ce trafic avec un minimum de temps mort.

Pré-requis

- Nouvelle installation de cluster : Section 19.5.2, « Configurer le sous-système **`mod_cluster`** » (nous appellerons ce cluster : NOUVEAU Cluster).
- Une ancienne installation de cluster a été rendue obsolète (nous appellerons ce cluster : ANCIEN Cluster).

Procédure 19.10. Mise à niveau du processus pour les clusters

1. Installez votre nouveau cluster en suivant les étapes décrites dans les conditions préalables.

2. Pour les NOUVEAUX et ANCIENS clusters à la fois, assurez-vous que l'option de configuration **`sticky-session`** est définie sur **`true`** (**`true`** par défaut). L'activation de cette option signifie que toutes les nouvelles demandes présentées à un nœud de cluster dans un de ces clusters continueront d'aller vers ce nœud.

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=sticky-
session,value=true)
```

3. Ajouter les nœuds dans le NOUVEAU cluster à la configuration de mod_cluster individuellement à l'aide du processus décrit ici : Section 19.5.6, « Configurer un nœud de worker de mod_cluster »

4. Configurer l'équilibrage de la charge (mod_cluster) pour arrêter les contextes individuels dans l'ANCIEN Cluster. L'arrêt des contextes (par opposition à leur désactivation) dans l'ANCIEN cluster permettra aux contextes individuels de s'arrêter gracieusement (et éventuellement à un arrêt total). Les sessions existantes seront toujours servies, mais aucune nouvelle session ne sera dirigée vers ces nœuds. Les contextes arrêtés peuvent prendre plusieurs minutes ou même plusieurs heures pour s'arrêter.

   Vous pouvez utiliser le CLI suivant pour stopper un contexte. Remplacer les valeurs de paramètre par des valeurs adaptées à votre environnement.

```
[standalone@localhost:9999 subsystem=modcluster] :stop-
context(context=/myapp, virtualhost=default-host, waittime=50)
```

Résultat

Vous avez effectué la mise à niveau de JBoss EAP 6 Cluster avec succès.

Rapporter un bogue

19.6. APACHE MOD_JK

19.6.1. Le connecteur Apache mod_jk HTTP

Apache **`mod_jk`** est un connecteur HTTP fourni aux clients qui en ont besoin pour des raisons de compatibilité. Il fournit un équilibrage des charges et fait partie des **`jboss-eap-native-webserver-connectors`** contenus dans JBoss Web Container. Pour les platerformes prises en charge, consulter https://access.redhat.com/site/articles/111663. Le connecteur **`mod_jk`** est maintenu par Apache, et sa documentation se trouve à l'adresse suivante http://tomcat.apache.org/connectors-doc/.

JBoss EAP 6 peut accepter des charges de travail en provenance d'un serveur proxy Apache HTTP. Le serveur proxy accepte les requêtes des clients en provenance des serveurs frontaux web, et passe le travail à des serveurs JBoss Enterprise Application Platform participant. Quand les sessions sticky sont activées, une requête en provenance d'un même client va toujours vers le même serveur JBoss EAP 6, à moins que celui-ci ne soit pas rendu disponible.

À la différence du JBoss HTTP connector **`mod_cluster`**, un connecteur **`mod_jk`** HTTP ne connaît pas le statut des déploiements sur les serveurs ou groupes de serveurs, et ne peut donc pas ajuster les envois en fonction.

Tout comme **mod_cluster**, **mod_jk** communique à travers le protocole AJP 1.3.

NOTE

mod_cluster est un équilibreur de charges plus avancé que **mod_jk**. **mod_cluster** fournit toute la fonctionnalité de **mod_jk** et quelques fonctionnalités supplémentaires. Pour plus d'informations sur **mod_cluster**, consulter Section 19.5.1, « Le connecteur HTTP **mod_cluster** ».

Prochaine étape : configurer une instance de plateforme JBoss EAP 6 pour qu'elle puisse participer à un groupe d'équilibrage des charges mod_jk

- Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »
- Section 19.6.3, « Installer le module jk_mod dans un serveur Apache HTTP (ZIP) »

Rapporter un bogue

19.6.2. Configurer JBoss EAP 6 pour qu'il communique avec Apache Mod_jk

Aperçu

Le connecteur mod_jk HTTP possède un simple composant, le module **mod_jk.so**, qui est chargé par le serveur web. Ce module reçoit les demandes des clients et les transfère vers le conteneur, en l'occurrence JBoss EAP 6. JBoss EAP 6 doit également être configuré pour accepter ces demandes et envoyer leurs réponses vers le serveur web.

La configuration du serveur Apache HTTPD est couverte dans Section 19.6.3, « Installer le module jk_mod dans un serveur Apache HTTP (ZIP) ».

Pour que JBoss Enterprise Application Platform puisse communiquer avec HTTP Apache, il doit avoir le connecteur AJP/1.3 activé. Ce connecteur sera présent par défaut dans les configurations suivantes :

- Dans un domaine géré, dans les groupes de serveurs qui utilisent les profils **ha** et **full-ha**, et le groupe de liaisons de sockets **ha** ou **full-ha**. Le groupe de serveurs **other-server-group** est configuré correctement dans une installation par défaut.
- Dans un serveur autonome, les profils **standalone-ha** et **standalone-full-ha** sont configurés pour les configurations en cluster. Pour démarrer le serveur autonome avec un de ces profils, lancer la commande ci-dessous, à partir du répertoire ***EAP_HOME/***. Remplacer par le nom de profil approprié.

```
[user@host bin]$ ./bin/standalone.sh --server-config=standalone-ha.xml
```

Rapporter un bogue

19.6.3. Installer le module jk_mod dans un serveur Apache HTTP (ZIP)

Conditions préalables

- Pour cette tâche, vous devrez utiliser Apache HTTP installé dans un environnement pris en charge ou le serveur Apache HTTP installé sur JBoss Enterprise Web Server. Notez que le

serveur Apache HTTP installé dans JBoss Enterprise Web Server fait partie de la distribution JBoss EAP 6.

- Si vous devez installer un serveur Apache HTTP, utilisez les instructions qui se trouvent dans *Red Hat Enterprise Linux Deployment Guide*.
- Si vous avez besoin d'installer JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*.
- Si vous utilisez le serveur Apache HTTP, télécharger le package JBoss EAP 6 Native Components pour votre plate-forme du portail clients de Red Hat à https://access.redhat.com. Ce paquet contient à la fois les binaires **`mod_jk`** et **`mod_cluster`** qui sont précompilés pour Red Hat Enterprise Linux. Si vous utilisez JBoss Enterprise Web Server, il comprend déjà le binaire pour **`mod_jk`**.
- Si vous utilisez Red Hat Enterprise Linux (RHEL) 5 et le serveur natif Apache HTTP (httpd 2.2.3), commencez par télécharger le module mod_perl pour charger le module mod_jk.
- Vous devez être connectés avec des privilèges administratifs (root).

Procédure 19.11. Installer le module mod cluster

1. **Configurer le module mod_jk.**

   a. Créer un nouveau fichier nommé ***`HTTPD_HOME`*`/conf.d/mod-jk.conf`** et y ajouter ce qui suit.

NOTE

La directive **`JkMount`** indique quels URL Apache doivent aller vers le module mod_jk. Sur la base de la configuration de la directive, mod_jk transfère l'URL reçu aux conteneurs de servlet qui conviennent.

Pour servir le contenu directement, et pour n'utiliser que l'équilibreur de charges pour les applications Java, le chemin URL doit être **`/application/*`**. Pour utiliser mod_jk en tant qu'équilibreur des charges, utiliser la valeur **`/*`** pour transférer tous les URL au mod_jk.

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat  "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
```

```
JkOptions +ForwardKeySize -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
# The default setting only sends Java application data to mod_jk.
# Use the commented-out line to send all URLs through mod_jk.
# JkMount /* loadbalancer
JkMount /application/* loadbalancer

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

Observer les valeurs et vérifier qu'elles conviennent à votre installation. Quand vous serez satisfait, sauvegarder le fichier.

b. **Spécifier une directive JKMountFile**
En plus de la directive JKMount de **`mod-jk.conf`**, vous pourrez spécifier un fichier qui contienne des modèles URL multiples à transférer au mod_jk.

i. Ajouter ce qui suit au fichier ***`HTTPD_HOME`*`/conf/mod-jk.conf`** :

```
# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkermap.properties
```

ii. Créer un nouveau fichier intitulé ***`HTTPD_HOME`*`/conf/uriworkermap.properties`** avec une ligne pour chaque modèle URL à faire correspondre. L'exemple suivant montre des exemples de syntaxe pour ce fichier.

```
# Simple worker configuration file
/*=loadbalancer
```

c. **Copier le fichier mod_jk.so dans le répertoire de modules d'httpd**

NOTE

Cela n'est utile que si le serveur HTTP Apache n'a pas de **`mod_jk.so`** dans son répertoire **`modules/`**. Vous pourriez éviter cette étape si vous utilisez le serveur Apache HTTP inclus comme téléchargement de JBoss EAP 6.

Extraire le paquet Native Web Server Connectors Zip. Localiser le fichier **mod_jk.so** soit dans le répertoire ***EAP_HOME*/modules/system/layers/base/native/lib/httpd/modules/** ou dans le répertoire ***EAP_HOME*/modules/system/layers/base/native/lib64/httpd/modules/** suivant que votre système d'exploitation est de 32-bit ou de 64-bit.

Copier le fichier dans le répertoire ***HTTPD_HOME*/modules/**.

2. **Configurer les noeuds de worker mod_jk.**

   a. Créer un nouveau fichier nommé ***HTTPD_HOME*/conf/workers.properties**. Utiliser l'exemple suivant comme point de départ, et modifier le fichier selon vos besoins.

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=node1.mydomain.com
worker.node1.type=ajp13
worker.node1.ping_mode=A
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=node2.mydomain.com
worker.node2.type=ajp13
worker.node2.ping_mode=A
worker.node2.lbfactor=1

# Load-balancing behavior
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1

# Status worker for managing load balancer
worker.status.type=status
```

   Pour obtenir une description détaillée de la syntaxe du fichier **workers.properties**, et pour obtenir des options de configuration avancées, consulter Section 19.6.5, « Référence de configuration pour les Apache Mod_jk Workers ».

3. **Redémarrer le serveur web.**
   La façon de redémarrer le serveur web dépend de savoir si vous utilisez le serveur Apache HTTP de Red Hat Enterprise Linux ou le serveur HTTP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

   - **Serveur Apache HTTPD de Red Hat Enterprise Linux**
     Exécuter la commande suivante :

```
[root@host]# service httpd restart
```

- **Serveur de JBoss Enterprise Web Server HTTP**
  JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du serveur web est différente pour chacun.

  - **Red Hat Enterprise Linux, installé avec RPM**
    Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son serveur web en tant que service. Pour redémarrer le serveur web, lancer les deux commandes suivantes :

    ```
    [root@host ~]# service httpd stop
    [root@host ~]# service httpd start
    ```

  - **Red Hat Enterprise Linux, installé avec Zip**
    Si vous avez installé le serveur HTTP Apache de JBoss Enterprise Web à partir d'une archive ZIP, utiliser la commande **apachectl** pour redémarrer le serveur web. Remplacer *EWS_HOME* par le répertoire où vous avez décompressé le serveur JBoss Enterprise Web Server Apache HTTP.

    ```
    [root@host ~]# EWS_HOME/httpd/sbin/apachectl restart
    ```

  - **Microsoft Windows Server**
    Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

    ```
    C:\> net stop Apache2.2
    C:\> net start Apache2.2
    ```

  - **Solaris**
    Lancer les commandes suivantes dans l'invite de commandes avec des permissions admin. Remplacer *EWS_HOME* par le répertoire dans lequel vous avez décompressé le serveur HTTP Apache de JBoss Enterprise Web.

    ```
    [root@host ~] EWS_HOME/httpd/sbin/apachectl restart
    ```

Résultat

Le serveur Apache HTTP est maintenant configuré pour pouvoir utiliser l'équilibreur de charges de mod_jk. Pour configurer JBoss EAP 6 pour qu'il soit au fait de mod_jk, consulter Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes ».

Rapporter un bogue

19.6.4. Installer le Module_jk_mod dans Apache HTTPD Server (RPM)

Conditions préalables

- Pour cette tâche, vous devrez utiliser Apache HTTPD installé dans un environnement pris en charge ou l'Apache HTTP installé sur JBoss Enterprise Web Server. Notez que l'Apache HTTP installé dans JBoss Enterprise Web Server fait partie de la distribution JBoss Enterprise Web Server faisant partie de la distribution JBoss EAP 6.
- Si vous devez installer Apache HTTP Server, utilisez les instructions qui se trouvent dans *Red Hat Enterprise Linux Deployment Guide*, dans https://access.redhat.com/site/documentation/.

- Si vous avez besoin d'installer le serveur JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*, dans https://access.redhat.com/site/documentation/.
- Vous devez être connectés avec des privilèges administratifs (root).

Procédure 19.12. Red Hat Enterprise Linux 5 : mod_jk with Apache HTTP Server 2.2.3

1. Installer mod_jk-ap22 1.2.37 et ses dépendances mod_perl de **`jbappplatform-6-*-server-5-rpm`** :

```
yum install mod_jk
```

2. **Option :** Copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

NOTE

Le message erreur suivant indique que votre module mod_jk a été chargé avant que mod_erl ne soit présent :

```
Cannot load /etc/httpd/modules/mod_jk.so into server:
/etc/httpd/modules/mod_jk.so: undefined symbol:
ap_get_server_description
```

Pour s'assurer que le module mod_perl soit chargé avant mod_jk, ajouter ce qui suit à **`/etc/httpd/conf.d/mod_jk.conf`** :

```
 <IfModule !perl_module>
        LoadModule perl_module modules/mod_perl.so
</IfModule>
LoadModule jk_module modules/mod_jk.so
```

Procédure 19.13. Red Hat Enterprise Linux 5 : mod_jk with JBoss EAP Apache HTTP Server 2.2.26

1. Installer à la fois mod_jk et le dernier Apache HTTP Server 2.2.26 fourni par le canal **`jbappplatform-6-*-server-5-rpm`** à l'aide de cette commande :

```
yum install mod_jk httpd
```

2. **Option :** Copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 19.14. Red Hat Enterprise Linux 6 : mod_jk et JBoss EAP Apache HTTP Server 2.2.26

1. Installer mod_jk-ap22 1.2.37 et le package Apache HTTP Server 2.2.26 httpd du canal **`jbappplatform-6-*-server-6-rpm`** (toutes les versions existantes doivent être mises à jour) :

```
yum install mod_jk httpd
```

2. **Option :** Copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 19.15. Red Hat Enterprise Linux 6 : mod_jk with Apache HTTP Server 2.2.15

1. Installer mod_jk with Apache HTTP Server 2.2.15 par la commande suivante :

```
yum install mod_jk
```

2. **Option :** Copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd/conf/workers.properties
```

Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
service httpd start
```

Procédure 19.16. Red Hat Enterprise Linux 7: mod_jk et JBoss EAP Apache HTTP Server 2.2.26

1. Installer mod_jk-ap22 1.2.37 et le package Apache HTTP Server 2.2.26 httpd22 du canal **`jbappplatform-6-*-server-6-rpm`** (tous les versions existantes doivent être mises à jour) :

```
yum install mod_jk
```

2. **Option :** Copier l'exemple de fichiers de configuration à utiliser :

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/mod_jk.conf.sample
/etc/httpd22/conf.d/mod_jk.conf
```

```
cp /usr/share/doc/mod_jk-ap22-1.2.37/workers.properties.sample
/etc/httpd22/conf/workers.properties
```

   Ces fichiers devront être modifiés pour pouvoir correspondre à vos besoins.

3. Démarrer le serveur :

```
systemctl start httpd22.service
```

Rapporter un bogue

19.6.5. Référence de configuration pour les Apache Mod_jk Workers

Le fichier **`workers.properties`** définit le comportement des noeuds de workers à qui mod_jk passe les requêtes de clients. Dans Red Hat Enterprise Linux, le fichier se trouve dans **`/etc/httpd/conf/workers.properties`**. Le fichier **`workers.properties`** définit où les différents conteneurs de servlet se trouvent, et la façon dont la charge de travail doit être distribuée parmi eux.

La configuration est divisée en trois sections. La première section traite des propriétés globales qui s'appliquent à tous les nœuds de worker. La deuxième section contient des paramètres qui s'appliquent à un worker spécifique. La troisième section contient les paramètres qui s'appliquent à un nœud spécifique, équilibré par le worker.

La structure générale d'une propriété est **`worker.`*`WORKER_NAME`*`.`*`DIRECTIVE`***, avec *WORKER_NAME* comme nom unique de worker, et *DIRECTIVE* comme paramètre de configuration à appliquer au worker.

Référence de configuration des Apache Mod_jk Workers

Les modèles de noeuds spécifient les paramètres par défaut par noeud. Vous pouvez remplacer le modèle contenu dans le paramètre de nœud lui-même. Vous pouvez voir un exemple de modèle de nœud dans Exemple 19.5, « Exemple de fichier **`workers.properties`** ».

Tableau 19.11. Propriétés globales

Property	Description
worker.list	La liste des noms de worker utilisés par mod_jk. Ces workers sont prêts à recevoir des requêtes.

Tableau 19.12. Propriétés per-worker

Property	Description
type	Le type de worker. Le type par défaut est **`ajp13`**. Autres valeurs possibles **`ajp14`**, **`lb`**, **`status`**. Pour plus d'informations sur ces directives, voir la référence de protocole Apache Tomcat Connector AJP à http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html.
balance_workers	Spécifie les nœuds de worker que l'équilibreur de charges doit gérer. Vous pouvez utiliser la directive plusieurs fois pour un même équilibrage de charge. Il se compose d'une liste séparée par des virgules des noms de workers. Ceci est défini par worker, et non pas par nœud. Elle affecte tous les nœuds équilibrés par ce type de worker.
sticky_session	Indique si les demandes d'une même session sont toujours acheminées vers le même worker. La valeur par défaut est **`0`**, ce qui signifie que les sticky sessions sont désactivées. Pour activer des sticky sessions, définir à **`1`**. Les sticky sessions doivent habituellement être activées, à moins que toutes vos demandes soient vraiment stateless. Ceci est défini par worker, et non pas par nœud. Affecte tous les nœuds équilibrés par ce type de worker.

Tableau 19.13. Propriétés per-node

Property	Description
host	Le nom d'hôte ou l'adresse IP du worker. Le noeud de worker doit supporter la pile de protocole **`ajp`**. La valeur par défaut est **`localhost`**.
Important	Le numéro de port de l'instance de serveur éloigné qui écoute les requêtes de protocoles définis. La valeur par défaut est **`8009`**, qui correspond au port d'écoute des workers AJP13. La valeur par défaut des workers AJP14 est **`8011`**.

Property	Description
ping_mode	Les conditions dans lesquelles les connexions sont interrogées pour le statut du réseau. La sonde utilise un paquet AJP13 vide pour CPing et s'attend à un CPong en réponse. Spécifier les conditions à l'aide d'une combinaison d'indicateurs de la directive. Les indicateurs ne sont pas séparés par une virgule ou un espace blanc. Le ping_mode peut être n'importe quelle combinaison de **C**, **P**, **I**, ou **A**. • **C** - Connect. Sonde la connexion une fois seulement suite à la connexion au serveur. Spécifier le timeout en utilisant la valeur de **connect_timeout**. Sinon, la valeur de **ping_timeout** sera utilisée. • **P** - Prepost. Sonde la connexion avant d'envoyer une requête au serveur. Spécifier le timeout en utilisant la directive **prepost_timeout**. Sinon, la valeur **ping_timeout** sera utilisée. • **I** - Interval. Interroge la connexion à un intervalle spécifié par **connection_ping_interval**, si présent. Sinon, utilise la valeur de **ping_timeout**. • **A** - All. Un raccourci pour **CPI**, qui indique que toutes les sondes de connexion sont utilisées.
ping_timeout, connect_timeout, prepost_timeout, connection_ping_interval	Les valeurs de timeout pour les paramètres de sonde de connexion ci-dessus. La valeur est spécifiée en millisecondes, et la valeur par défaut pour **ping_timeout** est de 10000.
lbfactor	Spécifie le facteur d'équilibrage des charge d'un worker individuel et ne s'applique qu'à un worker membre d'un équilibreur de charge. Ceci est utile pour donner à un serveur plus puissant, une charge de travail supplémentaire. Pour donner à un worker 3 fois la charge de la valeur par défaut, définir cette valeur à **3**: **worker.my_worker.lbfactor=3**

Exemple 19.5. Exemple de fichier `workers.properties`

```
worker.list=node1, node2, node3

worker.balancer1.sticky_sessions=1
worker.balancer1.balance_workers=node1
worker.balancer2.sticky_session=1
worker.balancer2.balance_workers=node2,node3

worker.nodetemplate.type=ajp13
worker.nodetemplate.port=8009

worker.node1.template=nodetemplate
worker.node1.host=localhost
worker.node1.ping_mode=CI
worker.node1.connection_ping_interval=9000
worker.node1.lbfactor=1

worker.node2.template=nodetemplate
```

```
worker.node2.host=192.168.1.1
worker.node2.ping_mode=A

worker.node3.template=nodetemplate
worker.node3.host=192.168.1.2
```

Les détails de configuration de ce document sont limités. Voir la documentation Apache à http://tomcat.apache.org/connectors-doc/ pour obtenir des instructions supplémentaires.

Rapporter un bogue

19.7. APACHE MOD_PROXY

19.7.1. Le connecteur Apache mod_proxy HTTP

Apache offre deux modules différents d'équilibrage de charge et de proxying pour ses démons httpd : **mod_proxy** et **mod_jk**. Pour en savoir plus sur **mod_jk**, consulter Section 19.6.1, « Le connecteur Apache mod_jk HTTP ». La plate-forme JBoss EAP 6 prend en charge l'utilisation de l'un d'entre eux, bien que **mod_cluster**, le connecteur HTTP de JBoss, couple plus étroitement JBoss EAP 6 et httpd externe, et est le connecteur HTTP recommandé. Reportez-vous à Section 19.1.3, « Connecteurs HTTP - Aperçu général » pour une vue d'ensemble des connecteurs HTTP pris en charge, y compris les avantages et les inconvénients.

À la différence de **mod_jk**, **mod_proxy** supporte les connexions via les protocoles HTTP et HTTPS. Chacun d'entre eux supporte le protocole AJP.

mod_proxy peut être configuré en autonome ou en configurations d'équilibrage de charge, et il prend en charge la notion de sticky sessions.

Le module **mod_proxy** nécessite que JBoss EAP 6 ait le connecteur web HTTP, HTTPS ou AJP configuré. Cela fait partie du sous-système web. Consulter Section 17.1, « Configurer le sous-système web » pour obtenir des informations sur la façon de configurer le sous-système web.

Rapporter un bogue

19.7.2. Installer Mod_proxy HTTP Connector sur le serveur Apache HTTPD

Aperçu

mod_proxy est un module d'équilibrage de charges fourni par Apache. Cette tâche présente une configuration de base. Pour plus d'informations sur la configuration avancée, ou pour plus de détails, reportez-vous à la documentation Apache **mod_proxy** à https://httpd.apache.org/docs/2.2/mod/mod_proxy.html. Pour plus d'informations sur **mod_proxy** d'une perspective JBoss EAP 6, consulter Section 19.7.1, « Le connecteur Apache mod_proxy HTTP » et Section 19.1.3, « Connecteurs HTTP - Aperçu général ».

Conditions préalables

- JBoss Enterprise Web Server httpd ou Apache HTTP doivent être installés. Un serveur Apache HTTP autonome est fourni séparément dans le portail clients Red Hat, dans la zone de téléchargement de JBoss EAP 6. Voir Section 19.3.2, « Installer le serveur Apache HTTP inclus dans JBoss EAP 6 » pour obtenir des informations sur ce serveur Apache HTTP si vous souhaitez l'utiliser.

- Les modules **mod_proxy** doivent être installés. Le serveur Apache HTTPD est généralement livré avec les modules **mod_proxy** déjà inclus. C'est le cas sur Red Hat Enterprise Linux et sur le serveur Apache HTTP qui vient avec le serveur Web de JBoss Enterprise.
- Vous devez avoir les permissions **root** ou administrateur pour pouvoir modifier la configuration de Serveur Apache HTTP.
- Dans notre exemple, on assume que JBoss EAP 6 est configuré avec le connecteur web HTTP ou HTTPS. Cela fait partie de la configuration du sous-système web. Voir Section 17.1, « Configurer le sous-système web » pour obtenir des informations sur la façon de configurer le sous-système web.

1. **Activer les modules mod_proxy dans le démon httpd**
   Recherchez les lignes suivantes dans votre fichier ***HTTPD_HOME*/conf/httpd.conf**. Si elles ne sont pas présentes, ajoutez-les en bas. Si elles sont présentes, mais que les lignes commencent par un caractère de commentaire (#), supprimer le caractère. Enregistrez le fichier par la suite. Habituellement, les modules sont déjà présents et activés.

   ```
   LoadModule proxy_module modules/mod_proxy.so
   LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
   LoadModule proxy_http_module modules/mod_proxy_http.so
   # Uncomment these to proxy FTP or HTTPS
   #LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
   #LoadModule proxy_connect_module modules/mod_proxy_connect.so
   ```

2. **Ajouter un proxy non équilibreur de charges.**
   Ajouter la configuration suivante à votre fichier ***HTTPD_HOME*/conf/httpd.conf**, directement sous une directive **<VirtualHost>** que vous possédez sans doute. Remplacer les valeurs par des valeurs appropriées à votre installation.

   Cet exemple utilise un hôte virtuel. Voir la nouvelle étape pour utiliser la configuration httpd par défaut.

   ```
   <VirtualHost *:80>
   # Your domain name
   ServerName Domain_NAME_HERE

   ProxyPreserveHost On

   # The IP and port of JBoss EAP 6
   # These represent the default values, if your httpd is on the same
   host
   # as your JBoss EAP 6 managed domain or server

   ProxyPass / http://localhost:8080/
   ProxyPassReverse / http://localhost:8080/

   # The location of the HTML files, and access control information
   DocumentRoot /var/www
   <Directory /var/www>
   Options -Indexes
   Order allow,deny
   Allow from all
   </Directory>
   </VirtualHost>
   ```

Après avoir appliqué vos changements, sauvegarder le fichier.

3. **Ajouter le proxy d'équilibrage des charges.**
Pour utiliser **mod_proxy** comme équilibreur de charges, et pour envoyer du travail à des serveurs multiples de JBoss EAP 6, ajouter la configuration suivante à votre fichier ***HTTPD_HOME*/conf/httpd.conf**. Remplacer les par les valeurs qui correspondent à votre environnement.

```
<Proxy balancer://mycluster>

Order deny,allow
Allow from all

# Add each JBoss Enterprise Application Server by IP address and
port.
# If the route values are unique like this, one node will not fail
over to the other.
BalancerMember http://192.168.1.1:8080 route=node1
BalancerMember http://192.168.1.2:8180 route=node2
</Proxy>

<VirtualHost *:80>
 # Your domain name
 ServerName YOUR_DOMAIN_NAME

 ProxyPreserveHost On
 ProxyPass / balancer://mycluster/

 # The location of the HTML files, and access control information
DocumentRoot /var/www
 <Directory /var/www>
  Options -Indexes
  Order allow,deny
  Allow from all
 </Directory>

</VirtualHost>
```

Les exemples ci-dessus communiquent tous par le protocole HTTP. Vous pouvez également utiliser les protocoles AJP ou HTTPS si vous chargez les modules **mod_proxy**. Voir la **mod_proxy** documentation http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations.

4. **Activer les sticky sessions.**
Sticky sessions signifie que si la demande d'un client va initialement à un nœud spécifique de JBoss EAP 6, toutes les demandes futures seront envoyées au même nœud, sauf si le nœud n'est plus disponible. C'est presque toujours le comportement correct.

Pour activer des sticky sessions du **mod_proxy**, ajoutez le paramètre **stickysession** à l'énoncé **ProxyPass**. Cet exemple montre également d'autres paramètres que vous pouvez utiliser. Reportez-vous à documentation **mod_proxy** Apache à http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations à leur sujet.

```
ProxyPass /MyApp balancer://mycluster stickysession=JSESSIONID
lbmethod=bytraffic nofailover=Off
```

5. **Redémarrer le serveur web.**
   Redémarrer le serveur web pour appliquer les changements.

Résultat

Votre serveur http Apache est configuré pour utiliser le **`mod_proxy`** pour envoyer des demandes de client aux serveurs ou clusters de JBoss EAP 6, en configuration standard ou équilibrage de charge. Pour configurer la plate-forme JBoss EAP 6 pour répondre à ces demandes, reportez-vous à Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes ».

Rapporter un bogue

19.8. MICROSOFT ISAPI

19.8.1. Internet Server API (ISAPI) HTTP Connector

Internet Server API (ISAPI) est le connecteur HTTP du serveur web d'IIS (Internet INformation Services) de Microsoft. Vous pouvez utiliser JBoss EAP 6 en tant que noeud de worker dans le cluster IIS.

Pour configurer JBoss EAP 6 pour qu'il participe à un cluster IIS, voir Section 19.8.3, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI ». Pour plus d'informations sur ISAPI, voir http://msdn.microsoft.com/en-us/library/ms524911(v=VS.90).aspx.

Rapporter un bogue

19.8.2. Téléchargement et extraction de Webserver Connector Natives dans Microsoft IIS

1. Ouvrir un navigateur et connectez-vous au Portail clients de Red Hat à l'adresse suivante https://access.redhat.com.

2. Naviguez dans **`Downloads`**, puis **`Red Hat JBoss Middleware Download Software`**, et enfin, sélectionner **`Enterprise Application Platform`** à partir du menu déroulant **`Product`**.

3. Sélectionner la version qui convient à partir du menu déroulant **`Version`**.

4. Choisissez entre l'option **`Download`** et **`Red Hat JBoss Enterprise Application Platform 6.3.0 Webserver Connector Natives for Windows Server 2008 x86_64`** ou encore **`Red Hat JBoss Enterprise Application Platform 6.3.0 Webserver Connector Natives for Windows Server 2008 i686`** suivant l'architecture du serveur.

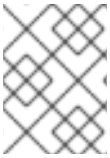
5. Extraire le fichier zip et copier son contenu dans le répertoire **`jboss-eap-6.3/modules/system/layers/base/native/sbin`** vers une location sur votre serveur. Le reste de cette tâche assume que vous avez copié le contenu**`C:\connectors\`**.

Rapporter un bogue

19.8.3. Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI

Conditions préalables :

- Section 19.8.2, « Téléchargement et extraction de Webserver Connector Natives dans Microsoft IIS »



NOTE

Voir https://access.redhat.com/site/articles/111663 pour une liste complète des configurations prises en charge par Microsoft Windows Server et IIS.

Procédure 19.17. Configurer IIS Redirector à l'aide du IIS Manager (IIS 7)

1. Ouvrir le IIS Manager en cliquant sur **Start** → **Run**, puis, saisissez **`inetmgr`**.
2. Dans le panneau de vue d'aroborescence, développer **`IIS 7`**.
3. Cliquer à deux fois sur **`ISAPI and CGI Registrations`** pour l'ouvrir sous forme d'une fenêtre séparée.
4. Dans le panneau **`Actions`**, cliquer sur **`Add`**. La fenêtre **`Add ISAPI or CGI Restriction`** s'ouvrira.
5. Indiquer les valeurs suivantes :
   - **`ISAPI or CGI Path`**: **`c:\connectors\isapi_redirect.dll`**
   - **`Description`**: **`jboss`**
   - **`Allow extension path to execute`**: sélectionner la case à cocher.
6. Cliquer sur **`OK`** pour fermer la fenêtre **`Add ISAPI or CGI Restriction`**.
7. **Définir un répertoire virtuel JBoss Native**
   a. Cliquer à droite sur **`Default Web Site`**, puis cliquer sur **`Add Virtual Directory`**. La fenêtre **`Add Virtual Directory`** va s'ouvrir.
   b. Indiquer les valeurs suivantes pour ajouter un répertoire virtuel :
      - **`Alias`**: **`jboss`**
      - **`Physical Path`**: **`C:\connectors\`**
   c. Cliquer sur **`OK`** pour sauvegarder les valeurs et fermer la fenêtre **`Add Virtual Directory`**.
8. **Définir un filtre JBoss Native ISAPI Redirect**
   a. Dans le panneau de vue d'arborescence, développer **Sites** → **Default Web Site**.
   b. Cliquer à deux fois sur **`Filtres ISAPI`**. L'affichage **`ISAPI Filters Features`** apparaîtra.
   c. Dans le panneau **`Actions`**, cliquer sur **`Add`**. La fenêtre **`Add ISAPI Filter`** s'ouvrira.
   d. Indiquer les valeurs suivantes dans la fenêtre **`Add ISAPI Filter`**:
      - **`Filter name`**: **`jboss`**

- **Executable**: **C:\connectors\isapi_redirect.dll**

e. Cliquer **OK** pour sauvegarder les valeurs et pour fermer la fenêtre **Add ISAPI Filters**.

9. **Activer le handler ISAPI-dll**

a. Cliquer deux fois sur l'élément **IIS 7** qui se trouve sur le panneau d'affichage : **IIS 7 Home Features View** s'ouvrira.

b. Cliquer deux fois sur **Handler Mappings**: **Handler Mappings Features View** s'ouvrira.

c. Dans la liste déroulante modifiable **Group by** sélectionner **State**, les **Handler Mappings** s'affichent dans **Enabled and Disabled Groups**.

d. Trouver **ISAPI-dll**. S'il se trouve dans le groupe **Disabled**, cliquer à droite, et sélectionner **Edit Feature Permissions**.

e. Activer les permissions suivantes :

- Read
- Script
- Execute

f. Cliquer sur **OK** pour sauvegarder les valeurs, et fermer la fenêtre **Edit Feature Permissions**.

Résultat

Microsoft IIS est maintenant configuré pour utiliser le re-directeur ISAPI Redirector. Ensuite, Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes », puis Section 19.8.4, « Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 » ou Section 19.8.5, « Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 ».

Rapporter un bogue

19.8.4. Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6

Aperçu

Cette tâche configure un groupe de serveurs de JBoss EAP 6 pour qu'ils puissent accepter les demandes du redirecteur ISAPI. Il n'inclut pas la configuration d'équilibrage de charge ou de haute disponibilité avec basculement. Si vous avez besoin de ces fonctionnalités, reportez-vous à Section 19.8.5, « Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6 ».

Cette configuration est faîte sur le serveur IIS, et assume que JBoss Enterprise Application Platform est déjà configurée, comme dans Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes ».

Conditions préalables

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS

- Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »
- Section 19.8.3, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI »

Procédure 19.18. Modifier les fichiers de propriété et configurer la redirection

1. **Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.**
   Le reste de cette procédure suppose que vous utilisez le répertoire **`C:\connectors\`** à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. **Créer le fichier `isapi_redirect.properties`.**
   Créer un nouveau fichier intitulé **`C:\connectors\isapi_redirect.properties`**. Copier les contenus suivants dans le fichier.

   ```
   # Configuration file for the ISAPI Redirector
   # Extension uri definition
   extension_uri=/jboss/isapi_redirect.dll

   # Full path to the log file for the ISAPI Redirector
   log_file=c:\connectors\isapi_redirect.log

   # Log level (debug, info, warn, error or trace)
   # Use debug only testing phase, for production switch to info
   log_level=debug

   # Full path to the workers.properties file
   worker_file=c:\connectors\workers.properties

   # Full path to the uriworkermap.properties file
   worker_mount_file=c:\connectors\uriworkermap.properties

   #Full path to the rewrite.properties file
   rewrite_rule_file=c:\connectors\rewrite.properties
   ```

   Si vous ne souhaitez pas utiliser un fichier **`rewrite.properties`**, dé-commentez la dernière ligne en plaçant un caractère # au début de la ligne. Voir Étape 5 pour plus d'informations.

3. **Créer le fichier `uriworkermap.properties`**
   Le fichier **`uriworkermap.properties`** contient les mappages entre les URL de l'application déployée et quel worker gère leurs demandes vers eux. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez votre fichier **`uriworkermap.properties`** dans **`C:\connectors\`**.

   ```
   # images and css files for path /status are provided by worker01
   /status=worker01
   /images/*=worker01
   /css/*=worker01

   # Path /web-console is provided by worker02
   # IIS (customized) error page is used for http errors with number
   greater or equal to 400
   # css files are provided by worker01
   /web-console/*=worker02;use_server_errors=400
   /web-console/css/*=worker01
   ```

```
# Example of exclusion from mapping, logo.gif won't be displayed
# !/web-console/images/logo.gif=*

# Requests to /app-01 or /app-01/something will be routed to
worker01
/app-01|/*=worker01

# Requests to /app-02 or /app-02/something will be routed to
worker02
/app-02|/*=worker02
```

4. **Créer le fichier `workers.properties`.**
   Le fichier **`workers.properties`** contient des définitions de mappage entre les étiquettes de workers et les instances de serveur. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez ce fichier dans le répertoire **`C:\connectors\`**.

```
# An entry that lists all the workers defined
worker.list=worker01, worker02

# Entries that define the host and port associated with these
workers

# First JBoss EAP 6 server definition, port 8009 is standard port
for AJP in EAP
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

# Second JBoss EAP 6 server definition
worker.worker02.host=127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

5. **Créer le fichier `rewrite.properties`.**
   Le fichier **`rewrite.properties`** contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire **`C:\connectors\`**.

```
#Simple example
# Images are accessible under abc path
/app-01/abc/=/app-01/images/
```

6. **Redémarrer le serveur IIS.**
   Redémarrer votre serveur IIS par les commandes **`net stop`** et **`net start`**.

```
C:\> net stop was /Y
C:\> net start w3svc
```

Résultat

Le serveur IIS est configuré pour envoyer des demandes de clients à des serveurs spécifiques de JBoss EAP 6 que vous aurez configurés, sur une base spécifique à l'application.

Rapporter un bogue

19.8.5. Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP 6

Aperçu

Cette configuration équilibre les requêtes des clients entre les serveurs de JBoss EAP 6 que vous spécifiez. Si vous préférez envoyer des demandes de client à des serveurs JBoss EAP 6 spécifiques sur une base «par-déploiement», reportez-vous plutôt à Section 19.8.4, « Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP 6 ».

Cette configuration est faîte sur le serveur IIS, et assume que JBoss Enterprise Application Platform est déjà configurée, comme dans Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes ».

Conditions préalables

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS.
- Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes »
- Section 19.8.3, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI »

Procédure 19.19. Équilibrage des requêtes de clients entre des serveurs multiples.

1. **Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.**
   Le reste de cette procédure suppose que vous utilisez le répertoire **`C:\connectors\`** à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. **Créer le fichier `isapi_redirect.properties`.**
   Créer un nouveau fichier intitulé **`C:\connectors\isapi_redirect.properties`**. Copier les contenus suivants dans le fichier.

```
# Configuration file for the ISAPI Redirector
# Extension uri definition
extension_uri=/jboss/isapi_redirect.dll

# Full path to the log file for the ISAPI Redirector
log_file==c:\connectors\isapi_redirect.log

# Log level (debug, info, warn, error or trace)
# Use debug only testing phase, for production switch to info
log_level=debug

# Full path to the workers.properties file
worker_file=c:\connectors\workers.properties

# Full path to the uriworkermap.properties file
worker_mount_file=c:\connectors\uriworkermap.properties

#OPTIONAL: Full path to the rewrite.properties file
rewrite_rule_file=c:\connectors\rewrite.properties
```

Si vous ne souhaitez pas utiliser un fichier **rewrite.properties**, dé-commentez la dernière ligne en plaçant un caractère # au début de la ligne. Voir Étape 5 pour plus d'informations.

3. **Créer le fichier uriworkermap.properties.**
   Le fichier **uriworkermap.properties** contient les mappages entre les URL de l'application déployée et quel worker gère les demandes à leur intention. Le fichier exemple suivant illustre la syntaxe du fichier, avec une configuration d'équilibrage de charge. Le caractère générique (*****) envoie toutes les requêtes de divers sous-répertoires d'URL vers l'équilibreur de charges nommé **router**. La configuration de l'équilibreur de charges est couverte dans Étape 4.

   Mettez votre fichier **uriworkermap.properties** dans **C:\connectors**.

```
# images, css files, path /status and /web-console will be
# provided by nodes defined in the load-balancer called "router"
/css/*=router
/images/*=router
/status=router
/web-console|/*=router

# Example of exclusion from mapping, logo.gif won't be displayed
!/web-console/images/logo.gif=*

# Requests to /app-01 and /app-02 will be routed to nodes defined
# in the load-balancer called "router"
/app-01|/*=router
/app-02|/*=router

# mapping for management console, nodes in cluster can be enabled or
disabled here
/jkmanager|/*=status
```

4. **Créer le fichier workers.properties.**
   Le fichier **workers.properties** contient les définitions de mappage entre les étiquettes de workers et les instances de serveur. Le fichier exemple suivant illustre la syntaxe du fichier. L'équilibrage de charge est configuré vers la fin du fichier, et comprend les workers **worker01** et **worker02**. Le fichier **workers.properties** suit la syntaxe du même fichier que celui utilisé pour la configuration d'Apache mod_jk. Pour plus d'informations sur la syntaxe du fichier **workers.properties**, reportez-vous à Section 19.6.5, « Référence de configuration pour les Apache Mod_jk Workers ».

   Mettez ce fichier dans le répertoire **C:\connectors**.

```
# The advanced router LB worker
worker.list=router,status

# First EAP server definition, port 8009 is standard port for AJP in
EAP
#
# lbfactor defines how much the worker will be used.
# The higher the number, the more requests are served
# lbfactor is useful when one machine is more powerful
# ping_mode=A - all possible probes will be used to determine that
# connections are still working

worker.worker01.port=8009
```

```
worker.worker01.host=127.0.0.1
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second EAP server definition
worker.worker02.port=8009
worker.worker02.host=127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the LB worker
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker for jkmanager
worker.status.type=status
```

5. **Créer le fichier `rewrite.properties`.**
   Le fichier **`rewrite.properties`** contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire **`C:\connectors\`**.

```
#Simple example
# Images are accessible under abc path
/app-01/abc/=/app-01/images/
```

6. **Redémarrer le serveur IIS.**
   Redémarrer votre serveur IIS par les commandes **`net stop`** et **`net start`**.

```
C:\> net stop was /Y
C:\> net start w3svc
```

Résultat

Le serveur IIS est configuré pour envoyer des demandes de clients à des serveurs de JBoss EAP 6 référencés dans le fichier **`workers.properties`**, équilibrant la charge équitablement à travers les serveurs.

Rapporter un bogue

19.9. ORACLE NSAPI

19.9.1. Netscape Server API (NSAPI) HTTP Connector

Netscape Server API (NSAPI) est un connecteur HTTP qui permet à la plateforme JBoss EAP 6 de participer en tant que noeud dans le serveur Oracle iPlanet Web Server (anciennement Netscape Web Server). Pour configurer ce connecteur, consulter Section 19.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges ».

Rapporter un bogue

19.9.2. Configurer le connecteur NSAPI dans Oracle Solaris

Résumé

Le connecteur NSAPI est un module qui exécute dans le serveur Oracle iPlanet Web Server.

Conditions préalables

- Votre serveur exécute Oracle Solaris 10 ou version supérieure, soit en architecture 32-bit ou 64-bit Intel, soit en architecture SPARC64.
- Oracle iPlanet Web Server 7.0.15 ou versions supérieures pour architectures Intel, ou 7.0.14 ou versions supérieures pour les architectures SPARC, est installé ou configuré, indépendamment du connecteur NSAPI.
- La plate-forme JBoss EAP 6 est installée et configurée sur chaque serveur qui servira en tant que noeud de worker. Voir Section 19.3.6, « Configurer JBoss EAP 6 pour accepter des requêtes en provenance des serveurs web externes ».
- Le package JBoss Native Components ZIP peut être téléchargé à partir du portail clients à https://access.redhat.com.

Procédure 19.20. Extraire et installer le connecteur NSAPI

1. **Extraire le package JBoss Native Components.**
   Le reste de cette procédure assume que le package de Native Components est extrait d'un répertoire nommé *EAP_HOME*. Pour le reste de cette procédure, le répertoire **`/opt/oracle/webserver7/config/`** sera identifié comme *IPLANET_CONFIG*. Si votre répertoire de configuration Oracle iPlanet est différent, modifier la procédure en fonction.
2. **Désactiver les mappages du servlet.**
   Ouvrir le fichier ***`IPLANET_CONFIG`*`/default.web.xml`** et chercher la section avec l'en-tête **`Built In Server Mappings`**. Désactiver les mappages pour les trois servlets suivantes, en les enveloppant entre lignes de commentaires XML (**`<!--`** et **`-->`**).
   - défaut
   - invoker
   - jsp

   L'exemple de configuration suivant montre les mappages désactivés.

```
<!-- ============== Built In Servlet Mappings =============== -->
<!-- The servlet mappings for the built in servlets defined above. -
->
<!-- The mapping for the default servlet -->
<!--servlet-mapping>
 <servlet-name>default</servlet-name>
 <url-pattern>/</url-pattern>
</servlet-mapping-->
<!-- The mapping for the invoker servlet -->
<!--servlet-mapping>
 <servlet-name>invoker</servlet-name>
```

```
 <url-pattern>/servlet/*</url-pattern>
</servlet-mapping-->
<!-- The mapping for the JSP servlet -->
<!--servlet-mapping>
 <servlet-name>jsp</servlet-name>
 <url-pattern>*.jsp</url-pattern>
</servlet-mapping-->
```

Sauvegarder et sortir du fichier.

3. **Configurer iPlanet Web Server pour qu'il puisse charger le module de connecteur NSAPI.** Ajouter les lignes suivantes à la fin de ce fichier ***IPLANET_CONFIG*/magnus.conf**, en modifiant les chemins de fichiers pour qu'ils s'accordent avec votre configuration. Ces lignes définissent l'emplacement du module **nsapi_redirector.so**, ainsi que celle du fichier **workers.properties** qui liste les worker nodes et leurs propriétés.

```
Init fn="load-modules" funcs="jk_init,jk_service"
shlib="EAP_HOME/modules/system/layers/base/native/lib/nsapi_redirect
or.so" shlib_flags="(global|now)"

Init fn="jk_init"
worker_file="IPLANET_CONFIG/connectors/workers.properties"
log_level="info" log_file="IPLANET_CONFIG/connectors/nsapi.log"
shm_file="IPLANET_CONFIG/connectors/tmp/jk_shm"
```

La configuration ci-dessus est basée sur une architecture 32-bit. Si vous utilisez 64-bit Solaris, changez le string **lib/nsapi_redirector.so** en **lib64/nsapi_redirector.so**.

Sauvegarder et sortir du fichier.

4. **Configurer le connecteur NSAPI**
Vous pouvez configurer le connecteur NSAPI pour une configuration de base, avec aucun équilibrage des charges, ou une configuration d'équilibrage des charges. Choisissez l'une des options suivantes, après quoi votre configuration sera terminée

   - Section 19.9.3, « Configurer NSAPI en connecteur de base HTTP »
   - Section 19.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges »

Rapporter un bogue

19.9.3. Configurer NSAPI en connecteur de base HTTP

Aperçu

Cette tâche configure le connecteur NSAPI à rediriger les demandes des clients aux serveurs JBoss EAP 6 sans aucun équilibrage de charge ou basculement. La redirection se fait sur la base d'un déploiement (est donc basé-URL). Pour une configuration d'équilibrage des charges, consultez Section 19.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges » à la place.

Conditions préalables

- Vous devez compléter Section 19.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris » avant de continuer avec la tâche suivante.

Procédure 19.21. Installer le connecteur HHP de base

1. **Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP 6.**

   NOTE

   Dans ***IPLANET_CONFIG*/obj.conf**, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

   Modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Chercher la section qui commence par **<Object name="default">**, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string **jknsapi** fait référence au connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractères génériques pour la correspondance de modèles.

   ```
   <Object name="default">
   [...]
   NameTrans fn="assign-name" from="/status" name="jknsapi"
   NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
   NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
   NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
   NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
   </Object>
   ```

2. **Définir le worker qui sert chaque chemin d'accès.**
   Continuer à modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer : **</Object>**.

   ```
   <Object name="jknsapi">
   ObjectType fn=force-type type=text/plain
   Service fn="jk_service" worker="worker01" path="/status"
   Service fn="jk_service" worker="worker02" path="/nc(/*)"
   Service fn="jk_service" worker="worker01"
   </Object>
   ```

   L'exemple ci-dessus redirige les requêtes vers le chemin URL **/status** et le worker nommé **worker01**, et tous les URL en-dessous **/nc/** vers le worker **worker02**. La troisième ligne indique que tous les URL assignés à l'objet **jknsapi** qui n'ont pas de correspondance avec les lignes précédentes sont servis à **worker01**.

   Sauvegarder et sortir du fichier.

3. **Définir les workers et leurs attributs.**
   Créer un fichier intitulé **workers.properties** dans le répertoire **IPLANET_CONFIG/connectors/**. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

   ```
   # An entry that lists all the workers defined
   worker.list=worker01, worker02

   # Entries that define the host and port associated with these
   workers
   ```

```
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

worker.worker02.host=127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

Le fichier **workers.properties** utilise la même syntaxe qu'Apache mod_jk. Pour obtenir plus d'informations sur les options disponibles, voir Section 19.6.5, « Référence de configuration pour les Apache Mod_jk Workers ».

Sauvegarder et sortir du fichier.

4. **Redémarrer le serveur iPlanet Web Server.**
Lancer les commandes suivantes pour redémarrer le serveur iPlanet Web Server.

```
IPLANET_CONFIG/../bin/stopserv
IPLANET_CONFIG/../bin/startserv
```

Résultat

iPlanet Web Server envoie maintenant les requêtes clients aux URL que vous avez configurés vers les déploiements de JBoss EAP 6.

Rapporter un bogue

19.9.4. Configurer NSAPI en tant que Cluster d'équilibrage des charges

Aperçu

Cette tâche configure le connecteur NSAPI à rediriger les demandes des clients aux serveurs JBoss EAP 6 dans une configuration d'équilibrage des charges. Pour utiliser NSAPI comme simple connecteur HTTP sans équilibrage des charges, voir Section 19.9.3, « Configurer NSAPI en connecteur de base HTTP ».

Conditions préalables

- Vous devez compléter Section 19.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris » avant de continuer avec la tâche suivante.

Procédure 19.22. Configurer le connecteur pour l'équilibrage des charges

1. **Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP 6.**

NOTE

Dans ***IPLANET_CONFIG*/obj.conf**, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

Modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Chercher la section qui commence par **<Object name="default">**, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string **jknsapi** fait référence au

connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractères génériques pour la correspondance de modèles.

```
<Object name="default">
[...]
NameTrans fn="assign-name" from="/status" name="jknsapi"
NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jkmanager/*" name="jknsapi"
</Object>
```

2. **Définir le worker qui sert chaque chemin d'accès.**
   Continuer à modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer dans l'étape précédente (**</Object>**), et modifiez là suivant vos besoins :

```
<Object name="jknsapi">
ObjectType fn=force-type type=text/plain
Service fn="jk_service" worker="status" path="/jkmanager(/*)"
Service fn="jk_service" worker="router"
</Object>
```

   Cet objet **jksnapi** définit les noeuds de worker utilisés pour servir chaque chemin relié au mappage **name="jksnapi"** de l'objet **default**. Tout sauf les URL correspondant à **/jkmanager/*** sont redirigés vers le worker nommé **router**.

3. **Définir les workers et leurs attributs.**
   Créer un fichier intitulé **workers.properties** dans le répertoire **IPLANET_CONFIG/connector/**. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

```
# The advanced router LB worker
# A list of each worker
worker.list=router,status

# First JBoss EAP server
# (worker node) definition.
# Port 8009 is the standard port for AJP
#

worker.worker01.port=8009
worker.worker01.host=127.0.0.1
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second JBoss EAP server
worker.worker02.port=8009
worker.worker02.host=127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
```

```
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the load-balancer called "router"
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker
worker.status.type=status
```

Le fichier **`workers.properties`** utilise la même syntaxe qu'Apache mod_jk. Pour obtenir plus d'informations sur les options disponibles, voir Section 19.6.5, « Référence de configuration pour les Apache Mod_jk Workers ».

Sauvegarder et sortir du fichier.

4. **Redémarrer le serveur iPlanet Web Server.**
   Choisir une des procédures suivantes, suivant que vous souhaitez exécuter iPlanet Web Server 6.1 ou 7.0.

   - **iPlanet Web Server 6.1**

     ```
     IPLANET_CONFIG/../stop
     IPLANET_CONFIG/../start
     ```

   - **iPlanet Web Server 7.0**

     ```
     IPLANET_CONFIG/../bin/stopserv
     IPLANET_CONFIG/../bin/startserv
     ```

Résultat

iPlanet Web Server redirige les modèles d'URL que vous avez configurés à vos serveurs JBoss EAP 6 dans une configuration d'équilibrage des charges.

Rapporter un bogue

CHAPITRE 20. MESSAGERIE

20.1. INTRODUCTION

20.1.1. HornetQ

HornetQ est un système de messagerie multiprotocole, asynchrone développé par Red Hat. HornetQ procure une haute disponibilité (HA) avec basculement automatique des clients pour garantir la fiabilité du message dans le cas d'une panne de serveur. HornetQ prend également en charge des solutions de clustering flexibles avec équilibrage de charge des messages.

HornetQ est le fournisseur JMS (Java Message Service) de JBoss EAP 6 et est configuré comme **`Messaging Subsystem`**

Rapporter un bogue

20.1.2. Java Messaging Service (JMS)

Les systèmes de messagerie vous permettent de coupler de façon informelle des systèmes hétérogènes avec une fiabilité supplémentaire. Les fournisseurs de Service JMS (Java Messaging) utilisent un système de transactions pour valider ou annuler les modifications atomiquement. Contrairement aux systèmes basés sur un modèle d'échange de données informatisé (RPC, Remote Procedure Call), les systèmes de messagerie utilisent principalement un modèle de passage de messages asynchrone avec aucune relation réelle entre les demandes et les réponses. La plupart des systèmes de messagerie supportent également un mode de requête-réponse, mais ce n'est pas une caractéristique principale des systèmes de messagerie.

Les systèmes de messagerie découplent les expéditeurs des messages des consommateurs de messages. Les expéditeurs et les consommateurs de messages sont complètement indépendants et ne savent rien l'un de l'autre. Cela vous permet de créer des systèmes flexible et faiblement couplés. Souvent, les grandes entreprises utilisent un système de messagerie pour mettre en place un bus de messages qui couple légèrement des systèmes hétérogènes. Les bus de messages forment souvent la base d'une Enterprise Service Bus (ESB). En utilisant un Message Bus pour découpler des systèmes disparates, on peut permettre au système de croître et de s'adapter plus facilement. Cela permet également plus de souplesse pour ajouter de nouveaux systèmes ou en retirer des "anciens", puisqu'ils n'ont pas de dépendances fragiles les uns avec les autres.

Rapporter un bogue

20.1.3. Styles de messagerie pris en compte

HornetQ prend en charge les styles de messagerie suivants :

Modèle de file d'attente de messages

Le modèle de file d'attente de message consiste à envoyer un message à une file d'attente. Une fois dans la file d'attente, le message est normalement rendu persistant pour en garantir sa livraison. Une fois que le message s'est déplacé dans la file, le système de messagerie le livre à un consommateur de messages. Le consommateur de messages accuse réception de la livraison du message une fois qu'il a été traité.

En messagerie PPP, le modèle de file d'attente de messagerie autorise plusieurs consommateurs pour une même file d'attente, mais chaque message ne peut être reçu que par un seul consommateur.

Modèle Publish-Subscribe

Le modèle Publish-Subscribe permet à plusieurs émetteurs d'envoyer des messages vers une seule entité sur le serveur. Cette entité est connue sous le nom de "topic". Chaque topic peut être traité par plusieurs consommateurs, ce que l'on appelle des "abonnements" (ou "subscriptions" en anglais)

Chaque abonnement reçoit une copie des messages envoyés au topic. La différence avec le modèle de file d'attente de messages, c'est que chaque message n'est consommé que par un seul consommateur.

Les abonnements qui sont durables conservent une copie de chaque message envoyé à ce topic jusqu'à ce que l'abonné les consomme. Ces copies sont conservées même en cas d'un redémarrage du serveur. Les abonnements non durables durent le temps de la connexion qui les a créés.

Rapporter un bogue

20.2. CONFIGURATION DES TRANSPORTS

20.2.1. Accepteurs et connecteurs

HornetQ utilise le concept de connecteurs et d'accepteurs comme un élément clé du système de messagerie.

Accepteurs et connecteurs

`Acceptor`

Un accepteur définit quels types de connections sont acceptées par le serveur HornetQ.

`Connector`

Un connecteur définit comment se connecter au serveur HornetQ, et est utilisé par le client HornetQ.

Il y a deux sortes de connecteurs et d'accepteurs, suivant que le connecteur ou l'accepteur qui correspond sont dans la même JVM ou non.

Invm et Netty

`Invm`

Invm est un acronyme pour Intra Virtual Machine. Peut être utilisé quand le client et le serveur exécutent en même temps dans la même JVM.

`Netty`

Le nom d'un projet JBoss. Doit être utilisé quand le client et le serveur exécutent dans des JVM différentes.

Un client HornetQ doit utiliser un connecteur compatible avec un des accepteurs du serveur. Seul un connecteur Invm peut se connecter à un accepteur Invm, et seul un connecteur netty peut se connecter à un accepteur de netty. Les connecteurs et les accepteurs sont configurés sur le serveur dans un **`standalone.xml`** et **`domain.xml`**. Vous pouvez utiliser la console de gestion ou l'interface CLI pour les définir.

Rapporter un bogue

20.2.2. Configuration de Netty TCP

Netty TCP est un simple transport de sockets basées TCP non chiffré. Netty TCP peut être configuré pour utiliser l'ancien blocage Java IO ou Java NIO non bloquant. Java NIO est recommandé pour une meilleure évolutivité avec un grand nombre de connexions simultanées sur le serveur. Si le nombre de connexions simultanées est moindre, l'ancien Java IO peut donner une meilleure latence que NIO.

Netty TCP n'est pas conseillé pour les connexions défilant sur un réseau non sécurisé car il est encodé. Avec le transport Netty TCP, toutes les connexions sont initiées côté client.

Exemple 20.1. Exemple de configuration de Netty TCP à partir de la configuration EAP par défaut

```
<connectors>
  <netty-connector name="netty" socket-binding="messaging"/>
  <netty-connector name="netty-throughput" socket-binding="messaging-
throughput">
    <param key="batch-delay" value="50"/>
  </netty-connector>
  <in-vm-connector name="in-vm" server-id="0"/>
</connectors>
<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging"/>
  <netty-acceptor name="netty-throughput" socket-binding="messaging-
throughput">
    <param key="batch-delay" value="50"/>
    <param key="direct-deliver" value="false"/>
  </netty-acceptor>
  <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>
```

La configuration de l'exemple montre également comment l'application JBoss EAP 6 de HornetQ utilise des liaisons de socket dans la configuration du connecteur et l'accepteur. Cela diffère de la version autonome de HornetQ, qui vous oblige à déclarer les ports et les hôtes spécifiques.

Le tableau suivant décrit les propriétés de configuration de Netty TCP :

Tableau 20.1. Propriétés de configuration de Netty TCP

Property	Par défaut	Description
batch-delay	0 millisecondes	Avant d'inscrire les paquets au tranport, HornetQ peut être configuré pour regrouper les écritures pour un maximum de milisecondes de batch-delay. Cela augmentera le débit total pour les petits messages en augmentant la latence de transfert de messages.

Property	Par défaut	Description
direct-deliver	true	Lorsqu'un message arrive sur le serveur et est livré aux consommateurs qui attendent, par défaut, la livraison se fait sur le même thread que celui sur lequel le message est arrivé. Cela donne une bonne latence dans des environnements avec des messages relativement peu volumineux et un petit nombre de consommateurs, mais réduit le débit et la latence. Pour un débit plus élevé, vous pouvez définir cette propriété comme « false »
local-address	[local address available]	Pour un connecteur netty, c'est utilisé pour indiquer l'adresse locale que le client va utiliser quand il se connectera à l'adresse distante. Si une adresse locale n'est pas spécifiée, le connecteur utilisera n'importe quelle adresse locale disponible.
local-port	0	Pour un connecteur netty, c'est utilisé pour indiquer le port local que le client va utiliser quand il se connectera à l'adresse distante. Si le port local par défaut est utilisé (0), le connecteur laissera le système collecter un port éphémère. Les ports valides sont 0 à 65535
nio-remoting-threads	-1	Si configuré pour utiliser NIO, HornetQ, par défaut, utilise un nombre de threads égal à trois fois le nombre de noyaux (ou hyper-threads) rapporté par Runtime.getRuntime().availableProcessors() pour le traitement des paquets entrants. Pour substituer cette valeur, vous pouvez définir une valeur personnalisée pour le nombre de threads
tcp-no-delay	true	Si sur true, alors l'agorithme sera activé. Cet algorithme aide à améliorer l'efficacité des réseaux TCP/IP en réduisant le nombre de paquets à envoyer sur le réseau

Property	Par défaut	Description
tcp-send-buffer-size	32768 bytes	Ce paramètre détermine la taille du tampon d'envoi TCP en octets
tcp-receive-buffer-size	32768 bytes	Ce paramètre détermine la taille du tampon de réception TCP en octets
use-nio	false	Si cela est sur true, alors les NIO Java non bloquantes seront utilisées. Si sur false, alors les anciens e/s non bloquantes de Java seront utilisées. Si vous avez besoin que le serveur utilise plusieurs connexions NIO Java concourantes non bloquantes, sinon, choisissez les anciennes e/s bloquantes

NOTE

Les propriétés de Netty TCP sont valides pour tous les types de transport (Netty SSL, Netty HTTP et Netty Servlet).

Rapporter un bogue

20.2.3. Configuration de Netty Secure Sockets Layer (SSL)

Netty TCP est un simple transport de sockets basé TCP non chiffré. Netty ressemble à Netty TCP mais il procure une sécurité supplémentaire en chiffrant les connexions TCP par SSL (Secure Sockets Layer). L'exemple suivant montre la configuration Netty pour SSL One Way :

NOTE

La plupart des paramètres suivants peuvent être utilisés avec des accepteurs comme connecteurs. Toutefois, certains paramètres fonctionnent uniquement avec les accepteurs. La description du paramètre explique la différence entre l'utilisation de ces paramètres dans des connecteurs ou des accepteurs.

```
<acceptors>
 <netty-acceptor name="netty" socket-binding="messaging"/>
   <param key="ssl-enabled" value="true"/>
   <param key="key-store-password" value="[keystore password]"/>
   <param key="key-store-path" value="[path to keystore file]"/>
 </netty-acceptor>
</acceptors>
```

Tableau 20.2. Propriétés de configuration de Netty SSL

Nom de propriété	Par défaut	Description
ssl-enabled	true	Active SSL
key-store-password	[keystore password]	Lorsqu'il est utilisé sur un accepteur, c'est le mot de passe du keystore côté serveur. Lorsqu'il est utilisé sur un connecteur, c'est le mot de passe du keystore côté client. C'est pertinent pour un connecteur si vous utilisez SSL Two Ways (authentification dans les deux sens). Cette valeur peut être configurée sur le serveur, mais elle sera téléchargée et utilisée par le client
key-store-path	[chemin vers le fichier keystore]	Lorsqu'il est utilisé sur un accepteur, c'est le chemin du trust store SSL côté serveur qui détient le clés de tous les clients que le serveur contient et à qui il fait confiance. Lorsqu'il est utilisé sur un connecteur, c'est le chemin du trust store SSL Two Ways côté client qui détient les clés publiques de tous les serveurs à qui le client fait confiance. C'est pertinent pour un connecteur si vous utilisez SSL Two Ways (authentification dans les deux sens). Ce chemin peut être configuré sur le serveur, mais sera téléchargé et utilisé par le client

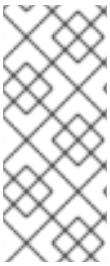
Si vous configurez Netty en SSL Two Ways (authentification mutuelle entre serveur et client), il y a trois paramètres supplémentaires en plus de ceux décrits dans l'exemple ci-dessus pour SSL One Way :

- ***need-client-auth*** : indique le besoin de l'authentification Two Way (dans les deux sens) pour les connexions client.
- ***trust-store-password*** : lorsqu'il est utilisé sur un accepteur, c'est le mot de passe du trust store côté serveur. Lorsqu'il est utilisé sur un connecteur, c'est le mot de passe du trust store côté client. C'est pertinent pour un connecteur à la fois en SSL One Way et SSL Two Ways. Cette valeur peut être configurée sur le serveur, mais elle sera téléchargée et utilisée par le client
- ***trust-store-path*** : lorsqu'il est utilisé sur un accepteur, c'est le chemin du keystore SSL côté serveur qui détient le clés de tous les clients à qui le serveur fait confiance. Lorsqu'il est utilisé sur un connecteur, c'est le chemin du keystore SSL côté client qui détient les clés publiques de tous les serveurs à qui le client fait confiance. C'est pertinent pour un connecteur si vous utilisez SSL Two Ways ou One Way. Ce chemin peut être configuré sur le serveur, mais sera téléchargé et utilisé par le client

Rapporter un bogue

20.2.4. Configuration de Netty HTTP

Netty HTTP conduit des paquets sur le protocole HTTP. Il peut être utile dans les scénarios où les pare-feux permettre uniquement le trafic HTTP. Netty HTTP utilise les mêmes propriétés que Netty TCP, ainsi que les quelques propriétés supplémentaires suivantes :



NOTE

Les paramètres suivants peuvent être utilisés par les accepteurs ainsi que par les connecteurs. Le transport Netty HTTP ne permet pas la réutilisation du standard HTTP port (8080 par défaut). L'utilisation du port HTTP standard entraîne une exception. Vous pouvez utiliser Section 20.2.5, « Configuration de Netty Servlet » (Netty Servlet Transport) pour le tunneling des connexions HornetQ via un port HTTP standard.

```
<socket-binding name="messaging-http" port="7080" />
```

```
<acceptors>
  <netty-acceptor name="netty" socket-binding="messaging-http">
    <param key="http-enabled" value="false"/>
    <param key="http-client-idle-time" value="500"/>
    <param key="http-client-idle-scan-period" value="500"/>
    <param key="http-response-time" value="10000"/>
    <param key="http-server-scan-period" value="5000"/>
    <param key="http-requires-session-id" value="false"/>
  </netty-acceptor>
</acceptors>
```

Le tableau suivant décrit les propriétés supplémentaires de configuration de Netty HTTP :

Tableau 20.3. Propriétés de configuration de Netty HTTP

Nom de propriété	Par défaut	Description
http-enabled	false	Si défini à true, HTTP est activé
http-client-idle-time	500 millisecondes	La durée pendant laquelle un client peut être inactif avant d'envoyer une demande de connexion HTTP vide pour conserver la connexion active
http-client-idle-scan-period	500 millisecondes	La fréquence (en ms) de balayage des clients inactifs
http-response-time	10000 millisecondes	La durée pendant laquelle le serveur peut patienter avant d'envoyer une réponse HTTP vide pour conserver la connexion active

Nom de propriété	Par défaut	Description
http-server-scan-period	5000 millisecondes	La fréquence, en millisecondes, de balayage des clients en attente de réponses
http-requires-session-id	false	Si défini sur true, le client devra attendre après le premier appel avant de recevoir une ID de session

AVERTISSEMENT

Failover automatique non pris en charge pour les clients se connectant via un transport Netty HTTP.

Rapporter un bogue

20.2.5. Configuration de Netty Servlet

Le transport de servlet autorise le trafic HornetQ via HTTP à un servlet exécutant dans un moteur de servlet qui le redirige ensuite vers un serveur de HornetQ in-VM. Le transport Netty HTTP agit comme un serveur web à l'écoute du trafic HTTP sur des ports spécifiques. Avec le transport de servlet, le trafic HornetQ est mandaté par un moteur de servlet qui puisse déjà servir à un site web ou à d'autres applications.

Pour pouvoir configurer un moteur de servlet qui puisse fonctionner sur un transport de servet Netty, vous devrez suivre les étapes suivantes :

- Déployer le servlet : l'exemple suivant décrit une application web qui utilise le servlet :

```
<web-app>
  <servlet>
    <servlet-name>HornetQServlet</servlet-name>
    <servlet-
class>org.jboss.netty.channel.socket.http.HttpTunnelingServlet</serv
let-class>
    <init-param>
      <param-name>endpoint</param-name>
      <param-value>local:org.hornetq</param-value>
    </init-param>
      <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>HornetQServlet</servlet-name>
```

```
    <url-pattern>/HornetQServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

Le paramètre init ***endpoint*** spécifie l'attribut d'hôte de l'accepteur Netty à qui le servlet va envoyer ses packages.

- Insérer l'accepteur de Netty servlet sur la configuration côté serveur : l'exemple suivant montre la définition d'un accepteur en serveur de fichiers de configuration (**`standalone.xml`** et **`domain.xml`**) :

```
<acceptors>
   <acceptor name="netty-servlet">
      <factory-class>
         org.hornetq.core.remoting.impl.netty.NettyAcceptorFactory
      </factory-class>
      <param key="use-servlet" value="true"/>
      <param key="host" value="org.hornetq"/>
   </acceptor>
</acceptors>
```

- La dernière étape consiste à définir un connecteur pour le client dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) :

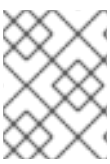
```
<netty-connector name="netty-servlet" socket-binding="http">
   <param key="use-servlet" value="true"/>
   <param key="servlet-path" value="/messaging/HornetQServlet"/>
</netty-connector>
```

- Il est également possible d'utiliser le transport de servlet via SSL en ajoutant la configuration suivante au connecteur :

```
<netty-connector name="netty-servlet" socket-binding="https">
   <param key="use-servlet" value="true"/>
   <param key="servlet-path" value="/messaging/HornetQServlet"/>
   <param key="ssl-enabled" value="true"/>
   <param key="key-store-path" value="path to a key-store"/>
   <param key="key-store-password" value="key-store password"/>
</connector>
```

AVERTISSEMENT

Failover automatique non pris en charge pour les clients se connectant via HTTP tunneling servlet.



NOTE

Le servlet Netty ne peut pas être utilisé pour configurer les serveurs EAP 6 pour mettre en place un cluster HornetQ.

Rapporter un bogue

20.3. JNDI (JAVA NAMING AND DIRECTORY INTERFACE)

L'API *Java Naming and Directory Interface (JNDI)* est un API standard Java pour les services de répertoire et de nommage. Il permet aux technologies basées-Java de découvrir ou d'organiser des composants de noms dans un environnement informatique distribué.

Rapporter un bogue

20.4. TRAVAILLER AVEC DES MESSAGES VOLUMINEUX

20.4.1. Travailler avec des messages volumineux

HornetQ prend en charge l'utilisation de messages volumineux, même lorsque le client ou le serveur a limité la quantité de mémoire. Les messages volumineux peuvent être traités tels quels, ou comprimés davantage pour un transfert plus efficace. Un utilisateur peut envoyer un message volumineux en définissant un **`InputStream`** dans le corps du message. Lorsque le message est envoyé, HornetQ lit ce **`InputStream`** et transmet les données au serveur par fragments.

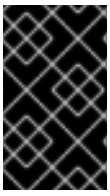
Le client et le serveur ne stockent jamais le corps complet d'un message volumineux en mémoire. Le consommateur reçoit initialement un message volumineux avec un corps vide et affecte par la suite un **`OutputStream`** au message pour obtenir des fragments dans un fichier disque.

Rapporter un bogue

20.4.2. Configurer des messages volumineux d'HornetQ

Configurer le serveur

En mode autonome, les messages volumineux sont stockés dans le répertoire **`EAP_HOME/standalone/data/largemessages`**. En mode domaine, les messages volumineux sont stockés dans le répertoire **`EAP_HOME/domain/servers/SERVERNAME/data/largemessages`**. La propriété de configuration **`large-messages-directory`** indique l'endroit où les messages volumineux sont stockés.



IMPORTANT

Pour une meilleure performance, nous vous recommandons de stocker le répertoire de messages volumineux sur un volume physique différent du journal de messages ou sur le répertoire de pagination.

Rapporter un bogue

20.4.3. Configurer les paramètres

Vous pouvez configurer des messages volumineux d'HornetQ en définissant les divers paramètres suivants :

Avec Hornet Core API côté client

Si vous utilisez HornetQ Core API côté client, vous devez définir le paramètre **`ServerLocator.setMinLargeMessageSize`** pour spécifier la taille minimale des messages volumineux. La taille minimale de messages (min-large-message-size) a la valeur 100KiB par défaut.

```
ServerLocator locator = HornetQClient.createServerLocatorWithoutHA(new
TransportConfiguration(NettyConnectorFactory.class.getName()))

locator.setMinLargeMessageSize(25 * 1024);

ClientSessionFactory factory =
HornetQClient.createClientSessionFactory();
```

Configurer le serveur pour les clients JMS (Java Messaging Service)

Si vous utilisez Java Messaging Service (JMS) vous devez spécifier la taille minimale des messages volumineux avec l'attribut **`min-large-message-size`** de vos fichiers de configuration de serveur (**`standalone.xml`** et **`domain.xml`**). La taille minimale de messages(min-large-message-size) a la valeur 100KiB par défaut.

NOTE

La valeur de l'attribut **`min-large-message-size`** doit être en octets

Vous pouvez choisir de compresser les messages volumineux pour un transfert rapide et efficace. Toutes les opérations de compression/de-compression sont gérées sur le côté client. Si le message compressé est plus petit que **`min-large-message-size`**, il sera envoyé au serveur comme un message ordinaire. À l'aide de Java Messaging Service (JMS), vous pouvez compresser des messages volumineux en définissant la propriété booléenne de **`compress-large-messages`** à "true" sur l'indice de serveur ou ConnectionFactory.

```
<connection-factory name="ConnectionFactory">
   <connectors>
      <connector-ref connector-name="netty"/>
   </connectors>
...
   <min-large-message-size>204800</min-large-message-size>
   <compress-large-messages>true</compress-large-messages>
</connection-factory>
```

Rapporter un bogue

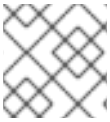
20.5. PAGINATION

20.5.1. La pagination

HornetQ supporte plusieurs files d'attente, comprenant chacune des millions de messages. Le serveur HornetQ exécute avec une mémoire limitée, ce qui rend le stockage de toutes les files d'attente de messages en mémoire à la fois difficile.

La pagination est un mécanisme utilisé par le serveur HornetQ pour paginer les messages vers et en provenance de la mémoire selon les besoins afin d'accommoder les files d'attente volumineuses dans un espace mémoire limité.

HornetQ commence par paginer les messages dans des disques quand la taille des messages en mémoire d'une adresse particulière commence à dépasser la taille de message configurée maximum.



NOTE

La pagination HornetQ est activée par défaut.

Rapporter un bogue

20.5.2. Les fichiers de pagination

Il y a un dossier individuel pour chaque adresse sur le système de fichiers qui stocke les messages dans plusieurs fichiers. Ces fichiers qui stockent les messages sont appelés fichiers de pagination. Chaque fichier contient des messages avec une taille maximale de message configurée (**`page-size-bytes`**).

Le système navigue dans les fichiers de pagination selon les besoins, et supprime les fichiers de pagination dès que tous les messages figurant sur la page sont reçus par le client.

Les consommateurs ayant des sélecteurs devront également naviguer à travers les fichiers de pagination et ignoreront les messages qui ne correspondent pas au critère.

Rapporter un bogue

20.5.3. Configuration d'un dossier de pagination

Des paramètres de pagination globaux sont spécifiés dans les fichiers de configuration du serveur (standalone.xml et domain.xml). Vous pourrez configurer l'emplacement du répertoire/dossier de pagination en utilisant le paramètre ***`paging-directory`***.

```
<hornetq-server>
 ...
 <paging-directory>/location/paging-directory</paging-directory>
 ...
</hornetq-server>
```

Le paramètre ***`paging-directory`*** est utilisé pour spécifier un emplacement/dossier où stocker les fichiers de pagination. HornetQ crée un dossier pour chaque adresse de pagination dans le répertoire de pagination. Les fichiers de pagination sont stockés dans ces dossiers.

Le répertoire de pagination par défaut est **`EAP_HOME/standalone/data/messagingpaging`** (standalone mode) et **`EAP_HOME/domain/servers/SERVERNAME/data/messagingpaging`** (en mode de domaine).

Rapporter un bogue

20.5.4. Mode de pagination

Quand les messages qui sont envoyés à une adresse dépassent la taille configurée, cette adresse va en "page/paging mode".

NOTE

La pagination est faîte individuellement par adresse. Si vous configurez un ***max-size-bytes*** d'adresse, cela signifie que chaque adresse correspondante aura une taille maximum que vous aurez spécifiée. Cependant, cela ne veut pas dire que la taille de toutes les adresses correspondantes soit limitée à ***max-size-bytes***,

Vous pouvez configurer la taille maximum en bytes (***max-size-bytes***) pour une adresse dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**) :

```
<address-settings>
   <address-setting match="jms.someaddress">
      <max-size-bytes>104857600</max-size-bytes>
      <page-size-bytes>10485760</page-size-bytes>
      <address-full-policy>PAGE</address-full-policy>
   </address-setting>
</address-settings>
```

Le tableau suivant décrit les paramètres utilisés pour la configuration de l'adresse :

Tableau 20.4. Configuration des paramètres de l'adresse de pagination

Élément	Valeur par défaut	Description
max-size-bytes	10485760	Ceci est utilisé pour indiquer la taille maximum de la mémoire que l'adresse peut avoir avant d'entrer en mode de pagination.
page-size-bytes	2097152	Cela est utilisé pour spécifier la taille de chaque fichier de pagination sur le système de pagination.
address-full-policy	PAGE	Cette valeur de cet attribut est utilisée pour les décisions de pagination. Vous pouvez définir n'importe quelle de ces valeurs pour cet attribut : **PAGE** : pour activer la pagination et les messages de pagination au delà de la limite définie sur le disque, **DROP** : pour supprimer silencieusement les messages qui excèdent la linite définie, **FAIL** : pour supprimer les messages et envoyer une exception aux producteurs de messages au client, **BLOCK** : pour bloquer les producteurs de messages client quand ils envoient des messages au delà de la limite définie
page-max-cache-size	5	Le système conservera des fichiers de pagination à hauteur de ***page-max-cache-size*** en mémoire pour optimiser l'Input/Output pendant la navigation de pagination

IMPORTANT

Si vous ne souhaitez pas paginer des messages quand la taille maximum est atteinte, vous pouvez configurer une adresse pour pouvoir supprimer les messages tout simplement, supprimer les messages avec une exception côté client ou bloquer les producteurs de messages d'envoyer des messages supplémentaires, en définissant ***address-full-policy*** à **DROP**, **FAIL** ou **BLOCK** respectivement. Dans la configuration par défaut, toutes les adresses devront être configurées pour paginer les messages une fois qu'une adresse atteint ***max-size-bytes***.

Adresses avec files d'attente multiples

Quand un message est rerouté vers une adresse ayant plusieurs files d'attente, il n'y aura qu'un message en mémoire. Chaque file d'attente gère une référence du message. Ainsi, la mémoire n'est libérée que quand toutes les files d'attente référençant le message auront délivré le message.

NOTE

Un abonnement/ ou file d'attente lazy peuvent réduire la performance Input/Output de toute l'adresse car toutes les files d'attente auront des messages envoyés par l'intermédiaire d'un stockage supplémentaire dans le système de pagination.

Rapporter un bogue

20.6. CONFIGURATION

20.6.1. Configurer le serveur JMS

Pour configurer le JMS Server d'HornetQ, modifier le fichier de configuration du serveur. La configuration du serveur se trouve dans le fichier ***EAP_HOME*/domain/configuration/domain.xml** pour les serveurs de domaine, ou dans le fichier ***EAP_HOME*/standalone/configuration/standalone-full.xml** pour les serveurs autonomes.

L'élément **<subsystem xmlns="urn:jboss:domain:messaging:1.4">** du fichier de configuration du serveur contient toute la configuration JMS. Ajouter les instances **ConnectionFactory**, **Queue**, ou **Topic** requises pour le JNDI.

1. **Activer le sous-système JMS dans JBoss EAP 6**
   Dans l'élément **<extensions>**, vérifier que la ligne suivante est bien présente et n'est pas dé-commentée :

   ```
   <extension module="org.jboss.as.messaging"/>
   ```

2. **Ajouter le sous-système JMS de base.**
   Si le sous-système de messagerie n'est pas présent dans votre fichier de configuration, ajoutez-le.

   a. Cherchez le **<profile>** qui correspond à celui que vous utilisez, et chercher sa balise de **<subsystems>**.

   b. Ajouter le XML suivant à la suite de la balise suivante **<profile>**.

   ```
   <subsystem xmlns="urn:jboss:domain:messaging:1.4">
       <hornetq-server>
       <!-- ALL XML CONFIGURATION IS ADDED HERE -->
       </hornetq-server>
   </subsystem>
   ```

   Toutes les configurations supplémentaires pourront être ajoutées à la ligne vide ci-dessus.

3. **Ajouter la configuration de base à JMS.**
   Ajouter l'XML suivant dans les lignes restées vides juste aprés la balise **<subsystem xmlns="urn:jboss:domain:messaging:1.4"><hornetq-server>** :

```
<journal-min-files>2</journal-min-files>
<journal-type>NIO</journal-type>
<persistence-enabled>true</persistence-enabled>
```

Personnaliser les valeurs ci-dessus pour qu'elles correspondent à vos besoins.

AVERTISSEMENT

La valeur de **`journal-file-size`** doit être plus élevée ou égale à **`min-large-message-size`** (100KiB par défaut), ou bien le serveur ne pourra pas stocker le message.

4. **Ajouter les instances de fabrique de connexion à HornetQ**
   Le client utilise un objet **`ConnectionFactory`** JMS pour faire des connexions au serveur. Pour ajouter un objet de fabrique de connexions JMS à HornetQ, inclure une simple balise **`<jms-connection-factories>`** et un élément **`<connection-factory>`** pour chaque fabrique de connexion comme suit :

```
  <jms-connection-factories>
    <connection-factory name="InVmConnectionFactory">
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="java:/ConnectionFactory"/>
        </entries>
    </connection-factory>
    <connection-factory name="RemoteConnectionFactory">
        <connectors>
            <connector-ref connector-name="netty"/>
        </connectors>
        <entries>
            <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
        </entries>
    </connection-factory>
    <pooled-connection-factory name="hornetq-ra">
        <transaction mode="xa"/>
        <connectors>
            <connector-ref connector-name="in-vm"/>
        </connectors>
        <entries>
            <entry name="java:/JmsXA"/>
        </entries>
    </pooled-connection-factory>
</jms-connection-factories>
```

5. **Configurer les connecteurs et accepteurs `netty`**
   La fabrique de connexion JMS utilise des connecteurs et accepteurs **`netty`**. Il s'agit de

références à des objets de connecteurs ou d'accepteurs déployés dans le fichier de configuration du serveur. L'objet de connecteur détermine le transport et les paramètres utilisés pour vous connecter au serveur HornetQ. L'accepteur identifie le type de connexions acceptées par le seveur HornetQ.

Pour configurer les connecteurs **`netty`**, inclure les paramètres suivants :

```
<connectors>
    <netty-connector name="netty" socket-binding="messaging"/>
    <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
        <param key="batch-delay" value="50"/>
    </netty-connector>
    <in-vm-connector name="in-vm" server-id="0"/>
</connectors>
```

Pour configurer les accepteurs **`netty`**, inclure les paramètres suivants :

```
<acceptors>
    <netty-acceptor name="netty" socket-binding="messaging"/>
    <netty-acceptor name="netty-throughput" socket-
binding="messaging-throughput">
        <param key="batch-delay" value="50"/>
        <param key="direct-deliver" value="false"/>
    </netty-acceptor>
    <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>
```

6. **Vérifier la configuration**
   Si vous avez suivi les étapes suivantes, votre système de messagerie devra ressembler à ce qui suit :

```
<subsystem xmlns="urn:jboss:domain:messaging:1.4">
    <hornetq-server>
        <journal-min-files>2</journal-min-files>
        <journal-type>NIO</journal-type>
        <persistence-enabled>true</persistence-enabled>
        <jms-connection-factories>
            <connection-factory name="InVmConnectionFactory">
                <connectors>
                    <connector-ref connector-name="in-vm"/>
                </connectors>
                <entries>
                    <entry name="java:/ConnectionFactory"/>
                </entries>
            </connection-factory>
            <connection-factory name="RemoteConnectionFactory">
                <connectors>
                    <connector-ref connector-name="netty"/>
                </connectors>
                <entries>
                    <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
                </entries>
            </connection-factory>
```

```
            <pooled-connection-factory name="hornetq-ra">
                <transaction mode="xa"/>
                <connectors>
                    <connector-ref connector-name="in-vm"/>
                </connectors>
                <entries>
                    <entry name="java:/JmsXA"/>
                </entries>
            </pooled-connection-factory>
        </jms-connection-factories>
        <connectors>
            <netty-connector name="netty" socket-
binding="messaging"/>
            <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
                <param key="batch-delay" value="50"/>
            </netty-connector>
            <in-vm-connector name="in-vm" server-id="0"/>
        </connectors>
        <acceptors>
            <netty-acceptor name="netty" socket-
binding="messaging"/>
            <netty-acceptor name="netty-throughput" socket-
binding="messaging-throughput">
                <param key="batch-delay" value="50"/>
                <param key="direct-deliver" value="false"/>
            </netty-acceptor>
            <in-vm-acceptor name="in-vm" server-id="0"/>
        </acceptors>
    </hornetq-server>
</subsystem>
```

7. **Configurer les groupes de liaison de sockets**
   Les connecteurs netty référencent les liaisons de socket de **`messaging`** et de **`messaging-throughput`**. La liaison de socket de **`messaging`** utilise le port 5445, et la liaison de socket **`messaging-throughput`** utilise le port 5455. La balise **`<socket-binding-group>`** se situe dans une section séparée du fichier de configuration du serveur. Veillez à ce que les liaisons de socket suivantes soient présentes dans l'élément **`<socket-binding-groups>`** :

   ```
   <socket-binding-group name="standard-sockets" default-
   interface="public" port-offset="${jboss.socket.binding.port-
   offset:0}">
       ...
       <socket-binding name="messaging" port="5445"/>
       <socket-binding name="messaging-throughput" port="5455"/>
       ...
     </socket-binding-group>
   ```

8. **Ajouter les instances de file d'attente à HornetQ**
   Il y a quatre façons de configurer les instances de files d'attente (ou destinations JMS) pour HornetQ.

   - Utiliser la console de gestion

     Pour utiliser la console de gestion, le serveur devra être démarré sous le mode **`Message-`**

Enabled. Vous y parviendrez en utilisant l'option **-c** et en forçant l'utilisation du fichier de configuration **standalone-full.xml** (pour les serveurs autonomes). Ainsi, en mode autonome, ce qui suit démarrera le serveur en mode activation de message.

```
./standalone.sh -c standalone-full.xml
```

Une fois que le serveur aura démarré, connectez-vous à la console de gestion, et sélectionner l'onglet **Configuration**. Étendre le menu **Subsystems**, puis le menu **Messaging** et cliquer sur **Destinations**. À côté de **Default** dans le tableau JMS Messaging Provider, cliquer sur **View**, puis cliquer sur **Add** pour saisir les détails de la destination JSM.

- Utiliser l'interface CLI :

  Tout d'abord, connectez-vous à l'interface CLI :

  ```
  bin/jboss-cli.sh --connect
  ```

  Puis, passez au sous-système de messagerie :

  ```
  cd /subsystem=messaging/hornetq-server=default
  ```

  Finalement, exécuter une opération « add », en remplaçant les exemples de valeurs données ci-dessous par les vôtres :

  ```
  ./jms-queue=testQueue:add(durable=false,entries=
  ["java:jboss/exported/jms/queue/test"])
  ```

- Créer un fichier de configuration JMS et l'ajouter au dossier de déploiements

  Commencer à créer un fichier de configuration JMS : *example-jms.xml*. Ajouter y les entrées suivantes, en remplaçant les valeurs par les vôtres.

  ```
  <?xml version="1.0" encoding="UTF-8"?>         <messaging-
  deployment xmlns="urn:jboss:messaging-deployment:1.0">
      <hornetq-server>
          <jms-destinations>
              <jms-queue name="testQueue">
                  <entry name="queue/test"/>
                  <entry
  name="java:jboss/exported/jms/queue/test"/>
              </jms-queue>
              <jms-topic name="testTopic">
                  <entry name="topic/test"/>
                  <entry
  name="java:jboss/exported/jms/topic/test"/>
              </jms-topic>
          </jms-destinations>
      </hornetq-server>
  </messaging-deployment>
  ```

  Sauvegardez ce fichier dans le dossier de déploiements et faire un déploiement.

- Ajoutez les entrées dans le fichier de configuration de JBoss EAP 6.

En utilisant *standalone-full.xml* comme exemple, cherchez le sous-système de messagerie dans ce fichier.

```
<subsystem xmlns="urn:jboss:domain:messaging:1.4">
```

Ajoutez y les entrées suivantes, encore une fois, en remplaçant les valeurs de l'exemple par les vôtres. Vous devez ajouter ces entrées après la balise de fin </jms-connection-factories> mais avant l'élément </hornetq-server> :

```
<jms-destinations>
        <jms-queue name="testQueue">
            <entry name="queue/test"/>
            <entry name="java:jboss/exported/jms/queue/test"/>
        </jms-queue>
        <jms-topic name="testTopic">
            <entry name="topic/test"/>
            <entry name="java:jboss/exported/jms/topic/test"/>
        </jms-topic>
    </jms-destinations>
```

9. **Procéder à une configuration supplémentaire**
   Si vous avez besoin de davantage de paramètres de configuration, revoir DTD dans ***EAP_HOME*/docs/schema/jboss-as-messaging_1_4.xsd**.

Rapporter un bogue

20.6.2. Configuration des paramètres de l'adresse JMS

Le sous-système JMS comprend plusieurs options configurables qui gèrent différents aspects de la transmission des messages, le nombre de tentatives d'envoi, et quand le message devra expirer. Ces options de configuration sont contenues dans l'élément de configuration **<address-settings>**.

Une des caractéristiques des configurations d'adresse est la syntaxe commune pour faire correspondre des adresses diverses, connue également sous le nom de Wildcard (caractères génériques).

Syntaxe Wildcard

Les adresses en syntaxe wildcard peuvent être utilisées pour faire correspondre plusieurs adresses similaires avec une seule instruction, ce qui est semblable à la façon dont nombreux systèmes utilisent le caractère astérisque (*****) pour faire correspondre plusieurs fichiers ou chaînes avec une seule recherche. Les caractères suivants ont une signification particulière dans un énoncé wildcard.

Tableau 20.5. Syntaxe Wildcard JMS

Caractère	Description
. (point simple)	Marque l'espace entre les mots au sein d'une expression wildcard.
# (a pound or hash symbol)	Fait correspondre une séquence de zéros ou de plusieurs mots.
* (un astérisque)	Faire correspondre à un mot unique.

Tableau 20.6. Exemples de JMS Wildcards

Exemple	Description
news.europe.#	Correspond à **news.europe**, **news.europe.sport**, **news.europe.politic**, mais pas à **news.usa** ou **europe**.
news.*	Correspond à **news.europe** mais pas à **news.europe.sport**.
news.*.sport	Correspond à **news.europe.sport** et **news.usa.sport**, mais pas à **news.europe.politics**.

Exemple 20.2. Configuration des paramètres d'adresse par défaut

Les valeurs de cet exemples sont utilisées pour illustrer le reste de ce topic.

```
<address-settings>
    <!--default for catch all-->
    <address-setting match="#">
        <dead-letter-address>jms.queue.DLQ</dead-letter-address>
        <expiry-address>jms.queue.ExpiryQueue</expiry-address>
        <redelivery-delay>0</redelivery-delay>
        <max-size-bytes>10485760</max-size-bytes>
        <address-full-policy>BLOCK</address-full-policy>
        <message-counter-history-day-limit>10</message-counter-history-
day-limit>
    </address-setting>
</address-settings>
```

Tableau 20.7. Description de la configuration des paramètres de l'adresse JMS

Élément	Description	Valeur par défaut	Type
address-full-policy	Détermine ce qui se passe quand une adresse dont la max-size-bytes est spécifiée, est remplie.	PAGE	STRING

Élément	Description	Valeur par défaut	Type
`dead-letter-address`	Si une adresse de lettres mortes est spécifiée, les messages seront déplacés vers l'adresse de lettres mortes si les tentatives de livraison **`max-livraison-tentatives`** ont échoué. Dans le cas contraire, ces messages non remis sont ignorés. Les caractères génériques (wildcard) sont autorisés.	jms.queue.DLQ	STRING
`expiry-address`	Si l'adresse d'expiration est présente, les messages expirés seront envoyés à l'adresse ou aux adresses correspondantes, au lieu d'être jetés. Les caractères génériques (wildcards) sont autorisés.	jms.queue.ExpiryQueue	STRING
`last-value-queue`	Définit si une file d'attente utilise uniquement les dernières valeurs ou non.	false	BOOLÉEN
`max-delivery-attempts`	Le nombre max de tentatives d'envoi d'un message avant qu'il soit envoyé à **`dead-letter-address`** ou qu'il soit ignoré.	10	INT
`max-size-bytes`	La taille maximum d'octets.	10485760L	LONG
`message-counter-history-day-limit`	Limite en Jours de l'historique du compteur de messages.	10	INT
`page-max-cache-size`	Le nombre de pages de fichiers à conserver en mémoire pour optimiser IO en cours de navigation de pagination.	5	INT
`page-size-bytes`	La taille de pagination.	5	INT

Élément	Description	Valeur par défaut	Type
`redelivery-delay`	La durée entre les tentatives de re-livraison, exprimée en millisecondes. Si défini sur la valeur **`0`**, les tentatives de re-livraison auront lieu indéfiniment.	0L	LONG
`redistribution-delay`	Définit la durée à attendre lorsque le dernier consommateur est fermé dans une file d'attente avant de pouvoir redistribuer des messages.	-1L	LONG
`send-to-dla-on-no-route`	Un paramètre pour une adresse qui définit la condition d'un message non acheminé vers une file d'attente pour qu'il soit envoyé à la place vers une file d'attente des messages morts ou DLA (de l'anglais Dead Letter Queue) indiquée pour cette adresse.	false	BOOLÉEN

- **Configurer les paramètres de l'adresse et les attributs du modèle**
  Choisir l'interface CLI ou la Console de gestion pour configurer vos attributs de modèle selon les besoins.

  - **Configurer les paramètres de l'adresse par l'interface CLI**
    Utiliser l'interface CLI pour configurer les paramètres de l'adresse.

    a. **Ajouter un nouveau Modèle**
    Utiliser l'opération **`add`** pour créer un nouveau paramètre d'adresse, si nécessaire. Vous pouvez exécuter cette commande à partir de la racine de la session d'interface CLI, qui, dans les exemples suivants, crée un nouveau modèle ou motif intitulé *patternname* avec un attribut **`max-delivery-attempts`** déclaré à *5*. Voici des exemples pour les modifications sur le serveur autonome et le domaine géré pour le profil **`full`**.

    ```
    [standalone@localhost:9999 /] /subsystem=messaging/hornetq-server=default/address-setting=patternname/:add(max-delivery-attempts=5)
    ```

    ```
    [domain@localhost:9999 /] /profile=full/subsystem=messaging/hornetq-server=default/address-setting=patternname/:add(max-delivery-attempts=5)
    ```

    b. **Modifier les attributs de modèle**
    Utiliser l'opération **`write`** pour écrire une nouvelle valeur dans un attribut. Vous pouvez

utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur de **`max-delivery-attempts`** à *10*

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-server=default/address-setting=patternname/:write-attribute(name=max-delivery-attempts,value=10)
```

```
[domain@localhost:9999 /] /profile=full/subsystem=messaging/hornetq-server=default/address-setting=patternname/:write-attribute(name=max-delivery-attempts,value=10)
```

c. **Confirmer les attributs de modèle**
Confirmer que les valeurs ont changé en exécutant **`read-resource`** accompagné du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-server=default/address-setting=patternname/:read-resource
```

```
[domain@localhost:9999 /] /profile=full/subsystem=messaging/hornetq-server=default/address-setting=patternname/:read-resource
```

- **Configurer les paramètres de l'adresse par la console de gestion**
Utiliser la console de gestion pour configurer les paramètres de l'adresse.

a. Connectez-vous à la console de gestion de votre domaine géré ou de votre serveur autonome.

b. Sélectionner l'onglet **`Configuration`** en haut de l'écran. En mode de domaine, sélectionner un profil à partir du menu **`Profile`** en haut et à gauche. Seuls les profils **`full`** et **`full-ha`** ont le sous-système **`messaging`** activé.

c. Étendre le menu **`Messaging`**, et sélectionner **`Destinations`**.

d. Une liste de fournisseurs JMS s'affiche. Dans la configuration par défaut, on ne voit que le fournisseur par **`default`**. Cliquer sur **`View`** pour afficher les paramètres de ce fournisseur en détail.

e. Cliquer sur l'onglet **`Address Settings`**. Ensuite, vous pourrez soit ajouter un nouveau modèle en cliquant sur **`Add`**, ou sélectionner un modèle existant et cliquer sur **`Edit`** pour mettre les paramétres à jour.

f. Si vous ajoutez un modèle, le champ **`Pattern`** s'en référera au paramètre **`match`** de l'élément **`address-setting`**. Vous pourrez aussi modifier **`Dead Letter Address`**, **`Expiry Address`**, **`Redelivery Delay`**, et **`Max Delivery Attempts`**. Les autres options doivent être configurées par l'interface CLI.

Rapporter un bogue

20.6.3. Configurer la messagerie dans HornetQ

La méthode recommandée pour configurer la messagerie dans JBoss EAP 6 est soit la console de gestion, soit par l'interface CLI. Vous pouvez effectuer des modifications persistantes avec l'un ou l'autre de ces outils de gestion sans avoir besoin de modifier manuellement les fichiers de configuration **`standalone.xml`** ou **`domain.xml`**. Cependant, il est utile de se familiariser avec les composants de messagerie des fichiers de configuration par défaut, où les exemples de documentation utilisant des outils de gestion donnent des extraits de fichiers de configuration comme référence.

Rapporter un bogue

20.6.4. Activer la journalisation dans HornetQ

Vous pouvez activer la journalisation pour HornetQ dans EAP 6.x en utilisant une des approches suivantes :

- Éditing des fichiers de configuration du serveur (**`standalone-full.xml`** et **`standalone-full-ha.xml`**) manuellement
- Éditing des fichiers de configuration du serveur par le CLI

Procédure 20.1. Définir la journalisation d'HornetQ en modifiant les fichiers de configuration du serveur manuellement

1. Ouvrir le(s) fichier(s) de configuration du serveur pour l'éditing. Par exemple, **`standalone-full.xml`** ou **`standalone-full-ha.xml`**
2. Naviguez dans la configuration du sous-système de journalisation dans le(s) fichier(s). La configuration par défaut ressemble à ceci :

```
<logger category="com.arjuna">
 <level name="TRACE"/>
</logger>
...
<logger category="org.apache.tomcat.util.modeler">
 <level name="WARN"/>
</logger>
....
```

3. Ajouter la catégorie d'enregistreur d'événement **`org.hornetq`** ainsi que le niveau de journalisation désiré, comme le montre l'exemple suivant :

```
<logger category="com.arjuna">
  <level name="TRACE"/>
</logger>
...
<logger category="org.hornetq">
  <level name="INFO"/>
</logger>
....
```

Résultat

La journalisation HornetQ est active et les messages de journalisation sont traités selon le niveau de journalisation défini.

Définir la journalisation d'HornetQ en modifiant les fichiers de configuration du serveur par l'interface CLI

Vous pouvez également utiliser l'interface CLI pour ajouter la catégorie d'enregistreur d'événement (logger) **`org.hornetq`** ainsi que le niveau de journalisation désiré dans le(s) fichier(s) de configuration du serveur. Pour plus d'informations : Section 14.3.2, « Configurer une Catégorie dans l'interface CLI »

Rapporter un bogue

20.6.5. Configurer HornetQ Core Bridge

Exemple 20.3. Exemple de configuration d'Hornet Core Bridge :

Les valeurs de cet exemples sont utilisées pour illustrer le reste de ce topic.

```
<bridges>
   <bridge name="myBridge">
        <queue-name>jms.queue.InQueue</queue-name>
        <forwarding-address>jms.queue.OutQueue</forwarding-address>
 <ha>true</ha>
        <reconnect-attempts>-1</reconnect-attempts>
        <use-duplicate-detection>true</use-duplicate-detection>
        <static-connectors>
         <connector-ref>
                bridge-connector
                </connector-ref>
        </static-connectors>
        </bridge>
</bridges>
```

Tableau 20.8. Attributs d'HornetQ Core Bridge

Attribut	Description
name	Tous les ponts doivent posséder un nom unique dans le serveur :
queue-name	Ce paramètre obligatoire est le nom unique de la file d'attente locale utilisée par le pont. La file d'attente doit déjà exister au moment où que le pont est instancié au démarrage.
forwarding-address	Il s'agit de l'adresse qui se trouve sur le serveur cible où le message sera envoyé. Si aucune adresse n'est spécifiée, alors, l'adresse d'origine du message sera retenue.
ha	Ce paramètre optionnel déterminera si ce pont doit prendre en charge HA ou non. **`true`** indique qu'il se connectera à tout serveur disponible faisant partie d'un groupement ou qu'il supportera le basculement. La valeur par défaut est false.

Attribut	Description
reconnect-attempts	Ce paramètre optionnel détermine le nombre total de tentatives de reconnexions que le pont doit faire avant d'abandonner et se fermer. Une valeur -1 indique un nombre d'essais illimité. La valeur par défaut correspond à -1.
use-duplicate-detection	Ce paramètre optionnel détermine si un pont doit ou non insérer automatiquement une propriété d'identifiant en duplicata dans chaque message qu'il fait suivre.
static-connectors	static-connectors correspond à une liste d'éléments connector-ref pointant vers des éléments de connecteur définis par ailleurs. Un connecteur encapsule les informations sur le transport à utiliser (TCP, SSL, HTTP etc.) ainsi que les paramètres de connexion de serveur (host, port, etc.).

Rapporter un bogue

20.6.6. Configurer un pontage JMS

HornetQ inclut un pontage de messages JMS qui fonctionne bien. La fonction de ce pontage est de comsommer des messages à partir d'un sujet ou d'une file d'attente source, et de les envoyer vers un sujet ou une file d'attente cible, se trouvant normalement sur un serveur différent.

Les serveurs sources et cibles ne doivent pas forcément être dans le même cluster, ce qui permet d'envoyer des messages d'un cluster à l'autre de façon fiable, via un WAN ou quand il y a une connexion fiable.

On peut déployer un pontage en tant qu'application autonome, avec le serveur autonome HornetQ ou à l'intérieur de l'instance JBoss AS. La source et la cible peuvent se trouver dans la même machine virtuelle ou dans une autre.

Exemple 20.4. Exemple de configuration de pontage JMS :

Les valeurs de cet exemples sont utilisées pour illustrer le reste de ce topic.

```
<subsystem>
  <subsystem xmlns="urn:jboss:domain:messaging:1.3">
             <hornetq-server>
             ...
             </hornetq-server>

             <jms-bridge name="myBridge">
                <source>
                    <connection-factory name="ConnectionFactory"/>
                    <destination name="jms/queue/InQueue"/>
                </source>
                <target>
                    <connection-factory
```

```
name="jms/RemoteConnectionFactory"/>
                    <destination name="jms/queue/OutQueue"/>
                    <context>
                        <property key="java.naming.factory.initial"
value="org.jboss.naming.remote.client.InitialContextFactory"/>
                        <property key="java.naming.provider.url"
value="remote://192.168.40.1:4447"/>
                    </context>
                </target>
                <quality-of-service>AT_MOST_ONCE</quality-of-service>
                <failure-retry-interval>1000</failure-retry-interval>
                <max-retries>-1</max-retries>
                <max-batch-size>10</max-batch-size>
                <max-batch-time>100</max-batch-time>
                <add-messageID-in-header>true</add-messageID-in-header>
            </jms-bridge>
...
</subsystem>
```

Tableau 20.9. Attributs JMS d'HornetQ Core

Attribut	Description
name	Tous les ponts doivent posséder un nom unique dans le serveur :
source connection-factory	Injecte le bean SourceCFF (comme défini dans le fichier de beans). Ce bean crée la ConnectionFactory source.
source destination name	Injecte le bean SourceDestinationFactory (comme défini dans le fichier de beans). Ce bean crée la Destination source.
target connection-factory	Injecte le bean TargetCFF (comme défini dans le fichier de beans). Ce bean crée la ConnectionFactory cible.
target destination name	Injecte le bean TargetDestinationFactory (comme défini dans le fichier de beans). Ce bean crée la Destination cible.
quality-of-service	Ce paramètre représente la qualité de mode de service requise. Les valeurs possibles sont : AT_MOST_ONCE, DUPLICATES_OK, ONCE_AND_ONLY_ONCE
failure-retry-interval	Représente la durée en millisecondes pendant laquelle il faut attendre pour créer à nouveau des connexions vers les serveurs source ou cible quand le pontage a détecté qu'ils ont échoué.

Attribut	Description
max-retries	Représente le nombre de tentatives pour créer à nouveau des connexions vers les serveurs source ou cible quand le pontage a détecté qu'ils ont échoué. Le pontage échouera après un certain nombre de tentatives. -1 représente un nombre d'éssais indéfini.
max-batch-size	Représente le nombre maximum de messages à consommer de la destination source avant de les envoyer en groupe vers une destination cible. Sa valeur doit >= 1.
max-batch-time	Cela représente la durée d'attente maximum en millisecondes avant d'envoyer un lot vers la cible, même si le nombre de messages consommés n'atteint pas la taille MaxBatchSize. Sa valeur doit être -1 pour correspondre à 'wait forever', ou bien >= 1 pour indiquer une durée précise.
add-messageID-in-header	Si défini sur true, alors l'id du message original sera ajouté dans le message envoyé à la destination dans l'en-tête HORNETQ_BRIDGE_MSG_ID_LIST. Si le message est envoyé plus d'une fois, chaque id de message sera ajouté. Cela permettra l'utilisation d'un modèle de requête-réponse distribué. Quand vous recevrez le message, vous pourrez envoyer une réponse par l'id de corrélation de l'id du premier message, ce qui fait que quand l'émetteur d'origine recevra le message, il sera facile de faire une corrélation.

Pour obtenir des informations plus en détail, consultez Section 20.9.2, « Créer un pontage JMS » .

Rapporter un bogue

20.6.7. Configurer la re-livraison différée

Introduction

La re-livraison différée est définie dans l'élément **<redelivery-delay>**, qui est un élément dépendant de l'élément de configuration **<address-setting>** de la configuration du sous-système JMS (Java Messaging Service).

```
<!-- delay redelivery of messages for 5s -->
<address-setting match="jms.queue.exampleQueue">
  <redelivery-delay>5000</redelivery-delay>
</address-setting>
```

Si un retard de livraison est spécifié, le système JMS attendra durant cette valeur de délai avant de re-livrer les messages. Si **<redelivery-delay>** est défini à **0**, il n'y aura pas de livraison à nouveau. Les caractères génériques (wildcards) peuvent être utilisés sur l'attribut***match*** de l'élément **<address-match>** pour configurer la livraison à nouveau des adresses qui correspondent au(x) caractère(s) générique(s).

Rapporter un bogue

20.6.8. Configurer les adresses de lettres mortes

Introduction

Une adresse de lettre morte est définie dans l'élément **<address-setting>** de configuration du sous-système de JMS (Java Messaging Service).

```
<!-- undelivered messages in exampleQueue will be sent to the dead letter
address
deadLetterQueue after 3 unsuccessful delivery attempts
-->
<address-setting match="jms.queue.exampleQueue">
  <dead-letter-address>jms.queue.deadLetterQueue</dead-letter-address>
  <max-delivery-attempts>3</max-delivery-attempts>
</address-setting>
```

Si **<dead-letter-address>** n'est pas spécifié, les messages sont supprimés au bout de **<max-delivery-attempts>** envois. Par défaut, les messages sont envoyés 10 fois. Si vous définissez **<max-delivery-attempts>** à **-1**, vous autorisez un nombre d'envois indéterminé. Ainsi, une lettre morte peut être définie globalement pour un ensemble d'adresses correspondantes et vous pouvez définir **<max-delivery-attempts>** à **-1** pour qu'une adresse particulière soit configurée sur un nombre d'envois indéfini. Les astérisques peuvent aussi être utilisés pour faire correspondre à un ensemble d'adresses particuliéres.

Rapporter un bogue

20.6.9. Configurer les adresses d'expiration de messages

Introduction

Les adresses d'expiration de messages sont définies dans la configuration address-setting de JMS (Java Messaging Service). Ainsi :

```
<!-- expired messages in exampleQueue will be sent to the expiry address
expiryQueue -->
<address-setting match="jms.queue.exampleQueue">
  <expiry-address>jms.queue.expiryQueue</expiry-address>
</address-setting>
```

Si les messages sont expirés et qu'aucune adresse d'expiration n'est spécifiée, les messages sont tout simplement retirés de la file d'attente et abandonnés. *Address wildcards* peut également être utilisé pour configurer des plages de données d'adresses d'expiration spécifiques pour un ensemble d'adresses. Voir Section 20.6.2, « Configuration des paramètres de l'adresse JMS » pour la syntaxe de wildcard JMX et quelques exemples.

Rapporter un bogue

20.6.10. Référence pour les attributs de configuration d'HornetQ

L'implémentation d'HornetQ de JBoss EAP 6 expose les attributs de configuration suivants. Vous pouvez utiliser l'interface CLI pour exposer plus particulièrement les attributs configurables ou affichables par l'opération **read-resource**.

-

Exemple 20.5. Exemple

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default:read-resource
```

Tableau 20.10. Attributs HornetQ

Attribut	Valeur par défaut	Type	Description
allow-failback	true	BOOLÉEN	Il indique si le serveur de sauvegarde se fermera automatiquement si le serveur live d'origine revient.
async-connection-execution-enabled	true	BOOLÉEN	Indique si les paquets entrants sur le serveur doivent être remis à un thread du pool de threads pour le traitement
address-setting			Un paramétrage de l'adresse définit certains attributs avec un caractère générique d'adresse plutôt qu'une file d'attente spécifique
acceptor			Un accepteur définit une façon dont les connections sont acceptées par le serveur HornetQ.
backup-group-name		STRING	Le nom d'un ensemble de live/sauvegardes qui doivent se répliquer ensemble.
backup	false	BOOLÉEN	Indique si ce serveur est un serveur de sauvegarde
check-for-live-server	false	BOOLÉEN	Il indique si un serveur répliqué live doit vérifier le cluster actuel pour voir s'il existe déjà un serveur live avec le même ID de nœud
clustered	false	BOOLÉEN	[Deprécié] Indique si le serveur est clusterisé

Attribut	Valeur par défaut	Type	Description
cluster-password	*CHANGE ME!!*	STRING	Le mot de passe utilisé par les connexions du cluster pour communiquer entere les noeuds clusterisés
cluster-user	HORNETQ.CLUSTER.ADMIN.USER	STRING	L'utilisateur utilisé par les connexions du cluster pour communiquer entre les noeuds clusterisés
cluster-connection			Les connexions du cluster groupe les serveurs en clusters pour que les messages puissent être équilibrés entre les noeuds du clsuter
create-bindings-dir	true	BOOLÉEN	Indique si le serveur doit créer le répertoire de liaisons au démarrage
create-journal-dir	true	BOOLÉEN	Indique si le serveur doit créer le journal de liaisons au démarrage
connection-ttl-override	-1L	LONG	Si défini, cela remplacera la longueur (en ms) qu'il faut pour conserver une connexion vivante sans recevoir de ping
connection-factory			Définit une usine de connexions
connector			Un connecteur peut être utilisé par un client pour définir comment il doit se connecter à un serveur
connector-service			
divert			Une ressource de messagerie qui vous permet de détourner en toute transparence les messages routés vers une adresse ou à une autre adresse, sans apporter aucune modification à toute logique d'application client

Attribut	Valeur par défaut	Type	Description
discovery-group			Groupe multidiffusion à écouter pour recevoir les informations de la part des autres serveurs lorsqu'ils annonceront leurs connecteurs.
failback-delay	5000	LONG	Le temps, en millisecondes, qu'il faut attendre avant qu'un failback (restauration automatique) puisse avoir lieu en cas de redémarrage d'une serveur live
failover-on-shutdown	false	BOOLÉEN	Indique si ce serveur de sauvegarde (s'il s'agit d'un seveur de sauvegarde) doit devenir live lors d'un arrêt normal du serveur.
grouping-handler			Prend des décisions sur quels noeuds d'un cluster doit gérer un message lié à une id de groupe
id-cache-size	20000	INT	La taille du cache pour pré-créer les ID de message
in-vm-acceptor			Définit la façon dont les connexions in-VM peuvent être faîtes au serveur HornetQ
in-vm-connector			Utilisé par un client in-VM pour définir comment il doit se connecter à un serveur
jmx-domain	org.hornetq	STRING	Le domaine JMX utilisé pour enregistrer des MBeans HornetQ dans le MBeanServer
jmx-management-enabled	false	BOOLÉEN	Indique si HornetQ doit exposer sa gestion interne API via JMX. Cela n'est pas recommandé, car accéder à ces MBeans peut conduire à une configuration incohérente
journal-buffer-size	501760 (490KiB)	LONG	La taille du tampon interne sur la journal

Attribut	Valeur par défaut	Type	Description
journal-buffer-timeout	500000 (0.5 milliseconds) for ASYNCIO journal and 3333333 (3.33 milliseconds) for NIO journal	LONG	Le timeout (en nanoseconds) utilisé pour vider des tampons intrnes dans le journal
journal-compact-min-files	10	INT	Le nombre minimum de fichiers de données de journaux avant que nous démarrions le compactage
journal-compact-percentage	30	INT	Le pourcentage de données live sur lequel on considère compacter le journal
journal-file-size	10485760	LONG	La taille (en octets) de chaque fichier de journal
journal-max-io	1	INT	Le nombre maximal de requêtes d'écriture que vous pourrez avoir dans la file d'attente de l'AIO à chaque instant. La valeur par défaut passe à 500 lorsque le journal ASYNCIO est utilisé
journal-min-files	2	INT	Le nombre de fichiers de journaux à pré-créer
journal-sync-non-transactional	true	BOOLÉEN	Indique si on doit attendre que les données non transactionnelles soient synchronisées au journal avant de renvoyer une réponse au client
journal-sync-transactional	true	BOOLÉEN	Indique si on doit attendre que les données de transaction soient synchronisées au journal avant de renvoyer une réponse au client
journal-type	ASYNCIO	String	Le type de journal à utiliser. Cet attribut peut prendre les valeurs "ASYNCIO" ou "NIO"
jms-topic			Définit un sujet JMS

Attribut	Valeur par défaut	Type	Description
live-connector-ref	référence	STRING	[Déprécié] Le nom du connecteur utilisé pour se connecter au connecteur live. Si ce serveur n'est pas une sauvegarde qui utilise «shared nothing HA», sa valeur est «undefined» (indéfinie)
log-journal-write-rate	false	BOOLÉEN	Indique si on doit consigner périodiquement le taux d'écriture au journal et le taux de vidage
mask-password	true	BOOLÉEN	
management-address	jms.queue.hornetq.management	STRING	Adresse à laquelle envoyer des messages de gestion
management-notification-address	hornetq.notifications	STRING	Le nom de l'adresse que les consommateurs utilisent pour recevoir des notifications de la part du management
max-saved-replicated-journal-size	2	INT	Le nombre maximum de journaux de sauvegarde à conserver suite à une restaurauration automatique.
memory-measure-interval	-1	LONG	Fréquence d'échantillonage de mémoire JVM en ms (ou -1 pour désactiver l'échantillonage de mémoire)
memory-warning-threshold	25	INT	Pourcentage de mémoire disponible qui si dépassée résulte en journmalisation d'avertissement
message-counter-enabled	false	BOOLÉEN	Indique si le compteur de messages est activé
message-counter-max-day-history	10	INT	Le nombre de jours qu'il faut conserver l'historique du compteur de messages

Attribut	Valeur par défaut	Type	Description
message-counter-sample-period	10000	LONG	La période d'échantillonage (en ms) à utiliser pour les compteurs de messages
message-expiry-scan-period	30000	LONG	La fréquence (en ms) de balayage des messages expirés
message-expiry-thread-priority	3	INT	La priorité du thread de messages expirés
page-max-concurrent-io	5	INT	Le nombre maximal de lectures simultanées autorisées sur la pagination
perf-blast-pages	-1	INT	
persist-delivery-count-before-delivery	false	BOOLÉEN	Si le nombre de livraison est rendu persistant avant la livraison. False signifie que cela se produit uniquement après qu'un message ait été annulé
persist-id-cache	true	BOOLÉEN	Indique si les ID sont persistés dans le journal
persistence-enabled	true	BOOLÉEN	Indique si le serveur utilisera le journal basé fichier pour la persistance
pooled-connection-factory			Définit une usine de connexions gérées
remoting-interceptors	Non défini	LIST	[Deprecated] La liste de classes d'intercepteur utilisée par ce serveur
remoting-incoming-interceptors	Non défini	LIST	La liste de classes d'intercepteurs entrants utilisée par ce serveur

Attribut	Valeur par défaut	Type	Description
remoting-outgoing-interceptors	Non défini	LIST	La liste de classes d'intercepteurs sortants utilisée par ce serveur
run-sync-speed-test	false	BOOLÉEN	Indique si l'on doit procéder à un diagnostique de la vitesse de sync de votre disque au démarrage. Utile quand vous déterminez les problèmes de performance.
replication-clustername		STRING	Le nom de la connexion de cluster à répliquer si plus d'une connexion de serveur est configurée
runtime-queue			Une file d'attente de runtime
remote-connector			Utilisé par un client distant pour définir comment il doit se connecter à un serveur
remote-acceptor			Définit la façon dont les connexions distantes peuvent être faîtes au serveur HornetQ
scheduled-thread-pool-max-size	5	INT	Le nombre de threads contenu dans le thread pool principal programmé
security-domain	autre	STRING	Le domaine de sécurité à utiliser pour pouvoir vérifier les informations rôle et utilisateur
security-enabled	true	BOOLÉEN	Indique si la sécurité est activée
security-setting			Un paramètre de sécurité permet à un groupe de permissions d'être définies en fonction de files d'attentes sur la base de leur adresse.

Attribut	Valeur par défaut	Type	Description
security-invalidation-interval	10000	LONG	Le temps, en millisecondes, qu'il faut attendre avant d'invalider la cache de sécurité
server-dump-interval	-1	LONG	La fréquence de vidage d'information de runtime de base dans le journal du serveur. Une valeur inférieure à 1 désactive cette fonction
shared store	true	BOOLÉEN	Indique si le serveur utilise ou non un store partagé en cas de basculement
thread-pool-max-size	30	INT	Le nombre de threads contenu dans le thread pool. -1 signifie qu'il n'y a pas de limite
transaction-timeout	300000	LONG	Le temps, en millisecondes (ms), qu'il faut attendre avant qu'une transaction puisse être retirée du gestionnaire de ressources après sa création
transaction-timeout-scan-period	1000	LONG	La fréquence (en ms) de balayage des transactions de timeout
wild-card-routing-enabled	true	BOOLÉEN	Indique si le serveur prend en charge le le routage de wild-card

AVERTISSEMENT

La valeur de **journal-file-size** doit être plus élevée que celle de la taille du message envoyé au serveur, ou bien le serveur ne pourra pas stocker le message.

Rapporter un bogue

20.6.11. Définir l'expiration des messages

Introduction

Les messages envoyés peuvent être définis de façon à pouvoir expirer sur le serveur, s'ils ne sont pas livrées au consommateur après certain lapse de temps (en millisecondes). À l'aide de Java Messaging Service (JMS) ou de l'API de base de HornetQ, le délai d'expiration est réglable sur le message directement. Par exemple :

```
// message will expire in 5000ms from now
message.setExpiration(System.currentTimeMillis() + 5000);
```

JMS **`MessageProducer`** inclut un paramètre **`TimeToLive`** qui contrôle l'expiration de message du message qu'il envoie :

```
// messages sent by this producer will be retained for 5s (5000ms) before
expiration
producer.setTimeToLive(5000);
```

Messages expirés qui sont consommés à partir d'une adresse d'expiration ont les propriétés suivantes :

- _HQ_ORIG_ADDRESS

Une propriété de string qui contient l'adresse d'origine du message expiré.

- _HQ_ACTUAL_EXPIRY

Une propriété longue qui contient l'heure d'expiration du message expiré.

Configurer les adresses d'expiration

Les adresses d'expiration de messages sont définies dans la configuration address-setting :

```
<!-- expired messages in exampleQueue will be sent to the expiry address
expiryQueue -->
<address-setting match="jms.queue.exampleQueue">
   <expiry-address>jms.queue.expiryQueue</expiry-address>
</address-setting>
```

Si les messages sont expirés et qu'aucune adresse d'expiration n'est spécifiée, les messages sont tout simplement retirés de la file d'attente et abandonnés.

Configurer Expiration Thread Reaper

Un thread reaper inspecte périodiquement les files d'attente pour valider si les messages ont expiré.

- message-expiry-scan-period

La fréquence de balayage des files d'attente pour détecter les messages expirés (en millisecondes, valeur par défaut 30000ms, définie à -1 pour désactiver le thread reaper).

- message-expiry-thread-priority

Priorité de thread reaper. Doit se situer en entre 0 et 9, 9 étant la plus haute priorité, et la valeur par défaut étant 3.

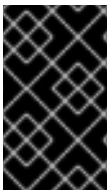
Rapporter un bogue

20.7. GROUPEMENT DES MESSAGES

20.7.1. Groupement des messages

Un groupement de messages est un ensemble de messages qui partagent certaines caractéristiques :

- Tous les messages d'un groupe de messages sont groupés sous un id de groupe commun. Cela signifie qu'ils peuvent être identifiés par une propriété de groupe en commun.
- Tous les messages dans un groupe de messages sont traités en série et consommés par le même consommateur quel que soit le nombre de clients présents dans la file d'attente. Cela signifie qu'un groupe de messages spécifique avec un id de groupe unique est toujours traité par un consommateur lorsque le consommateur l'ouvre. Si le consommateur ferme le groupe de messages, le groupe de messages entier sera redirigé vers un autre consommateur dans la file d'attente.



IMPORTANT

Les groupes de messages sont particulièrement utiles lorsque des messages possédant une certaine valeur de la propriété (id de groupe) doivent être traités en série par un même consommateur.

Rapporter un bogue

20.7.2. Avec Hornet Core API côté client

La propriété **`_HQ_GROUP_ID`** est utilisée pour identifier un groupe de messages dans l'API d'HornetQ côté client. Pour choisir un id de groupe unique de groupe de message, vous pouvez également définir la propriété **`autogroup`** sur "true" dans la SessionFactory.

Rapporter un bogue

20.7.3. Configurer le serveur pour les clients JMS (Java Messaging Service)

La propriété **`JMSXGroupID`** est utilisée pour identifier un groupe de messages pour les clients JMS (Java Messaging Service). Si vous souhaitez envoyer un groupe de messages avec des messages différents vers un consommateur, vous pouvez définir le même **`JMSXGroupID`** pour des messages différents :

```
Message message = ...
message.setStringProperty("JMSXGroupID", "Group-0");
producer.send(message);

message = ...
message.setStringProperty("JMSXGroupID", "Group-0");
producer.send(message);
```

La deuxième approche consiste à définir la propriété **`autogroup`** sur « true » sur le HornetQConnectonFactory. Le HornetQConnectionFactory reprendra ensuite un id de groupe de message unique aléatoire. Vous pouvez définir la propriété **`autogroup`** dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) comme suit :

```
<connection-factory name="ConnectionFactory">
```

```
   <connectors>
      <connector-ref connector-name="netty-connector"/>
   </connectors>
   <entries>
      <entry name="ConnectionFactory"/>
   </entries>
   <autogroup>true</autogroup>
</connection-factory>
```

Une alternative aux deux approches ci-dessus consiste à définir un id de groupe de messages spécifique par le biais de la fabrique de connexions. Cela définiera la propriété **`JMSXGroupID`** à la valeur spécifiée pour tous les messages envoyés par le biais de cette fabrique de connexions. Pour définir un id de groupe de messages spécifique sur la fabrique de connexion, modifiez la propriété **`groupe-id`** dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) comme suit :

```
<connection-factory name="ConnectionFactory">
   <connectors>
      <connector-ref connector-name="netty-connector"/>
   </connectors>
   <entries>
      <entry name="ConnectionFactory"/>
   </entries>
   <group-id>Group-0</group-id>
</connection-factory>
```

Rapporter un bogue

20.7.4. Groupement clusterisés

Un regroupement clusterisé suit une approche différente par rapport au groupement de messages normal. Dans un cluster, les groupes de messages avec des id de groupe spécifiques peuvent arriver sur n'importe quel nœud. Il est important pour un nœud de déterminer quels identifiants de groupes sont liés aux consommateurs et sur quel noeud. Chaque nœud est responsable de diriger les groupes de messages correctement vers le nœud qui possède le consommateur qui traite ces ID de groupes quel que soit l'endroit où les groupes de messages arrivent par défaut.

Cette situation a été corrigée par un handler de groupement. Chaque nœud possède un handler de groupement et ce gestionnaire de groupement (avec d'autres handlers) est responsable de l'acheminement des groupes de messages vers le nœud qui convient. Il existe deux types de regroupement des handlers, à savoir **`local`** (local) et **`remote`** (distant).

Le handler local est chargé de décider du chemin qu'un groupe de messages doit prendre. Les handlers distants communiquent avec le handler local et fonctionnent ainsi. Chaque cluster doit choisir un nœud spécifique pour avoir un handler de groupement local et tous les autres nœuds doivent avoir des handlers distants.

Vous pouvez configurer les handlers de groupements "local" et "remote" dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) comme suit :

```
<grouping-handler name="my-grouping-handler">
   <type>LOCAL</type>
   <address>jms</address>
   <timeout>5000</timeout>
</grouping-handler>
```

```
<grouping-handler name="my-grouping-handler">
   <type>REMOTE</type>
   <address>jms</address>
   <timeout>5000</timeout>
</grouping-handler>
```

L'attribut « timeout » veille à ce qu'une décision de routage soit faite rapidement dans un délai imparti. Si une décision n'est pas faite dans ce délai, une exception sera levée.

Le nœud qui commence par recevoir un groupe de messages prend la décision de routage selon les conditions de routage de cluster ordinaire (disponibilité de la file d'attente de round robin). Le nœud propose cette décision au handler de groupement respectif, qui achemine ensuite les messages vers la file d'attente proposée, s'il accepte la proposition.

Si le handler de groupement rejette la proposition, il proposera un autre itinéraire et le routage aura lieu en conséquence. Les autres nœuds suivront et renverront les groupes de messages dans la file d'attente choisie. Après l'arrivée d'un message dans une file d'attente, ce message sera alloué à un client sur cette file d'attente.

Rapporter un bogue

20.7.5. Meilleures pratiques avec les groupements clusterisés

Voici quelques unes des meilleures pratiques avec les groupements clusterisés :

- Si vous créez et fermez des consommateurs régulièrement, assurez-vous que vos clients soient répartis uniformément sur les différents nœuds. Une fois qu'une file d'attente est épinglée, les messages seront transférés automatiquement vers cette file d'attente indépendamment du fait que l'on puisse supprimer des visiteurs dessus.
- Si vous souhaitez supprimer une file d'attente qui dispose d'un groupe de messages qui lui soit liée, assurez-vous que la file d'attente soit supprimée par la session qui envoie les messages. Cela fera en sorte que les autres nœuds ne tenteront pas de router les messages dans cette file d'attente suite à la suppression.
- Comme un mécanisme de failover réplique toujours le noeud qui a le handler de groupement local

Rapporter un bogue

20.8. LA DÉTECTION DE MESSAGES DUPLIQUÉS

20.8.1. Détection de messages dupliqués

La détection de messages dupliqués permet le filtrage des messages dupliqués sans besoin de codage de la logique de détection des doublons au sein de l'application. Vous pouvez configurer la détection de messages dupliqués dans HornetQ.

Lorsqu'un émetteur(client/server) envoie un message vers un autre serveur, on peut assister à une une situation où le serveur(receveur) de la cible ou la connexion échoue suite à l'envoi du message, mais avant l'envoi d'une réponse à l'expéditeur indiquant que le processus a réussi. Dans de telles situations, il est très difficile pour l'émetteur(client) de déterminer si le message a été envoyé avec succès au destinataire prévu.

L'envoi de message peut ou peut ne pas réussir si le récepteur cible ou la connexion échouent (avant ou après l'envoi du message). Si l'expéditeur (client/serveur) décide de renvoyer le dernier message, cela peut provoquer un message envoyé deux fois à la même adresse.

HornetQ fournit une détection des messages doubles envoyés aux adresses.

Rapporter un bogue

20.8.2. Utiliser la détection des messages en double pour l'envoi des messages

Pour activer la détection des messages en double des messages envoyés, vous devrez définir une propriété spéciale sur le message à une valeur unique. Vous pourrez créer la valeur comme vous le souhaitez, mais cette valeur doit être unique.

Lorsque le serveur cible reçoit ce message, il vérifie si la propriété spéciale est définie. Si la propriété est définie, alors le serveur cible vérifiera son cache de mémoire pour vérifier la présence d'un message reçu avec cette valeur d'en-tête. Si le serveur détecte un message de cette valeur d'en-tête, il ignorera le message envoyé par un client.

Si vous envoyez des messages dans une transaction, alors vous n'aurez pas à définir la propriété pour chaque message que vous envoyez dans cette transaction ; Il suffira de le mettre une fois pour toute dans la transaction. Si le serveur détecte un message dupliqué pour chaque message de la transaction, il ignorera l'ensemble de la transaction.

Le nom de la propriété que vous avez définie est donné par la valeur du ***`org.hornetq.api.core.HDR_DUPLICATE_DETECTION_ID`***, qui est ***`_HQ_DUPL_ID`***. La valeur de cette propriété peut être de type **`byte[]`** ou **`SimpleString`** pour l'API core. Pour les clients JMS (Java Messaging Service), il doit être de type **`String`** avec une valeur unique. Une simple façon de créer un id unique consiste à créer un UUID.

L'exemple suivant vous montre comment définir la propriété pour l'API core :

```
...

ClientMessage message = session.createMessage(true);

SimpleString myUniqueID = "This is my unique id";   // Can use a UUID for
this

message.setStringProperty(HDR_DUPLICATE_DETECTION_ID, myUniqueID);

...
```

L'exemple suivant vous montre comment définir la propriété pour les clients JMS :

```
...

Message jmsMessage = session.createMessage();

String myUniqueID = "This is my unique id";   // Could use a UUID for this

message.setStringProperty(HDR_DUPLICATE_DETECTION_ID.toString(),
myUniqueID);

...
```

Rapporter un bogue

20.8.3. Configurer un cache d'ID dupliqué

Le serveur maintient les caches des valeurs reçues de la propriété ***org.hornetq.core.message.impl.HDR_DUPLICATE_DETECTION_ID*** envoyée à chaque adresse. Chaque adresse maintient son propre cache d'adresses.

Le cache est déterminé par rapport à sa taille. La taille maximum du cache est configurée par le paramètre ***id-cache-size*** dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**). La valeur par défaut de ce paramètre est de 2000 éléments. Si le cache a une taille maximum de n éléments, alors le (n + 1)ième ID stocké remplacera le 0ème élément du cache.

Les caches peuvent également être configurés pour persister dans un disque ou non. On doit pour cela configurer le paramètre ***persist-id-cache*** dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**). Si la valeur est sur "true" alors chaque ID sera persisté en stockage permanent au fur et à mesure qu'il sera reçu. La valeur par défaut de ce paramètre est true.

NOTE

Définir la taille du cache de l'ID dupliqué à un nombre élevé pour pouvoir veiller à ce que le renvoi à nouveau des messages n'écrase pas les messages déjà envoyés qui sont stockés dans le cache.

Rapporter un bogue

20.8.4. Utilisation de la détection dupliquée avec Bridges et les connexions de cluster

Les core bridges peuvent être configurés pour ajouter automatiquement une valeur d'ID unique en double (si celle-ci n'existe pas déjà dans le message) avant de transférer le message à la cible. Pour configurer un core bridge pour la détection de duplication des messages, définir la propriété ***use-duplicate-detection*** sur "true" dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**). La valeur par défaut de ce paramètre est "true".

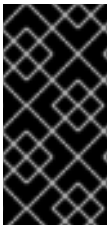
Les connexions de cluster utilisent des core bridges en interne pour déplacer les messages entre les noeuds du cluster. Pour configurer une connexion de cluster à la détection de duplication de messages, définir la propriété ***use-duplicate-detection*** sur "true" dans les fichiers de configuration du serveur (**standalone.xml** et **domain.xml**). La valeur par défaut de ce paramètre est "true".

Rapporter un bogue

20.9. PONTAGES JMS

20.9.1. Les ponts

La fonction des ponts est de consommer des messages à partir d'une file d'attente de source, et de les envoyer vers une adresse cible, qui se trouve normalement sur un serveur HornetQ. Les ponts s'accommodent de connections non fiables, et se reconnectent automatiquement quand les connexions sont rendues disponibles à nouveau. Les ponts HornetQ peuvent être configurés avec des expressions de filtre pour n'envoyer que certains messages.



IMPORTANT

Un pont JMS ne peut être déployé vers serveur EAP 6, qui inclut HornetQ configuré comme une sauvegarde dédiée. La raison est que Transaction Manager sur un serveur de sauvegarde dédié est incapable de réparer les opérations précédemment démarrées sur le serveur HornetQ.

Rapporter un bogue

20.9.2. Créer un pontage JMS

Résumé

Un pontage JMS consomme des messages d'une file d'attente ou une topic JMS source et les envoie à une file d'attente JMS de cible ou sujet, se trouvant en général sur un autre serveur. Il peut être utilisé pour faire un pontage entre les messages entre les serveurs JMS, tant qu'ils sont compatibles avec JMS 1.1. Les ressources JMS de source et de destination sont cherchées à l'aide de JNDI et les classes de client doivent être regroupées dans un module pour la recherche JNDI. Le nom du module est ensuite déclaré dans la configuration de pontage JMS.

Procédure 20.2. Créer un pontage JMS

Cette procédure montre comment configurer un pontage JMS pour faire migrer des messages d'un serveur JBoss EAP 5.x vers un serveur JBoss EAP 6.

1. **Configurer le pontage sur le serveur de messagerie JMS source**
   Configurer le pontage JMS sur le serveur source grâce aux instructions fournies par ce type de serveur. Pour trouver un exemple sur la façon de configurer un pontage JMS sur un serveur JBoss EAP 5.x, voir la rubrique intitulée *Create a JMS Bridge* dans le *Migration Guide* de JBoss EAP 6.

2. **Configurer un pontage déployé dans de serveur JBoss EAP 6.x de destination**
   Dans JBoss EAP 6.1 et dans les versions supérieures, le pontage JMS peut servir à combler des messages depuis n'importe quel serveur compatible JMS 1.1. Comme les ressources JMS sources et cibles sont recherchées par JNDI, les classes de recherche JNDI du fournisseur de messagerie source ou fournisseur de messages doivent être regroupées dans un module de JBoss. Les étapes suivantes utilisent le fournisseur de messages fictive « MyCustomMQ » à titre d'exemple.

   a. Créer le module JBoss pour le fournisseur de messagerie.

      i. Créer une structure de répertoire sous ***EAP_HOME*/modules/system/layers/base/** pour le nouveau module. Le sous-répertoire **main/** contiendra les JAR du client et le fichier **module.xml**. L'exemple suivant est un exemple de structure de répertoires créé pour le fournisseur de messageries MyCustomMQ : ***EAP_HOME*/modules/system/layers/base/org/mycustommq/main/**

      ii. Dans le sous-répertoire **main/**, créer un fichier **module.xml** qui contienne la définition de module suivante pour le fournisseur de messagerie. Ce qui suit est un exemple de **module.xml** créé pour le fournisseur de messagerie MyCustomMQ.

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.mycustommq">
    <properties>
        <property name="jboss.api" value="private"/>
```

```
    </properties>

    <resources>
        <!-- Insert resources required to connect to the
source or target   -->
        <resource-root path="mycustommq-1.2.3.jar" />
        <resource-root path="mylogapi-0.0.1.jar" />
    </resources>

    <dependencies>
       <!-- Add the dependencies required by JMS Bridge code
-->
       <module name="javax.api" />
       <module name="javax.jms.api" />
       <module name="javax.transaction.api"/>
       <!-- Add a dependency on the org.hornetq module since
we send         -->
       <!-- messages tothe HornetQ server embedded in the
local EAP instance -->
       <module name="org.hornetq" />
    </dependencies>
</module>
```

iii. Copier les JAR de fournisseur de messagerie requises pour la recherche JNDI des ressources source vers le sous-répertoire **`main/`** du module. La structure du répertoire du module MyCustomMQ ne devra pas ressembler à ce qui suit.

```
modules/
`-- system
    `-- layers
        `-- base
            `-- org
                  `-- mycustommq
                      `-- main
                          |-- mycustommq-1.2.3.jar
                          |-- mylogapi-0.0.1.jar
                          |-- module.xml
```

b. Configurer le pontage JMS dans le sous-système de **`messaging`** du serveur JBoss EAP.

i. Avant de commencer, arrêtez le serveur et sauvegardez les fichiers de configuration du serveur actuel. Si vous exécutez un serveur autonome, il s'agira du fichier ***`EAP_HOME`*** **`/standalone/configuration/standalone-full-ha.xml`**. Si vous exécutez un domaine géré, sauvegardez les fichiers ***`EAP_HOME`*** **`/domain/configuration/host.xml`** et ***`EAP_HOME`*** **`/domain/configuration/domain.xml`**.

ii. Ajouter l'élément **`jms-bridge`** au sous-système **`messaging`** dans le fichier de configuration du serveur. Les éléments **`source`** et **`target`** procurent les noms des ressources JMS utilisées pour les recherches JNDI. Si les informations d'authentification **`user`** et **`password`** sont spécifiées, elles seront passées comme arguments quand une connexion JMS est créée.

Ce qui suit est un exemple d'élément **`jms-bridge`** configuré pour le fournisseur de messagerie MyCustomMQ :

```
<subsystem xmlns="urn:jboss:domain:messaging:1.3">
   ...
   <jms-bridge name="myBridge" module="org.mycustommq">
      <source>
         <connection-factory name="ConnectionFactory"/>
         <destination name="sourceQ"/>
         <user>user1</user>
         <password>pwd1</password>
         <context>
            <property key="java.naming.factory.initial"
value="org.mycustommq.jndi.MyCustomMQInitialContextFactory"/>
            <property key="java.naming.provider.url"
value="tcp://127.0.0.1:9292"/>
         </context>
      </source>
      <target>
         <connection-factory name="java:/ConnectionFactory"/>
         <destination name="/jms/targetQ"/>
      </target>
      <quality-of-service>DUPLICATES_OK</quality-of-service>
      <failure-retry-interval>500</failure-retry-interval>
      <max-retries>1</max-retries>
      <max-batch-size>500</max-batch-size>
      <max-batch-time>500</max-batch-time>
      <add-messageID-in-header>true</add-messageID-in-header>
   </jms-bridge>
</subsystem>
```

Dans l'exemple suivant, les propriétés JNDI sont définies dans l'élément **`context`** pour la **`source`**. Si l'élément **`context`** est omis, comme dans l'exemple **`target`** ci-dessus, les ressources JMS seront recherchées dans l'instance locale.

Rapporter un bogue

20.10. PERSISTANCE

20.10.1. Persistance dans HornetQ

HornetQ gère sa propre persistance. Il est livré avec un journal de haute performance, qui est optimisé pour les cas d'utilisation de messagerie spécifique.

Le journal HornetQ journal est en «append» (ajout) uniquement avec une taille de fichier configurable, ce qui améliore les performances en permettant des opérations d'écriture simples. Il se compose d'un ensemble de fichiers sur le disque, qui sont initialement pré-créés à une taille fixe et remplis au fur et à mesure que les opérations de serveur (ajouter un message, supprimer le message, mise à jour de message, etc.) sont effectuées, les enregistrements des opérations sont ajoutés au journal jusqu'à ce que le fichier journal soit plein, moment à partir duquel le fichier journal suivant est utilisé.

Un algorithme de nettoyage de la mémoire détermine si les fichiers journaux peuvent être extraits à nouveau ou réutilisés quand toutes les données auront été supprimées. Un algorythme de compaction supprime les espaces vides des fichiers journaux et compriment les données.

Le journal supporte également les transaction locales et XA.

La majorité du journal est imprimée en Java, mais l'interaction avec le système de fichier est rendue abstrait pour autoriser plusieurs implémentations enfichables. Les deux implémentations livrées avec HornetQ sont :

- *Java New I/O (NIO)*

  Utilise Java NIO pour l'interface avec le système de fichiers. Cela donne une excellence performance et exécute sur n'importe quelle plate-forme avec Java 6 ou un runtime plus récent.

- *Linux Asynchronous IO (AIO)*

  Utilise un encapsuleur de code natif pour parler à la bibliothèque d'e/s asynchrone Linux (AIO). Avec AIO, HornetQ reçoit un message lorsque les données ont été rendues persistantes. Cette commande supprime le besoin de synchronisation explicite. AIO fournira généralement une meilleure performance que Java NIO, mais nécessite le noyau Linux 2.6 ou version ultérieure et le package libaio.

  AIO nécessite également les systèmes de fichiers ext2, ext3, ext4, jfs ou xfs.

Le serveur standard HornetQ utilise les instances de journaux suivants :

- *bindings journal*

  Stocke les données relatives à des liaisons, y compris l'ensemble des files d'attente déployées sur le serveur et leurs attributs. Il stocke également des données comme les compteurs de séquence ID. Le journal de liaisons est toujours un journal NIO, car il a généralement un débit faible en comparaison au journal de messages.

  Les fichiers de ce journal ont comme préfixe hornetq-bindings. Chaque fichier a des extensions de liaison. La taille du fichier est de 1048576 octets, et le fichier se trouve dans le dossier de liaisons.

- *JMS journal*

  Stocke toutes les données liées à JMS, comme les files d'attente JMS, les sujets ou fabriques de connexions et toutes les liaisons JNDI de ces ressources. Toutes les ressources JMS créées avec l'API de gestion sont persistées dans ce journal. Toutes les ressources configurées par des fichiers de configuration ne le sont pas. Ce journal n'est créé que si JMS est utilisé.

- *message journal*

  Stocke toutes les données liées à des messages, y compris les messages eux-mêmes et les duplicate-id caches. Par défaut, HornetQ utilise AIO pour son journal. Si AIO n'est pas disponible, ce sera automatiquement NIO.

Les gros messages sont persistés en dehors du journal de messages. Dans les situations de moindre mémoire, configurer HornetQ pour qu'il envoie les messages sur le disque. Si la persistance n'est pas requise, HornetQ peut être configuré pour ne persister aucune donnée.

Rapporter un bogue

20.11. HORNETQ CLUSTERING

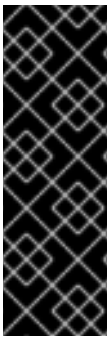
Les clusters HornetQ sont utilisés pour créer des groupes de serveurs HornetQ qui partagent la charge de traitement des messages. Chaque noeud actif du cluster agit comme serveur HornetQ indépendant et gère ses propres messages et connexions.

Pour former un cluster, chaque noeud (serveur indépendant HornetQ) déclare des connexions au cluster par un autre noeud avec des paramètres de configuration dans les fichiers de configuration (**`standalone.xml`** et **`domain.xml`**).

En clustering, les Core Bridges sont utilisés pour le pontage/routage des messages d'un cluster à l'autre. Les ponts principaux consomment des messages à partir d'une file d'attente de la source et ensuite, transfèrent ces messages vers un serveur cible HornetQ (nœud) qui peut ou peut ne pas être dans le même cluster.

Lorsqu'un nœud forme une connexion avec un autre nœud de cluster, il crée un Core Bridge en interne. Chaque nœud crée un Core Bridge explicite et vous n'avez pas besoin de le déclarer. Ces connexions au cluster permettent la transmission des messages entre les nœuds dans divers clusters d'équilibrage de la charge de traitement des messages.

Vous pouvez configurer des noeuds de cluster dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**).



IMPORTANT

Vous pouvez configurer un nœud par le biais de fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) et copier cette configuration aux autres nœuds pour générer un cluster symétrique. Toutefois, vous devez être prudent lorsque vous copiez les fichiers de configuration de serveur. Vous ne devez pas copier les données de HornetQ (c'est-à-dire les liaisons, le journal et les répertoires de messages volumineux) d'un nœud à l'autre. Lorsqu'un nœud est démarré pour la première fois, il persiste un identificateur unique pour le répertoire de journal utile à la bonne formation des clusters.

Rapporter un bogue

20.11.1. Server Discovery

Les serveurs utilisent un mécanisme qui s'appelle "server discovery" pour :

- Transmettre leurs informations de connexion aux clients de messagerie : le but de la messagerie clients est de se connecter aux serveurs d'un cluster sans détails précis sur les serveurs en cours d'exécution à un moment donné
- Se connecter aux autres serveurs : les serveurs d'un cluster veulent établir des connexions au cluster avec d'autres serveurs sans détail précis sur les autres serveurs d'un cluster

L'information sur les serveurs est envoyée à Messaging Clients par les connexions HornetQ habituelles et aux autres services par l'intermédiaire des connexions du cluster.

La première connexion doit être établie et peut être établie par les techniques de découverte de serveur dynamique (discovery) comme UDP (User Datagram Protocol), JGroups ou en fournissant une liste de connecteurs.

Rapporter un bogue

20.11.2. Broadcast Groups

Des connecteurs sont utilisés sur le client pour définir comment et de quelle manière il se connecte au serveur. Les serveurs utilisent des groupes de diffusion (broadcast groups) pour diffuser des connecteurs sur le réseau. Le groupe de diffusion prend un groupe de paires de connecteur et les diffuse

sur le réseau. Chaque paire de connecteurs contient des paramètres de connexion pour un serveur live et de sauvegarde.

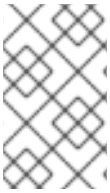
Vous pouvez définir des groupes de diffusion dans l'élément ***broadcast-groups*** des fichiers de configuration du serveur (**standalone.xml** et **domain.xml**). Un seul serveur HornetQ peut posséder plusieurs groupes de diffusion. Vous pouvez définir un UDP (User Datagram Protocol) ou un groupe de diffusion JGroup.

Rapporter un bogue

20.11.2.1. Groupe de diffusion UDP (User Datagram Protocol)

L'exemple ci-dessous montre comment définir un groupe de diffusion UDP :

```
<broadcast-groups>
   <broadcast-group name="my-broadcast-group">
      <local-bind-address>172.16.9.3</local-bind-address>
      <local-bind-port>5432</local-bind-port>
      <group-address>231.7.7.7</group-address>
      <group-port>9876</group-port>
      <broadcast-period>2000</broadcast-period>
      <connector-ref>netty</connector-ref>
  </broadcast-group>
</broadcast-groups>
```



NOTE

Dans l'exemple de configuration ci-dessus, les attributs "local-bind-address", "local-bind-port", "group-address" et "group-port" sont dépréciés. Vous pourrez choisir d'utilliser l'attribut "socket-binding" à la place.

L'exemple indiqué ci-dessous définit un groupe de diffusion UDP qui remplace tous les attributs dépréciés par l'attribut "socket-binding" :

```
<broadcast-groups>
   <broadcast-group name="my-broadcast-group">
      <socket-binding>messaging-group</socket-binding>
      <broadcast-period>2000</broadcast-period>
      <connector-ref>netty</connector-ref>
   </broadcast-group>
</broadcast-groups>
```

Le tableau ci-dessous décrit tous les paramètres importants utilisés dans les exemples ci-dessus et, qui servent, en général, à définir un groupe de diffusion UDP :

Tableau 20.11. Paramètres de groupe de diffusion UDP

Attribut	Description
attibut de nom	Dénote le nom de chaque groupe de diffusion d'un serveur. Chaque groupe de diffusion doit posséder un nom unique.

Attribut	Description
local-bind-address	[Déprécié] C'est un attribut spécifique UDP qui spécifie l'adresse de liaison locale à laquelle se relie le paquet de datagramme. Vous devez définir cette propriété pour définir l'interface que vous souhaitez utiliser pour vos diffusions. Si cette propriété n'est pas spécifiée, alors la liaison se relie à une adresse générique (une adresse générée par le noyau au hasard).
local-bind-port	[Déprécié] C'est un attribut spécifique UDP qui spécifie un port local auquel le socket de datagramme se relie. Une valeur "-1" par défaut indique un port anonyme à utiliser.
group-address	[Déprécié] C'est une adresse multidiffusion spécifique à UDP où les messages sont diffusés. Cette adresse IP possède un large éventail qui va de 224.0.0.0 à 239.255.255.255, inclus. L'adresse IP de 224.0.0.0 est réservée et ne peut pas être utilisée.
group-port	[Déprécié] Dénote le numéro de port UDP de diffusion.
socket-binding	Dénote la liaison de socket de groupe de diffusion
broadcast-period	Ce paramètre indique la durée entre deux diffusion (en millisecondes). Optionnel.
connector-ref	Se réfère au connecteur qui sera diffusé.

Rapporter un bogue

20.11.2.2. Groupe de diffusion JGroups

Vous pouvez utiliser JGroups pour la diffusion en spécifiant deux attributs ***`jgroups-stack`*** et ***`jgroups-channel`***. L'exemple ci-dessous définit un groupe de diffusion JGroups :

```
<broadcast-groups>
  <broadcast-group name="bg-group1">
    <jgroups-stack>udp</jgroups-stack>
    <jgroups-channel>udp</jgroups-channel>
    <broadcast-period>2000</broadcast-period>
    <connector-ref>netty</connector-ref>
  </broadcast-group>
</broadcast-groups>
```

La définition du groupe de diffusion JGroups utilise deux attributs principaux :

- L'attribut ***`jgroups-stack`*** qui indique le nom de la pile définie dans le sous-système **`org.jboss.as.clustering.jgroups`**

- L'attribut ***jgroups-channel*** qui indique le canal auquel les canaux de JGroups se connectent pour la multidiffusion

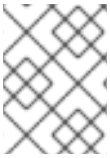
Rapporter un bogue

20.11.3. Les groupes discovery

Les groupes de diffusion sont utilisés pour diffuser les connecteurs sur le réseau. D'autre part, les groupes discovery définissent comment l'information sur les connecteurs est reçue à partir des points de terminaison de diffusion (Groupes de diffusion JGroups ou UDP) .Un groupe discovery maintient une liste de paires de connecteurs, un par diffusion par un serveur différent.

Quand un groupe discovery reçoit des diffusions à partir d'un point de terminaison pour un serveur particulier, il met à jour les paires de connecteur dans la liste du serveur spécifique. S'il ne reçoit pas de diffusion en provenance d'un serveur spécifique pendant un certain temps, il supprime d'entrée du serveur de la liste.

Les groupes discovery sont surtout utilisés par les connexions du cluster et les clients JMS (Java Messaging Service) afin d'obtenir les informations de connexion de départ pour pouvoir télécharger la topologie qui convient.



NOTE

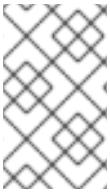
Vous devez configurer chaque groupe discovery avec un point de terminaison de diffusion qui convient correspondant à son homologue de groupe de diffusion (UDP ou JGroups).

Rapporter un bogue

20.11.3.1. Configurer un groupe de diffusion UDP (User Datagram Protocol) sur le serveur

L'exemple ci-dessous montre comment définir un groupe discovery UDP :

```
<discovery-groups>
   <discovery-group name="my-discovery-group">
      <local-bind-address>172.16.9.7</local-bind-address>
      <group-address>231.7.7.7</group-address>
      <group-port>9876</group-port>
      <refresh-timeout>10000</refresh-timeout>
   </discovery-group>
</discovery-groups>
```



NOTE

Dans l'exemple de configuration ci-dessus, les attributs "local-bind-address", "group-address" et "group-port" sont dépréciés. Vous pourrez choisir d'utilliser l'attribut "socket-binding" à la place.

L'exemple indiqué ci-dessous définit un groupe de discovery UDP qui remplace tous les attributs dépréciés par l'attribut "socket-binding" :

```
<discovery-groups>
   <discovery-group name="my-discovery-group">
```

```
        <socket-binding>messaging-group</socket-binding>
        <refresh-timeout>10000</refresh-timeout>
    </discovery-group>
</discovery-groups>
```

Le tableau ci-dessous décrit tous les paramètres importants utilisés dans l'exemple ci-dessus et, qui servent, en général, à définir un groupe de diffusion :

Tableau 20.12. Paramètres de groupe discovery UDP

Attribut	Description
name attribute	Cet attribut indique le nom du groupe discovery. Chaque nom discovery doit avoir un nom unique par serveur.
local-bind-address	[Déprécié] C'est un attribut spécifique UDP optionnel. Il est utilisé pour configurer un groupe discovery pour qu'il écoute une interface spécifique lors de l'utilisation d'interfaces multiples sur la même machine.
group-address	[Déprécié] C'est un attribut spécifique UDP obligatoire. Il est utilisé pour configurer un groupe discovery pour qu'il écoute une adresse IP spécifique de groupe. La valeur de cet attribut doit correspondre à l'attribut ***`group-address`*** du groupe de diffusion que vous souhaitez écouter.
group-port	[Déprécié] C'est un attribut spécifique UDP obligatoire. Il est utilisé pour configurer le port UDP du groupe multidiffusion. La valeur de cet attribut doit correspondre à l'attribut ***`group-port`*** du groupe de diffusion que vous souhaitez écouter.
socket-binding	Dénote la liaison de socket de groupe discovery
refresh-timeout	Il s'agit d'un attribut spécifique UDP facultatif. Il est utilisé pour configurer la durée (en millisecondes) pendant laquelle le groupe discovery patientra avant de retirer l'entrée de paire de connecteurs d'un serveur de la liste après avoir reçu la dernière diffusion en provenance de ce serveur. La valeur de ***`refresh-timeout`*** doit être définie à un niveau nettement supérieur à la valeur de l'attribut ***`broadcast-period`*** sur le groupe de diffusion pour empêcher le déplacement rapide de serveurs de la liste lorsque le processus de diffusion est encore actif. La valeur par défaut de cet attribut est de 10 000 millisecondes.

Rapporter un bogue

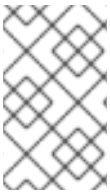
20.11.3.2. Configurer un groupe discovery JGroups sur le serveur

L'exemple ci-dessous montre comment définir un groupe discovery JGroups :

```
<discovery-groups>
  <discovery-group name="dg-group1">
    <jgroups-stack>udp</jgroups-stack>
    <jgroups-channel>udp</jgroups-channel>
    <refresh-timeout>10000</refresh-timeout>
  </discovery-group>
</discovery-groups>
```

La définition du groupe discovery JGroups utilise deux attributs principaux :

- L'attribut ***`jgroups-stack`*** qui indique le nom de la pile définie dans le sous-système **`org.jboss.as.clustering.jgroups`**
- L'attribut ***`jgroups-channel`*** qui indique le canal auquel les canaux de JGroups se connectent pour la multidiffusion



NOTE

Les attributs JGroups et les attributs spécifiques UDP sont exclusifs. Vous pouvez utiliser les groupes d'attributs JGroups ou UDP dans la configuration d'un groupe discovery ou de diffusion

Rapporter un bogue

20.11.3.3. Configurer les groupes discovery pour les clients JMS (Java Messaging Service)

Les groupes discovery peuvent être configurés pour les clients JMS et pour les clients principaux. Vous pouvez spécifier le groupe discovery à utiliser pour une usine de connexions JMS dans les fichiers de configuration de serveur (**`standalone.xml`** et **`domain.xml`**) :

```
<connection-factory name="ConnectionFactory">
 <discovery-group-ref discovery-group-name="my-discovery-group"/>
  <entries>
    <entry name="ConnectionFactory"/>
  </entries>
</connection-factory>
```

L'élément ***`discovery-group-ref`*** est utilisé pour spécifier le nom d'un groupe discovery. Lorsqu'une application client télécharge cette usine de connexions de JNDI (Java Naming and Directory Interface) et crée des connexions JMS, ces connexions sont équilibrées sur tous les serveurs que le groupe discovery préserve en écoutant l'adresse de multidiffusion spécifiée dans la configuration de groupe discovery.

Si vous utilisez JMS et non pas JNDI pour chercher une usine de connexions, alors vous pourrez spécifier les paramètres de groupe de discovery directement quand vous créerez l'usine de connexions JMS :

```
final String groupAddress = "231.7.7.7";
final int groupPort = 9876;
ConnectionFactory jmsConnectionFactory =
HornetQJMSClient.createConnectionFactory(new
DiscoveryGroupConfiguration(groupAddress, groupPort, new
UDPBroadcastGroupConfiguration(groupAddress, groupPort, null, -1)),
```

```
JMSFactoryType.CF);
Connection jmsConnection1 = jmsConnectionFactory.createConnection();
Connection jmsConnection2 = jmsConnectionFactory.createConnection();
```

La valeur par défaut de l'attribut ***refresh-timeout*** peut être définie sur DiscoveryGroupConfiguration en utilisant la méthode setter **setDiscoveryRefreshTimeout()**. Pour que la fabrique de connexions attende un certain temps avant de créer la première connexion, vous pouvez utiliser la méthode setter **setDiscoveryInitialWaitTimeout()** sur DiscoveryGroupConfiguration.

Cela garantit que la fabrique de connexions ait suffisemment de temps pour recevoir les diffusions de tous les nœuds du cluster. La valeur par défaut de ce paramètre est de 10 000 millisecondes.

Rapporter un bogue

20.11.3.4. Configuration de discovery pour l'API principal

Si vous utilisez l'API principal pour instancier les instances de ***ClientSessionFactory***, alors vous pouvez spécifier les paramètres du groupe discovery directement quand vous créez une usine de session :

```
final String groupAddress = "231.7.7.7";
final int groupPort = 9876;
ServerLocator factory = HornetQClient.createServerLocatorWithHA(new
DiscoveryGroupConfiguration(groupAddress, groupPort, new
UDPBroadcastGroupConfiguration(groupAddress, groupPort, null, -1))));
ClientSessionFactory factory = locator.createSessionFactory();
ClientSession session1 = factory.createSession();
ClientSession session2 = factory.createSession();
```

La valeur par défaut de l'attribut ***refresh-timeout*** peut être définie sur DiscoveryGroupConfiguration en utilisant la méthode setter **setDiscoveryRefreshTimeout()**. Vous pouvez utiliser la méthode setter **setDiscoveryInitialWaitTimeout()** sur DiscoveryGroupConfiguration pour que l'usine de sessions patiente pendant un certain temps avant de créer une session.

Rapporter un bogue

20.11.4. Équilibrage des charges côté serveur

Il y a une topologie de clusterisation importante :

- Cluster symétrique : dans un cluster symétrique, chaque noeud de cluster est connecté directement à chaque noeud du cluster. Pour créer un cluster symétrique, chaque noeud du cluster définit une connexion de cluster avec l'attribut ***max-hops*** défini sur 1.

NOTE

Dans un cluster symétrique, chaque nœud connaît toutes les files d'attente qui existent sur tous les autres nœuds et les consommateurs qu'ils ont. Avec ces connaissances, il peut déterminer comment équilibrer la charge et redistribuer les messages vers les nœuds.

Rapporter un bogue

20.11.4.1. Configuration des connexions du cluster

Les connexions du cluster sont configurées dans les fichiers de configuration du serveur (**`standalone.xml`** et **`domain.xml`**) dans l'élément ***`cluster-connection`***. Il peut y avoir zéro ou plusieurs connexions définies par serveur HornetQ.

```
<cluster-connections>
  <cluster-connection name="my-cluster">
    <address>jms</address>
    <connector-ref>netty-connector</connector-ref>
    <check-period>1000</check-period>
    <connection-ttl>5000</connection-ttl>
    <min-large-message-size>50000</min-large-message-size>
    <call-timeout>5000</call-timeout>
    <retry-interval>500</retry-interval>
    <retry-interval-multiplier>1.0</retry-interval-multiplier>
    <max-retry-interval>5000</max-retry-interval>
    <reconnect-attempts>-1</reconnect-attempts>
    <use-duplicate-detection>true</use-duplicate-detection>
    <forward-when-no-consumers>false</forward-when-no-consumers>
    <max-hops>1</max-hops>
    <confirmation-window-size>32000</confirmation-window-size>
    <call-failover-timeout>30000</call-failover-timeout>
    <notification-interval>1000</notification-interval>
    <notification-attempts>2</notification-attempts>
     <discovery-group-ref discovery-group-name="my-discovery-group"/>
   </cluster-connection>
</cluster-connections>
```

Le tableau suivant décrit les attributs configurables :

Tableau 20.13. Attributs configurables de connexions de cluster

Attribut	Description	Par défaut
`address`	Chaque connexion de cluster ne s'applique qu'aux messages envoyés à une adresse qui commence par cette valeur. L'adresse peut être n'importe quelle valeur et vous pouvez avoir plusieurs connexions au cluster avec des valeurs différentes des adresses, équilibrant simultanément les messages de ces adresses, potentiellement à différents clusters de serveurs. Cela n'utilise pas de correspondance de wild-card.	
`connector-ref`	C'est un attribut obligatoire qui fait référence à un connecteur envoyé dans d'autres noeuds du cluster pour qu'ils adoptent la topologie de cluster qui convient.	

Attribut	Description	Par défaut
check-period	Il d'agit d'une durée (en millisecondes) utilisée pour vérifier si une connexion de cluster a manqué des pings en provenance d'un autre cluster.	30 000 millisecondes
connection-ttl	Indique la durée pendant laquelle une connexion de cluster doit rester vivante s'il cesse de recevoir des messages en provenance d'un noeud spécifique du cluster.	60 000 millisecondes
min-large-message-size	Si la taille du message (en octets) est plus élevée que cette valeur, alors il sera divisé en plusieurs segments après avoir été envoyé à d'autres membres du cluster du réseau.	102400 octets
call-timeout	Indique la durée (en millisecondes) pendant laquelle un paquet envoyé sur une connexion de cluster doit attendre (une réponse) avant de lancer une exception.	30 000 millisecondes
retry-interval	Si la connexion de cluster est créée entre les noeuds d'un cluster et que le noeud cible n'a pas été démarré ou doit être redémarré, les connexions de cluster d'autres nœuds retenteront de se reconnecter à la cible jusqu'à ce qu'il revienne. Le paramètre ***retry-interval*** définit l'intervalle (en millisecondes) entre les tentatives.	500 millisecondes
retry-interval-multiplier	Utilisé pour incrémenter ***retry-interval*** après chaque tentative	1
max-retry-interval	Se réfère à la durée maximum (en millisecondes) des nouvelles tentatives	2000 millisecondes

Attribut	Description	Par défaut
reconnect-attempts	Définit le nombre de fois que le système va essayer de connecter un noeud au cluster	-1 (infinite retries)
use-duplicate-detection	Les connexions au cluster utilisent des ponts pour relier les nœuds, et les ponts peuvent être configurés pour ajouter une propriété id en double dans chaque message qui est transmis. Si le nœud cible du pont se bloque et puis récupère, les messages peuvent être renvoyés à partir du nœud de la source. En permettant la détection des doublons, tous les messages en double vont être filtrés et ignorés lors de la réception au niveau du nœud cible.	True
forward-when-no-consumers	Ce paramètre détermine si oui ou non les messages seront distribués de façon alternée entre d'autres nœuds du cluster indépendamment du fait qu'ils puissent correspondre à un consommateur sur d'autres nœuds	False
max-hops	Détermine comment les charges de messages sont équilibrées sur d'autres serveurs HornetQ connectés à ce serveur.	-1
confirmation-window-size	La taille (en octets) de la fenêtre utilisée pour envoyer des confirmations du serveur connecté à	1048576
call-failover-timeout	Utilisé quand un appel est fait lors d'une tentative de basculement	-1 (no timeout)
notification-interval	Détermine la fréquence (en millisecondes) à laquelle la connexion du cluster doit se diffuser quand elle s'attache au cluster.	1000 millisecondes

Attribut	Description	Par défaut
notification-attempts	Définit le nombre de fois que la connexion du cluster doit se diffuser quand elle se connecte au cluster.	2
discovery-group-ref	Ce paramètre détermine quel groupe discovery est utilisé pour obtenir la liste des autres serveurs du cluster vers laquelle la connexion de cluster va faire des connexions	

Lorsque vous créez des connexions entre les noeuds d'un cluster pour former une connexion de cluster, HornetQ utilise un cluster utilisateur et un mot de passe de cluster qui est défini dans les fichiers de configuration de serveur (**`standalone.xml`** et **`domain.xml`**) :

```
<cluster-user>HORNETQ.CLUSTER.ADMIN.USER</cluster-user>
<cluster-password>NEW USER</cluster-password>
```

AVERTISSEMENT

Il est important de changer les valeurs par défaut de ces informations d'identification pour empêcher les clients distants d'effectuer les connexions au serveur en utilisant les valeurs par défaut.

Rapporter un bogue

20.12. HAUTE DISPONIBILITÉ

20.12.1. Introduction à la haute disponibilité

HornetQ supporte la possibilité de continuer à fonctionner après un basculement d'un ou de plusieurs serveurs. Ceci est en partie réalisé grâce au support de basculement lorsque des connexions de clients migrent d'un serveur live vers un serveur de sauvegarde en cas d'un basculement de serveur live. Pour conserver le serveur de sauvegarde actuel, les messages sont répliqués du serveur live vers le serveur de sauvegarde en continu par deux stratégies: store partagé et réplication.

Il existe deux types de topologies en haute disponibilité :

- **`Topologie dédié`** : cette topologie est composée de deux serveurs EAP. Dans le premier serveur, HornetQ est configuré comme un serveur live. Dans le second serveur, HornetQ est configuré comme un serveur de sauvegarde. Le serveur EAP ayant HornetQ configuré comme un serveur de sauvegarde agit seulement en tant que conteneur pour HornetQ. Ce serveur est inactif et ne peut pas héberger des déploiements comme les EJB, les MDB ou des servlets.

- **`Topologie colocalisé`** : cette topologie contient deux serveurs d'EAP. Chaque serveur EAP contient deux serveurs de HornetQ (un serveur live et un serveur de sauvegarde). Le serveur HornetQ sur le premier serveur EAP et le serveur de sauvegarde de HornetQ sur le second serveur EAP forment une paire de serveurs live de sauvegarde, tandis que le serveur live HornetQ sur le second serveur EAP et le serveur de sauvegarde HornetQ sur le premier serveur EAP forment une autre paire de serveurs de sauvegarde live.

En topologie colocalisée, dès qu'un serveur HornetQ (faisant partie de la paire live-sauvegarde) échoue, le serveur de sauvegarde HornetQ prend la relève et redevient actif. Lorsque le serveur de sauvegarde HornetQ s'arrête en cas de restauration automatique (failback), alors les usines de connexions et destinations configurées sur le serveur de sauvegarde sont dissociées de JNDI (Java Naming and Directory Interface).

L'interface JNDI (Java Naming and Directory Interface) est partagée avec l'autre serveur live HornetQ (faisant partie de l'autre paire live-sauvegarde). Ainsi, la dissociation des usines de connexion et destinations de JNDI a pour effet de dissocier également les fabriques de connexions et de destinations de ce serveur live HornetQ.

IMPORTANT

La configuration des serveurs de sauvegarde colocalisés ne peuvent pas contenir de configurations de destination ou d'usines de connexion.

NOTE

Les informations suivantes référencient **`standalone-full-ha.xml`**. Les changements de configuration peuvent s'appliquer à **`standalone-full-ha.xml`** ou à tout fichier de configuration dérivé.

Rapporter un bogue

20.12.2. HornetQ Shared Stores

Lorsque vous utilisez un magasin partagé (Shared Store), les serveurs live et de sauvegarde partagent le répertoire de données même, ensemble, à l'aide d'un système de fichiers partagé. Cela inclut le répertoire de pagination, le répertoire de journaux, des messages volumineux et le journal de liaison. Lorsque le basculement et le serveur de sauvegarde reprennent, il chargent le stockage persistant de système de fichiers partagé. Les clients peuvent alors s'y connecter.

Cette forme de haute disponibilité diffère de la réplication de données, car elle requiert que le système de fichiers soit accessible à la fois par les noeuds de sauvegarde live et de sauvegarde. Cela correspondra le plus souvent à un SAN de haute performance.

L'avantage de share-store haute disponibilité est qu'aucune réplication ne se produit entre les nœuds live et de sauvegarde. Autrement dit, il n'y a pas de dégradation des performances en raison de la surcharge de réplication pendant le fonctionnement normal.

L'inconvénient la réplication shared-store est qu'elle nécessite un système de fichiers partagé, et que lorsque le serveur de sauvegarde est activé, il faut charger le journal à partir d'un shared-store. Cela peut prendre un certain temps, selon la quantité de données dans le store.

S'il est exigé d'avoir des performances élevées durant le fonctionnement normal, il y a accès à un réseau SAN rapide et un taux de basculement légèrement plus lent est acceptable (en fonction de la quantité de données). Shared-store haute disponibilité est recommandé.

Rapporter un bogue

20.12.3. Configurations de stockage d'HornetQ

HornetQ prend en charge deux styles de configuration différents pour les stores partagés :

- GFS2 sur un SAN, pour le type de journal ASYNCIO.
- NFSv4, pour les types de journaux ASYNCIO ou NIO.

IMPORTANT

NFS est pris en charge sous des directives strictes de configuration qui sont soulignées ci-dessous.

L'implémentation NFS de Red Hat Linux utilisée supporte à la fois le direct I/O (ouverture des fichiers avec l'indicateur O_DIRECT défini), et l'I/O asynchrone basé noyau. Avec ces deux fonctionnalités présentes, il est possible d'utiliser NFS comme option de stockage, sous conditions de configuration strictes :

- Le cache client Red Hat Enterprise Linux NFS doit être désactivé.

NOTE

Il est recommandé que, si vous utilisez NFS en vertu des stipulations ci-dessus, une configuration NFS hautement disponible soit utilisée.

Rapporter un bogue

20.12.4. Types de journaux HornetQ

Deux types de journaux sont disponibles pour HornetQ :

- ASYNCIO
- NIO

Le type de journal ASYNCIO, également connu sous le nom de AIO, est un wrapper de code natif mince autour de la bibliothèque Linux d'e/s asynchrones (AIO). L'utilisation de fonctionnalités natives peut fournir de meilleures performances que NIO. Ce type de journal n'est supporté que sur Red Hat Enterprise Linux et exige que **`libaio`** et le package de composants natifs soient installés où JBoss EAP 6 est en cours d'exécution. Consultez le *Guide d'Installation* pour obtenir des instructions sur l'installation du package de composants natifs.

IMPORTANT

Consultez le journal de serveur après le démarrage de JBoss EAP 6, pour vous assurer que la bibliothèque native est chargée avec succès, et que le type de journal ASYNCIO est utilisé. Si la bibliothèque native ne se charge pas, HornetQ retournera à un type de journal NIO, et cela sera précisé dans le journal du serveur.

Le type de journal NIO utilise Java NIO pour créer l'interface avec le système de fichiers. Il fournit une très bonne performance et exécute sur toutes les plateformes prises en charge.

Pour spécifier le type de journal HornetQ, définir le paramètre **<journal-type>** dans le sous-système de **Messaging**.

Rapporter un bogue

20.12.5. Configurer HornetQ avec une topologie dédiée et un store partagé

Pour configurer les serveurs live ou de backup des stores partagés dans la topologie dédiée, configurer les fichiers **standalone-X.xml** sur chaque serveur avec ce qui suit :

```
<shared-store>true</shared-store>
<paging-directory path="${shared.directory}/journal"/>
<bindings-directory path="${shared.directory}/bindings"/>
<journal-directory path="${shared.directory}/journal"/>
<large-messages-directory path="${shared.directory}/large-messages"/>
.
.
.
<cluster-connections>
   <cluster-connection name="my-cluster">
      ...
   </cluster-connection>
</cluster-connections>
```

Tableau 20.14. Les attributs de configuration des serveurs HornetQ (pour serveurs live et de backup à la fois)

Attribut	Description
shared-store	Indique si le serveur utilise un store partagé ou non. La valeur par défaut est false.
paging-directory path	Indique le chemin d'accès vers le répertoire de pagination. Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire
bindings-directory path	Indique le chemin d'accès vers le journal des liaisons Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce journal.
journal-directory path	Indique le chemin d'accès vers le répertoire de journalisation. Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire
large-messages-directory path	Indique le chemin d'accès vers le répertoire de messages volumineux Ce chemin est le même pour les serveurs live ou de backup car ils partagent ce répertoire
failover-on-shutdown	Indique si ce serveur devient actif quand un serveur de sauvegarde actuellement actif se ferme

Le serveur de sauvegarde doit également être marqué explicitement en tant que serveur de sauvegarde.

```
<backup>true</backup>
```

L'attribut de configuration dédié au serveur de sauvegarde d'HornetQ est : **`allow-failback`**. Il indique si le serveur de sauvegarde se ferme automatiquement quand le serveur live d'origine est de retour.

Rapporter un bogue

20.12.6. La réplication de messages HornetQ

AVERTISSEMENT

Seuls les messages persistés peuvent être répliqués. Tout message non persistant ne peut pas survivre à un basculement.

La réplication de messages entre un serveur direct et un serveur de sauvegarde est effectué par le biais du trafic réseau car les serveurs live et de sauvegarde ne partagent pas les mêmes stores de données. Toutes les revues sont répliquées entre les deux serveurs, tant que les deux serveurs sont dans le même cluster et ont le même nom d'utilisateur et mot de passe de cluster. Tout le trafic de données persistantes reçu par le serveur live est répliqué sur le serveur de sauvegarde.

Quand le serveur de sauvegarde est en ligne, il cherche à trouver et à se connecter à un serveur live pour tenter la synchronisation. Une fois synchronisé, il n'est plus disponible en tant que serveur de sauvegarde. La synchronisation peut prendre un long moment selon le volume de données à synchroniser et la vitesse du réseau. Si le seveur de sauvegarde apparaît en ligne et qu'aucun serveur live est disponible, le serveur de sauvegarde attendra jusqu'à ce que le serveur live soit disponible dans le cluster.

Pour activer les serveurs qui répliquent les données, il faudra définir un lien entre eux dans le fichier **`standalone-full-ha.xml`**. Un serveur de sauvegarde ne se répliquera qu'avec un serveur live du même nom de groupe. Le nom de groupe doit être défini dans le paramètre ***`backup-group-name`*** qui se trouve dans le fichier **`standalone-full-ha.xml`** de chaque serveur.

Dans le cas d'un serveur live ayant échoué, le serveur de sauvegarde correctement configuré et totalement synchronisé reprendra ses fonctions. Le serveur de sauvegarde ne s'activera que si le serveur live a échoué, et si le serveur de sauvegarde est en mesure de se connecter à plus de la moitié des serveurs dans le cluster. Si plus de la moitié des autres serveurs du cluster manquent également de répondre, cela indique une panne générale de réseau et le serveur de sauvegarde attendra pour réessayer la connexion au serveur live.

Pour accéder à l'état initial après un basculement, il faut démarrer le serveur et attendre qu'il soit entièrement synchronisée avec le serveur de sauvegarde. Lorsque cela aura été réalisé, vous pourrez arrêter le serveur de sauvegarde pour que le serveur de départ s'active à nouveau. Cela se fait automatiquement si l'attribut ***`allow-failback`*** est définie sur true.

Rapporter un bogue

20.12.7. Configurer les serveurs HornetQ pour la réplication

Pour configurer les serveurs live et de sauvegarde en tant que paire de réplication, configurer les fichiers **`standalone-full-ha.xml`** sur chaque serveur pour qu'ils aient les paramètres de configuration suivants :

```
<shared-store>false</shared-store>
<backup-group-name>NameOfLiveBackupPair</backup-group-name>
<check-for-live-server>true</check-for-live-server>
.
.
.
<cluster-connections>
   <cluster-connection name="my-cluster">
      ...
   </cluster-connection>
</cluster-connections>
```

Tableau 20.15. Attributs de configuration pour la réplication HornetQ

Attribut	Description
shared-store	Indique si le serveur utilise un store partagé ou non. La valeur par défaut est false.
backup-group-name	Il s'agit du nom unique qui identifie une paire live/sauvegarde qui doivent se répliquer ensemble
check-for-live-server	Indique si un serveur répliqué live doit vérifier le cluster actuel pour voir s'il existe déjà un serveur live avec le même id de nœud. La valeur par défaut est false.
failover-on-shutdown	Indique si ce serveur de sauvegarde (s'il s'agit d'un seveur de sauvegarde) doit devenir le serveur live lors d'un arrêt normal du serveur. La valeur par défaut est false.

Le serveur de sauvegarde doit également être marqué explicitement en tant que serveur de sauvegarde.

```
<backup>true</backup>
```

Tableau 20.16. Attributs de configuration de serveur de sauvegarde HornetQ

Attribut	Description
allow-failback	Indique si le serveur de sauvegarde se fermera automatiquement si le serveur live d'origine revient. La valeur par défaut est true.

Attribut	Description
max-saved-replicated-journal-size	Le nombre maximal de journaux de sauvegarde à conserver après que la restauration automatique se soit produit. La spécification de cet attribut n'est nécessaire que si allow-failback est sur true. La valeur par défaut est 2, ce qui signifie que, après 2 restaurations, le serveur de sauvegarde doit être redémarré pour pouvoir répliquer le journal du serveur live et redevenir la sauvegarde.

Rapporter un bogue

20.12.8. High-availability (HA) Failover

High-availability Failover est disponible, soit en basculement automatique des clients, ou en basculement au niveau des applications, grâce à une structure de live-backup. Chaque serveur live a un serveur de sauvegarde. Une sauvegarde par serveur live uniquement est prise en charge.

Le serveur de sauvegarde ne prend le relais que si le serveur se plante ou en cas de basculement. Après que le serveur a été redémarré, et si l'attribut de **`allow-failback`** est défini sur true, il redevient le serveur live. Lorsque le serveur live d'origine prend la relève, le serveur de sauvegarde rétablit sa fonction de sauvegarde pour le serveur live.

IMPORTANT

La mise en cluster doit être activée même si vous n'utilisez pas les fonctionnalités de clustering. C'est parce que chaque nœud du cluster HA doit avoir une connexion de cluster pour tous les autres nœuds, afin de négocier des rôles avec d'autres serveurs.

La topologie de cluster haute disponibilité est atteinte par les serveurs directs et de sauvegarde car ils envoient des informations sur leurs informations de connexion en utilisant la multidiffusion IP. Si la multidiffusion IP ne peut pas être utilisée, il est également possible d'utiliser une configuration statique des connexions initiales. Après la connexion initiale, le client est informé de la topologie. Si la connexion en cours est périmée, le client établit une connexion vers un autre nœud.

Après qu'un serveur live a échoué et qu'un serveur de sauvegarde a pris la relève, vous devrez redémarrer le serveur live et procéder à une restauration automatique des clients. Pour ce faire, redémarrez le serveur d'origine et arrêter (kill) le nouveau serveur. Vous pouvez le faire en supprimant le processus proprement dit ou attendre que le serveur se plante par lui-même. Vous pouvez également provoquer un basculement lors de l'arrêt normal du serveur. Pour cela, définir la propriété **`failover-on-shutdown`** à true dans le fichier de configuration **`standalone.xml`** :

```
<failover-on-shutdown>true</failover-on-shutdown>
```

Par défaut, la propriété **`failover-on-shutdown`** est définie à false.

Vous pouvez également forcer l'arrêt du nouveau serveur live quand l'ancien serveur live revient, ce qui permet au serveur live d'origine de prendre la relève automatiquement en définissant la propriété **`allow-failback`** à true dans le fichier de configuration **`standalone.xml`** :

```
<allow-failback>true</allow-failback>
```

En mode de réplication HA, pour forcer l'arrêt du nouveau serveur live quand l'ancien serveur live revient, définir la propriété **`check-for-live-server`** à true dans le fichier de configuration **`standalone.xml`** :

```
<check-for-live-server>true</check-for-live-server>
```

Rapporter un bogue

20.12.9. Déploiements sur les serveurs de sauvegarde HornetQ

Dans un environnement dédié de HA, un serveur JBoss EAP 6 avec HornetQ configuré comme une sauvegarde ne doit pas servir à héberger les déploiements qui utilisent ou se connectent à la sauvegarde HornetQ sur ce serveur. Cela inclut les déploiements comme les Enterprise Java Beans (Stateless Session Beans, Message Driven Beans) ou servlets.

Si un serveur JBoss EAP 6 a une configuration de sauvegarde de cohabitation HornetQ (avec le serveur HornetQ configuré en « live » dans le sous-système de messagerie et un autre serveur HornetQ configuré en sauvegarde), alors le serveur JBoss EAP 6 peut héberger des déploiements tant qu'ils sont configurés pour se connecter au serveur HornetQ « live ».

Rapporter un bogue

CHAPITRE 21. SOUS-SYSTÈME DE TRANSACTION

21.1. CONFIGURATION DE SOUS-SYSTÈME DE TRANSACTION

21.1.1. Configuration des transactions

Introduction

Les procédures suivantes vous montrent comment configurer le sous-système de transactions de JBoss EAP 6.

- Section 21.1.3, « Configurez votre base de données pour pouvoir utiliser les transactions JTA »
- Section 21.1.4, « Configuration d'une source de données XA »
- Section 21.1.2, « Configurer le gestionnaire de transactions (TM) »
- Section 21.1.6, « Configurer la journalisation des sous-systèmes de transactions »

Rapporter un bogue

21.1.2. Configurer le gestionnaire de transactions (TM)

Vous pouvez configurer le gestionnaire de transactions (TM) à l'aide de la console de gestion sur le web ou l'interface CLI en ligne de commandes. Pour chaque commande ou option donnée, on assume que vous exécutez JBoss EAP 6 comme un domaine géré. Si vous utilisez un serveur autonome ou que vous souhaitez modifier un profil différent de la valeur par **`default`**, il se peut que vous ayez à modifier les étapes et les commandes de la manière suivante.

Notes sur les commandes d'exemple

- Pour la console de gestion, le profil par **`default`** est celui qui sera sélectionné quand vous vous connectez à la console. Si vous souhaitez modifier la configuration du gestionnaire de transactions dans un autre profile, sélectionnez votre profile à la place, et non pas **`default`**, pour chaque instruction.

  De même, substituez votre profil à la place du profil par défaut **`default`** pour les commandes CLI de l'exemple.

- Si vous utilisez un serveur autonome, un seul profil existe. Ignorer toute instruction pour choisir un profil spécifique. Dans les commandes CLI, retirer la partie **`/profile=default`** des commandes d'échantillon.

NOTE

Pour que les options du gestionnaire de transactions (TM) soient visibles dans la console de gestion ou dans l'interface CLI, le sous-système **`transactions`** doit être activé. Il est activé par défaut, et il faut pour cela qu'un certain nombre d'autres sous-systèmes fonctionnent correctement, donc il est improbable qu'il soit désactivé.

Configurer le gestionnaire de transactions (TM) par la console de gestion

Pour configurer le gestionnaire de transactions (TM) à l'aide de la console de gestion sur le web, sélectionnez l'onglet **`Configuration`** en haut de l'écran. Si vous utilisez un domaine géré, vous avez le

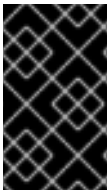
choix de plusieurs profils. Choisir le bon **Profile** de la boîte de sélection dans la partie supérieure gauche de l'écran. Étendez le menu **Container**, et sélectionnez **Transactions**.

Vous verrez la plupart des options dans la page de configuration du gestionnaire de transactions. Les options **Recovery** sont cachées par défaut. Cliquer sur l'onglet **Recovery** pour voir les options de recouvrement. Cliquer sur **Edit** pour éditer une des options. Les changements prendront place immédiatement.

Cliquer sur l'étiquette **Need Help?** pour afficher le texte d'aide en ligne.

Configurer le gestionnaire de transactions (TM) par l'interface CLI

Dans l'interface CLI, vous pouvez configurer le TM en utilisant une série de commandes. Les commandes commencent toutes par **/profile=default/subsystem=transactions/** pour un domaine géré avec **default** de profil, ou par **/subsystem=transactions** pour un serveur autonome.



IMPORTANT

HornetQ n'autorise pas plusieurs instances à partager un message log store. Si vous configurez plusieurs instances d'HornetQ, chaque instance devra avoir son propre message log store.

Tableau 21.1. Options Configuration Gestionnaire de transactions (TM)

Option	Description	Commande CLI
Activer les statistiques	Indique s'il faut activer les statistiques de transaction. Ces statistiques se trouvent dans la console de gestion dans la section **Subsystem Metrics** de l'onglet **Runtime**.	**/profile=default/subsystem=transactions/:write-attribute(name=enable-statistics,value=true)**
Délai d'attente par défaut	Délai d'attente de transaction par défaut. La valeur par défaut est de **300** secondes. Vous pouvez la remplacer par programmation, sur la base d'une transaction.	**/profile=default/subsystem=transactions/:write-attribute(name=default-timeout,value=300)**
Chemin de Store Objet	Un chemin de système de fichiers relatif ou absolu où le store objet TM stocke des données. Relatif, par défaut, à la valeur du paramètre **object-store-relative-to**.	**/profile=default/subsystem=transactions/:write-attribute(name=object-store-path,value=tx-object-store)**

Option	Description	Commande CLI
Chemin de Store Objet Relatif à	Référence une configuration de chemin global dans le modèle du domaine. La valeur par défaut correspond au répertoire de données de JBoss EAP 6, qui correspond à la valeur de la propriété **`jboss.server.data.dir`**, et qui a pour valeur par défaut ***`EAP_HOME`*`/domain/data/`** pour un domaine géré, ou ***`EAP_HOME`*`/standalone/data/`** pour une instance de serveur autonome. La valeur de l'attribut **`object-store-path`** de l'object store est relative à ce chemin.	**`/profile=default/subsystem=transactions/:write-attribute(name=object-store-relative-to,value=jboss.server.data.dir)`**
Liaisons de sockets	Indique le nom de la liaison du socket utilisé par le gestionnaire de transactions pour la récupération et la création des identificateurs de transactions, lorsque le mécanisme du socket est utilisé. Se référer à **`processus-id-socket-max-ports`** pour plus d'informations sur la génération de l'identificateur unique. Les liaisons de socket sont spécifiées par le groupe de serveurs dans l'onglet **`Serveur`** de la console de gestion.	**`/profile=default/subsystem=transactions/:write-attribute(name=socket-binding,value=txn-recovery-environment)`**
Listener de recouvrement	Indique si oui ou non le processus de recouvrement de transactions doit écouter au socket de réseau. La valeur par défaut est **`false`**.	**`/profile=default/subsystem=transactions/:write-attribute(name=recovery-listener,value=false)`**

Les options suivantes sont pour une utilisation avancée et ne peuvent être modifiées qu'à l'aide de l'interface CLI. Soyez prudent lors de leur modification à partir de la configuration par défaut. Communiquer avec Red Hat Global Support Services pour plus d'informations.

Tableau 21.2. Options de configuration TM avancées

Option	Description	Commande CLI
jts	Indique si l'on doit utiliser les transactions Java Transaction Service (JTS). La valeur par défaut est **`false`**, qui utilise des transactions JTA uniquement.	**`/profile=default/subsystem=transactions/:write-attribute(name=jts,value=false)`**

Option	Description	Commande CLI
Identificateur de nœud	L'identificateur de noeud de gestionnaire de transactions. Cette option est requise dans les situations suivantes : • Pour les communications de JTS à JTS • Quand deux gestionnaires de transactions accèdent à des gestionnaires de ressources partagées • Quand deux gestionnaires de transactions accèdent à des object stores partagés Le **`node-identifier`** doit être unique pour chaque gestionnaire de transaction car il est requis pour assurer l'intégrité des données pendant le recouvrement. Le **`node-identifier`** doit également être unique à JTA car plusieurs noeuds peuvent entrer en interraction avec le même gestionnaire de ressources ou partager un object store de transations.	**`/profile=default/subsystem=transactions/:write-attribute(name=node-identifier,value=1)`**
process-id-socket-max-ports	Le gestionnaire de transactions crée un identifiant unique pour chaque journal des transactions. Deux mécanismes différents sont fournis pour générer des identificateurs uniques : un mécanisme basé sur le socket et un mécanisme fondé sur l'identificateur de processus du processus. Dans le cas de l'identifiant basé-socket, le socket est ouvert et son numéro de port est utilisé pour l'identifiant. Si le port est déjà utilisé, on cherchera le port suivant, jusqu'à ce qu'un port libre soit trouvé. Le **`processus-id-socket-max-ports`** représente le nombre maximal de sockets que le gestionnaire de transactions (TM) va essayer avant d'abandonner. La valeur par défaut est **`10`**.	**`/profile=default/subsystem=transactions/:write-attribute(name=process-id-socket-max-ports,value=10)`**

Option	Description	Commande CLI
process-id-uuid	Définir à **`true`** avec un identifiant de processus pour créer un identifiant unique pour chaque transaction. Sinon, le mécanisme basé socket sera utilisé. La valeur par défaut est **`true`**. Se référer à **`process-id-socket-max-ports`** pour obtenir davantage d'informations.	**`/profile=default/subsystem=transactions/:write-attribute(name=process-id-uuid,value=true)`**
use-hornetq-store	Utiliser les mécanismes de stockage journalisés de HornetQ au lieu du stockage basé sur des fichiers, pour les journaux de transactions. Ceci est désactivé par défaut, mais peut améliorer les performances I/O. Il n'est pas recommandé pour les transactions JTS sur les gestionnaires de transactions séparés. Quand vous changez cette option, le serveur doit démarrer à nouveau à l'aide de la commande **`shutdown`** pour que le changement puisse prendre effet.	**`/profile=default/subsystem=transactions/:write-attribute(name=use-hornetq-store,value=false)`**

Rapporter un bogue

21.1.3. Configurez votre base de données pour pouvoir utiliser les transactions JTA

Résumé

Cette tâche vous montre comment activer JTA (Java Transactions API) sur votre source de données.

Pré-requis

Vous devez remplir les conditions suivantes avant de continuer cette tâche :

- Votre base de données ou autre ressource devra supporter JTA. Dans le doute, veuillez consulter la documentation.
- Créer une source de données. Veuillez vous référer à Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion ».
- Stopper le serveur JBoss EAP 6
- Obtenir un accès pour pouvoir éditer les fichiers de configuration directement, dans un éditeur de texte.

Procédure 21.1. Configurer la source de données pour utiliser les transactions JTA.

1. **Ouvrir le fichier de configuration dans l'éditeur de texte.**

Selon si vous exécutez JBoss EAP 6 sur un domaine géré ou un serveur autonome, votre fichier de configuration ne se trouvera pas au même endroit.

- **Domaine géré**
  Le fichier de configuration par défaut d'un domaine géré se trouve dans ***EAP_HOME*/domain/configuration/domain.xml** pour Red Hat Enterprise Linux, et ***EAP_HOME*\domain\configuration\domain.xml** pour Microsoft Windows Server.

- **Serveur autonome**
  Le fichier de configuration par défaut d'un serveur autonome se trouve dans ***EAP_HOME*/standalone/configuration/domain.xml** pour Red Hat Enterprise Linux, et ***EAP_HOME*\standalone\configuration\domain.xml** pour Microsoft Windows Server.

2. **Chercher la balise `<datasource>` qui correspond à votre source de données.**
   La source de données aura un attribut **`jndi-name`** correspondant à celui que vous aviez indiqué quand vous l'avez créée. Par exemple, la source de données ExampleDS ressemble à ceci :

```
<datasource jndi-name="java:jboss/datasources/ExampleDS" pool-
name="H2DS" enabled="true" jta="true" use-java-context="true" use-
ccm="true">
```

3. **Définir l'attribut `jta` à `true`.**
   Ajouter l'élément suivant au contenu de votre balise **`<datasource>`**, tel qu'il apparaît à l'étape précédente : **`jta="true"`**

4. **Sauvegarder le fichier de configuration.**
   Sauvegarder le fichier de configuration et sortir de l'éditeur de texte.

5. **Démarrer JBoss EAP 6.**
   Relancer le serveur JBoss EAP 6.

Résultat :

JBoss EAP 6 démarre, et votre source de données est configurée pour utiliser les transactions JTA.

Rapporter un bogue

21.1.4. Configuration d'une source de données XA

Pré-requis

Pour pouvoir ajouter une source de données XA, vous devrez vous connecter à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion » pour plus d'informations.

1. **Ajouter une nouvelle source de données.**
   Ajouter une nouvelle source de données à la plateforme JBoss EAP 6. Suivre les instructions qui se trouvent dans Section 6.3.1, « Créer une source de données non-XA avec les interfaces de gestion », puis, cliquer sur l'onglet **`XA Datasource`** en haut.

2. **Configurer les propriétés supplémentaires suivant les besoins.**
   Tous les paramètres de la source de données se trouvent dans Section 6.6.1, « Paramètres de source de données ».

Résultat

Votre source de données XA est configurée et prête à l'utilisation.

Rapporter un bogue

21.1.5. Messages de journalisation de transactions

Pour suivre le statut de la transaction tout en gardant les fichiers de journalisation lisibles, utiliser le niveau de journalisation **DEBUG** pour le logger de transaction. Pour un débogage détaillé, utiliser le niveau de journalisation **TRACE**. Veuillez consulter Section 21.1.6, « Configurer la journalisation des sous-systèmes de transactions » pour plus d'informations sur la configuration du logger de transaction.

Le gestionnaire de transaction peut générer beaucoup d'informations de journalisation si configuré pour se connecter au niveau de journalisation **TRACE**. Vous trouverez ci-dessous quelques-uns des messages les plus courants. Cette liste n'est pas exhaustive, il se peut que vous rencontriez d'autres messages.

Tableau 21.3. Changement d'état de transaction

Début de transaction	Quand une transaction commence, le code suivant s'exécute :
	`com.arjuna.ats.arjuna.coordinator.BasicAction::Begin:1342`
	`tsLogger.logger.trace("BasicAction::Begin() for action-id "+ get_uid());`

Validation de transaction	Quand une transaction est validée, le code suivant s'exécute :
	`com.arjuna.ats.arjuna.coordinator.BasicAction::End:1342`
	`tsLogger.logger.trace("BasicAction::End() for action-id "+ get_uid());`

Restauration de transaction	Quand une transaction est restaurée, le code suivant s'exécute :
	`com.arjuna.ats.arjuna.coordinator.BasicAction::Abort:1575`
	`tsLogger.logger.trace("BasicAction::Abort() for action-id "+ get_uid());`

Délai d'expiration de transaction	Quand une transaction expire, le code suivant s'exécute :
	
	```com.arjuna.ats.arjuna.coordinator.TransactionReaper::doCancellations:349```
	
	```tsLogger.logger.trace("Reaper Worker " + Thread.currentThread() + " attempting to cancel " + e._control.get_uid());```
	
	Vous verrez ensuite le même thread restaurer la transaction comme montré ci-dessus.

Rapporter un bogue

21.1.6. Configurer la journalisation des sous-systèmes de transactions

Résumé

Utiliser cette procédure pour contrôler la quantité d'informations enregistrées sur les transactions, indépendamment des autres paramètres de journalisation dans JBoss EAP 6. La procédure montre comment procéder dans la console de gestion sur le web. La commande de gestion CLI est donnée par la suite.

Procédure 21.2. Configurer l'enregistreur de transactions (Transaction Logger) par la console de gestion

1. **Naviguer vers la zone de configuration de la journalisation**
   Dans la console de gestion, cliquer sur l'onglet **`Configuration`**. Si vous utilisez un domaine géré, fermer le profil du serveur que vous souhaitez configurer, à partir de la case de sélection **`Profile`** qui se trouve en haut et à gauche.

   Étendre le menu **`Core`**, et sélectionner **`Logging`**.

2. **Modifier les attributs de `com.arjuna`.**
   Sélectionner l'onglet **`Log Categories`**. Sélectionner **`com.arjuna`**, puis **`Edit`** dans **`Details`**. Vous pourrez ajouter ici les informations de journalisation spécifiques à la classe. La classe **`com.arjuna`** est déjà présente. Vous pourrez modifier le niveau de journalisation et décider si vous souhaitez utiliser les handlers parents.

   Niveau de journalisation

   Le niveau de journalisation est **`WARN`** par défaut. Comme les transactions peuvent produire une grande quantité de messages de journalisation, la signification des niveaux de journalisation standard est légèrement différente pour l'enregistreur de transactions. En général, les messages avec des niveaux de gravité moins élevés que le niveau choisi seront ignorés.

   Niveaux de journalisation des transactions, du plus au moins détaillé.

- TRACE
- DEBOG
- INFO
- AVERTISSEMENT
- ERREUR
- FAILURE

Utiliser les gestionnaires parents

Indique si l'enregistreur d'événements doit envoyer ses sorties vers l'enregistreur d'événements parent. Le comportement par défaut est **`true`**.

3. Les changements prennent effet immédiatement.

Rapporter un bogue

21.2. ADMINISTRATION DES TRANSACTIONS

21.2.1. Naviguer et gérer les transactions

Le CLI prend en charge la capacité de naviguer et de manipuler les enregistrements des transactions. Cette fonctionnalité est fournie par l'interaction entre le gestionnaire de transactions et l'API de gestion de JBoss EAP 6.

Le gestionnaire de transactions stocke des informations sur chaque transaction en attente et les participants impliqués dans la transaction, dans un stockage persistant appelé *object store*. L'API de gestion expose le store objet sous forme de ressource appelée **`log-store`**. Une opération API nommée **`probe`** lit les journaux de transaction et crée un noeud pour chaque journal. Vous pouvez invoquer la commande **`probe`** manuellement, quand vous souhaitez réactualiser le **`log-store`**. Il est normal pour les journaux de transaction d'apparaitre ou de disparaitre rapidement.

Exemple 21.1. Réactualiser le log store

Cette commande réactualise le log store des groupes de serveurs qui utilisent le profil par défaut **`default`** dans un domaine géré. Dans le cas d'un serveur autonome, supprimer **`profile=default`** de la commande.

```
/profile=default/subsystem=transactions/log-store=log-store/:probe
```

Exemple 21.2. Voir toutes les transactions préparées

Pour voir toutes les transactions préparées, commencer par réactualiser le log store (voir Exemple 21.1, « Réactualiser le log store »), puis exécuter la commande suivante, qui fonctionne de la même manière qu'une commande **`ls`** de système de fichiers.

```
ls /profile=default/subsystem=transactions/log-store=log-
store/transactions
```

Chaque transaction est visible, ainsi que son identifiant unique. Les opérations individuelles peuvent être exécutées contre une transaction individuelle (voir Gérer une transaction).

Gérer une transaction

Voir des attributs de transaction.

Pour voir des informations sur une transaction, comme son nom JNDI, son nom de produit EIS ou sa version, ou statut, utiliser la commande CLI **`:read-resource`**.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:read-resource
```

Voir tous les participants d'une transaction.

Chaque journal de transaction contient un élément enfant nommé **`participants`**. Utiliser la commande CLI **`read-resource`** sur cet élément pour voir les participants des transactions. Les participants sont identifiés par leur nom JNDI.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-
b66efc2\:4f9e6f8f\:9/participants=java\:\/JmsXA:read-resource
```

Le résultat devrait ressembler à ceci :

```
{
   "outcome" => "success",
   "result" => {
       "eis-product-name" => "HornetQ",
       "eis-product-version" => "2.0",
       "jndi-name" => "java:/JmsXA",
       "status" => "HEURISTIC",
       "type" => "/StateManager/AbstractRecord/XAResourceRecord"
   }
}
```

Le statut du résultat affiché ici est dans un état **`HEURISTIC`** et est susceptible d'être recouvré. Voir Recouvrement d'une transaction. pour plus d'informations.

Supprimer une transaction.

Chaque journal de transaction supporte une opération **`:delete`** pour effacer l'enregistrement qui représente la transaction.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:delete
```

Recouvrement d'une transaction.

Chaque journal de transaction supporte le recouvrement par la commande CLI **`:recover`**.

Recouvrement des transactions heuristiques et des participants

- Si le statut de la transaction est **`HEURISTIC`**, l'opération de recouvrement change l'état en **`PREPARE`** et déclenche un recouvrement.

- Si l'un des participants de la transaction est heuristique, l'opération de recouvrement tente de reprendre l'opération **`commit`** (validation) à nouveau. En cas de succès, le participant est retiré du journal des transactions. Vous pouvez vérifier cela en exécutant à nouveau l'opération **`:probe`** sur le **`log-store`** et en vérifiant que le participant n'est plus inscrit. Si c'est le dernier participant, la transaction sera également supprimée.

Réactualiser le statut de la transaction qui a besoin d'être recouvrée.

Si une transaction a besoin d'être recouvrée, vous pourrez utiliser la commande CLI **`:refresh`** pour vous assurer qu'elle a toujours besoin d'être recouvrée, avant de tenter le recouvrement.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:refresh
```

Voir les statistiques de transaction

Si les statistiques de Transaction manager sont activées, vous pouvez consulter les statistiques à propos du Gestionnaire de transactions et du sous-système de transaction. Veuillez consulter Section 21.1.2, « Configurer le gestionnaire de transactions (TM) » pour plus d'informations sur l'activation des statistiques de Transaction manager.

Vous pouvez consulter les statistiques soit par la console de gestion basée-web, soit par le CLI. de gestion Dans la console de gestion basée-web, les statistiques de transaction seront disponibles via **Runtime** → **Status** → **Subsystems** → **Transactions**. Les statistiques de transaction sont disponibles pour chaque serveur dans un domaine géré, également. Pour voir le statut d'un autre serveur, sélectionner **`Change Server`** situé en haut à gauche du menu, et sélectionner un serveur de la liste.

La table suivante affiche chaque statistique disponible, sa description et la commande CLI de gestion pour afficher le statistique.

Tableau 21.4. Les statistiques de sous-système de transaction

Statistique	Description	Commande CLI
Total	Le nombre total de transactions exécutées par le gestionnaire de transactions sur ce serveur.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-transactions,include-defaults=true)`

Statistique	Description	Commande CLI
Validé	Le nombre de transactions validées exécutées par le gestionnaire de transactions sur ce serveur.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-committed-transactions,include-defaults=true)`
Abandonné	Le nombre de transactions interrompues exécutées par le gestionnaire de transactions sur ce serveur.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-aborted-transactions,include-defaults=true)`
Délai expiré	Le nombre de transactions expirées exécutées par le gestionnaire de transactions sur ce serveur.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-timed-out-transactions,include-defaults=true)`
Heuristiques	Pas disponible dans la console de gestion. Nombre de transactions dans un état heuristique.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-heuristics,include-defaults=true)`

Statistique	Description	Commande CLI
Transactions In-Flight	Pas disponible dans la console de gestion. Nombre de transactions commencées mais pas encore achevées.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-inflight-transactions,include-defaults=true)`
Origine de l'échec - Applications	Le nombre de transactions échouées dont l'origine de l'échec était une application.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-application-rollbacks,include-defaults=true)`
Origine de l'échec - Ressources	Le nombre de transactions échouées dont l'origine de l'échec était une ressource.	`/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-resource-rollbacks,include-defaults=true)`

Rapporter un bogue

21.3. RÉFÉRENCES DE TRANSACTIONS

21.3.1. Erreurs et exceptions pour les transactions JBoss

Pour obtenir des informations lancées par les méthodes de la classe **`UserTransaction`**, voir la spécification *UserTransaction API* dans http://download.oracle.com/javaee/1.3/api/javax/transaction/UserTransaction.html.

Rapporter un bogue

21.3.2. Limitations sur les transactions JTA

Les transactions JTA ne peuvent pas être au courant de la distribution à travers les instances multiples de JBoss EAP 6. Pour ce comportement, utiliser les transactions JTS.

Pour pouvoir utiliser les transactions JTS, vous devrez configurer l'ORB, qui inclut l'activation des transactions dans le sous-système JacORB, puis configurez le sous-système JTS.

- Section 21.4.2, « Configurer l'ORB pour les transactions JTS »

Rapporter un bogue

21.4. CONFIGURATION ORB

21.4.1. CORBA (Common Object Request Broker Architecture)

Common Object Request Broker Architecture (CORBA) est une norme qui autorise les applications et services à travailler ensemble même lorsqu'ils sont écrits dans de multiples langages normalement incompatibles ou lorsqu'ils sont hébergés sur des plateformes différentes. Les requêtes CORBA sont émises par un composant côte serveur appelé *Object Request Broker (ORB)*. JBoss EAP 6 fournit une instance ORB au moyen du composant JacORB.

L'ORB est utilisé en interne pour les transactions *Java Transaction Service (JTS)*, et peut également être utilisé par votre propre application.

Rapporter un bogue

21.4.2. Configurer l'ORB pour les transactions JTS

Dans une installation JBoss EnAP 6 par défaut, l'ORB est désactivé. Vous pouvez activer l'ORB en utilisant l'interface CLI de ligne de commande.

NOTE

Dans un domaine géré, le sous-système JacORB est disponible dans les profils **`full`** et **`full-ha`** uniquement. Dans un serveur autonome, il est disponible uniquement quand vous utilisez les configurations **`standalone-full.xml`** ou **`standalone-full-ha.xml`**.

Procédure 21.3. Configurer l'ORB par la console de gestion

1. **Voir les paramètres de configuration du profil.**
   Sélectionner **`Configuration`** dans la partie supérieure de la console de gestion. Si vous utilisez un domaine géré, sélectionner soit le profil **`full`** ou **`full-ha`** à partir de la boîte de dialogue de sélection en haut à gauche.

2. **Modifier les paramètres `Initialisateurs`**
   Étendre sur le menu **`Subsystems`**. Étendre le menu **`Container`**, et sélectionner **`JacORB`**.

   Sur le formulaire qui apparaît sur l'écran principal, sélectionner l'onglet **`Initializers`**, et cliquer sur le bouton **`Edit`**.

   Activer les intercepteurs de sécurité en configurant la valeur de **`Security`** à **`active`**.

   Pour activer ORB sur JTS, définir la valeur des **`Transaction Interceptorss`** à **`active`**, au lieu de la valeur par défaut **`spec`**.

   Voir le lien **`Need Help?`** sur le formulaire pour accéder à des explications sur ces valeurs. Cliquer sur **`Save`** quand vous aurez fini de modifier les valeurs.

3. **Configuration ORB avancée**

Voir les autres sections du formulaire pour les options de configuration avancées. Chaque section inclut un lien **Need Help?** avec des informations détaillées sur les paramètres.

Configurer l'ORB par l'interface CLI

Vous pouvez configurer chaque aspect de l'ORB à l'aide de l'interface CLI. Les commandes suivantes configurent les initialisateurs aux mêmes valeurs que celles de la procédure ci-dessus, pour la console de gestion. Il s'agit de la configuration minimale pour l'ORB, si utilisé avec JTS.

Ces commandes sont configurées pour un domaine de sécurité utilisant le profil **full**. Si nécessaire, modifier le profil pour qu'il convienne mieux à celui que vous aurez besoin de configurer. Si vous utilisez un serveur autonome, n'utilisez pas la portion **/profile=full** des commandes.

Exemple 21.3. Activer les intercepteurs de sécurité

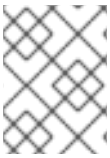
```
/profile=full/subsystem=jacorb/:write-attribute(name=security,value=on)
```

Exemple 21.4. Activer les transactions dans le sous-système JacORB

```
/profile=full/subsystem=jacorb/:write-
attribute(name=transactions,value=on)
```

Exemple 21.5. Activer JTS dans le sous-système de transactions

```
/profile=full/subsystem=transactions:write-
attribute(name=jts,value=true)
```



NOTE

Pour l'activation JTS, le serveur doit être démarré à nouveau car le rechargement ne suffira pas.

Rapporter un bogue

21.5. JDBC OBJECT STORE SUPPORT

21.5.1. JDBC Store de Transactions

Conditions préalables :

- Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI »

Les transactions peuvent utiliser une source de données JDBC comme magasin d'objets. Si la base de données à utiliser est configuré pour le basculement et la récupération, c'est peut-être une meilleure option que l'utilisation d'espace disque sur un serveur d'applications. Les avantages doivent être pesés contre le fait qu'un magasin d'objets JDBC brut est un magasin d'objets spéciaux et ne peut pas exécuter de la mçeme façon qu'un système de fichiers ou qu'un HornetQ journal object store.

NOTE

Une source de données JDBC utilisée comme Transactions object store *doit* specifier **`jta="false"`** dans la section **`datasource`** du fichier de configuration du serveur.

Procédure 21.4. Active l'utilisation d'une source de données JDBC comme Transaction Object Store

1. Définir **`use-jdbc-store`** à **`true`**.

```
/subsystem=transactions:write-attribute(name=use-jdbc-store,
value=true)
```

2. Définir **`jdbc-store-datasource`** au nom JNDI pour la source de données à utiliser.

```
/subsystem=transactions:write-attribute(name=jdbc-store-datasource,
value=java:jboss/datasources/TransDS)
```

3. Démarrer à nouveau le serveur JBoss EAP 6 pour que les changements puissent prendre effet.

```
shutdown --restart=true
```

L'ensemble des attributs sont fournis ci-dessous.

Tableau 21.5. Propriétés des StoresJDBC de transactions

Property	Description
`use-jdbc-store`	Le définir à true pour activer le store JDBC de transactions.
`jdbc-store-datasource`	Le nom JNDI de la source de données JDBC utilisée pour le stockage.
`jdbc-action-store-drop-table`	Supprimez et recréez les tables de stores d'actions lors du lancement. En option, par défaut « false ».
`jdbc-action-store-table-prefix`	Le préfixe des noms de tables de stores d'actions. En option.
`jdbc-communication-store-drop-table`	Supprimez et recréez les tables de stores de communications lors du lancement. En option, par défaut « false ».
`jdbc-communication-store-table-prefix`	Le préfixe des noms de tables de stores de communications. En option.

Property	Description
`jdbc-state-store-drop-table`	Supprimez et recréez les tables de stores d'états lors du lancement. En option, par défaut « false ».
`jdbc-state-store-table-prefix`	Le préfixe des noms de tables de stores d'états. En option.

Voir également :

- Section 21.1.3, « Configurez votre base de données pour pouvoir utiliser les transactions JTA »

Rapporter un bogue

CHAPITRE 22. SOUS-SYSTÈME DE MESSAGERIE

22.1. UTILISER DES TRANSPORTS PERSONNALISÉS DANS LES SOUS-SYSTÈMES DE MESSAGERIE

Quand on utilise un serveur de messagerie standard (POP3, IMAP), le serveur possède un ensemble d'attributs qui peuvent être définis, dont certains obligatoires.

Le plus important étant **`outbound-socket-binding-ref`** qui fait référence à une liaison de socket de mail sortante et qui est défini par l'adresse d'hôte et le numéro de port.

Ce n'est pas la meilleure solution pour certains utilisateurs car leur configuration utilise les hôtes multiples à but d'équilibrage des charges. Cette configuration, cependant, n'est pas prise en charge par le JavaMail standard, ce qui signifie que certains utilisateurs devront mettre en place des moyens de transport pour leur mail personnalisés.

Ces transports personnalisés ne requièrent pas de **`outbound-socket-binding-ref`** et autorisent les formats de propriétés d'hôte personnalisés.

Un transport personnalisé peut être configuré par le CLI à l'aide des commandes suivantes :

Procédure 22.1.

1. Ajouter une nouvelle session mail. La commande ci-dessous créer une nouvelle session nommée mySession et définit JNDI à **`java:jboss/mail/MySession`** :

```
/subsystem=mail/mail-session=mySession:add(jndi-
name=java:jboss/mail/MySession)
```

2. Ajouter une liaison de socket sortante. La commande ci-dessous ajoute une liaison de socket nommée ***`my-smtp-binding`*** qui pointe vers **`localhost:25`**.

```
/socket-binding-group=standard-sockets/remote-destination-outbound-
socket-binding=my-smtp-binding:add(host=localhost, port=25)
```

3. Ajouter un serveur SMTP avec **`outbind-socket-binding-ref`**. La commande suivante ajoute un SMTP nommé **`my-smtp-binding`** et définit un nom d'utilisateur, un mot de passe et une configuration TLS.

```
/subsystem=mail/mail-session=mySession/server=smtp:add(outbound-
socket-binding-ref= my-smtp-binding, username=user, password=pass,
tls=true)
```

4. Répéter ce processus pour POP3 et IMAP :

```
/socket-binding-group=standard-sockets/remote-destination-outbound-
socket-binding=my-pop3-binding:add(host=localhost, port=110)
```

```
/subsystem=mail/mail-session=mySession/server=pop3:add(outbound-
socket-binding-ref=my-pop3-binding, username=user, password=pass)
```

```
/socket-binding-group=standard-sockets/remote-destination-outbound-
socket-binding=my-imap-binding:add(host=localhost, port=143)
```

```
/subsystem=mail/mail-session=mySession/server=imap:add(outbound-
socket-binding-ref=my-imap-binding, username=user, password=pass)
```

5. Pour utiliser un serveur personnalisé, créer un nouveau serveur mail personnalisé sans la liaison de socket sortante (comme c'est optionnel) et fournir à la place l'information hôte comme faisant partie des propriétés.

```
/subsystem=mail/mail-
session=mySession/custom=myCustomServer:add(username=user,password=p
ass, properties={"host" => "myhost", "my-property" =>"value"})
```

Lorsque vous définissez les protocoles personnalisés, n'importe quel nom de propriété qui contient un point (.) est considéré comme un nom complet et est passé tel qu'il est fourni. N'importe quel autre format (*my-property*, par exemple) sera traduit dans le format suivant : **`mail.`*`server-name`*`.`*`my-property`***.

Vous trouverez ci-dessous un exemple de configuration XML complète qui montre un format personnalisé de l'attribut custom-server :

```
<subsystem xmlns="urn:jboss:domain:mail:1.1">
    <mail-session jndi-name="java:/Mail" from="user.name@domain.org">
        <smtp-server outbound-socket-binding-ref="mail-smtp" tls="true">
            <login name="user" password="password"/>
        </smtp-server>
        <pop3-server outbound-socket-binding-ref="mail-pop3"/>
        <imap-server outbound-socket-binding-ref="mail-imap">
            <login name="nobody" password="password"/>
        </imap-server>
    </mail-session>
    <mail-session debug="true" jndi-name="java:jboss/mail/Default">
        <smtp-server outbound-socket-binding-ref="mail-smtp"/>
    </mail-session>
    <mail-session debug="true" jndi-name="java:jboss/mail/Custom">
        <custom-server name="smtp">
            <login name="username" password="password"/>
            <property name="host" value="mail.example.com"/>
        </custom-server>
        <custom-server name="pop3" outbound-socket-binding-ref="mail-
pop3">
            <property name="custom_prop" value="some-custom-prop-value"/>
            <property name="some.fully.qualified.property" value="fully-
qualified-prop-name"/>
        </custom-server>
    </mail-session>
    <mail-session debug="true" jndi-name="java:jboss/mail/Custom2">
        <custom-server name="pop3" outbound-socket-binding-ref="mail-
pop3">
            <property name="custom_prop" value="some-custom-prop-value"/>
        </custom-server>
    </mail-session>
</subsystem>
```

Rapporter un bogue

CHAPITRE 23. ENTERPRISE JAVABEANS

23.1. INTRODUCTION

23.1.1. Entreprise JavaBeans

Enterprise JavaBeans (EJB) 3.1 est une API pour développer des applications de Java EE distribuées, transactionnelles, sécurisées et portables grâce à l'utilisation des composants côté serveur appelés Enterprise Beans. Enterprise Beans implémente la logique métier d'une application, de manière découplée, qui encourage la réutilisation. Enterprise JavaBeans 3.1 est documenté dans la spécification Java EE JSR-318.

JBoss EAP 6 prend en charge la génération d'applications qui utilisent la spécification Enterprise JavaBeans 3.1.

Rapporter un bogue

23.1.2. Enterprise JavaBeans pour Administrateurs

Les administrateurs JBoss ont de nombreuses options de configuration disponibles pour contrôler la performance des Beans Enterprise dans JBoss EAP 6. Ces options sont accessibles par la console de gestion ou par l'outil de configuration de ligne de commande. Éditer le fichier de configuration du serveur XML pour appliquer les modifications est également possible mais non recommandé

Les options de configuration EJB se situent dans des endroits légèrement différents de la console de gestion, selon que le serveur exécute ou non.

1. Cliquez sur l'onglet **`Configuration`** qui se trouve en haut de la console de gestion.
2. Si vous êtes en mode de domaine, sélectionner un profil à partir du menu déroulant **`Profiles`** en haut et à gauche.
3. Étendre le menu **`Subsystems`**.
4. Étendre le menu **`Container`**, puis sélectionner **`EJB 3`**.

Rapporter un bogue

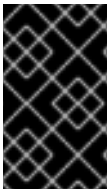
23.1.3. Beans Enterprise

Les beans enterprise sont des composants d'applications côté serveur, ainsi définis dans la spécification Enterprise JavaBeans (EJB) 3.1, JSR-318. Les beans enterprise sont conçus pour l'implémentation d'une logique commerciale d'application d'une manière découplée, pour encourager sa réutilisation.

Les beans enterprise sont écrits comme des classes Java et sont annotés avec les annotations EJB appropriées. Ils peuvent être déployés sur le serveur d'applications dans leurs propres archives (un fichier JAR) ou être déployés dans le cadre d'une application Java EE. Le serveur d'applications gère le cycle de vie de chaque bean entreprise et leur fournit des services comme la sécurité, les transactions et la gestion de concurrence.

Un bean enterprise peut également définir un nombre d'interfaces de métier. Les interfaces de métier vous proposent un plus grand contrôle sur les méthodes bean disponibles aux clients et peut également vous donner accès aux clients qui exécutent dans les JVM à distance.

Il existe trois types de beans enterprise : les session beans, les message-driven beans et les entity beans.



IMPORTANT

Les entity beans sont maintenant obsolètes dans EJB 3.1 et Red Hat recommande d'utiliser des entités JPA à la place. Red Hat ne recommande d'utiliser des entity beans que pour les compatibilités rétroactives avec les systèmes hérités.

Rapporter un bogue

23.1.4. Session Beans

Les session beans sont des beans enterprise qui encapsulent un ensemble de processus métier connexes ou des tâches, et qui sont injectés dans les classes qui en ont fait la demande. Il existe trois types de session beans : sans état, avec état et singleton.

Rapporter un bogue

23.1.5. Message-Driven Beans

Les Message-driven Beans (MDB) fournissent un modèle basé-événement pour le développement de l'application. Les méthodes des MDB ne sont pas injectées ou invoquées du code client mais sont déclenchées par la réception de messages d'un service de messagerie tel que le serveur Java Messaging Service (JMS). La spécification Java EE 6 exige que JMS soit pris en charge, mais les autres systèmes de messagerie peuvent être supportés également.

Rapporter un bogue

23.2. CONFIGURER LES BEAN POOLS

23.2.1. Bean Pools

JBoss EAP 6 maintient un certain nombre d'instances de beans stateless enterprise déployés en mémoire pour procurer une performance plus rapide. Cette technique s'appelle le Bean Pooling. Quand on a besoin d'un bean, le serveur de l'application peut en prendre un du pool qui convient parmi les beans déjà existants au lieu d'en instancier un nouveau. Quand le bean n'est plus requis, il est renvoyé dans le pool en vue d'être réutilisé.

Les bean pools sont configurés et maintenus séparément dans le cas des stateless session beans et des beans basés messages.

Rapporter un bogue

23.2.2. Créer un bean pool

Les bean pools peuvent être créés par l'intermédiaire de la console de gestion ou du CLI.

Les bean pools peuvent également être créés en ajoutant la configuration du bean pool requis au fichier de configuration du serveur en utilisant l'éditeur de texte. Exemple 23.2, « Exemple de configuration XML » est un exemple de configuration.

Procédure 23.1. Créer un bean pool par la console de gestion

1. Se connecter à la console de gestion. Consulter Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur **Add**. Le dialogue **Add EJB3 Bean Pools** apparaîtra.
4. Donnez les informations requises, les valeurs de **Name**, **Max Pool Size**, **Timeout** et l'unité de **Timeout**.
5. Cliquer sur **Save** pour terminer.

Procédure 23.2. Créer un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:add(max-pool-size=MAXSIZE, timeout=TIMEOUT, timeout-unit="UNIT")
```

   - Remplacer *BEANPOOLNAME* par le nom requis de bean pool.
   - Remplacer *MAXSIZE* par le nom requis de bean pool.
   - Remplacer *TIMEOUT*
   - Remplacer *UNIT* par l'unité de temps requise. Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.

3. Utiliser l'opération **read-resource** pour confirmer la création d'un bean pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-resource
```

Exemple 23.1. Créer un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-pool=ACCTS_BEAN_POOL:add(max-pool-size=500, timeout=5000, timeout-unit="SECONDS")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 23.2. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">

   <pools>

      <bean-instance-pools>
```

```
            <strict-max-pool  name="slsb-strict-max-pool" max-pool-
size="20"
               instance-acquisition-timeout="5"
               instance-acquisition-timeout-unit="MINUTES" />

            <strict-max-pool name="mdb-strict-max-pool" max-pool-size="20"
               instance-acquisition-timeout="5"
               instance-acquisition-timeout-unit="MINUTES" />

         </bean-instance-pools>

      </pools>

   </subsystem>
```

Rapporter un bogue

23.2.3. Supprimer un bean pool

Les bean pool non utilisés peuvent être supprimés par la console de gestion.

Prérequis :

- Le bean pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Consulter Section 23.2.5, « Assigner des beans pools aux beans de session et aux beans basés messages » pour vérifier qu'ils ne sont pas en cours d'utilisation.

Procédure 23.3. Supprimer un bean pool par la console de gestion.

1. Se connecter à la console de gestion. Consulter Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur le bean pool que vous souhaiter supprimer de la liste.
4. Cliquer sur **`Remove`**. La boîte de dialogue **`Remove Item`** apparaîtra.
5. Cliquer sur **`Confirm`** pour confirmer.

Procédure 23.4. Supprimer un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **`remove`** avec la syntaxe suivante.

   ```
   /subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:remove
   ```

   - Remplacer *BEANPOOLNAME* par le nom requis de bean pool.

Exemple 23.3. Supprimer un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-
pool=ACCTS_BEAN_POOL:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

23.2.4. Modifier un bean pool

Les bean pools peuvent être modifiés par la console de gestion.

Procédure 23.5. Modifier un bean pool par la console de gestion

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion »
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Bean Pools**.
3. Cliquer sur le bean pool que vous souhaiter modifier.
4. Cliquer sur **`Edit`**.
5. Modifier les informations que vous souhaitez. Seules les valeurs de **`Max Pool Size`**, **`Timeout`**, et de l'unité de **`Timeout Unit`** peuvent être modifiées.
6. Cliquer sur le bouton **`Save`** pour terminer.

Procédure 23.6. Modifier un bean pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante pour chaque attribut du bean pool à modifier.

   ```
   /subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:write-
   attribute(name="ATTRIBUTE", value="VALUE")
   ```

   - Remplacer *BEANPOOLNAME* par le nom requis de bean pool.
   - Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont **`max-pool-size`**, **`timeout`**, et **`timeout-unit`**.
   - Remplacer *VALUE* par la valeur requise de l'attribut.
3. Utiliser l'opération **`read-resource`** pour confirmer les changements au bean pool.

   ```
   /subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-
   resource
   ```

Exemple 23.4. Définir la valeur de timeout d'un bean pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-
pool=HSBeanPool:write-attribute(name="timeout", value="1500")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

23.2.5. Assigner des beans pools aux beans de session et aux beans basés messages

Les administrateurs de systèmes JBoss peuvent assigner des beans pools individuels que les session beans et les bean basés-messages peuvent utiliser. Les beans pools peuvent être distribués par la console de gestion ou le CLI.

Par défaut, deux beans pools sont fournis, **`slsb-strict-max-pool`** et **`mdb-strict-max-pool`** pour les stateless sessions beans et les beans basés-messages respectivement.

Pour créer ou modifier des beans pools, consulter Section 23.2.2, « Créer un bean pool » et Section 23.2.4, « Modifier un bean pool ».

Procédure 23.7. Allouer des beans pools pour les session beans ou pour les beans basés-message par la console de gestion.

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion »
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **`Edit`**.
4. Sélectionner le bean pool à utiliser pour chaque type de bean à partir de la combo-box appropriée.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 23.8. Allouer des beans pools pour les session beans ou pour les beans basés-message par le CLI.

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

   ```
   /subsystem=ejb3:write-attribute(name="BEANTYPE", value="BEANPOOL")
   ```

   - Remplacer *BEANTYPE* par **`default-mdb-instance-pool`** pour les beans basés-messages ou **`default-slsb-instance-pool`** pour les stateless sessions beans.
   - Remplacer *BEANPOOL* par le nom du bean pool à assigner.
3. Utiliser l'opération **`read-resource`** pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

Exemple 23.5. Assigner un bean pool pour les sessions beans par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-
attribute(name="default-slsb-instance-pool", value="LV_SLSB_POOL")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 23.6. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
   <session-bean>
      <stateless>
         <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
      </stateless>
      <stateful default-access-timeout="5000" cache-ref="simple"/>
      <singleton default-access-timeout="5000"/>
   </session-bean>
   <mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
   </mdb>


</subsystem>
```

Rapporter un bogue

23.3. CONFIGURER LES EJB THREAD POOLS

23.3.1. Enterprise Bean Thread Pools

JBoss EAP 6 maintient un certain nombre d'instances d'objets de thread Java en mémoire à utiliser par les services de beans enterprise, y compris l'invocation à distante, le service de minuteur et l'invocation asynchrone.

Cette technique s'appelle le «thread pooling». Elle fournit une meilleure performance en éliminant l'étape de création de threads et procure à l'administrateur de services un moyen de contrôler l'utilisation des ressources.

On peut créer des pools de threads multiples par divers paramètres et chaque service peut recevoir ainsi un thread de pool différent.

Rapporter un bogue

23.3.2. Créer un thread pool

Les thread pool EJB peuvent être créés par la console de gestion ou le CLI.

Procédure 23.9. Créer un thread pool EJB par la console de gestion

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion »
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur **Add**. Le dialogue **Add EJB3 Thread Pools** apparaîtra.
4. Donnez les informations requises, les valeurs de **Name**, **Max Threads**, et **Keep-Alive Timeout**.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 23.10. Créer un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:add(max-threads=MAXSIZE,
keepalive-time={"time"=>"TIME", "unit"=>UNIT"})
```

   - Remplacer *BEANPOOLNAME* par le nom requis de thread pool.
   - Remplacer *MAXSIZE* par la taille maximum de thread Pool.
   - Remplacer *UNIT* par l'unité de temps requise de «keep-alive time». Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.
   - Remplacer *TIME* par la valeur (entier relatif) du «keep-alive time». Cette valeur doit correspondre à un nombre d'unités *UNIT*.

3. Utiliser l'opération **read-resource** pour confirmer la création d'un bean pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=THREADPOOLNAME:read-
resource
```

Exemple 23.7. Créer un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=testmepool:add(max-threads=50, keepalive-time={"time"=>"150",
"unit"=>"SECONDS"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 23.8. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
```

```
    <thread-pools>
       <thread-pool name="default" max-threads="20" keepalive-
time="150"/>
    </thread-pools>

</subsystem>
```

Rapporter un bogue

23.3.3. Supprimer un thread pool

Les thread pool non utilisés peuvent être supprimés par la console de gestion.

Conditions préalables

- La thread pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Voir les tâches suivantes pour vérifier que le thread pool n'est pas en cours d'utilisation :
  - Section 23.6.2, « Configurer le Service de la minuterie EJB3 »
  - Section 23.7.2, « Configurer le thread pool du service d'invocations asynchrones EJB3 »
  - Section 23.8.2, « Configurer EJB3 Remote Service »

Procédure 23.11. Supprimer un thread pool EJB par la console de gestion

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur le thread pool que vous souhaiter supprimer.
4. Cliquer sur **`Remove`**. La boîte de dialogue **`Remove Item`** apparaîtra.
5. Cliquez sur **`OK`** pour confirmer.

Procédure 23.12. Supprimer un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **`remove`** avec la syntaxe suivante.

   ```
   /subsystem=ejb3/thread-pool=THREADPOOLNAME:remove
   ```

   - Remplacer *THREADPOOLNAME* par le nom requis de thread pool.

Exemple 23.9. Supprimer un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=ACCTS_THREADS:remove
```

```
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

23.3.4. Modifier un thread pool

Les administrateurs JBoss peuvent modifier les thread pools par la console de gestion ou le CLI.

Procédure 23.13. Modifier un thread pool par la console de gestion

1. Connectez-vous à la console de gestion. Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Thread Pools**.
3. Cliquer sur le thread pool que vous souhaiter modifier.
4. Cliquer sur **Edit**.
5. Modifier les détails que vous souhaitez modifier. Vous ne pourrez modifier que les valeurs suivantes : **Thread Factory**, **Max Threads**, **Keepalive Timeout**, et **Keepalive Timeout Unit**.
6. Cliquer sur le bouton **Save** pour terminer.

Procédure 23.14. Modifier un thread pool par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **write_attribute** avec la syntaxe suivante pour chaque attribut du thread pool à modifier.

   ```
   /subsystem=ejb3/thread-pool=THREADPOOLNAME:write-
   attribute(name="ATTRIBUTE", value="VALUE")
   ```

   - Remplacer *THREADPOOLNAME* par le nom requis de thread pool.
   - Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont **keepalive-time**, **max-thread**, et **thread-factory**.
   - Remplacer *VALUE* par la valeur requise de l'attribut.
3. Utiliser l'opération **read-resource** pour confirmer les changements au thread pool.

   ```
   /subsystem=ejb3/thread-pool=THREADPOOLNAME:read-resource
   ```

IMPORTANT

Quand vous changez la valeur de l'attribut **`keepalive-time`** par le CLI, la valeur requise correspond à une représentations d'objet. La syntaxe sera la suivante :

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:write-
attribute(name="keepalive-time", value={"time" =>
"VALUE","unit" => "UNIT"}
```

Exemple 23.10. Définir la xaleur maximum d'un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=HSThreads:write-attribute(name="max-threads", value="50")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 23.11. Définir la valeur `keepalive-time` d'un thread pool par le CLI

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=HSThreads:write-attribute(name="keepalive-time", value=
{"time"=>"150"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Rapporter un bogue

23.4. CONFIGURER LES SESSION BEANS

23.4.1. Session Bean Access Timeout

Les Stateful beans et les Singleton Session Beans ont une valeur de délai d'accès précisée pour les accès simultanés. Cette valeur correspond à la période pendant laquelle une demande de méthode de bean de session peut être bloquée avant qu'il y ait timeout.

La valeur de timeout et l'unité de temps utilisées peut être spécifiée grâce à l'annotation **`@javax.ejb.AccessTimeout`** sur la méthode. Elle peut être spécifiée sur le bean de session (qui s'applique à toutes les méthodes de beans) et sur certaines méthodes pour remplacer la configuration du bean.

Si non spécifié, JBoss EAP 6 fournit une valeur de timeout de 5000 millisecondes.

Consulter Javadocs pour AccessTimeout à l'adresse suivante :
http://docs.oracle.com/javaee/6/api/javax/ejb/AccessTimeout.html

Rapporter un bogue

23.4.2. Définir les valeurs de timeout d'accès aux beans de session par défaut

Les administrateurs de systèmes JBoss peuvent spécifier les valeurs de timeout par défaut des beans de session Stateful ou Singleton. Les valeurs de timeout par défaut peuvent être modifiées par la console de gestion ou le CLI. La valeur par défaut est de 5000 millisecondes.

Procédure 23.15. Définir les valeurs de timeout d'accès aux beans de session par défaut par la console de gestion

1. Connectez-vous à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **Edit**. Le champ de la zone **Details** est maintenant modifiable.
4. Saisir les valeurs qui conviennent dans **Stateful Access Timeout** et/ou dans les cases de texte **Singleton Access Timeout**.
5. Cliquer sur le bouton **Save** pour terminer.

Procédure 23.16. Définir les valeurs de timeout d'accès aux beans de session par le CLI.

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **write-attribute** avec la syntaxe suivante.

   ```
   /subsystem=ejb3:write-attribute(name="BEANTYPE", value=TIME)
   ```

   - Remplacer *BEANTYPE* par **default-stateful-bean-access-timeout** pour les sessions beans stateful, ou **default-singleton-bean-access-timeout** pour les sessions bean singleton.
   - Remplacer *TIME* par la valeur de timeout qui convient.
3. Utiliser l'opération **read-resource** pour confirmer les changements.

   ```
   /subsystem=ejb3:read-resource
   ```

Exemple 23.12. Définir la valeur de timeout d'accès aux beans stateful par le CLI à 9000.

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-
attribute(name="default-stateful-bean-access-timeout", value=9000)
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Exemple 23.13. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
   <session-bean>
      <stateless>
         <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
```

```
        </stateless>
        <stateful default-access-timeout="5000" cache-ref="simple"/>
        <singleton default-access-timeout="5000"/>
    </session-bean>

</subsystem>
```

Rapporter un bogue

23.5. CONFIGURER LES MESSAGE-DRIVEN BEANS

23.5.1. Définir l'Adaptateur de ressources par défaut des Beans basés-messages

Les administrateurs de systèmes JBoss peuvent spécifier l'adaptateur de ressources par défaut utilisé par les beans basés-message. L'adaptateur de ressources par défaut peut être spécifié par la Console de gestion ou le CLI. L'adaptateur de ressources fourni par défaut dans JBoss EAP 6 est **`hornetq-ra`**.

Procédure 23.17. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par la Console de gestion.

1. Connectez-vous à la Console de gestion. Section 3.4.2, « Se conncecter à la console de gestion »
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Container**.
3. Cliquer sur **`Edit`**. Le champ de la zone **`Details`** est maintenant modifiable.
4. Saisir le nom de l'adaptateur de la ressource à utiliser dans la case de texte **`Default Resource Adapter`**.
5. Cliquer sur le bouton **`Save`** pour terminer.

Procédure 23.18. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par l'interface CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

   ```
   /subsystem=ejb3:write-attribute(name="default-resource-adapter-
   name", value="RESOURCE-ADAPTER")
   ```

   Remplacer *RESOURCE-ADAPTER* par le nom de l'adaptateur de ressources à utiliser.
3. Utiliser l'opération **`read-resource`** pour confirmer les changements.

   ```
   /subsystem=ejb3:read-resource
   ```

Exemple 23.14. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI

```
[standalone@localhost:9999 subsystem=ejb3] /subsystem=ejb3:write-
attribute(name="default-resource-adapter-name", value="EDIS-RA")
{"outcome" => "success"}
[standalone@localhost:9999 subsystem=ejb3]
```

Exemple 23.15. Exemple de configuration XML

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">

   <mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
   </mdb>


</subsystem>
```

Rapporter un bogue

23.6. CONFIGURER LE SERVICE EJB3 TIMER

23.6.1. Service de minuterie EJB3

Le service de minuterie EJB3 est un service standard Java EE 6 pour programmer l'invocation de méthodes à partir de beans enterprise. Les beans de sessions Stateless, les beans de sessions Singleton et les beans basés-messages peuvent tous programmer un rappel de n'importe quelle méthode qui leur appartient à un moment précis, après un intervalle de temps, ou à un intervalle récurrent, ou encore sur la base d'un calendrier.

Rapporter un bogue

23.6.2. Configurer le Service de la minuterie EJB3

Les administrateurs JBoss peuvent configurer le Service de la minuterie EJB3 dans la Console de gestion de JBoss EAP 6. Les fonctions pouvant être configurées sont le thread pool utilisé pour l'invocation de beans programmés et le répertoire où les données du Service de la minuterie sont stockées.

Procédure 23.19. Configurer le Service du timer EJB3

1. Connectez-vous à la Console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».

2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Timer Services**.

3. Cliquer sur Edit. Les champs de la zone **`Details`** sont maintenant modifiables.

4. Vous pouvez sélectionner un Thread Pool EJB3 différent pour le Service de minuterie si les Thread Pools supplémentaires sont été configurés, et vous pouvez changer le répertoire utilisé pour sauvegarder les données du Service de minuterie. La configuration de répertoire de

données du Service de minuterie comprend deux valeurs: **Path**, le répertoire qui contient les données, et **Relative To**, le répertoire qui contient **Path**. Par défaut **Relative To** est défini à une variable de chemin de système de fichiers.

5. Cliquer sur le bouton **Save** pour terminer.

Rapporter un bogue

23.7. CONFIGURER LE SERVICE D'INVOCATION ASYNCHRONE EJB

23.7.1. Service d'invocations asynchrones EJB3

Le service d'invocations asynchrones est un service de conteneurs JavaBeans Enterprise qui gère l'invocation asynchrone des méthodes de beans de sessions. Ce service maintient un certain nombre d'invocations asynchrones configurables (Thread Pool) qui sont allouées pour l'exécution de méthodes asynchrones.

Enterprise JavaBeans 3.1 permet à toute méthode de bean de session (stateful, stateless, ou singleton) d'être annotée pour permettre l'exécution asynchrone.

Rapporter un bogue

23.7.2. Configurer le thread pool du service d'invocations asynchrones EJB3

Les administrateurs JBoss peuvent configurer le service d'invocations asynchrones EJB3 dans la console de gestion JBoss EAP 6 pour permettre l'utilisation d'un thread pool spécifique.

Procédure 23.20. Configurerhttp://francegourmet.com.au/product-category/snails-and-mushrooms/

1. Connectez-vous à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Async Service**.
3. Cliquer sur **Edit**.
4. Sélectionner le thread pool à utiliser à partir de la liste. Le thread pool devra déjà avoir été créé.
5. Cliquer sur le bouton **Save** pour terminer.

Rapporter un bogue

23.8. CONFIGURER EJB3 REMOTE INVOCATION SERVICE

23.8.1. EJB3 Remote Service

Le service EJB3 Remote gère l'exécution à distance des Beans Enterprise dans les interfaces commerciales à distance.

Rapporter un bogue

23.8.2. Configurer EJB3 Remote Service

Les administrateurs JBoss peuvent configurer EJB3 Remote Service dans la console de gestion de JBoss EAP 6. Les fonctions pouvant être configurées sont le thread pool utilisé pour l'invocation de beans programmés et le connecteur sur lequel le réseau EJB3 Remoting est enregistré.

Procédure 23.21. Configurer EJB3 Remote Service

1. Connectez-vous à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Container** et sélectionner **EJB 3**. Sélectionner l'onglet **Services**, cliquer sur **Remote Service**.
3. Cliquer sur **`Edit`**.
4. Vous pouvez sélectionner un EJB3 Thread Pool différent pour Remote Service si les Thread Pools supplémentaires ont été configurés. Vous pouvez changer le connecteur utilisé pour enregistrer le Canal EJB Remoting.
5. Cliquer sur le bouton **`Save`** pour terminer.

Rapporter un bogue

23.9. CONFIGURER LES EJB 2.X ENTITY BEANS

23.9.1. EJB Entity Beans

Les EJB Entity Beans sont un type de bean enterprise de la version 2.x de la spécification EJB qui représentait des données persistantes maintenues dans une base de données. Les entity beans ont été remplacées par les entités JPA et ont été officiellement listées pour être retirées (nettoyées) des versions futures de la spécification. Red Hat ne recommande pas l'utilisation des entity beans, sauf pour raison de compatibilité rétro-active.

Le support des entity beans est désactivé par défaut dans JBoss EAP 6.

Rapporter un bogue

23.9.2. Container-Managed Persistence

Container-Managed Persistence (CMP) est un service fourni par un serveur d'applications qui procure la persistance des données pour les beans entity.

Rapporter un bogue

23.9.3. Activer EJB 2.x Container-Managed Persistence

Container-Managed Persistence (CMP) est géré par l'extension **`org.jboss.as.cmp`**. CMP est activé par défaut dans le domaine géré et dans la configuration complète du serveur autonome, par ex. **`standalone-full.xml`**.

Pour activer CMP dans une configuration différente, ajouter le module **`org.jboss.as.cmp`** à la liste d'extensions actives dans le fichier de configuration du serveur.

```
<extensions>
        <extension module="org.jboss.as.cmp"/>
</extensions>
```

Une fois que l'extension est ajoutée, vous devrez aussi ajouter l'élément suivant au fichier de configuration XML du profil sous l'élément <profile>.

```
<subsystem xmlns="urn:jboss:domain:cmp:1.1"/>
```

Pour désactiver CMP dans une configuration de serveur, supprimer l'entrée d'extension du module **`org.jboss.as.cmp`**.

Rapporter un bogue

23.9.4. Configurer EJB 2.x Container-Managed Persistence

Le sous-système EJB 2.x Container Managed Persistence (CMP) peut être configuré pour spécifier un certain nombre de générateurs de clés. Les générateurs de clés sont utilisés pour produire des clés uniques pour identifier chaque entité persistée par le service CMP.

Il existe deux types de générateurs de clés : les générateurs de clés basés-UUID et les générateurs de clés HiLO

Les générateurs de clés basés-UUID

Un générateur de clés basé-UUID crée des clés qui utilisent un Identifiant Unique Universel (UUI). Les générateurs de clés UUID ont besoin uniquement d'avoir un nom unique; ils n'ont pas d'autre configuration.

Les générateurs de clés basé-UUID peuvent être ajoutés par le CLI avec la syntaxe suivante.

```
/subsystem=cmp/uuid-keygenerator=UNIQUE_NAME:add
```

Exemple 23.16. Ajouter le générateur de clés UUID

Pour ajouter un générateur de clés basé-UUID ayant pour nom **`uuid_identities`**, utiliser cette commande CLI :

```
/subsystem=cmp/uuid-keygenerator=uuid_identities:add
```

La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.0">
   <key-generators>
      <uuid name="uuid_identities" />
   </key-generators>
</subsystem>
```

Générateurs de clés HiLo

Les générateurs de clés HiLo utilisent une base de données pour créer et stocker des clés d'identité des entités. Le générateur de clés HiLo doivent posséder des noms uniques et sont configurés avec des propriétés qui indiquent la source de données utilisée pour stocker les données, ainsi que les

noms du tableau et colonnes qui stockent les clés.

Les générateurs de clés HiLo peuvent être ajoutés par le CLI grâce à la syntaxe de commande suivante:

```
/subsystem=cmp/hilo-keygenerator=UNIQUE_NAME/:add(property=value,
property=value, ...)
```

Exemple 23.17. Ajouter un générateur de clés HiLo

```
/subsystem=cmp/hilo-keygenerator=HiLoKeyGeneratorFactory:add(create-
table=true,create-table-ddl="create table HILOSEQUENCES (SEQUENCENAME
varchar(50) not null, HIGHVALUES integer not null, constraint hilo_pk
primary key (SEQUENCENAME))",data-
source=java:jboss/datasources/ExampleDS, id-
column=HIGHVALUES,sequence-column=SEQUENCENAME,table-
name=HILOSEQUENCES,sequence-name=general,block-size=10)
```

La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.1">
     <key-generators>
         <hilo name="HiLoKeyGeneratorFactory">
             <block-size>10</block-size>
             <create-table>true</create-table>
             <create-table-ddl>create table HILOSEQUENCES
(SEQUENCENAME varchar(50) not null, HIGHVALUES integer not null,
constraint hilo_pk primary key (SEQUENCENAME))</create-table-ddl>
             <data-source>java:jboss/datasources/ExampleDS</data-
source>
             <id-column>HIGHVALUES</id-column>
             <sequence-column>SEQUENCENAME</sequence-column>
             <sequence-name>general</sequence-name>
             <table-name>HILOSEQUENCES</table-name>
         </hilo>
     </key-generators>
 </subsystem>
```

NOTE

La taille de bloc doit être sur une valeur! = 0, sinon la clé PKey générée ne sera pas incrémentée et, par conséquent, la création d'entités échouera accompagnée de l'exception DuplicateKeyException.

NOTE

Le select-hi-ddl doit être défini avec 'FOR UPDATE' en cas de cluster pour veiller à l'homogénéité. Toutes les bases de données ne prennent pas en charge la fonctionnalité de verrouillage.

Rapporter un bogue

23.9.5. Les propriétés de sous-système CMP pour les générateurs de clés HiLo

Tableau 23.1. Les propriétés de sous-système CMP pour les générateurs de clés HiLo

Propriété	Type des données	Description
`block-size`	long	La taille du bloc.
`create-table`	booléen	Si défini sur **TRUE**, le tableau **`table-name`** sera créé avec le contenu **`create-table-ddl`** si le tableau n'est pas trouvé.
`create-table-ddl`	chaîne	Les commandes DDL utilisées pour créer le tableau spécifié dans **`table-name`** si on ne trouve pas le tableau et **`create-table`** est défini à **TRUE**.
`data-source`	token	La source de données utilisée pour se connecter à la base de données.
`drop-table`	booléen	Pour déterminer si on doit supprimer les tableaux.
`id-column`	token	Le nom de la colonne ID.
`select-hi-ddl`	chaîne	La commande SQL qui retournera la plus grande clé actuellement stockée.
`sequence-column`	token	Le nom de la colonne de séquences.
`sequence-name`	token	Le nom de la séquence.
`table-name`	token	Nom de la table utilisée pour stocker les informations sur les clés

Rapporter un bogue

CHAPITRE 24. JAVA CONNECTOR ARCHITECTURE (JCA)

24.1. INTRODUCTION

24.1.1. Java EE Connector API (JCA)

JBoss EAP 6 fournit un support complet à la spécification Java EE Connector API (JCA). Voir JSR 322: Java EE Connector Architecture 1.6 pour obtenir plus d'informations sur la spécification JCA.

Un adaptateur de ressources est un composant qui implémente l'architecture de Java EE Connector API. Il ressemble à un objet de source de données, mais fournit une connectivité à partir d'EIS (Enterprise Information System) vers un grand nombre de systèmes hétérogènes, comme des bases de données, systèmes de messagerie, traitement de transactions, et systèmes ERP (Enterprise Resource Planning).

Rapporter un bogue

24.1.2. Java Connector Architecture (JCA)

La Java EE Connector Architecture (JCA) définit une architecture standard pour les systèmes de Java EE pour les systèmes externes hétérogènes Enterprise Information Systems (EIS). Exemples de systèmes EISs : Enterprise Resource Planning (ERP), transaction central de traitement (TP), bases de données et systèmes de messagerie.

JCA 1.6 fournit des fonctionnalités de gestion :

- connections
- transactions
- sécurité
- cycle de vie
- Instances de travail
- Flux interne de transactions
- Flux interne de messages

JCA 1.6 a été développé en tant que Java Community Process JSR-322, http://jcp.org/en/jsr/detail?id=313.

Rapporter un bogue

24.1.3. Adaptateurs de ressources

Un adaptateur de ressources est un composant Java EE déployable qui permet la communication entre une application Java EE et une entreprise d'informations système (EIE) à l'aide de la spécification Java Connector Architecture (JCA). Un adaptateur de ressources est souvent fourni par les fournisseurs de l'EIE pour permettre une intégration facile de leurs produits aux applications Java EE.

Un système d'information Enterprise peut être n'importe quel autre système de logiciel au sein d'une organisation. Les exemples incluent les systèmes ERP (Enterprise Resource Planning), les systèmes de base de données, les serveurs d'e-mails et les systèmes de messagerie propriétaires.

Un adaptateur de ressources est empaqueté dans un fichier de Ressources Adaptateur Archive (RAR) qui peut être déployé dans JBoss EAP 6. Un fichier RAR peut également être inclus dans un déploiement Enterprise Archive (EAR).

Rapporter un bogue

24.2. CONFIGURATION DU SOUS-SYSTÈME JAVA CONNECTOR ARCHITECTURE (JCA)

Le sous-système JCA du fichier de configuration de JBoss EAP 6 contrôle les paramètres de configuration généraux du conteneur JCA et déploiements d'adaptateurs de ressources.

Éléments clés du sous-système JCA

Validation d'archive

- Ce paramétrage indique si la validation d'archivage doit avoir lieu sur les unités de déploiement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour la validation d'archivage.

Tableau 24.1. Attributs de validation d'archivage

Attribut	Valeur par défaut	Description
`enabled`	true	Indique si la validation d'archivage est activée.
`fail-on-error`	true	Indique si un rapport d'erreur de validation d'archivage a fait échouer le développement.
`fail-on-warn`	false	Indique si un rapport d'avertissement de validation d'archivage a fait échouer le développement.

- Si une archive n'implémente pas la spécification Java EE Connector Architecture correctement, et que la validation d'archivage est activée, un message d'erreur s'affichera pendant le déploiement pour décrire le problème, comme par exemple :

```
Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public int
hashCode()" method.
Code: com.mycompany.myproject.ResourceAdapterImpl

Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public boolean
equals(Object)" method.
Code: com.mycompany.myproject.ResourceAdapterImpl
```

- Si la validation d'archivage n'est pas spécifiée, on la considérera comme présente et l'attribut **`enabled`** aura comme valeur true par défaut.

Validation de bean

- Ce paramètre indique si la validation de bean (JSR-303) aura lieu sur les unités de déploiement.
- Le tableau ci-dessous décrit les attributs que vous pouvez déterminer pour la validation de bean.

Tableau 24.2. Attributs de validation de bean

Attribut	Valeur par défaut	Description
`enabled`	true	Indique si la validation de bean est activée.

- Si la validation de bean n'est pas spécifiée, on la considérera comme présente et l'attribut **`enabled`** aura comme valeur true par défaut.

Work Managers

- Il y a deux types de Work Managers :

  Work Manager par défaut

  Le Work Manager par défaut et ses Thread Pools.

  Work Manager personnalisé

  Une définition de Work Manager et ses Thread Pools.

- Le tableau suivant décrit les attributs que vous pouvez définir pour les Work Managers.

Tableau 24.3. Attributs de Work Managers

Attribut	Description
`name`	Indique le nom du Work Manager. Requis pour les Work Managers personnalisés.
`short-running-threads`	Thread Pool pour les instances Work standards. Chaque Work Manager a un Thread Pool à exécution courte.
`long-running-threads`	Les instances Work de Thread pool de JCA 1.6 qui définissent **`LONG_RUNNING`**. Chaque Work Manager peut avoir un Thread pool de longue durée en option.

- Le tableau ci-dessous décrit les attributs que vous pouvez définir pour les Thread pools de Work Managers.

Tableau 24.4. Attributs de Thread pool

Attribut	Description
`allow-core-timeout`	Paramètre booléen qui détermine quels threads principaux risquent d'expirer. La valeur par défaut est false.
`core-threads`	La taille du pool de threads. Doit être inférieure à la taille de pool de threads maximum.
`queue-length`	La longueur maximum de la file d'attente.
`max-thread`	Taille de pool de threads maximum.
`keepalive-time`	Indique le durée pendant laquelle les threads de pool doivent être conservés après avoir complété leur tâche.
`thread-factory`	Référence à la fabrique de threads.

Bootstrap Contexts

- Utilisé pour définir les contextes de bootstapping (démarrage) personnalisés.
- Le tableau suivant décrit les attributs à définir pour les contextes de bootstraping.

Tableau 24.5. Attributs de contexte de bootstrapping

Attribut	Description
`name`	Indique le nom du contexte de bootstrapping
`workmanager`	Indique le nom du Work Manager à utiliser dans ce contexte.

Gestionnaire de connexion mis en cache

- Utilisé pour déboguer les connexions et pour supporter l'inscription tardive d'une connexion dans une transaction, pour vérifier leur bonne utilisation et fonctionnement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour le manager de connexions mis en cache.

Tableau 24.6. Attributs de manager de connexion mis en cache

Attribut	Valeur par défaut	Description
`debug`	false	Sorties avertissement en cas d'échec de fermeture explicite des connexions
`error`	false	Envoie une exception en cas d'échec de fermeture explicite des connexions.

Procédure 24.1. Configurer le sous-système JCA par la console de management

Le sous-système JCA de JBoss EAP 6 peut être configuré dans la console de gestion. Les options de configuration de JCA sont situées dans des endroits légèrement différents dans la console de gestion, selon la façon dont le serveur est exécuté.

1. Cliquer sur l'onglet **Configuration** en haut de l'écran. Étendre le menu **Connector** et sélectionner **JCA**.

2. Si le serveur exécute en mode de domaine, sélectionner un profil à partir du menu déroulant **`Profile`** en haut et à gauche.

3. Configurer les paramètres du sous-système JCA à l'aide des trois onglets.

   a. **Config courante**
   L'onglet **`Common Config`** contient des paramètres pour le gestionnaire de connexions en cache, la validation de l'archive et la validation de bean (JSR-303). Chacun d'entre eux est contenu dans son onglet propre. Ces réglages peuvent être changés en ouvrant l'onglet correspondant, en cliquant sur le bouton Edit, en effectuant les changements nécessaires et puis en cliquant sur le bouton Save de sauvegarde.

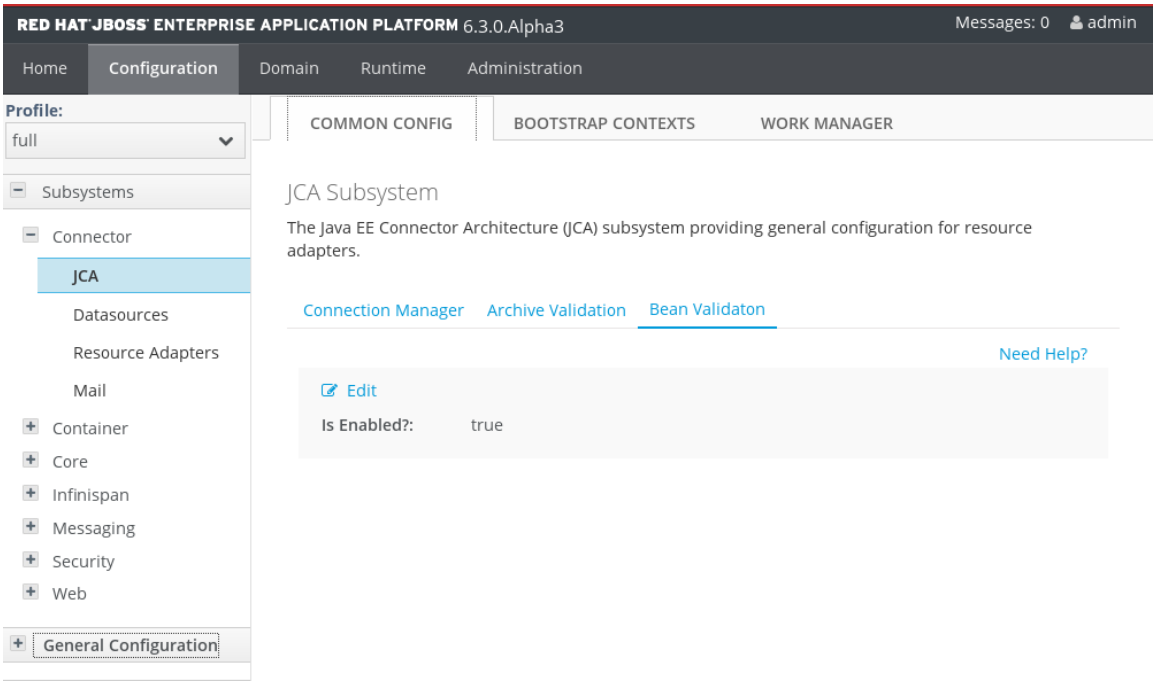


Figure 24.1. Configuration commune JCA

   b. **Work Managers**
   L'onglet **`Work Manager`** contient la liste des Work Managers (gestionnaires de travail) configurés. Les nouveaux Work Managers peuvent être ajoutés, supprimés, et leurs pools de threads configurés ici. Chaque Work Manager peut avoir un pool de threads à exécution courte et un pool de threads de longue durée en option.

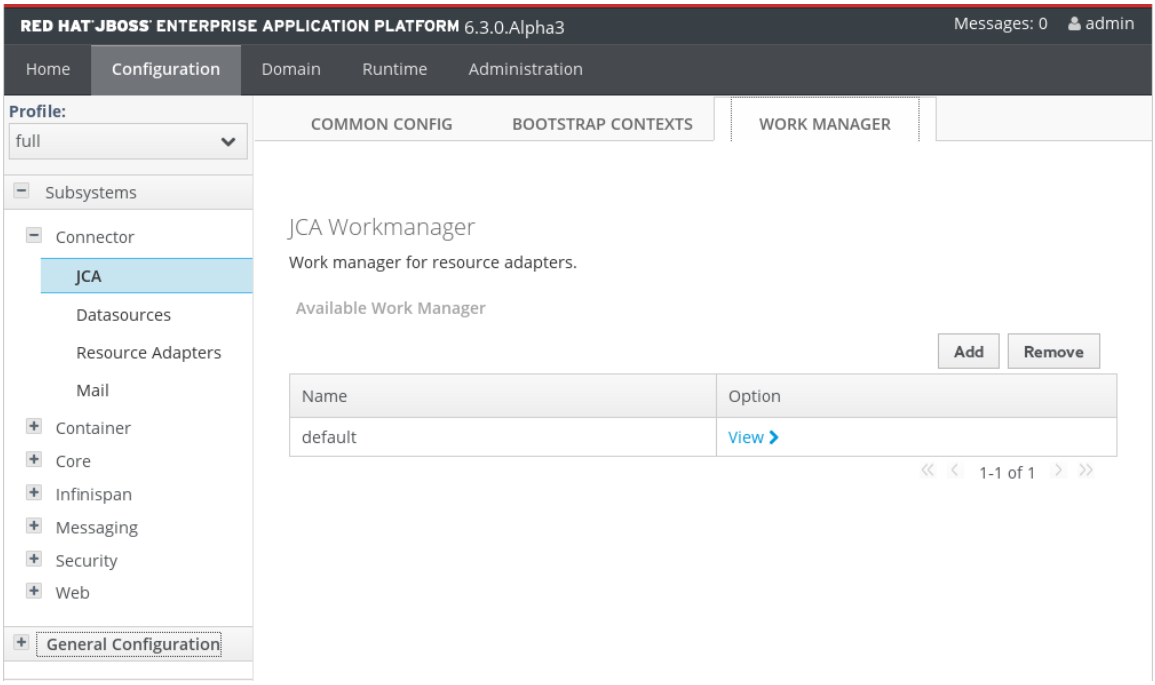


Figure 24.2. Work Managers

Les attributs de thread pool peuvent être configurés en cliquant sur **`View`** sur l'adaptateur de ressources sélectionnées.

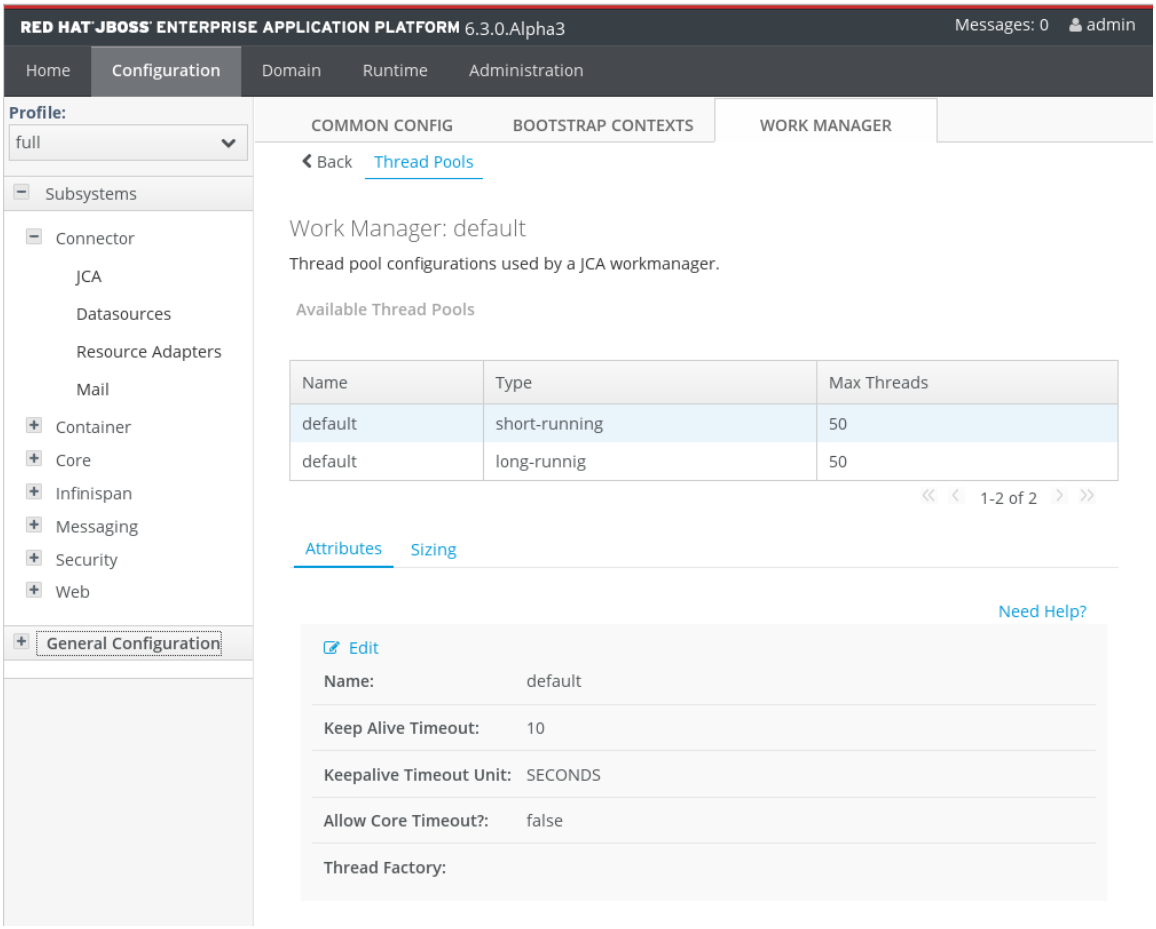


Figure 24.3. Work Manager Thread Pools

c. **Bootstrap Contexts**

L'onglet **`Bootstrap Contexts`** contient la liste des contextes d'amorçage configurés. De nouveaux objets de contexte d'amorçage peuvent être ajoutés, supprimés ou configurés. Un Work Manager doit être assigné à chaque contexte de Bootstrap.

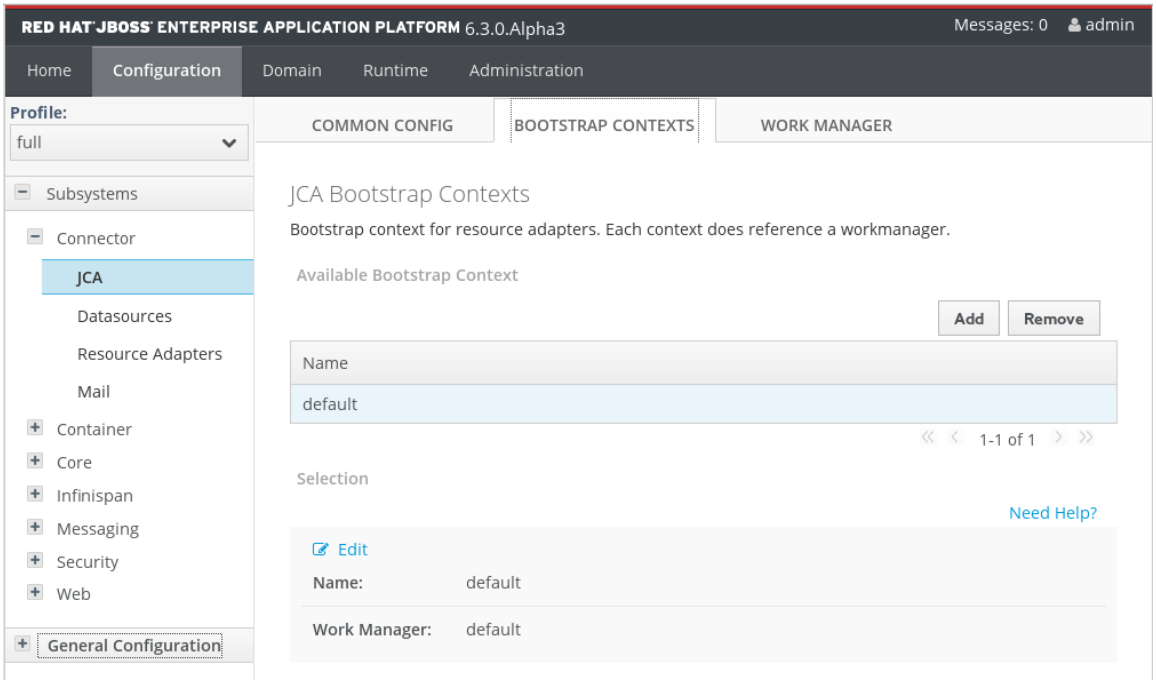


Figure 24.4. Bootstrap Contexts

Rapporter un bogue

24.3. DÉPLOYER UN ADAPTATEUR DE RESSOURCES

Les adaptateurs de ressources peuvent être déployés dans JBoss EAP 6 à l'aide du Management CLI, de la console de management basée-web, ou en copiant manuellement les fichiers. Le processus est le même que celui d'autres artefacts déployables.

Procédure 24.2. Déployer un adaptateur de ressources par la Management CLI

1. Ouvrir une invite de commande de votre système d'exploitation.
2. Connectez-vous au Management CLI.
   - Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

     ```
     $ EAP_HOME/bin/jboss-cli.sh --connect
     $ Connected to standalone controller at localhost:9999
     ```

   - Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

     ```
     C:\>EAP_HOME\bin\jboss-cli.bat --connect
     C:\> Connected to standalone controller at localhost:9999
     ```

3. Déployer l'adaptateur de ressources.
   - Pour déployer l'adaptateur de ressources dans un serveur autonome, saisir ce qui suit dans une ligne de commande :

```
$ deploy path/to/resource-adapter-name.rar
```

- Pour déployer l'adaptateur de ressources dans tous les serveurs d'un domaine géré, saisir ce qui suit dans une ligne de commande :

```
$ deploy path/to/resource-adapter-name.rar --all-server-groups
```

Procédure 24.3. Déployer un adaptateur de ressources par la console de gestion

1. Connectez-vous à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».
2. Cliquer sur l'onglet **Runtime** qui se trouve en haut de l'écran. Sélectionner **Manage Deployments**. Cliquer sur **Add**.
3. Naviguer dans l'archive d'adaptateur de ressources et le sélectionner. Puis, cliquer sur le bouton **Next**.
4. Vérifier les noms de déploiement, puis cliquer sur le bouton **Save**.
5. L'archive d'adaptateur de ressources doit maintenant apparaître dans la liste dans un état de désactivation.
6. Activer l'adaptateur de ressources.
   - Dans le mode de domaine, cliquer sur **Assign**. Sélectionner à quels groupes de serveurs il faut assigner l'adaptateur de ressouces. Cliquer sur **Save** pour terminer.
   - En mode autonome, sélectionner le composant d'application de la liste. Cliquer sur **En/Disable**. Cliquer sur **Confirm** après **Are You Sure?** pour activer le composant.

Procédure 24.4. Déployer un adaptateur de ressources manuellement

- Copier l'archive d'adaptateur de ressources dans le répertoire des déploiements de serveur,
  - Pour un serveur autonome, copier l'archive d'adptateur de ressources dans le répertoire ***EAP_HOME*/standalone/deployments/**.
  - Dans un domaine géré, vous devez utliliser la console de gestion ou le CLI pour déployer l'archive d'adaptateur de ressources dans les groupes de serveurs.

Rapporter un bogue

24.4. CONFIGURATION D'UN ADAPTATEUR DE RESSOURCES DÉPLOYÉES

Les administrateurs JBoss peuvent configurer les adaptateurs de ressources pour JBoss EAP 6 à l'aide du Management CLI, de la console de management basée-web, ou en modifiant manuellement la configuration des fichiers.

Voir le document du fournisseur pour votre adaptateur de ressources pour obtenir des informations sur les propriétés prises en charge et autres informations.

NOTE

Dans la procédure suivante, la ligne de commande que vous devez saisir suit l'invite suivante **`[standalone@localhost:9999 /]`**. Ne pas saisir le texte qui se trouve à l'intérieur des accolades. Voici la sortie que vous devriez apercevoir comme résultat, ainsi, **`{"outcome" => "success"}`**.

Procédure 24.5. Configurer un adaptateur de ressources par l'interface CLI

1. Ouvrir une invite de commande de votre système d'exploitation.
2. Connectez-vous au Management CLI.
   - Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

     ```
     $ EAP_HOME/bin/jboss-cli.sh --connect
     ```

     Vous devriez voir le résultat de sortie suivant :

     ```
     $ Connected to standalone controller at localhost:9999
     ```

   - Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

     ```
     C:\>EAP_HOME\bin\jboss-cli.bat --connect
     ```

     Vous devriez voir le résultat de sortie suivant :

     ```
     C:\> Connected to standalone controller at localhost:9999
     ```

3. Ajouter la configuration d'adaptateur de ressource.

   ```
   [standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
   adapter=eis.rar:add(archive=eis.rar, transaction-
   support=XATransaction)
   {"outcome" => "success"}
   ```

4. Configurer la <config-property> **`server`** niveau adaptateur de ressources.

   ```
   [standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
   adapter=eis.rar/config-properties=server/:add(value=localhost)
   {"outcome" => "success"}
   ```

5. Configurer la <config-property>**`port`** niveau adaptateur de ressources

   ```
   [standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
   adapter=eis.rar/config-properties=port/:add(value=9000)
   {"outcome" => "success"}
   ```

6. Ajouter une définition de connexion à la fabrique de connexions gérées.

   ```
   [standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
   adapter=eis.rar/connection-definitions=cfName:add(class-
   ```

```
name=com.acme.eis.ra.EISManagedConnectionFactory, jndi-
name=java:/eis/AcmeConnectionFactory)
{"outcome" => "success"}
```

7. Configurer <config-property> **`port`** niveau usine de connexions gérées.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/connection-definitions=cfName/config-
properties=name/:add(value=Acme Inc)
{"outcome" => "success"}
```

8. Ajouter un objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/admin-objects=aoName:add(class-
name=com.acme.eis.ra.EISAdminObjectImpl, jndi-
name=java:/eis/AcmeAdminObject)
{"outcome" => "success"}
```

9. Configurer la propriété **`threshold`** de l'objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/admin-objects=aoName/config-
properties=threshold/:add(value=10)
{"outcome" => "success"}
```

10. Activer l'adaptateur de ressource.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar:activate
{"outcome" => "success"}
```

11. Voir l'adaptateur de ressources nouvellement configuré et activé.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar:read-resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "archive" => "eis.rar",
        "beanvalidationgroups" => undefined,
        "bootstrap-context" => undefined,
        "transaction-support" => "XATransaction",
        "admin-objects" => {"aoName" => {
            "class-name" => "com.acme.eis.ra.EISAdminObjectImpl",
            "enabled" => true,
            "jndi-name" => "java:/eis/AcmeAdminObject",
            "use-java-context" => true,
            "config-properties" => {"threshold" => {"value" => 10}}
        }},
        "config-properties" => {
            "server" => {"value" => "localhost"},
            "port" => {"value" => 9000}
```

```
        },
        "connection-definitions" => {"cfName" => {
            "allocation-retry" => undefined,
            "allocation-retry-wait-millis" => undefined,
            "background-validation" => false,
            "background-validation-millis" => undefined,
            "blocking-timeout-wait-millis" => undefined,
            "class-name" =>
"com.acme.eis.ra.EISManagedConnectionFactory",
            "enabled" => true,
            "flush-strategy" => "FailingConnectionOnly",
            "idle-timeout-minutes" => undefined,
            "interleaving" => false,
            "jndi-name" => "java:/eis/AcmeConnectionFactory",
            "max-pool-size" => 20,
            "min-pool-size" => 0,
            "no-recovery" => undefined,
            "no-tx-separate-pool" => false,
            "pad-xid" => false,
            "pool-prefill" => false,
            "pool-use-strict-min" => false,
            "recovery-password" => undefined,
            "recovery-plugin-class-name" => undefined,
            "recovery-plugin-properties" => undefined,
            "recovery-security-domain" => undefined,
            "recovery-username" => undefined,
            "same-rm-override" => undefined,
            "security-application" => undefined,
            "security-domain" => undefined,
            "security-domain-and-application" => undefined,
            "use-ccm" => true,
            "use-fast-fail" => false,
            "use-java-context" => true,
            "use-try-lock" => undefined,
            "wrap-xa-resource" => true,
            "xa-resource-timeout" => undefined,
            "config-properties" => {"name" => {"value" => "Acme
Inc"}}
        }}
    }
}
```

Procédure 24.6. Configurer un adaptateur de ressources par la console de management basée-web

1. Connectez-vous à la console de gestion. Voir Section 3.4.2, « Se conncecter à la console de gestion ».

2. Cliquer sur l'onglet **`Configuration`** en haut de l'écran. Étendre le menu **Connectors** et sélectionner **`Resource Adapters`**.

   a. En mode de domaine, sélectionner **`Profile`** à partir du menu déroulant qui se trouve en haut et à gauche.

   Cliquer sur **`Ajouter`**.

3. Saisir le nom de l'archive et choisir le type de transaction **XATransaction** à partir du menu déroulant **TX:**. Ensuite, cliquer sur **Save**.

4. Sélectionner l'onglet **Properties**. Cliquer sur **Add**.

5. Saisir le **serveur** pour le **Name** (nom) et le nom d'ĥôte, par exemple **localhost**, pour la valeur **Value**. Puis cliquer sur **Save** pour terminer.

6. Cliquer sur **Add** à nouveau. Saisir le **port** pour le **Name** (nom) et le nom dde port, par exemple **9000**, pour la valeur **Value**. Puis cliquer sur **Save** pour terminer.

7. Les propriétés **server** et **port** apparaissent maintenant dans le panneau **Properties**. Cliquer sur le lien **View** (Vue) sous la colonne **Option** pour l'adaptateur de ressources listées pour visualiser les définitions de connexion or **Connection Definitions**.

8. Cliquer sur **Add** qui se trouve au dessus du tableau **Available Connection Definitions** pour ajouter une définition de connexion.

9. Saisir le **JNDI Name** et le nom de classe complet de la **Connection Class**. Puis cliquer sur **Save** pour terminer.

10. Sélectionner la nouvelle définition de connexion, puis sélectionner l'onglet **Properties**. Cliquer sur le bouton **Add** pour saisir les données de **Key** et **Value** pour cette définition de connexion. Cliquer sur **Save** pour terminer.

11. La définition de connexion est terminée, mais non activée. Sélectionner la définition de connexion et cliquer sur le bouton **Enable** pour activer la définition de connexion.

12. Un dialogue vous demande **Souhaitez-vous réellement modifier la définition de connexion?** du nom JNDI. Cliquer sur **Confirm**. La définition de connexion devrait maintenant afficher **Enabled** (activée).

13. Cliquer sur l'onglet **Admin Objects** qui se trouve dans la partie supérieure de la page pour créer et configurer des objets admin. Puis, cliquer sur **Add**.

14. Saisir le **JNDI Name** et le nom de classe **Class Name** complet de l'objet admin. Puis cliquer sur **Save**.

15. Sélectionner l'onglet **Properties**, puis cliquer sur **Add** pour ajouter des propriétés d'objet admin.

16. Saisir une propriété de configuration d'objet admin, comme par exemple la limite **threshold**, dans le champ **Name** (nom). Saisir la valeur de la propriété de configuration, comme par exemple **10**, pour la valeur **Value**. Puis cliquer sur **Save** pour sauvegarder la propriété.

17. L'objet admin est maintenant complété, mais non actif. CLiquer sur **Enable** pour activer l'objet admin.

18. Un dialogue vous demande **Souhaitez-vous réellement modifier l'Objet admin?** du nom JNDI. Cliquer sur **Confirm**. L'objet admin devrait maintenant afficher **Enabled** (activé).

19. Vous devez charger à nouveau la configuration du serveur pour terminer ce processus. Cliquer sur le lien **Runtime**. Étendre le menu **Server**. Sélectionner **Overview** dans le panneau de navigation de gauche.

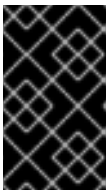
a. Charger à nouveau les serveurs

- En mode de domaine, faire glisser le curseur sur le groupe de serveurs. Sélectionner **`Restart Group`**.
- En mode standalone, il y aura un bouton **`Reload`** disponible. Cliquer sur **`Reload`**.

20. Un dialogue vous demande **`Souhaitez-vous charger à nouveau la configuration du serveur ?`** pour le serveur indiqué. Cliquer sur **`Confirm`**. La configuration du serveur sera à jour.

Procédure 24.7. Configurer un adaptateur de ressources manuellement

1. Stopper le serveur JBoss EAP 6.



IMPORTANT

Vous devez interrompre le serveur avant de modifier le fichier de configuration du serveur pour que votre changement puisse être persisté au redémarrage du serveur.

2. Ouvrir le fichier de configuration du serveur pour l'éditing.
   - Pour les serveurs autonomes, il s'agit du fichier ***`EAP_HOME`*`/standalone/configuration/standalone.xml`**.
   - Si vous exécutez dans un domaine géré, il s'agira du fichier ***`EAP_HOME`*`/domain/configuration/domain.xml`**.
3. Chercher le sous-système **`urn:jboss:domain:resource-adapters`** dans le fichier de configuration.
4. Il n'y a pas d'adaptateurs de ressources définis pour ce système. Veuillez commencer par remplacer :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1"/>
```

   par ceci :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">
    <resource-adapters>
        <!-- <resource-adapter> configuration listed below -->
    </resource-adapters>
</subsystem>
```

5. Remplacer la configuration **`<!-- <resource-adapter> listée ci-dessous -->`** par la définition XML de l'adaptateur de ressources. Ce qui suit représente la représentation XML de la configuration de l'adaptateur de ressources créé par l'interface CLI et la console de management basée-web décrite ci-dessus.

```
<resource-adapter>
    <archive>
        eis.rar
    </archive>
    <transaction-support>XATransaction</transaction-support>
    <config-property name="server">
        localhost
    </config-property>
    <config-property name="port">
        9000
    </config-property>
    <connection-definitions>
        <connection-definition class-
name="com.acme.eis.ra.EISManagedConnectionFactory"
                jndi-name="java:/eis/AcmeConnectionFactory"
                pool-name="java:/eis/AcmeConnectionFactory">
            <config-property name="name">
                Acme Inc
            </config-property>
        </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object class-
name="com.acme.eis.ra.EISAdminObjectImpl"
                jndi-name="java:/eis/AcmeAdminObject"
                pool-name="java:/eis/AcmeAdminObject">
            <config-property name="threshold">
                10
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>
```

6. **Démarrer le serveur**
   Lancer à nouveau le serveur JBoss EAP 6 pour qu'il commence à exécuter avec la nouvelle configuration.

Rapporter un bogue

24.5. RÉFÉRENCE DE DESCRIPTION D'ADAPTATEUR DE RESSOURCES

Les tableaux suivants décrivent les éléments de description d'adaptateurs de ressources.

Tableau 24.7. Éléments principaux

Élément	Description
`bean-validation-groups`	Indique le groupe de validation du bean qui doit être utilisé

Élément	Description
bootstrap-context	Indique le nom unique du contexte de bootstrapping qui doit être utilisé
config-property	Config-property spécifie les propriétés de configuration de l'adaptateur de ressources.
transaction-support	Indique le type de transactions pris en charge par l'adaptateur de ressources. La valeurs valides sont : **NoTransaction**, **LocalTransaction**, **XATransaction**
connection-definitions	Indique les définitions de connexion
admin-objects	Indique les objets d'administration

Tableau 24.8. Éléments de groupes de validation de beans

Élément	Description
bean-validation-group	Indique le nom de classe complet d'un groupe de validation de beans devant être utilisé pour la validation.

Tableau 24.9. Définition de connexion / attributs d'objets admin

Attribut	Description
class-name	Indique le nom de classe complet d'une usine de connexions gérée ou d'un objet admin
jndi-name	Indique le nom JNDI
enabled	L'objet doit-il être activé ?
use-java-context	Indique si on doit utiliser un contexte **java:/** JNDI
pool-name	Indique le nom de pool de l'objet
use-ccm	Active le gestionnaire de connexion mis en cache

Tableau 24.10. Éléments de définition de connexion

Élément	Description
config-property	Config-property spécifie les propriétés de configuration de l'usine de connexions.

Élément	Description
pool	Indique les paramètres de pooling
xa-pool	Indique les paramètres de pooling XA
security	Indique les paramètres de sécurité
timeout	Indique les paramètres de timeout
validation	Indique les paramètres de validation
recovery	Indique les paramètres de recouvrement XA

Tableau 24.11. Éléments de pooling

Élément	Description
min-pool-size	L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le sujet de la demande de connexion. La valeur par défaut est 0
max-pool-size	L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut est à **20**.
prefill	Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est **false**.
use-strict-min	Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est **false**.
flush-strategy	Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont : **FailingConnectionOnly** (default), **IdleConnections**, **EntirePool**

Tableau 24.12. Éléments de pool XA

Élément	Description
min-pool-size	L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le sujet de la demande de connexion. Cette valeur par défaut à **0**

Élément	Description
max-pool-size	L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut est à **20**.
prefill	Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est **false**.
use-strict-min	Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est **false**.
flush-strategy	Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont : **FailingConnectionOnly** (default), **IdleConnections**, **EntirePool**
is-same-rm-override	L'élément is-same-rm-override element permet de définir inconditionnellement si javax.transaction.xa.XAResource.isSameRM(XAResource) doit renvoyer **true** or **false**
interleaving	Élément qui permet ou non l'entrelacement des usines de connexions XA
no-tx-separate-pools	Oracle n'aime pas que les connexions XA soient utilisées à la fois à l'intérieur et à l'extérieur d'une connexion JTA. Pour résoudre ce problème, vous pourrez créer des sub-pools pour ces contextes différents.
pad-xid	Est-ce que le Xid doit être mis en tampon ?
wrap-xa-resource	Est-ce que les instances XAResource doivent être encapsulées dans une instance org.jboss.tm.XAResourceWrapper

Tableau 24.13. Éléments de sécurité

Élément	Description
application	Indique si les paramètres de sécurité fournis, (comme par exemple, à partir de **getConnection(user, pw)**, sont utilisés pour distinguer les connexions d'un pool.
security-domain	Indique si des sujets (de domaine de sécurité) sont utilisés pour distinguer les connexions d'un pool. Le contenu du domaine de sécurité correspond au nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom est en corrélation à l'attribut application-policy/name du descripteur JAAS **login-config.xml**.

Élément	Description
security-domain-and-application	Indique que les paramètres de l'application fournis (par exemple, à partir de **getConnection(user, pw)**) ou que le sujet (du domaine de la sécurité) soient utilisés pour distinguer les connexions du pool. Le contenu du domaine de sécurité est le nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom est en corrélation à l'attribut application-policy/name du descripteur JAAS **login-config.xml**.

Tableau 24.14. Éléments de timeout

Élément	Description
blocking-timeout-millis	L'élément « blocking-timeout-millis » indique la durée maximale en millisecondes de blocage pendant que vous attendez une connexion, avant de lever une exception. Notez que cela bloque uniquement pendant que vous attendez un permis de connexion, et ne soulèvera pas d'exception si la création d'une nouvelle connexion prend un temps excessivement long. La valeur par défaut est **30000** (30 secondes).
idle-timeout-minutes	Les éléments idle-timeout-minutes indiquent la durée maximum, en minutes, avant qu'une connexion inutile puisse être fermée. La durée maximum dépend du temps de balayage de l'idleRemover, qui correspond à la moitié du temps « idle-timeout-minutes » le plus petit de n'importe quel pool.
allocation-retry	Cet élément de tentative d'allocation indique le nombre de fois que l'on doit allouer un connexion avant de lancer une exception. La valeur par défaut est **0**.
allocation-retry-wait-millis	Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est **5000**, soit 5 secondes.
xa-resource-timeout	Passé à **XAResource.setTransactionTimeout()**. La valeur par défaut est 0 sans invoquer le setter. Indiqué en secondes.

Tableau 24.15. Éléments de validation

Élément	Description
background-validation	Élément pour spécifier que les connexions doivent être validées en arrière-plan plutôt qu'avant utilisation
background-validation-minutes	L'élément « background-validation-minutes » indique la durée, en minutes, d'exécution de la validation d'arrière-plan.

Élément	Description
use-fast-fail	Indique s'il y a échec d'allocation de connexion à la première connexion si invalide (true) ou s'il y a de nouvelles tentatives jusqu'à ce que le pool soit épuisé de tout essai de connexion possible (false). La valeur par défaut est **false**

Tableau 24.16. Éléments d'objets admin

Élément	Description
config-property	Spécifie une propriété de configuration d'objet d'administration.

Tableau 24.17. Éléments de recouvrement

Élément	Description
recover-credential	Indique la paire nom / mot de passe ou le domaine de sécurité qui doit être utilisé pour le recouvrement.
recover-plugin	Spécifie l'implémentation de org.jboss.jca.core.spi.recovery.RecoveryPlugin class.

Les schéma de déploiement sont définis dans **jboss-as-resource-adapters_1_0.xsd** et http://www.ironjacamar.org/doc/schema/ironjacamar_1_0.xsd pour l'activation automatique.

Rapporter un bogue

24.6. AFFICHAGES DES STATISTIQUES DE CONNEXION

Vous pouvez lire les statistiques d'une connexion définie à partir de la sous-arborescence **deployment=name.rar**.

Les statistiques sont définis à ce niveau et non pas au niveau **/subsystem** afin d'être accessibles à partir de tout **rar** non défini dans les configurations de fichiers **standalone.xml** ou **domain.xml**.

Par exemple :

Exemple 24.1.

```
/deployment=example.rar/subsystem=resource-
adapters/statistics=statistics/connection-
definitions=java\:\/testMe:read-resource(include-runtime=true)
```

NOTE

Veillez à spécifier l'argument ***`include-runtime=true`*** car tous les statistiques sont en runtime uniquement et la valeur par défaut est **`false`**.

Rapporter un bogue

24.7. STATISTIQUES D'ADAPTATEUR DE RESSOURCES

Statistiques principaux

Le tableau suivant contient une liste de statistiques principaux d'adaptateurs de ressources pris en charge :

Tableau 24.18. Statistiques principaux

Nom	Description
`ActiveCount`	Le nombre de connexions actives. Chacune de ces connexions est soit utilisée par une application, ou disponible via pool
`AvailableCount`	Le nombre de connexions disponibles dans le pool
`AverageBlockingTime`	Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes.
`AverageCreationTime`	Le durée moyenne passée à créer une connexion. La valeur est en millisecondes.
`CreatedCount`	Le nombre de connexions créées.
`DestroyedCount`	Le nombre de connexions détruites.
`InUseCount`	Le nombre de connexions actuellement utilisées.
`MaxCreationTime`	La durée maximum pour créer une connexion. La valeur est en millisecondes.
`MaxUsedCount`	Le nombre maximum de connexions utilisées
`MaxWaitCount`	Le nombre maximum de requêtes attendant une connexion en même temps.
`MaxWaitTime`	Le durée maximum à attendre un verrou exclusif sur le pool.
`TimedOut`	Le nombre de connexions expirées
`TotalBlockingTime`	Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes.
`TotalCreationTime`	La durée passée à créer des connexions. La valeur est en millisecondes.

Nom	Description
`WaitCount`	Le nombre de requêtes en attente de connexion.

Rapporter un bogue

24.8. DÉPLOYER L'ADAPTATEUR DE RESSOURCES WEBSPHERE MQ

Websphere MQ

WebSphere MQ est un logiciel de messagerie Oriented Middleware (MOM) d'IBM qui permet à des applications sur des systèmes distribués à communiquer entre eux. Ceci est accompli grâce à l'utilisation des messages et des files d'attente de messages. WebSphere MQ est chargé de remettre des messages à des files d'attente de messages et pour transférer des données à d'autres gestionnaires de file d'attente à l'aide de canaux de message. Pour plus d'informations sur WebSphere MQ, voir WebSphere MQ.

Résumé

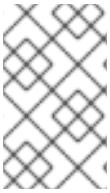
Cette section couvre les étapes à suivre pour déployer et configurer l'adaptateur de ressource Websphere MQ dans JBoss EAP 6. Cela peut se faire manuellement en modifiant les fichiers de configuration, par l'interface CLI, ou par la console de gestion basée-web.

Pré-requis

Avant de démarrer, vous devrez vérifier votre version d'adaptateur de ressource WebSphere MQ et comprendre certaines propriétés de configuration de WebSphere MQ.

- L'adaptateur de ressources WebSphere MQ est fourni en tant que fichier RAR (Resource Archive) nommé **`wmq.jmsra-VERSION.rar`**. Vous devrez utiliser la version **`7.5.0.0 ou version supérieure`**.
- Vous devez connaître les valeurs des propriétés de configuration Websphere MQ suivantes. Voir la documentation de produit WebSphere MQ pour obtenir des détails sur ces propriétés.
  - MQ.QUEUE.MANAGER: le nom du gestionnaire de files d'attentes de WebSphere MQ
  - MQ.HOST.NAME: le nom d'hôte utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.CHANNEL.NAME: le canal de serveur utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.QUEUE.NAME: le nom de la file d'attente de destination
  - MQ.TOPIC.NAME: le nom du sujet de destination
  - MQ.PORT: le port utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.CLIENT: le type de transport
- Pour les connexions sortantes, vous devrez vous familiariser avec la propriété de configuration suivante :

- MQ.CONNECTIONFACTORY.NAME: le nom de l'instance d'usine de connexion qui fournit la connexion vers le système à distance.



NOTE

Voici les configurations par défaut fournies par IBM. Elles sont assujetties au changement. Veuillez vous référer à la documentation Websphere MQ pour plus d'informations.

Procédure 24.8. Déployer l'adaptateur de ressources manuellement

1. Si vous avez besoin d'un support de transactions avec l'adaptateur de ressources de WebSphereMQ, vous devrez re-paquager l'archive **`wmq.jmsra-VERSION.rar`** pour qu'elle inclue **`mqetclient.jar`**. Vous devrez utiliser la commande suivante :

   ```
   [user@host ~]$ jar -uf wmq.jmsra-VERSION.rar mqetclient.jar
   ```

   Soyez certain de remplacer *VERSION* par le numéro de version correct.

2. Copier le fichier **`wmq.jmsra-VERSION.rar`** dans le répertoire **`EAP_HOME/standalone/deployments/`**.

3. Ajouter l'adaptateur de ressources au fichier de configuration du serveur.

   a. Ouvrir le fichier **`EAP_HOME/standalone/configuration/standalone-full.xml`** dans un éditeur.

   b. Chercher le sous-système **`urn:jboss:domain:resource-adapters`** dans le fichier de configuration.

   c. Il n'y a pas d'adaptateur de ressources défini pour ce système. Veuillez commencer par remplacer :

   ```
   <subsystem xmlns="urn:jboss:domain:resource-adapters:1.1"/>
   ```

   par ceci :

   ```
   <subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">
       <resource-adapters>
           <!-- <resource-adapter> configuration listed below -->
       </resource-adapters>
   </subsystem>
   ```

   d. La configuration de l'adaptateur de ressources dépend de si vous avez besoin de la restauration et du support de transactions. Si vous n'avez pas besoin de support de transaction, choisissez la première étape de configuration ci-dessous. Si vous avez besoin de support de transaction, choisissez la deuxième étape de la configuration.

   - Pour les déploiements non transactionnels, veuillez remplacer la configuration **`<!-- <resource-adapter> listée ci-dessous -->`** par ce qui suit :

   ```
   <resource-adapter>
       <archive>
           wmq.jmsra-VERSION.rar
   ```

```
    </archive>
    <transaction-support>NoTransaction</transaction-support>
    <connection-definitions>
        <connection-definition
                class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
                jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
                pool-name="MQ.CONNECTIONFACTORY.NAME">
            <config-property name="hostName">
                MQ.HOST.NAME
            </config-property>
            <config-property name="port">
                MQ.PORT
            </config-property>
            <config-property name="channel">
                MQ.CHANNEL.NAME
            </config-property>
            <config-property name="transportType">
                MQ.CLIENT
            </config-property>
            <config-property name="queueManager">
                MQ.QUEUE.MANAGER
            </config-property>
            <security>
                <security-domain>MySecurityDomain</security-
domain>
            </security>
       </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object
                class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
                jndi-name="java:jboss/MQ.QUEUE.NAME"
                pool-name="MQ.QUEUE.NAME">
            <config-property name="baseQueueName">
                MQ.QUEUE.NAME
            </config-property>
            <config-property name="baseQueueManagerName">
                MQ.QUEUE.MANAGER
            </config-property>
       <admin-object class-
name="com.ibm.mq.connector.outbound.MQTopicProxy"
                jndi-name="java:jboss/MQ.TOPIC.NAME" pool-
name="MQ.TOPIC.NAME">
            <config-property name="baseTopicName">
                MQ.TOPIC.NAME
            </config-property>
            <config-property name="brokerPubQueueManager">
                MQ.QUEUE.MANAGER
            </config-property>
       </admin-object>
```

```
      </admin-object>
    </admin-objects>
</resource-adapter>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

- Pour les déploiements transactionnels, veuillez remplacer la configuration **<!-- <resource-adapter> listée ci-dessous -->** par ce qui suit :

```
<resource-adapter>
    <archive>
        wmq.jmsra-VERSION.rar
    </archive>
    <transaction-support>XATransaction</transaction-support>
    <connection-definitions>
        <connection-definition
                class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
                jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
                pool-name="MQ.CONNECTIONFACTORY.NAME">
            <config-property name="hostName">
                MQ.HOST.NAME
            </config-property>
            <config-property name="port">
                MQ.PORT
            </config-property>
            <config-property name="channel">
                MQ.CHANNEL.NAME
            </config-property>
            <config-property name="transportType">
                MQ.CLIENT
            </config-property>
            <config-property name="queueManager">
                MQ.QUEUE.MANAGER
            </config-property>
           <security>
                <security-domain>MySecurityDomain</security-
domain>
            </security>
            <recovery>
                <recover-credential>
                    <user-name>USER_NAME</user-name>
                    <password>PASSWORD</password>
                </recover-credential>
            </recovery>
        </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object
                class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
                jndi-name="java:jboss/MQ.QUEUE.NAME"
                pool-name="MQ.QUEUE.NAME">
```

```
            <config-property name="baseQueueName">
                MQ.QUEUE.NAME
            </config-property>
            <config-property name="baseQueueManagerName">
                MQ.QUEUE.MANAGER
            </config-property>
        </admin-object>
        <admin-object class-
name="com.ibm.mq.connector.outbound.MQTopicProxy"
                jndi-name="java:jboss/MQ.TOPIC.NAME" pool-
name="MQ.TOPIC.NAME">
            <config-property name="baseTopicName">
                MQ.TOPIC.NAME
            </config-property>
            <config-property name="brokerPubQueueManager">
                MQ.QUEUE.MANAGER
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR. Vous devrez également remplacer *USER_NAME* et *PASSWORD* avec le nom et le mot de passe valides.

NOTE

Pour supporter les transactions, l'élément <transaction-support> a été défini à **`XATransaction`**. Pour supporter XA recovery, l'élément <recovery> a été ajouté à une définition de connexion.

e. Si vous souhaitez changer le fournisseur par défaut en système de messagerie EJB3 dans JBoss EAP 6 de HornetQ vers WebSphere MQ, modifier le sous-système **`urn:jboss:domain:ejb3:1.2`** comme suit :

Remplacer :

```
<mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

par :

```
<mdb>
    <resource-adapter-ref resource-adapter-name="wmq.jmsra-
VERSION.rar"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

Procédure 24.9. Modifier le code MDB pour utiliser l'adaptateur de ressources

- Configurer ActivationConfigProperty et ResourceAdapter du code MDB comme suit :

```
@MessageDriven( name="WebSphereMQMDB",
    activationConfig =
    {
        @ActivationConfigProperty(propertyName =
"destinationType",propertyValue = "javax.jms.Queue"),
        @ActivationConfigProperty(propertyName = "useJNDI",
propertyValue = "false"),
        @ActivationConfigProperty(propertyName = "hostName",
propertyValue = "MQ.HOST.NAME"),
        @ActivationConfigProperty(propertyName = "port",
propertyValue = "MQ.PORT"),
        @ActivationConfigProperty(propertyName = "channel",
propertyValue = "MQ.CHANNEL.NAME"),
        @ActivationConfigProperty(propertyName = "queueManager",
propertyValue = "MQ.QUEUE.MANAGER"),
        @ActivationConfigProperty(propertyName = "destination",
propertyValue = "MQ.QUEUE.NAME"),
        @ActivationConfigProperty(propertyName = "transportType",
propertyValue = "MQ.CLIENT")
    })
    @ResourceAdapter(value = "wmq.jmsra-VERSION.rar")
@TransactionAttribute(TransactionAttributeType.NOT_SUPPORTED)
public class WebSphereMQMDB implements MessageListener {
}
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

Rapporter un bogue

24.9. INSTALLER L'ADAPTATEUR DE RESSOURCES DE JBOSS ACTIVE MQ

Afin d'installer l'adaptateur de ressources JBoss Active MQ (A-MQ) dans JBoss EAP 6 pour qu'il puisse fontionner avec JBoss Fuse A-MQ 6.1.0, suivre les étapes suivantes : https://access.redhat.com/site/documentation/en-US/Red_Hat_JBoss_Fuse/6.1/html/Deploying_into_a_Web_Server/files/DeployRar.html.

Rapporter un bogue

24.10. CONFIGURER UN ADAPTATEUR DE RESSOURCES JMS STANDARD À UTILISER AVEC UN FOURNISSEUR JMS DE TIERCE PARTIE

Résumé

JBoss EAP 6 peut être configuré pour fonctionner avec des fournisseurs tiers de JMS, cependant tous les fournisseurs JMS ne produisent pas un adaptateur de ressources JMS JCA pour l'intégration avec les plateformes d'applications Java. Cette procédure couvre les étapes requises pour configurer l'adaptateur de ressources JMS générique inclus dans JBoss EAP 6 pour se connecter à un fournisseur JMS. Dans cette procédure, Tibco EMS 6.3 est utilisé comme un exemple de fournisseur JMS, mais d'autres fournisseurs JMS peuvent avoir besoin d'une configuration différente.

IMPORTANT

L'adaptateur de ressources JMS JCA générique ne devrait servir que lorsqu'un fournisseur JMS ne fournit pas son propre adaptateur de ressources. Avant d'utiliser l'adaptateur de ressources JMS générique, vous devez d'abord vérifier auprès du fournisseur JMS pour savoir s'il possède son propre adaptateur de ressources qui puisse être utilisé avec JBoss EAP 6.

Conditions préalables

Cette procédure suppose qu'un serveur de fournisseur JMS est déjà configuré et prêt à l'emploi. Les binaires nécessaires à l'implémentation du fournisseur JMS seront nécessaires. Vous devez également connaître les valeurs des propriétés de fournisseur JMS suivantes :

- *PROVIDER_HOST*:*PROVIDER_PORT*: le nom d'hôte et le numéro de port du serveur de fournisseur JMS.
- *PROVIDER_CONNECTION_FACTORY*: le nom de l'usine de connexion déployée sur le serveur de fournisseur JMS. Doit être XA.
- *PROVIDER_QUEUE*, *PROVIDER_TOPIC*: les noms des files d'attente et des sujets qui se trouvent sur le serveur de fournisseurs JMS qui doivent être utilisés.

Procédure 24.10. Configuration d'un adaptateur de ressources générique JMS

1. Créer une implémentation **`ObjectFactory`** pour la liaison de files d'attentes et de sujets JNDI :

   a. En utilisant le code ci-dessous comme modèle, remplacer les informations serveur par les valeurs de serveur du fournisseur JMS.

```
import java.util.Hashtable;
import java.util.Properties;

public class RemoteJMSObjectFactory implements ObjectFactory {

  private Context context = null;

  public RemoteJMSObjectFactory() {
  }

  public Object getObjectInstance(Object obj, Name name, Context
nameCtx,
      Hashtable<?, ?> environment) throws Exception {
    try {
      String jndi = (String) obj;

      final Properties env = new Properties();
      env.put(Context.INITIAL_CONTEXT_FACTORY,
          "com.tibco.tibjms.naming.TibjmsInitialContextFactory");
      env.put(Context.URL_PKG_PREFIXES,
"com.tibco.tibjms.naming");
      env.put(Context.PROVIDER_URL,
"tcp://TIBCO_HOST:TIBCO_PORT");

      context = new InitialContext(env);
      Object o = context.lookup(jndi);
```

```
        return o;
      } catch (NamingException e) {
        e.printStackTrace();
        throw e;
      }
    }
  }
```

b. Compiler le code ci-dessus, et sauvegarder le fichier de classe résultant dans un fichier JAR nommé **`remoteJMSObjectFactory.jar`**

2. Créer un module **`genericjms`** pour votre instance JBoss EAP 6 :

a. Créer la structure de répertoire suivante : ***`EAP_HOME`*`/modules/system/layers/base/org/jboss/genericjms/provider/main`**

b. Copier le fichier **`remoteJMSObjectFactory.jar`** dans ***`EAP_HOME`*`/modules/system/layers/base/org/jboss/genericjms/provider/main`**

c. Copier les binaires requis pour l'implémentation JMS du fournisseur dans ***`EAP_HOME`*`/modules/system/layers/base/org/jboss/genericjms/provider/main`**. Pour Tibco EMS, les binaires requis sont **`tibjms.jar`** et **`tibcrypt.jar`** du répertoire **`/lib`** de l'installation Tibco.

d. Créer un fichier **`module.xml`** dans ***`EAP_HOME`*`/modules/system/layers/base/org/jboss/genericjms/provider/main`** comme ci-dessous, en énumérant les fichiers JAR des étapes précédentes comme ressources :

```
<module xmlns="urn:jboss:module:1.1"
name="org.jboss.genericjms.provider">
  <resources>
      <resource-root path="tibjms.jar"/>
      <resource-root path="tibcrypt.jar"/>
      <resource-root path="remoteJMSObjectFactory.jar"/>
  </resources>

   <dependencies>
      <module name="javax.api"/>
      <module name="javax.jms.api"/>
  </dependencies>
</module>
```

3. Ajouter le module JMS générique comme une dépendance pour tous les déploiements en tant que modules globaux.

NOTE

Dans cette procédure, ***`EAP_HOME`*`/standalone/configuration/standalone-full.xml`** est utilisé comme fichier de configuration JBoss EAP 6.

Dans ***EAP_HOME*/standalone/configuration/standalone-full.xml**, sous **<subsystem xmlns="urn:jboss:domain:ee:1.1">**,ajouter:

```
<global-modules>
  <module name="org.jboss.genericjms.provider" slot="main"/>
  <module name="org.jboss.common-core" slot="main"/>
</global-modules>
```

4. Remplacer l'adaptateur de ressources HornetQ par défaut par l'adaptateur de ressources générique.

   Dans ***EAP_HOME*/standalone/configuration/standalone-full.xml**, remplacer **<subsystem xmlns="urn:jboss:domain:ejb3:1.4"> <mdb>**, par:

```
<mdb>
  <resource-adapter-ref resource-adapter-
name="org.jboss.genericjms"/>
  <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

5. Ajouter les liaisons pour vos files d'attente et sujets JMS en tant qu'objets selon les besoins.

   Dans ***EAP_HOME*/standalone/configuration/standalone-full.xml**, sous **<subsystem xmlns="urn:jboss:domain:naming:1.3">**, ajouter les liaisons, remplaçant *PROVIDER_QUEUE* et *PROVIDER_TOPIC* selon les besoins :

```
<bindings>
  <object-factory name="PROVIDER_QUEUE"
module="org.jboss.genericjms.provider"
class="org.jboss.qa.RemoteJMSObjectFactory"/>
  <object-factory name="PROVIDER_TOPIC"
module="org.jboss.genericjms.provider"
class="org.jboss.qa.RemoteJMSObjectFactory"/>
</bindings>
```

6. Dans ***EAP_HOME*/standalone/configuration/standalone-full.xml**, ajouter la configuration d'adaptateur de ressources générique dans **<subsystem xmlns="urn:jboss:domain:resource-adapters:1.1">**.

   Remplacer *PROVIDER_CONNECTION_FACTORY*, *PROVIDER_HOST*, et *PROVIDER_PORT* par les valeurs du fournisseur JMS.

```
<resource-adapters>
  <resource-adapter id="org.jboss.genericjms">
    <module slot="main" id="org.jboss.genericjms"/>
    <transaction-support>NoTransaction</transaction-support>
    <connection-definitions>
      <connection-definition class-
name="org.jboss.resource.adapter.jms.JmsManagedConnectionFactory"
jndi-name="java:/jms/PROVIDER_CONNECTION_FACTORY" pool-
name="PROVIDER_CONNECTION_FACTORY">
        <config-property name="JndiParameters">

java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialCon
```

```
textFactory;java.naming.provider.url=tcp://PROVIDER_HOST:PROVIDER_PO
RT
        </config-property>
        <config-property name="ConnectionFactory">
          PROVIDER_CONNECTION_FACTORY
        </config-property>
        <security>
          <application/>
        </security>
      </connection-definition>
    </connection-definitions>
  </resource-adapter>
</resource-adapters>
```

Résultat

L'adaptateur de ressources JMS est maintenant configuré et prêt à l'utilisation.

Quand on crée un nouveau message MDB (Message Driven Bean), il faut un code (ci-dessous) similaire pour utiliser l'adaptateur de ressources. Remplacer *PROVIDER_CONNECTION_FACTORY*, *PROVIDER_HOST*, et *PROVIDER_PORT* par les valeurs du fournisseur JMS.

En option, l'exemple ci-dessous configure également une connexion sécurisée pour Tibco EMS en spécifiant les propriétés d'utilisateur **`user`** et de mot de passe **`password`** (remplacer les valeurs de propriété en fonction des besoins).

```
@MessageDriven(activationConfig = {
  @ActivationConfigProperty(propertyName = "destinationType",
propertyValue = "javax.jms.Queue"),
  @ActivationConfigProperty(propertyName = "jndiParameters", propertyValue
=
"java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialContextF
actory;java.naming.provider.url=tcp://PROVIDER_HOST:PROVIDER_PORT")
  @ActivationConfigProperty(propertyName = "destination", propertyValue =
"PROVIDER_QUEUE"),
  @ActivationConfigProperty(propertyName = "connectionFactory",
propertyValue = "PROVIDER_CONNECTION_FACTORY"),
  @ActivationConfigProperty(propertyName = "user", propertyValue =
"USER"),
  @ActivationConfigProperty(propertyName = "password", propertyValue =
"PASSWORD"),
})

@ResourceAdapter("org.jboss.genericjms")
public class SampleMdb implements MessageListener {
  @Override

    public void onMessage(Message message) {

    }

}
```

Rapporter un bogue

CHAPITRE 25. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2

25.1. INTRODUCTION

25.1.1. Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) est un service exploité par amazon.com qui offre aux clients un environnement informatique virtuel personnalisable. Une Image de Machine Amazon (AMI) peut être démarrée en utilisant le service pour créer une instance ou une machine virtuelle. Les utilisateurs peuvent installer n'importe quel logiciel dont ils ont besoin sur une instance et sont facturés en fonction de l'usage. Amazon EC2 est conçu pour être flexible et permettre aux utilisateurs de déployer rapidement leurs applications.

Vous pourrez en savoir davantage sur le site web Amazon EC2, http://aws.amazon.com/ec2/.

Rapporter un bogue

25.1.2. Amazon Machine Instances (AMIs)

Une Amazon Machine Image (AMI) est un modèle d'instance de machine virtuelle EC2. Les utilisateurs créent des instances EC2 en sélectionnant une AMI appropriée pour créer l'instance. La composante primaire d'une AMI est un système de fichiers en lecture seule qui contient un système d'exploitation installé, mais aussi des autres logiciels. Chaque AMI a différents logiciels installés pour des cas d'utilisation différents. Amazon EC2 comprend beaucoup d'AMIs au choix offerts par amazon.com et des tierces parties. Les utilisateurs peuvent également créer leurs propres AMIs personnalisées.

Rapporter un bogue

25.1.3. JBoss Cloud Access

JBoss Cloud Access est une fonctionnalité de Red Hat qui fournit un support à JBoss EAP 6 aux fournisseurs cloud certifiés Red Hat comme Amazon EC2. JBoss Cloud Access vous permet de déplacer vos abonnements entre les serveurs traditionnels et les ressources publiques basées-cloud d'une façon simple et peu coûteuse.

Vous trouverez des informations supplémentaires à l'adresse suivante http://www.redhat.com/solutions/cloud/access/jboss/.

Rapporter un bogue

25.1.4. Fonctionnalités de JBoss Cloud Access

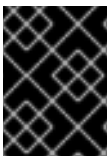
L'abonnement au programme JBoss Cloud Access donne accès aux AMI (Amazon Machine Images) privées créées par Red Hat.

Les AMI de Red Hat ont le logiciel suivant pré-installé et complètement pris en charge par Red Hat :

- Red Hat Enterprise Linux 6
- JBoss EAP 6
- L'agent JBoss Operations Network (JON) 3

- Mises à jour de produit par les RPM par l'intermédiaire de l'infrastructure de mise à jour de Red Hat.

Chaque AMI de Red Hat n'est qu'un point de départ, qui requiert une configuration supplémentaire pour se conformer aux besoins de votre application.



IMPORTANT

JBoss Cloud Access n'apporte pas actuellement de support au profil full-ha, ni pour les instances standalone, ni pour les domaines gérés.

Rapporter un bogue

25.1.5. Types d'instances Amazon EC2 prises en charge

JBoss Cloud Access prend en charge les types d'instance Amazon EC2 suivantes. Voir *Amazon EC2 User Guide* pour obtenir davantage de détails sur chaque type d'instance, http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/instance-types.html.

Tableau 25.1. Types d'instances Amazon EC2 prises en charge

Type d'instance	Description
Instance standard	Les instances standard sont des environnements d'ordre général ayant un ratio de mémoire-à-CPU équilibré.
Instance de mémoire élevée	Les instances de mémoire élevée possède davantage de mémoire allouée que les instances standard. Les instances de mémoire élevée conviennent aux applications à haut débit telles que les bases de données ou les applications de mise en cache de mémoire.
Instance Haut CPU	Les instance Haut CPU ont davantage de ressources CPU allouées que de mémoire et conviennent à des débits moindres et à des applications intensives en CPU.

IMPORTANT

Le type d'instance **`Micro (t1.micro)`** ne convient pas au déploiement de la plateforme JBoss EAP 6.

Rapporter un bogue

25.1.6. Les AMI Red Hat prises en charge

Les AMI Red Hat prises en charge peuvent être identifiées par leur nom AMI.

Les AMI de JBoss EAP 6 sont composées ainsi :

```
RHEL-osversion-JBEAP-6.0.0-arch-creationdate
```

osversion est le nom de version de Red Hat Enterprise Linux installé dans l'AMI. Exemple **`6.2`**.

arch est l'architecture de l'AMI. Correspondra à **`x86_64`** ou **`i386`**.

creationdate est la date de création de l'AMI sous le format *YYYYMMDD*. Exemple **`20120501`**.

Exemple de nom d'AMI : **`RHEL-6.2-JBEAP-6.0.0-x86_64-20120501`**.

Rapporter un bogue

25.2. DÉPLOYER JBOSS EAP 6 DANS AMAZON EC2

25.2.1. Aperçu du déploiement de JBoss EAP 6 sur Amazon EC2

JBoss EAP 6 peut être déployé avec l'AMI Amazon EC2. L'AMI contient tout ce qui est requis pour le déploiement des instances clusterisées ou non clusterisées.

Le déploiement d'instances non clusterisées est le scénario le plus facile. Cela exige uniquement quelques changements de configuration pour spécifier votre déploiement d'application quand vous créez l'instance.

Le déploiement d'instances en cluster est plus compliqué. En plus de vos instances en cluster, vous devez déployer une instance de JBoss EAP 6 qui puisse agir en tant que proxy mod_cluster et S3 Bucket pour le protocole de découverte S3_PING JGroups. Red Hat recommande également la création d'un cloud privé virtuel pour contenir votre cluster.

Chacune de ces étapes est expliquée ci-dessous mais on assume que vous avez de l'expérience avec JBoss EAP 6, Red Hat Enterprise Linux 6 et Amazon EC2.

La documentation supplémentaire suivante est recommandée :

- JBoss EAP 6, https://access.redhat.com/site/documentation/JBoss_Enterprise_Application_Platform/.
- Red Hat Enterprise Linux 6, https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.
- Amazon Web Services, http://aws.amazon.com/documentation/.

Rapporter un bogue

25.2.2. JBoss EAP 6 non clusterisée

25.2.2.1. Instances non-clusterisées

Une instance non clusterisée est une instance simple Amazon EC2 exécutant sur JBoss EAP 6. Elle ne fait pas partie d'un cluster.

Rapporter un bogue

25.2.2.2. Instances non clusterisées

25.2.2.2.1. Lancer l'instance de JBoss EAP 6 non clusterisée

Résumé

Ce sujet couvre les étapes requises pour lancer une instance de JBoss EAP 6 non clusterisée sur une AMI (Amazon Machine Image) Red Hat.

Conditions préalables

- Pour une AMI Red Hat qui convient, consulter Section 25.1.6, « Les AMI Red Hat prises en charge ».
- Groupe de sécurité pré-configuré qui autorise les requêtes entrantes sur les ports 22, 8080, et 9990 au moins.

Procédure 25.1. Lancer une instance non clusterisée de JBoss EAP 6 sur une AMI (Amazon Machine Image) de Red Hat.

1. Configurer le champ **`User Data`**. Les paramètres configurables sont disponibles ici : Section 25.4.1, « Paramètres de configuration permanente », Section 25.4.2, « Paramètres de scripts personnalisés ».

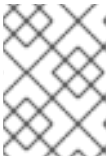
   Exemple 25.1. Exemple de champ de données utilisateur

   L'exemple montre le champ de données utilisateur d'une instance JBoss EAP 6 non clusterisée. Le mot de passe de l'utilisateur **`admin`** a été défini à **`adminpwd`**.

   ```
   JBOSSAS_ADMIN_PASSWORD=adminpwd
   JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

   # In production, access to these ports needs to be restricted for
   security reasons
   PORTS_ALLOWED="9990 9443"

   cat> $USER_SCRIPT << "EOF"

   # Get the application to be deployed from an Internet URL
   # mkdir -p /usr/share/java/jboss-ec2-eap-applications
   # wget https://<your secure storage hostname>/<path>/<app
   name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
   name>.war

   # Create a file of CLI commands to be executed after starting the
   server
   cat> $USER_CLI_COMMANDS << "EOC"
   # deploy /usr/share/java/jboss-ec2-eap-applications/<app name>.war
   EOC

   EOF
   ```

2. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

   ```
   yum -y update
   ```



NOTE

`yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

3. Lancement de l'instance AMI Red Hat

Résultat

Une instance non clusterisée de JBoss EAP 6 a été configurée, et lancée sur une AMI Red Hat.

Rapporter un bogue

25.2.2.2.2. Déployer une application sur une instance de JBoss EAP 6 non clusterisée

Résumé

Ce sujet couvre le déploiement d'une application sur une instance de JBoss EAP 6 sur une AMI Red Hat.

1. - **Déploiement d'un exemple d'application**
   Ajouter les lignes suivantes au champ **`User Data`** :

   ```
   # Deploy the sample application from the local filesystem
   deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
   ```

   Exemple 25.2. Champs de données d'utilisateur avec un exemple d'application

   Cet exemple utilise l'exemple d'application fourni sur l'AMI Red Hat. Il inclut également une configuration de base d'une instance non clusterisée de JBoss EAP 6. Le mot de passe **`admin`** de l'utilisateur a été défini à **`adminpwd`**.

   ```
   JBOSSAS_ADMIN_PASSWORD=adminpwd
   JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

   # In production, access to these ports needs to be restricted
   for security reasons
   PORTS_ALLOWED="9990 9443"

   cat> $USER_SCRIPT << "EOF"

   # Create a file of CLI commands to be executed after starting
   the server
   cat> $USER_CLI_COMMANDS << "EOC"

   # Deploy the sample application from the local filesystem
   deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
   EOC

   EOF
   ```

   - **Déployer une application personnalisée**
     Ajouter les lignes suivantes au champ **`User Data`** (données utilisateur), pour configurer le nom de l'URL de l'application :

```
# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://<your secure storage hostname>/<path>/<app name>.war
-O /usr/share/java/jboss-ec2-eap-applications/<app name>.war
```

Exemple 25.3. Exemple de champ de données utilisateur avec application personnalisée

Cet exemple utilise une application nommée **MyApp**, et inclut une configuration de base pour une instance JBoss EAP 6 non clusterisée. Le mot de passe **admin** de l'utilisateur a été défini à **adminpwd**.

```
JBOSSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted
for security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://PATH_TO_MYAPP/MyApp.war -O /usr/share/java/jboss-
ec2-eap-applications/MyApp.war

# Create a file of CLI commands to be executed after starting
the server
cat> $USER_CLI_COMMANDS << "EOC"
deploy /usr/share/java/jboss-ec2-eap-applications/MyApp.war
EOC

EOF
```

2. Lancement de l'instance AMI Red Hat

Résultat

L'application a été déployée avec succès dans JBoss EAP 6.

Rapporter un bogue

25.2.2.2.3. Lancer l'instance de JBoss EAP 6 non clusterisée

Résumé

Cette rubrique couvre les étapes nécessaires pour tester la plateforme non clusterisée de JBoss EAP 6.

Procédure 25.2. Vérifier que l'instance de JBoss EAP 6 non clusterisée exécute correctement

1. Déterminer le **Public DNS** de l'instance, qui se situe dans le panneau d'informations de l'instance.

2. Naviguer dans **http://<*public-DNS*>:8080**.

3. Confirmer que la page d'accueil de JBoss EAP apparaît, y compris le lien vers la console admin. Si la page d'accueil n'est pas disponible, consulter : Section 25.5.1, « Résolution de problèmes dans Amazon EC2 ».

4. Cliquer sur l'hyperlien **`Admin Console`**.

5. Se connecter :

   - Nom d'utilisateur : **`admin`**

   - Mot de passe : Spécifié dans le champ **`User Data`** ici : Section 25.2.2.2.1, « Lancer l'instance de JBoss EAP 6 non clusterisée ».

6. **Tester l'exemple d'application**
   Naviguer dans **`http://<public-DNS>:8080/hello`** pour tester que l'exemple d'application exécute avec succès. Le texte **`Hello World!`** doit apparaître dans le navigateur. Si le texte n'apparait pas, voir : Section 25.5.1, « Résolution de problèmes dans Amazon EC2 ».

7. Se déconnecter de la console admin de JBoss EAP.

Résultat

L'instance de JBoss EAP 6 exécute correctement.

Rapporter un bogue

25.2.2.3. Domaines gérés non clusterisés

25.2.2.3.1. Lancer une instance pour qu'elle serve de contrôleur de domaine

Résumé

Cette rubrique couvre les étapes requises pour lancer un domaine géré non clusterisé de JBoss EAP 6 sur une AMI Red Hat (Amazon Machine Image de Red Hat)

Conditions préalables

- Pour une Red Hat AMI qui convient, consulter Section 25.1.6, « Les AMI Red Hat prises en charge ».
- Section 25.2.3.4, « Créer un VPC (Virtual Private Cloud) »
- Section 25.2.3.5, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »
- Section 25.2.3.6, « Configurer le routage par défaut du sous-système privé VPC »
- Section 25.2.3.8, « Configurer l'installation IAM »
- Section 25.2.3.10, « Configurer l'installation S3 Bucket »

Procédure 25.3. Lancer un domaine géré de JBoss EAP 6 non clusterisé sur une Red Hat AMI

1. Dans l'onglet « groupe de sécurité », veillez bien à ce que tout le trafic soit autorisé. Les capacités de pare-feu intégrées de Red Hat Enterprise Linux peuvent être utilisées pour restreindre l'accès si nécessaire.

2. Définir le sous-réseau public du VPC à *running*.

3. Sélectionner un IP statique.

4. Configurer le champ **`User Data`**. Les paramètres configurables sont disponibles ici : Section 25.4.1, « Paramètres de configuration permanente », Section 25.4.2, « Paramètres de scripts personnalisés ». Pour plus d'informations sur le contrôleur de domaine discovery d'Amazon EC2, voir Section 25.2.2.3.4, « Configurer Domain Controller Discovery et Failover dans Amazon EC2 ».

Exemple 25.4. Exemple de champ de données utilisateur

L'exemple montre le champ de données utilisateur d'un domaine géré de JBoss EAP 6 non clusterisé. Le mot de passe de l'utilisateur **`admin`** a été défini sur **`admin`**.

```
## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=admin

## subnet prefix this machine is connected to
SUBNET=10.0.0.

## S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

#### to run the example no modifications below should be needed
####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## Create a file of CLI commands to be executed after starting the
server
cat> $USER_CLI_COMMANDS << "EOC"

# Add the modcluster subsystem to the default profile to set up a
proxy
/profile=default/subsystem=web/connector=ajp:add(name=ajp,protocol
=AJP/1.3,scheme=http,socket-binding=ajp)
/:composite(steps=[ {"operation" => "add", "address" => [
("profile" => "default"), ("subsystem" => "modcluster") ] },{
"operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration") ], "advertise" => "false", "proxy-list" =>
"${jboss.modcluster.proxyList}", "connector" => "ajp"}, {
"operation" => "add", "address" => [ ("profile" => "default"),
```

```
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration") ]},
{ "operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration"),
("load-metric" => "busyness")], "type" => "busyness"} ])

# Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/hello.war --server-
groups=main-server-group
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

5. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

   ```
   yum -y update
   ```

   NOTE

   `yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

6. Lancement de l'instance Red Hat AMI

Résultat

Un domaine géré non clusterisé de JBoss EAP 6 a été configuré, et lancé sur une Red Hat AMI.

Rapporter un bogue

25.2.2.3.2. Lancer une ou plusieurs instances pour qu'elles servent de contrôleurs hôtes

Résumé

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss EAP 6 en tant que contrôleurs hôtes non clusterisée sur Red Hat AMI (Amazon Machine Image de Red Hat).

Conditions préalables

- Configurer et lancer le contrôleur de domaine non clusterisé. Consulter Section 25.2.2.3.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine » .

- Section 25.2.3.8, « Configurer l'installation IAM »
- Section 25.2.3.10, « Configurer l'installation S3 Bucket »

Procédure 25.4. Lancer les contrôleurs hôtes

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

1. Sélectionner une AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe de sécurité.
5. Veillez à ce que tout le trafic en provenance du sous-système de JBoss EAP 6 soit autorisé.
6. Définir les autres restrictions suivant les besoins.
7. Ajouter ce qui suit dans le champ User Data :

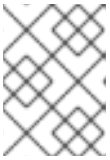
```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## host controller setup
### static domain controller discovery setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
### S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

JBOSS_HOST_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed ####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/other-server-group/main-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
```

```
done >> /etc/hosts

EOF
```

Pour plus d'informations sur le contrôleur de domaine discovery d'Amazon EC2, voir Section 25.2.2.3.4, « Configurer Domain Controller Discovery et Failover dans Amazon EC2 ».

8. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```



NOTE

`yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance Red Hat AMI

Résultat

Les contrôleurs hôtes non clusterisés de JBoss EAP 6 ont été configurés, et lancés sur une Red Hat AMI.

Rapporter un bogue

25.2.2.3.3. Tester le domaine géré de JBoss EAP 6 non clusterisée

Résumé

Cette rubrique couvre les étapes requises pour tester le domaine géré non clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTP et du contrôleur de domaines de JBoss EAP 6 à la fois.

Conditions préalables

- Configurer et lancer le contrôleur de domaines. Consulter Section 25.2.2.3.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine » .
- Configurer et lancer les contrôleur hôtes. Consulter Section 25.2.2.3.2, « Lancer une ou plusieurs instances pour qu'elles servent de contrôleurs hôtes » .

Procédure 25.5. Tester le Serveur Web

- Naviguer dans **`http://ELASTIC_IP_OF_APACHE_HTTPD`** avec un navigateur pour confirmer que le serveur web exécute avec succès.

Procédure 25.6. Tester le contrôleur de domaine

1. Naviguer dans **`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console`**
2. Connectez-vous en utilisant le nom d'utilisateur **`admin`** et le mot de passe spécifiés dans le

champ « données d'utilisateur » pour le contrôleur de domaine et la page d'accueil de la console admin du domaine géré s'affichera (**`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances`**).

3. Cliquer sur l'étiquette **`Server`** du serveur en haut et à droite de l'écran, et sélectionner un contrôleur hôte dans le menu déroulant **`Host`** en haut et à gauche de l'écran.

4. Vérifier que chaque contrôleur hôte ait deux configurations de serveurs nommées respectivement **`server-one`** et **`server-two`** qui appartiennent toutes deux au **`main-server-group`**.

5. Se déconnecter de la console admin de JBoss EAP.

Procédure 25.7. Tester les contrôleur hôte

1. Naviguer dans **`http://ELASTIC_IP_OF_APACHE_HTTPD/hello`** pour tester que l'exemple d'application exécute. Le texte **`Hello World!`** devrait apparaître dans le navigateur.

   Si le texte n'est pas visible, voir : Section 18.5.1, "About Troubleshooting Amazon EC2".

2. Connectez-vous à l'instance Apache HTTP :

   ```
   $ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTPD
   ```

3. Naviguer dans **`http://localhost:7654/mod_cluster-manager`** pour confirmer que toutes les instances exécutent correctement.

Résultat

Le serveur web de JBoss EAP 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur une Red Hat AMI.

Rapporter un bogue

25.2.2.3.4. Configurer Domain Controller Discovery et Failover dans Amazon EC2

Pour un domaine géré s'exécutant sur Amazon EC2, en plus de la découverte de contrôleur de domaine statique, les contrôleurs hôtes peuvent découvrir dynamiquement un contrôleur de domaine en utilisant le système de stockage d'Amazon S3. En particulier, les contrôleurs hôtes et le contrôleur de domaine peuvent être configurés avec les informations nécessaires pour accéder à un package Amazon S3.

En utilisant cette configuration, lorsqu'un contrôleur de domaine est démarré, il écrit ses coordonnées dans un fichier S3 dans le package. Chaque fois qu'un contrôleur hôte tente de contacter le contrôleur de domaine, il obtient des informations de contact du contrôleur de domaine du fichier S3.

Cela signifie que si les coordonnées du contrôleur de domaine changent (par exemple, il est fréquent que l'adresse IP de l'instance EC2 change quand il est arrêté et démarré), les contrôleurs hôtes n'ont pas besoin d'être reconfigurés. Les contrôleurs hôtes sont en mesure d'obtenir les nouvelles coordonnées du contrôleur de domaine dans le fichier S3.

Vous pouvez activer Domain Controller Discovery en passant les paramétres ***`JBOSS_DOMAIN_S3_ACCESS_KEY`***, ***`JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY`***, et ***`JBOSS_DOMAIN_S3_BUCKET`*** à l'instance de JBoss EAP 6 quand vous la lancez. Voir Section 25.4.1, « Paramètres de configuration permanente » pour voir les paramètres configurables. Sinon, vous pouvez configurer Domain Discovery manuellement par la configuration suivante.

La configuration manuelle de Domain Controler Discovery est spécifiée par les propriétés suivantes :

access-key

La clé d'accès au compte utilisateur Amazon AWS

secret-access-key

La clé d'accès secrète au compte utilisateur Amazon AWS

location

Le package Amazon S3 à utiliser

Voici des exemples de configuration de contrôleurs hôtes et de contrôleurs de domaines. Bien qu'une option discovery est illustrée dans les exemples ci-dessous, il est possible de configurer un nombre quelconque de discovery statique ou d'options discovery S3. Pour plus d'informations sur le processus de découverte et de basculement du domaine, voir Section 1.7, « Domain Controller Discovery et Failover ».

Exemple 25.5. Configuration du contrôleur hôte

```
<domain-controller>
  <remote security-realm="ManagementRealm">
    <discovery-options>
      <discovery-option name="s3-discovery"
code="org.jboss.as.host.controller.discovery.S3Discovery"
module="org.jboss.as.host-controller">
        <property name="access-key" value="S3_ACCESS_KEY"/>
        <property name="secret-access-key"
value="S3_SECRET_ACCESS_KEY"/>
        <property name="location" value="S3_BUCKET_NAME"/>
      </discovery-option>
    </discovery-options>
  </remote>
</domain-controller>
```

Exemple 25.6. Configuration du contrôleur de domaine

```
<domain-controller>
  <local>
    <discovery-options>
      <discovery-option name="s3-discovery"
code="org.jboss.as.host.controller.discovery.S3Discovery"
module="org.jboss.as.host-controller">
        <property name="access-key" value="S3_ACCESS_KEY"/>
        <property name="secret-access-key"
value="S3_SECRET_ACCESS_KEY"/>
        <property name="location" value="S3_BUCKET_NAME"/>
      </discovery-option>
    </discovery-options>
  </local>
</domain-controller>
```

-

Rapporter un bogue

25.2.3. JBoss EAP 6 clusterisé

25.2.3.1. Instances clusterisées

Une instance clusterisée est une instance Amazon EC2 exécutant sur JBoss EAP 6 avec le clustering activé. Une autre instance exécutant sur le serveur Apache HTTP agira en tant que proxy pour les instances dans le cluster.

Les AMI de JBoss EAP 6 comprennent deux fichiers de configuration à utiliser dans les instances en cluster, **`standalone-ec2-ha.xml`** et **`standalone-mod_cluster-ec2-ha.xml`**. Chacun de ces fichiers de configuration fournit du clustering sans multidiffusion car Amazon EC2 ne prend pas en charge la multidiffusion. Cela se fait par monodiffusion TCP pour les communications de clusters et S3_PING comme protocole de découverte. La configuration **`autonome-mod_cluster-ec2-ha.xml`** fournit également un enregistrement simple par les proxys de mod_cluster.

De même, le fichier de configuration **`domain-ec2.xml`** fournit deux profils à utiliser dans les domaines gérés clusterisés : ec2-ha, et mod_cluster-ec2-ha.

Rapporter un bogue

25.2.3.2. Créer une instance de base de données de service de bases de données relationnelles.

Résumé

Cette rubrique couvre les étapes nécessaires pour créer une instance de base de données de service de bases de données relationnelles, en utilisant MySQL comme exemple.

AVERTISSEMENT

Il est hautement conseillé que les fonctionnalités de sauvegarde et de maintenance demeurent actives dans les environnements de production.

IMPORTANT

Il est de bonne pratique de créer des paires séparées utilisateur/mot de passe pour chaque application qui accède à la base de données. Régler les options de configuration selon les besoins de votre application.

Procédure 25.8. Créer une instance de base de données de service de bases de données relationnelles.

1. Cliquer sur le **RDS** de la console AWS.
2. Abonnez-vous au service si nécessaire.

3. Cliquer sur **`Launch DB instance`**.
4. Cliquer sur **`MySQL`**.
    a. Sélectionner une version, comme **`5.5.12`**.
    b. Sélectionner **`small instance`**.
    c. Veillez à ce que **`Multi-AZ Deployment`** et **`Auto upgrade`** soient désactivés : **`off`**.
    d. Définir **`Storage`** à **`5GB`**.
    e. Définir le nom d'utilisateur et le mot de passe de l'administrateur de système et cliquer sur le bouton **`Next`**.
    f. Sélectionner un nom de base de données à créer avec l'instance, et cliquer sur **`Next`**.
    g. Désactiver les back-ups et la maintenance, si nécessaire.
    h. Confirmer les paramètres.

Résultat

La base de données est alors créée. Elle s'initialisera et sera prête à l'utilisation dans quelques minutes.

Rapporter un bogue

25.2.3.3. Clouds privés virtuels

Amazon VPC (Amazon Virtual Private Cloud) est une fonctionnalité d'AWS (Amazon Web Service) qui vous permet d'isoler un ensemble de ressources AWS dans un réseau privé. La topologie et la configuration de ce réseau privé peuvent être personnalisées.

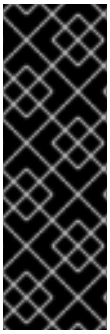
Voir le site Amazon Virtual Private Cloud pour obtenir plus d'informations http://aws.amazon.com/vpc/.

Rapporter un bogue

25.2.3.4. Créer un VPC (Virtual Private Cloud)

Résumé

Cette rubrique décrit les étapes requises pour créer un cloud privé virtuel, en prenant comme exemple une base de données externe au VPC. Vos stratégies de sécurité peuvent exiger la connexion à la base de données à crypter. Veuillez consulter *RDS FAQ* d'Amazon pour plus d'informations sur le cryptage des connexions de base de données.



IMPORTANT

Un VPC est recommandé pour une installation de cluster dans JBoss EAP 6 car cela simplifie grandement une communication sécurisée entre les nœuds du cluster, un Server JON et le proxy mod_cluster. Sans un VPC, ces canaux de communication doivent être chiffrés et authentifiés.

Pour obtenir des informations détaillées sur la façon de configurer SSL, voir : Section 11.12.1, « Implémentation du cryptage SSL pour le serveur de JBoss EAP 6. ».

1. Aller dans l'onglet VPC de la console AWS.
2. Abonnez-vous au service si nécessaire.
3. Cliquer sur "**`Create new VPC`**".
4. Sélectionner un VPC avec un sous-système public et un privé.
    a. Définir le sous-système public à **`10.0.0.0/24`**.
    b. Définir le sous-système privé à **`10.0.1.0/24`**.
5. Aller dans **`Elastic IPs`**.
6. Créer un IP élastique pour que l'instance mod_cluster proxy/NAT puisse l'utiliser.
7. Aller dans **`Security groups`** et créer un groupe de sécurité pour autoriser le trafic entrant et sortant.
8. Aller sur les ACL de réseau
    a. Créer un ACL pour autoriser le trafic entrant et sortant.
    b. Créer un ACL pour autoriser le trafic vers et depuis les ports TCP **`22`**, **`8009`**, **`8080`**, **`8443`**, **`9443`**, **`9990`** et **`16163`** uniquement.

Résultat

Le Cloud privé virtuel (VPC) a été créé.

Rapporter un bogue

25.2.3.5. Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

Résumé

Cette section couvre toutes les étapes requises pour lancer une instance de serveur Apache HTTP qui puisse servir de proxy mod_cluster et d'instance NAT au Virtuel Private Cloud.

Conditions préalables

- Section 25.2.3.2, « Créer une instance de base de données de service de bases de données relationnelles. ».
- Section 25.2.3.4, « Créer un VPC (Virtual Private Cloud) »

Procédure 25.9. Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

1. Créer un IP élastique pour cette instance.
2. Sélectionner une AMI.
3. Allez dans le **`Security Group`** et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).

4. Choisir "**`running`**" dans le sous-système public du VPC.

5. Sélectionner un IP statique (comme par ex **`10.0.0.4`**).

6. Mettez ce qui suit dans le champ **`User Data:`**

```
JBOSSCONF=disabled

cat > $USER_SCRIPT << "EOS"

echo 1 > /proc/sys/net/ipv4/ip_forward
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter

iptables -I INPUT 4 -s 10.0.1.0/24 -p tcp --dport 7654 -j ACCEPT
iptables -I INPUT 4 -p tcp --dport 80 -j ACCEPT

iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -I FORWARD -s 10.0.1.0/24 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 ! -s 10.0.0.4 -j MASQUERADE

# balancer module incompatible with mod_cluster
sed -i -e 's/LoadModule proxy_balancer_module/#\0/'
/etc/httpd/conf/httpd.conf

cat > /etc/httpd/conf.d/mod_cluster.conf << "EOF"
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so

Listen 7654

# workaround JBPAPP-4557
MemManagerFile /var/cache/mod_proxy/manager

<VirtualHost *:7654>
   <Location /mod_cluster-manager>
      SetHandler mod_cluster-manager
      Order deny,allow
      Deny from all
      Allow from 127.0.0.1
   </Location>

   <Location />
      Order deny,allow
      Deny from all
      Allow from 10.
      Allow from 127.0.0.1
   </Location>

   KeepAliveTimeout 60
   MaxKeepAliveRequests 0
   ManagerBalancerName mycluster
```

```
   ServerAdvertise Off
   EnableMCPMReceive On
</VirtualHost>
EOF

echo "`hostname | sed -e 's/ip-//' -e 'y/-/./'`        `hostname`"
>> /etc/hosts

semanage port -a -t http_port_t -p tcp 7654 #add port in the apache
port list for the below to work
setsebool -P httpd_can_network_relay 1 #for mod_proxy_cluster to
work
chcon -t httpd_config_t -u system_u
/etc/httpd/conf.d/mod_cluster.conf

#### Uncomment the following line when launching a managed domain
####
# setsebool -P httpd_can_network_connect 1

service httpd start

EOS
```

7. Décochez la case Amazon EC2 cloud source/destination pour cette instance pour qu'elle puisse agir en tant que router.

   a. Cliquer à droite sur l'instance de serveur Apache HTTP et sélectionner "**`Change Source/Dest check`**".

   b. Cliquer sur **`Yes, Disable`**.

8. Créer un IP élastique pour cette instance.

Résultat

L'instance de serveur Apache HTTP aura été lancée avec succès.

Rapporter un bogue

25.2.3.6. Configurer le routage par défaut du sous-système privé VPC

Résumé

Cette rubrique couvre les étapes requises pour configurer le routage par défaut du sous-système privé VPC. Les noeuds de cluster de JBoss EAP 6 exécuteront dans le sous-système privé du VPC, mais les noeuds de cluster ont besoin d'un accès internet pour la connectivité S3. Un routage par défaut doit être défini pour aller dans l'instance NAT.

Procédure 25.10. Configurer le routage par défaut du sous-système privé VPC

1. Naviguer dans l'instance de serveur Apache HTTP de la console Amazon AWS.
2. Naviguer dans **VPC** → **tables de routage**.
3. Cliquer sur le tableau de routage utilisé par le sous-système privé.
4. Dans le champ de nouveau routage, saisir **`0.0.0.0/0`**.

5. Cliquer sur "**`Select a target`**".
6. Sélectionner "**`Enter Instance ID`**".
7. Choisir l'ID de l'instance du serveur Apache HTTP en cours d'exécution.

Résultat

Le routage par défaut a été configuré correctement pour le sous-système VPC.

Rapporter un bogue

25.2.3.7. IAM (Identity and Access Management)

IAM (Identity and Access Management) fournit une sécurité configurable pour vos ressources AWS. IAM peut être configuré pour utiliser les comptes créés dans IAM ou pour fournir une fédération d'identité entre IAM et vos propres services d'identité.

Consultez le site web AWS Identity and Access Management pour plus d'informations http://aws.amazon.com/iam/.

Rapporter un bogue

25.2.3.8. Configurer l'installation IAM

Résumé

Cette rubrique couvre les étapes de configuration requises pour installer IAM pour les instances de JBoss EAP 6. Le protocole ***`S3_PING`*** utilise le compartiment S3 pour découvrir d'autres membres du cluster. La version 3.0.x de **JGroups** a besoin d'un compte d'accès Amazon AWS et de clés secrètes pour s'authentifier dans le service S3.

Comme la découverte de contrôleur de domaine S3 utilise le compartiment S3, il a besoin d'un accès à un compte Amazon AWS et à des clés secrètes pour authentifier le service S3 (similaire au protocole ***`S3_PING`*** utilisé par JGroups). L'utilisateur IAM et le compartiment S3 utilisés pour la découverte de S3 doivent être différents de l'utilisateur IAM et du compartiment S3 utilisés pour le clustering.

Il y a un risque de sécurité à entrer vos informations d'identification du compte principal dans le domaine de l'utilisateur des données, de les stocker en ligne ou dans une AMI. Pour contourner cela, un compte distinct peut être créé en utilisant la fonction Amazon IAM qui donnerait seulement accès à un seul compartiment de S3.

Procédure 25.11. Configurer l'installation IAM

1. Aller dans l'onglet IAM de la console AWS.
2. Cliquer sur **`users`** (utilisateurs).
3. Sélectionner **`Create New Users`** (Créer Nouveaux Utilisateurs).
4. Choisir un nom, et veillez à ce que l'option **`Generate an access key for each User`** (Générer une clé d'accès pour chaque utilisateur) soit cochée.
5. Sélectionner **`Download credentials`**, et les sauvegarder dans un emplacement sécurisé.
6. Fermer la fenêtre.

7. Cliquer sur un utilisateur nouvellement créé.

8. Prenez note de la valeur de **`User ARM`**. Cette valeur est requise pour configurer Bucket S3, et est documentée ici: Section 25.2.3.10, « Configurer l'installation S3 Bucket ».

Résultat

Le compte IAM a été créé avec succès.

Rapporter un bogue

25.2.3.9. S3 Bucket

Les compartiments S3 représentent une unité de stockage d'organisation de base d'Amazon S3 (Amazon Simple Storage System). Un compartiment peut stocker un certain nombre d'objets arbitraires et doit posséder un nom unique pour l'identifier avec Amazon S3.

Consulter le site web Amazon S3 pour obtenir plus d'informations, http://aws.amazon.com/s3/.

Rapporter un bogue

25.2.3.10. Configurer l'installation S3 Bucket

Résumé

Cette rubrique couvre les étapes nécessaires de configuration d'un nouveau compartiment S3.

Conditions préalables

- Section 25.2.3.8, « Configurer l'installation IAM ».

Procédure 25.12. Configurer l'installation S3 Bucket

1. Ouvrir l'onglet **`S3`** dans la console AWS.

2. Cliquer sur **`Créer Bucket`**.

3. Choisir un nom pour le compartiment et cliquer sur **`Create`** (Créer).

NOTE

Les noms de compartiments sont uniques dans tout S3. Les noms ne peuvent pas être réutilisés.

4. Cliquer à droite sur le nouveau compartiment et sélectionner **`Properties`** (propriétés).

5. Cliquer sur **`Add bucket policy`** (ajouter la règle de compartiment) dans l'onglet de permissions.

6. Cliquer sur **`New policy`** (Nouvelle police) pour ouvrir l'assistant de création de police.

   a. Copier le texte suivant dans la nouvelle police, en remplaçant **`arn:aws:iam::05555555555:user/jbosscluster*`** par la valeur définie ici : Section 25.2.3.8, « Configurer l'installation IAM ». Changer les deux instances de **`clusterbucket123`** au nom de compartiment défini dans l'étape 3 de cette procédure.

```
{
    "Version": "2008-10-17",
    "Id": "Policy1312228794320",
    "Statement": [
        {
            "Sid": "Stmt1312228781799",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::055555555555:user/jbosscluster"
                ]
            },
            "Action": [
                "s3:ListBucketVersions",
                "s3:GetObjectVersion",
                "s3:ListBucket",
                "s3:PutBucketVersioning",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:GetBucketVersioning"
            ],
            "Resource": [
                "arn:aws:s3:::clusterbucket123/*",
                "arn:aws:s3:::clusterbucket123"
            ]
        }
    ]
}
```

Résultat

Un nouveau compartiment a maintenant été créé, et configuré.

Rapporter un bogue

25.2.3.11. Instances clusterisées

25.2.3.11.1. Lancer les AMI de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes requises pour lancer les AMI JBoss EAP 6.

Conditions préalables

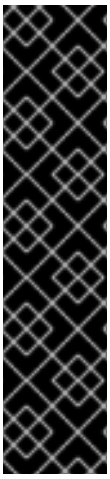
- Section 25.2.3.2, « Créer une instance de base de données de service de bases de données relationnelles. ».
- Section 25.2.3.4, « Créer un VPC (Virtual Private Cloud) ».
- Section 25.2.3.5, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC ».

- Section 25.2.3.6, « Configurer le routage par défaut du sous-système privé VPC ».
- Section 25.2.3.8, « Configurer l'installation IAM ».
- Section 25.2.3.10, « Configurer l'installation S3 Bucket ».

AVERTISSEMENT

Exécuter le cluster JBoss EAP 6 dans un sous-système avec un masque de réseau inférieur à 24 bits ou bien fractionner plusieurs sous-systèmes compliquent l'obtention d'un ID homologue unique pour chaque membre du cluster.

Voir la variable ***`JBOSS_CLUSTER_ID`*** pour obtenir des informations sur la façon d'effectuer ce travail de configuration de façon fiable : Section 25.4.1, « Paramètres de configuration permanente ».



IMPORTANT

La fonctionnalité de Amazon EC2 AutoScaling peut être utilisée avec des nœuds de cluster JBoss EAP 6. Cependant, s'assurer que c'est testé **avant** le déploiement. Vous devez vous assurer que vos charges de travail particulières sont à l'échelle du nombre de nœuds désirés, et que la performance répond à vos besoins avant d'envisager d'utiliser le type d'instance (des types d'instances différentes reçoivent une part de ressources cloud EC2 différentes).

De plus, l'emplacement de l'instance et l'utilisation machine/RDS réseau/stockage/machine hôte/RDS peuvent affecter la performance d'un cluster. Tester avec vos charges réelles attendues pour essayer de pallier à l'avance aux conditions inattendues.

AVERTISSEMENT

L'action *scale-down* Amazon EC2 termine les nœuds sans élégance, et, comme certaines transactions pourraient être interrompues, les autres nœuds de cluster (et équilibreurs de charge) auront besoin de temps pour basculer. Cela est susceptible d'influencer l'expérience des utilisateurs de votre application.

Il est recommandé que vous réduisiez le cluster d'applications manuellement en désactivant le serveur de l'interface de gestion du mod_cluster jusqu'à ce que toutes les sessions traitées soient complétées, ou bien que vous fermiez l'instance JBoss EAP 6 avec grâce (on peut utiliser l'accès SSH vers l'instance ou JON).

Vérifier que votre procédure de réduction choisie n'ait pas d'effets négatifs sur l'expérience utilisateur. Il est possible que vous ayez besoin de prendre des mesures supplémentaires pour certaines charges de travail, équilibrages de charge ou installations.

Procédure 25.13. Lancer les AMI de JBoss EAP 6 clusterisée

1. Sélectionner une AMI.
2. Définir le nombre d'instances voulues (la taille du cluster).
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe **`Security Group`**.
5. Veillez à ce que tout le trafic en provenance du cluster JBoss EAP 6 soit autorisé.
6. Définir les autres restrictions suivant les besoins.
7. Ajouter ce qui suit dans le champ **`User Data`** :

Exemple 25.7. Exemple de champ de données utilisateur

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## password to access admin console
JBOSSAS_ADMIN_PASSWORD=<your password for opening admin console>

## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed
####
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar
```

```
cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy sample application from local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/cluster-
demo.war

## ExampleDS configuration for MySQL database
data-source remove --name=ExampleDS
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name="mysql",driver-module-name="com.mysql")
data-source add --name=ExampleDS --connection-
url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/subsystem=datasources/data-source=ExampleDS:enable
/subsystem=datasources/data-source=ExampleDS:test-connection-in-
pool
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

Résultat

Les AMI JBoss EAP 6 clusteriseés ont été configurées et lancées avec succès.

Rapporter un bogue

25.2.3.11.2. Lancer l'instance de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes pour s'assurer que les instances de la plateforme clusterisée de JBoss EAP 6 exécutent correctement.

Procédure 25.14. Tester l'instance clusterisée

1. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD pour confirmer que le serveur web exécute correctement.

2. **Tester les noeuds clusterisés**

   a. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/put.jsp à l'aide d'un navigateur.

   b. Vérifier que l'un des noeuds de cluster journalise le message suivant :

   ```
   Putting date now
   ```

   c. Stopper le noeud de cluster qui a journalisé le message dans l'étape précédente.

   d. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/get.jsp à l'aide d'un navigateur.

   e. Vérifier que l'heure indiquée est la même que l'heure PUT (mise) par **`put.jsp`** à l'étape 2-a.

   f. Vérifier que l'un des noeuds de cluster en cours d'exécution journalise le message suivant :

   ```
   Getting date now
   ```

   g. Démarrer à nouveau le noeud clusterisé qui est arrêté.

   h. Connectez-vous à l'instance Apache HTTP :

   ```
   ssh -L7654:localhost:7654 <ELASTIC_IP_OF_APACHE_HTTPD>
   ```

   i. Naviguez dans http://localhost:7654/mod_cluster-manager pour confirmer que toutes les instances exécutent correctement.

Résultat

Les instances clusterisées de JBoss EAP 6 ont été testées, et il a été confirmé qu'elle exécutait correctement.

Rapporter un bogue

25.2.3.12. Domaines gérés clusterisés

25.2.3.12.1. Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster

Résumé

Cette rubrique couvre les étapes requises pour lancer un domaine géré clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Conditions préalables

- Une Red Hat AMI qui convient. Voir Section 25.1.6, « Les AMI Red Hat prises en charge » .
- Section 25.2.3.4, « Créer un VPC (Virtual Private Cloud) »

- Section 25.2.3.5, « Lancer une instance de serveur Apache HTTP pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »
- Section 25.2.3.6, « Configurer le routage par défaut du sous-système privé VPC »
- Section 25.2.3.8, « Configurer l'installation IAM »
- Section 25.2.3.10, « Configurer l'installation S3 Bucket »

Procédure 25.15. Lancer un contrôleur de domaine clusterisé

1. Créer un IP élastique pour cette instance.
2. Sélectionner une AMI.
3. Allez dans le groupe de sécurité et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).
4. Choisir "running" dans le sous-système public du VPC.
5. Sélectionner un IP statique (comme par ex **`10.0.0.5`**).
6. Mettez ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.0.

## S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

#### to run the example no modifications below should be needed ####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app name>.war
-O /usr/share/java/jboss-ec2-eap-applications/<app name>.war

## Install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar
```

```
cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/cluster-demo.war --
server-groups=other-server-group

## ExampleDS configuration for MySQL database
data-source --profile=mod_cluster-ec2-ha remove --name=ExampleDS
/profile=mod_cluster-ec2-ha/subsystem=datasources/jdbc-
driver=mysql:add(driver-name="mysql",driver-module-name="com.mysql")
data-source --profile=mod_cluster-ec2-ha add --name=ExampleDS --
connection-url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/profile=mod_cluster-ec2-ha/subsystem=datasources/data-
source=ExampleDS:enable
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

7. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

   ```
   yum -y update
   ```

   NOTE

   `yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

8. Lancement de l'instance Red Hat AMI

Résultat

Un domaine géré clusterisé de JBoss EAP 6 a été configuré, et lancé sur une Red Hat AMI.

Rapporter un bogue

25.2.3.12.2. Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs hôtes de cluster

Résumé

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss EAP 6 en tant que contrôleurs hôtes clusterisés sur une AMI de Red Hat (Amazon Machine Image de Red Hat).

Conditions préalables

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter Section 25.2.3.12.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster » .
- Section 25.2.3.8, « Configurer l'installation IAM »
- Section 25.2.3.10, « Configurer l'installation S3 Bucket »

Procédure 25.16. Lancer les contrôleur hôte

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

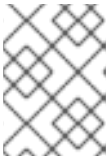
1. Sélectionner une AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe de sécurité.
5. Veillez à ce que tout le trafic en provenance du cluster JBoss EAP 6 soit autorisé.
6. Définir les autres restrictions suivant les besoins.
7. Ajouter ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## host controller setup
### static domain controller discovery setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
### S3 domain controller discovery setup
# JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY=<your secret key>
# JBOSS_DOMAIN_S3_ACCESS_KEY=<your access key>
# JBOSS_DOMAIN_S3_BUCKET=<your bucket name>

JBOSS_HOST_PASSWORD=<password for slave host controllers>
```

```
## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed ####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/main-server-group/other-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

8. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```



NOTE

`yum -y update` doit être exécuté régulièrement pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance AMI Red Hat

Résultat

Les contrôleurs hôtes clusterisés de JBoss EAP 6 ont été configurés, et lancés sur une AMI Red Hat.

Rapporter un bogue

25.2.3.12.3. Tester le domaine géré de JBoss EAP 6 clusterisée

Résumé

Cette rubrique couvre les étapes requises pour tester le domaine géré clusterisé de JBoss EAP 6 sur une Red Hat AMI (Amazon Machine Image de Red Hat)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTP et du contrôleur de domaines de JBoss EAP 6 à la fois.

Conditions préalables

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter Section 25.2.3.12.1, « Lancer une instance pour qu'elle serve de contrôleur de domaine de cluster » .
- Configurer et lancer les contrôleur hôte du cluster. Consulter Section 25.2.3.12.2, « Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs hôtes de cluster » .

Procédure 25.17. Tester l'instance de serveur Apache HTTP

- Naviguer dans **`http://ELASTIC_IP_OF_APACHE_HTTP_SERVER`** avec un navigateur pour confirmer que le serveur web exécute avec succès.

Procédure 25.18. Tester le contrôleur de domaine

1. Naviguez dans **`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console`**
2. Connectez-vous en utilisant le nom d'utilisateur **`admin`** et le mot de passe spécifiés dans le champ Données d'utilisateur pour le contrôleur de domaine. Une fois connecté, la page d'accueil de la console d'administration d'un domaine géré s'affichera (**`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances`**).
3. Cliquer sur l'étiquette **`Server`** du serveur en haut et à droite de l'écran. Sélectionner un contrôleur hôte dans le menu déroulant **`Host`** en haut et à gauche de l'écran.
4. Vérifier que ce contrôleur hôte ait deux configurations de serveurs nommées respectivement **`server-one`** et **`server-two`** et vérifier qu'elles appartiennent toutes deux à **`other-server-group`**.

Procédure 25.19. Tester les contrôleurs hôtes

1. Naviguez dans **`http://ELASTIC_IP_OF_APACHE_HTTP_SERVER/cluster-demo/put.jsp`**

2. Vérifier que l'un des contrôleurs hôtes journalise le message suivant : **`Putting date now.`**

3. Stopper l'instance du serveur qui a journalisé le message dans l'étape précédente (voir *Stop a Server Using the Management Console*).

4. Naviguez dans **`http://ELASTIC_IP_OF_APACHE_HTTP_SERVER/cluster-demo/get.jsp`**.

5. Vérifier que l'heure indiquée est la même que l'heure **`PUT`** (mise) par **`put.jsp`** à l'étape 2-a.

6. Vérifier que l'une des instances de serveur en cours d'exécution journalise le message suivant : **`Getting date now.`**

7. Re-démarrer l'instance du serveur arrêtée (voir la section 2.2.2, Démarrer un serveur par la console de gestion).

8. Connectez-vous à l'instance Apache HTTP .

```
$ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTP_SERVER
```

9. Naviguer dans **`http://localhost:7654/mod_cluster-manager`** pour confirmer que toutes les instances exécutent correctement.

Résultat

Le serveur web de JBoss EAP 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur une AMI de Red Hat.

Rapporter un bogue

25.3. METTRE EN PLACE LE MONITORING DANS JBOSS OPERATIONS NETWORK (JON)

25.3.1. AMI Monitoring

Une fois que vous avez votre application commerciale déployée dans une instance AMI configurée correctement, l'étape suivante est d'instaurer le monitoring de la plateforme avec JON (JBoss Operations Network).

Le serveur JON est généralement situé à l'intérieur d'un réseau d'entreprise, donc il est nécessaire d'établir une connexion sécurisée entre le serveur et chacun de ses agents. La création d'un réseau privé virtuel entre les deux points est la solution la plus courante, mais cela complique la configuration de réseau requise. Ce chapitre fournit des directives de configuration de réseau permettant d'établir la communication entre l'agent de JON et le serveur JON. Pour plus d'informations sur la configuration, la gestion et l'utilisation, veuillez vous référer à la documentation officielle de Red Hat pour JBoss Operations Network (JON).

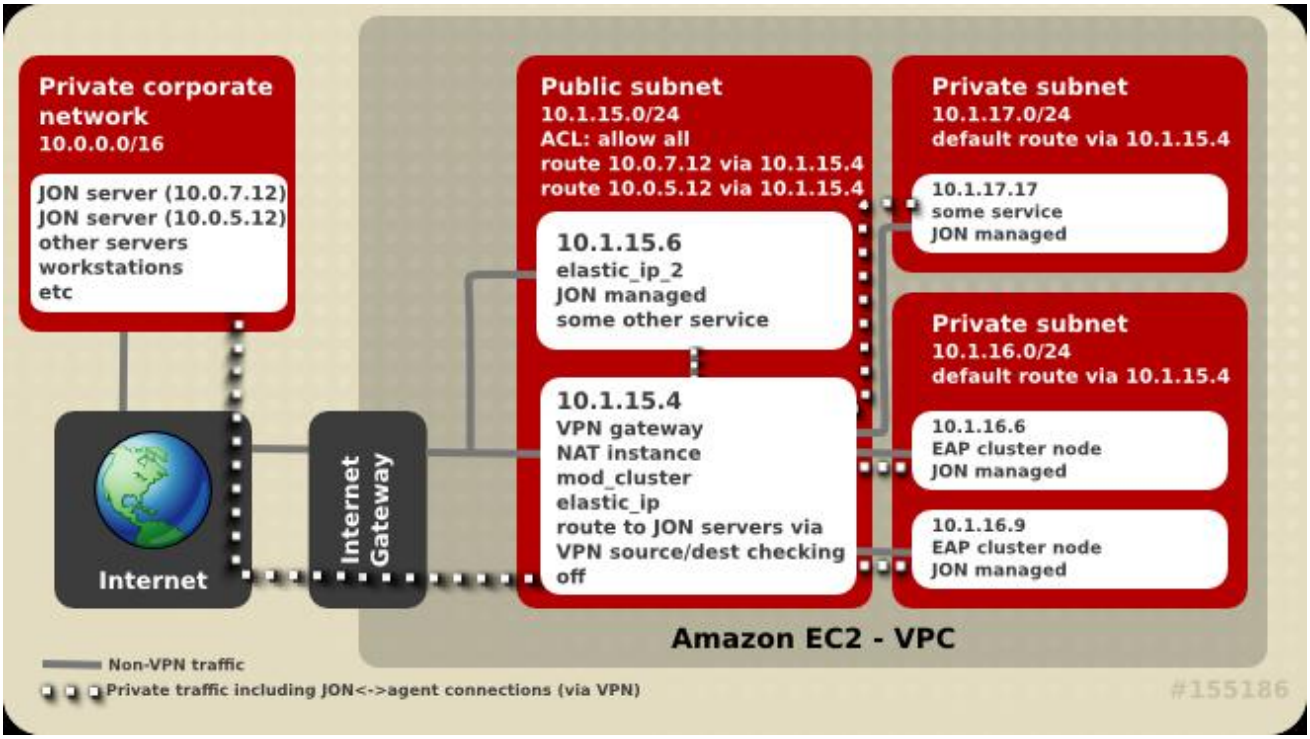


Figure 25.1. La connectivité du serveur JON

Rapporter un bogue

25.3.2. Prérequis de connectivité

L'inscription d'un agent JON avec ses serveurs requiert une communication bidirectionnelle entre l'agent et les serveurs. L'agent JON a besoin d'accéder au port **`7080`** sur tous les serveurs de JON, sauf dans le cas de SSL quand le port **`7443`** est utilisé. Chaque serveur de JON doit pouvoir accéder à chacun des agents connectés sur un hôte unique et son port homologue. Le port de l'agent est habituellement **`16163`**.

S'il y a plusieurs serveurs JON clusterisés, veillez à ce que chaque agent puisse communiquer avec tous les serveurs dans le cluster JON via les paires IP et nom d'hôte comme configurés par la console d'administration du serveur JON. Le serveur JON utilisé par l'agent à enregistrer n'est sans doute pas le serveur qu'il essaie d'utiliser après l'initialisation.

Rapporter un bogue

25.3.3. Network Address Translation (NAT)

Une passerelle VPN entreprise agissant en mode routé simplifie grandement la configuration du réseau. Si toutefois votre gateway VPN entreprise opère en mode NAT, le serveur JON n'a pas de visibilité directe des agents. Dans ce cas, le réacheminement de port devra être configuré pour chaque agent.

Les configurations NAT VPN ont besoin d'un port sur la passerelle à transmettre à l'adresse de port de l'agent JON sur l'ordinateur dont il s'agit. L'agent JON doit également être configuré pour indiquer au serveur le numéro de port transféré et l'adresse IP. Vous trouverez de plus amples informations dans la description de **`rhq.communications.connector.*`** du fichier de configuration de **`agent-configuration.xml`**

Rapporter un bogue

25.3.4. Amazon EC2 et DNS

Les serveurs JON et les agents JON doivent être en mesure de résoudre les noms d'hôtes des uns et des autres. La résolution DNS est plus compliquée dans le cas d'une configuration VPN. Les serveurs connectés ont plusieurs options possibles. Une des options consiste à utiliser les serveurs DNS du réseau de l'entreprise ou Amazon EC2. Une autre option est d'utiliser une configuration divisée de DNS avec les serveurs DNS de l'entreprise utilisés pour résoudre les noms dans des domaines particuliers, et les serveurs d'Amazon EC2 DNS utilisés pour la résolution de tous les autres noms.

Rapporter un bogue

25.3.5. Le routage dans EC2

Tous les serveurs de Amazon EC2 ont une fonctionnalité de routage **`source/destination checking`** activée par défaut. Cette fonction supprime tous les paquets envoyés au serveur qui ont une destination séparée de l'adresse IP de la machine. Si la solution VPN sélectionnée pour la connexion des agents au serveur JON inclut un routeur, cette fonctionnalité devra être désactivée pour le ou les serveur(s) agissant en tant que routeurs ou passerelles VPN. Ce paramètre de configuration est accessible via la console d'Amazon AWS. La désactivation de **`source/destination checking`** est également requise dans un cloud privé virtuel (VPC).

Certaines configurations VPN renvoient le trafic internet général par le VPN entreprise par défaut. Il est conseillé de l'éviter car cela risque de ralentir et de rendre moins la configuration moins performante pour vos besoins particuliers.

Malgré que l'utilisation d'un schéma d'adressage approprié n'est pas un problème spécifique à JON, les mauvais schémas peuvent l'affecter. Amazon EC2 attribue des adresses IP à partir du réseau **`10.0.0.0/8`**. Les instances ont généralement une adresse IP publique également, mais seul le trafic réseau sur l'adresse IP interne dans la même zone de disponibilité est gratuit. Pour éviter d'utiliser le réseau **`10.0.0.0/8`** en adressage privé, il y a plusieurs choses à considérer.

- Quand vous créez un VPC, évitez d'allouer des adresses déjà utilisées dans le réseau privé afin d'éviter les problèmes de connectivité.
- Si une instance a besoin d'accéder à des ressources locales de zone disponible, veillez à ce que les adresses privées Amazon EC2 soient utilisées et que le trafic ne soit pas dirigé par le VPN.
- Si une instance d'Amazon EC2 a accès à un petit sous-ensemble d'adresses de réseau privé d'entreprise (par exemple les serveurs JON uniquement), seules ces adresses devront être acheminées par le VPN. Cela augmentera la sécurité et réduira les chances de collision d'espaces d'adresse de réseau privé et Amazon EC2.

Rapporter un bogue

25.3.6. Quitter ou Re-démarrer JON

Un des avantages des environnements de Cloud Computing est la facilité avec laquelle vous pouvez résilier ou lancer une instance de la machine. Vous pouvez également lancer une instance identique à l'instance initiale. Cela peut entraîner des problèmes si la nouvelle instance tente de s'enregistrer par les serveurs JON en utilisant le même nom d'agent que celui de l'agent déjà en cours d'exécution. Dans un tel cas, le serveur JON ne permettra pas à un agent de reconnecter avec un token d'identification manquant ou non correspondant.

Afin d'éviter cela, veillez à ce que les agents qui ont fait leur travail soient retirés de l'inventaire JON avant d'essayer de connecter un agent du même nom ou de spécifier le token d'identification qui convient quand vous démarrerez un nouvel agent.

Un autre problème que vous pourriez rencontrer est lorsqu'une machine d'agent reçoit une nouvelle adresse IP VPN qui ne correspond plus à l'adresse enregistrée dans la configuration de JON. Un exemple pourrait inclure une machine qui redémarre ou lorsque une connexion VPN a été interrompue. Dans ce cas, il est recommandé que vous liez le cycle de vie de l'agent JON au cycle de vie de la connexion VPN. Si la connexion tombe, vous pouvez arrêter l'agent. Lorsque la connexion est rétablie à nouveau, mettre à jour **`JON_AGENT_ADDR`** dans **`/etc/sysconfig/jon-agent-ec2`** pour refléter la nouvelle adresse IP, puis redémarrez l'agent.

Les informations sur la façon de changer l'adresse IP de l'agent se trouve dans le guide «Configuring JON Servers and Agents Guide» à l'adresse suivante https://access.redhat.com/site/documentation/JBoss_Operations_Network/.

S'il existe un grand nombre d'instances lancées ou résiliées, cela risque d'être difficile d'ajouter ou de supprimer les instances manuellement dans l'inventaire de JON. Les capacités de scripting de JON peuvent être utilisées pour automatiser ces étapes. Voir la documentation JON pour plus d'informations.

Rapporter un bogue

25.3.7. Configurer une instance pour vous enregistrer dans le JBoss Operations Network

Utiliser la procédure suivante pour enregistrer une instance de JBoss EAP 6 dans JBoss Operations Network.

- Dans JBoss EAP 6, ajouter ceci dans le champ User Data (données utilisateur).

```
JON_SERVER_ADDR=jon2.it.example.com
## if instance not already configured to resolve its hostname
JON_AGENT_ADDR=`ip addr show dev eth0 primary to 0/0 | sed -n
's#.*inet \([0-9.]\+\)/.*#\1#p'`
PORTS_ALLOWED=16163
# insert other JON options when necessary.
```

  Voir les paramètres Section 25.4.1, « Paramètres de configuration permanente », commençant par ***`JON`***_ pour le format des options JON.

Rapporter un bogue

25.4. CONFIGURATION DU SCRIPT UTILISATEUR

25.4.1. Paramètres de configuration permanente

Résumé

Les paramètres suivants peuvent être utilisés pour influencer la configuration et les opérations de JBoss EAP 6. Leur contenu se trouve dans **`/etc/sysconfig/jbossas`** et **`/etc/sysconfig/jon-agent-ec2`**.

Tableau 25.2. Paramètres configurables

Nom	Description	Par défaut
JBOSS_JGROUPS_S3_PING_ACCESS_KEY	Clé d'accès de compte utilisateur Amazon AWS pour S3_PING Discovery quand on utilise le clustering.	S/O
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY	Clé d'accès secrète au compte utilisateur Amazon AWS	S/O
JBOSS_JGROUPS_S3_PING_BUCKET	Amazon S3 Bucket à utiliser dans S3_PING Discovery.	S/O
JBOSS_CLUSTER_ID	ID des noeuds de membres d'un groupement. Utilisé uniquement pour le clustering. Les valeurs acceptées sont (dans l'ordre) : • Un nombre d'ID de groupement valide entre **`0`** - **`1023`**. • Un nom d'interface de réseau, avec le dernier octet de l'IP utilisé comme valeur. • "S3" comme valeur coordonnerait l'utilisation de l'ID par la S3 Bucket utilisée par S3_PING des jgroups. Il est conseillé d'utiliser le dernier octet de l'IP (par défaut) quand tous les noeuds de cluster sont situés dans le sous-système de 24 octets ou davantage (par exemple, dans un sous-ensemble VPC).	Dernier octet de l'adresse IP d'eth0
MOD_CLUSTER_PROXY_LIST	Liste délimitée par des virgules d'IP/Noms d'hôte de proxies mod_cluster si mod_cluster doit être utilisé.	S/O
PORTS_ALLOWED	Liste des ports entrants qui seront utilisés par le pare-feu en plus des ports par défaut.	S/O
JBOSSAS_ADMIN_PASSWORD	Mot de passe pour l'utilisateur **`admin`**.	S/O

Nom	Description	Par défaut
JON_SERVER_ADDR	IP ou nom d'hôte du serveur JON dans lequel s'enregistrer. Uniquement utilisé pour l'enregistrement, ensuite, l'agent peut communiquer avec les autres serveurs dans le groupement JON.	S/O
JON_SERVER_PORT	Port utilisé par l'agent pour communiquer avec le serveur.	7080
JON_AGENT_NAME	Nom de l'agent JON, doit être unique.	ID de l'instance
JON_AGENT_PORT	Port que l'agent écoute.	16163
JON_AGENT_ADDR	Adresse IP à laquelle l'agent JON est relié. Utilisé quand le serveur a plus d'une adresse publique, (par ex VPN).	L'agent JON choisit l'IP ou le nom d'hôte local par défaut.
JON_AGENT_OPTS	Propriétés de système d'agent JON supplémentaire pouvant être utilisé pour configurer SSL, NAT et d'autres paramètres avancés.	S/O

Nom	Description	Par défaut
JBOSS_SERVER_CONFIG	Nom du fichier de configuraiton de serveur JBoss EAP 6 à utiliser. Si JBOSS_DOMAIN_CONTROLLER =true, alors **`domain-ec2.xml`** sera utilisé. Sinon : • Si la config S3 est présente, alors **`standalone-ec2-ha.xml`** sera utilisé. • Si MOD_CLUSTER_PROXY_LIST est spécifié, alors **`standalone-mod_cluster-ec2-ha.xml`** sera sélectionné. • Si aucune des deux premières options n'est utilisée, alors le fichier **`standalone.xml`** sera utilisé. • Peut également être défini à **`standalone-full.xml`**.	**`standalone.xml`**, **`standalone-full.xml`**, **`standalone-ec2-ha.xml`**, **`standalone-mod_cluser-ec2-ha.xml`**, **`domain-ec2.xml`** suivant les autres paramètres.
JAVA_OPTS	Valeurs personnalisées à ajouter à la variable avant que JBoss EAP 6 démarre.	JAVA_OPTS est créé à partir de valeurs appartenant à d'autres paramètres.
JBOSS_IP	Adresse IP à laquelle le serveur est lié.	127.0.0.1
JBOSSCONF	Le nom du profil de JBoss EAP 6 pour démarrer. Pour empêcher JBoss EAP 6 de démarrer, JBOSSCONF peut être défini à **`disabled`**	**`standalone`**
JBOSS_DOMAIN_CONTROLLER	Indique si cette instance exécute ou non comme contrôleur de domaine.	**`false`**
JBOSS_DOMAIN_MASTER_ADDRESS	Adresse IP de contrôleur de domaine distant.	S/O
JBOSS_HOST_NAME	Le nom d'hôte logique (du domaine). Doit être un nom séparé.	La valeur de la variable d'environnement HOSTNAME.

Nom	Description	Par défaut
JBOSS_HOST_USERNAME	Le nom d'utilisateur du contrôleur hôte doit être utilisé quand on enregistre dans le contrôleur de domaine. Si non fourni, le JBOSS_HOST_NAME sera utilisé à la place.	JBOSS_HOST_NAME
JBOSS_HOST_PASSWORD	Le mot de passe que le contrôleur hôte doit utiliser quand il s'enregistre avec le contrôleur de domaine.	S/O
JBOSS_HOST_CONFIG	Si JBOSS_DOMAIN_CONTROLLER =true, alors **`host-master.xml`** sera utilisé. Si JBOSS_DOMAIN_MASTER_ADDRESS est présent, alors **`host-slave.xml`** sera utilisé.	**`host-master.xml`** ou **`host-slave.xml`**, suivant les autres paramètres.
JBOSS_DOMAIN_S3_ACCESS_KEY	Clé d'accès de compte utilisateur AWS Amazon pour la découverte de contrôleurs de domaines S3.	S/O
JBOSS_DOMAIN_S3_SECRET_ACCESS_KEY	Clé secrète d'accès de compte utilisateur AWS Amazon pour la découverte de contrôleurs de domaines S3.	S/O
JBOSS_DOMAIN_S3_BUCKET	Amazon S3 Bucket à utiliser pour la découverte de contrôleur de domaine S3.	S/O

Rapporter un bogue

25.4.2. Paramètres de scripts personnalisés

Résumé

Les paramètres suivants peuvent être utilisés dans la section de personnalisation utlisateur du champ **`User Data:`**.

Tableau 25.3. Paramètres configurables

Nom	Description
JBOSS_DEPLOY_DIR	Déployer le répertoire du profil actif (par exemple, **`/var/lib/jbossas/standalone/deployments/`**). Les archives déployables de ce répertoire seront déployées. Red Hat recommande d'utiliser la console de gestion ou le CLI pour gérer les déploiements au lieu d'utiliser le répertoire de déploiement.

Nom	Description
JBOSS_CONFIG_DIR	Répertoire de config du profile actif (par exemple, **`/var/lib/jbossas/standalone/configuration`**).
JBOSS_HOST_CONFIG	Nom du fichier de configuration de l'hôte actif (par exemple, **`host-master.xml`**). Red Hat conseille d'utiliser la console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration.
JBOSS_SERVER_CONFIG	Nom du fichier de configuration du serveur actif (par exemple, **`standalone-ec2-ha.xml`**). Red Hat conseille d'utiliser la console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration.
USER_SCRIPT	Chemin d'accès au script de configuration personnalisé disponible avant de trouver la configuraiton user-data.
USER_CLI_COMMANDS	Chemin d'accès à un fichier personnalisé des commandes CLI, qui est disponible avant de trouver la configuration user-data.

Rapporter un bogue

25.5. RÉSOLUTION DE PROBLÈMES

25.5.1. Résolution de problèmes dans Amazon EC2

EC2 donne un statut « Alarm » pour chaque instance, indiquant un dysfonctionnement grave d'instance, mais l'absence de ce statut d'alarme ne donne aucune garantie que l'instance a démarré correctement ou que les services s'exécutent correctement. Il est possible d'utiliser Amazon Cloud Watch avec ses fonctionnalités de métriques personnalisées pour surveiller la santé des instance de services mais l'utilisation d'une solution de gestion d'entreprise est conseillée.

Pour simplifier la résolution de problème, Red Hat conseille de gérer l'instance EC2 par JON (JBoss Operations Network) qui peut automatiquement découvrir, surveiller et gérer de nombreux services sur une instance EC2 avec l'agent JON installé, y compris JBoss Enterprise Application Platform et ses services installés, avec : JBoss EAP 6, Tomcat, Apache HTTP Server, and PostgreSQL. Pour obtenir plus d'informaitons sur le monitoring de JBoss EAP par JON, consulter Section 25.3.1, « AMI Monitoring ».

Rapporter un bogue

25.5.2. Information de diagnostique

En cas de problème détecté par JBoss Operations Network, Amazon CloudWatch ou une inspection manuelle, voici les sources habituelles d'informations de diagnostique :

- **`/var/log/jboss_user-data.out`** est la sortie du script init de jboss-ec2-eap et du script de configuration personnalisée utilisateur.
- **`/var/cache/jboss-ec2-eap/`** contient les données utilisateur réelles, le script personnalisé, et les commandes CLI utilisées au démarrage de l'instance.

- **`/var/log`** contient également tous les journaux collectés au démarrage de la machine, JBoss EAP 6, httpd et la plupart des autres services.

L'accès à ces fichiers est disponible uniquement via une session SSH. Voir Amazon EC Getting Started Guide pour obtenir des informations sur la façon de configurer ou d'établir une session SSH avec une instance Amazon EC2.

Rapporter un bogue

ANNEXE A. RÉFÉRENCES SUPPLÉMENTAIRES

A.1. TÉLÉCHARGER LES FICHIERS DU PORTAIL DES CLIENTS DE RED HAT

Conditions préalables

- Avant de commencer cette tâche, vous aurez besoin d'un compte de Portail clients. Rendez-vous dans https://access.redhat.com et cliquer sur le lien **`Register`** qui se trouve en haut et à droite pour créer un compte.

Procédure A.1. Connectez-vous et téléchargez les fichiers du Portail clients de Red Hat

1. Aller dans https://access.redhat.com et cliquer sur le lien **`Log in`** en haut et à gauche. Saisir vos identifiants, et cliquer sur **`Log In`**.

   Résultat

   Vous êtes connecté dans RHN et on vous renvoie à la page web principale à https://access.redhat.com.

2. **Rendez-vous à la page `Downloads`.**
   Utiliser une des options de pour aller dans la page **`Downloads`**.

   - Cliquer sur le lien **`Downloads`** qui se trouve en haut de la barre de navigation.
   - Rendez-vous directement dans https://access.redhat.com/downloads/.

3. **Sélectionner le produit et la version à télécharger.**
   Utiliser une de ces méthodes pour choisir le produit et la version qui conviennent à télécharger.

   - Procéder une étape à la fois.
   - Chercher votre produit à partir de la zône de recherche qui se trouve en haut et à droite de l'écran.

4. **Télécharger le fichier qui convient pour votre système d'exploitation et la méthode d'installation de votre choix.**
   Suivant le produit, vous aurez le choix entre un installateur natif, un RPM ou une archive Zip suivant le système d'exploitation et l'architecture. Cliquer soit sur le nom de fichier ou sur le lien **`Download`** à droite du fichier que vous souhaitez télécharger.

Résultat

Le fichier sera téléchargé dans votre ordinateur.

Rapporter un bogue

A.2. CONFIGURER LE JDK PAR DÉFAUT DANS RED HAT ENTERPRISE LINUX

Il est possible d'avoir plusieurs Java Development Kits (JDKs) installés sur votre système Red Hat Enterprise Linux. Cette tâche vous montre comment spécifier quel kit est utilisé par votre environnement actuel. Nécessite la commande **`alternatives`**.

IMPORTANT

Cette tâche ne s'applique que pour Red Hat Enterprise Linux.

NOTE

Il est possible de ne pas avoir à passer par cette étape. Red Hat Enterprise Linux utilise OpenJDK 1.6 comme option par défaut. Si c'est ce qui vous convient et que vous utilisez votre système correctement, vous n'aurez pas besoin d'indiquer quel JDK utiliser.

Conditions préalables

- Pour compléter cette tâche, vous devrez avoir l'accès super utilisateur, soit en connexion directe, ou par la commande **sudo**.

Procédure A.2. Configurer le JDK par défaut

1. **Déterminez les chemins qui vous conviennent le mieux pour vos binaires `java` et `javac`.** Vous pouvez utiliser la commande **`rpm -ql packagename |grep bin`** pour trouver les emplacements des binaires installés à partir des RPM. Les locations par défaut des binaires **`java`** et **`javac`** des systèmes Red Hat Enterprise Linux 32-bit sont les suivantes :

   Tableau A.1. Emplacements par défaut des binaires `java` et `javac`

JDK	Chemin
OpenJDK 1.6	**`/usr/lib/jvm/jre-1.6.0-openjdk/bin/java`** **`/usr/lib/jvm/java-1.6.0-openjdk/bin/javac`**
Oracle JDK 1.6	**`/usr/lib/jvm/jre-1.6.0-sun/bin/java`** **`/usr/lib/jvm/java-1.6.0-sun/bin/javac`**

2. **Définir chaque solution alternative que vous souhaitez utiliser pour chacun.** Exécutez les commandes suivantes pour que votre système utilise **`java`** et **`javac`**: **`/usr/sbin/alternatives --config java`** ou **`/usr/sbin/alternatives --config javac`**. Suivez les instructions sur l'écran.

3. **Option: définir un choix alternatif `java_sdk_1.6.0`.** Si vous souhaitez utiliser Oracle JDK, vous devrez configurer la solution alternative **`java_sdk_1.6.0.`** également. Utiliser la commande suivante : **`/usr/sbin/alternatives --config java_sdk_1.6.0`**. Le chemin d'accès qui convient est normalement **`/usr/lib/jvm/java-1.6.0-sun`**. Vous pourrez faire une liste de fichiers pour vérifier.

Résultat :

Le JDK alternatif est sélectionné et actif.

Rapporter un bogue

ANNEXE B. HISTORIQUE DE RÉVISION

Version 6.3.0-38 **Monday August 4 2014** **Sande Gilda**

Red Hat JBoss Enterprise Application Platform 6.3.0.GA