



Red Hat JBoss Enterprise Application Platform 7.0 Guide de démarrage

À utiliser avec Red Hat JBoss Enterprise Application Platform 7

Red Hat Customer Content
Services

Red Hat JBoss Enterprise Application Platform 7.0 Guide de démarrage

À utiliser avec Red Hat JBoss Enterprise Application Platform 7

Notice légale

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Ce guide fournit des informations de base pour aider les utilisateurs à démarrer avec Red Hat JBoss Enterprise Application Platform 7.

Table des matières

| | |
|--|-----------|
| CHAPITRE 1. INTRODUCTION | 3 |
| 1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7 | 3 |
| 1.2. GUIDE DE DÉMARRAGE | 3 |
| CHAPITRE 2. ADMINISTRATION DE JBOSS EAP | 4 |
| 2.1. TÉLÉCHARGEMENT ET INSTALLATION DE JBOSS EAP | 4 |
| 2.2. DÉMARRAGE ET ARRÊT DU SERVICE JBOSS EAP 6 À PARTIR D'UN TERMINAL. | 5 |
| 2.3. JBOSS EAP MANAGEMENT | 7 |
| 2.4. RÉSEAU ET CONFIGURATION DE PORT | 18 |
| CHAPITRE 3. DÉVELOPPEMENT D'APPLICATIONS AVEC JBOSS EAP | 26 |
| 3.1. VUE D'ENSEMBLE | 26 |
| 3.2. CONFIGURER L'ENVIRONNEMENT DE DÉVELOPPEMENT | 26 |
| 3.3. UTILISATION DES EXEMPLES DE QUICKSTART | 32 |
| ANNEXE A. MATÉRIEL DE RÉFÉRENCE | 41 |
| A.1. ARGUMENTS DE TEMPS D'EXÉCUTION DU SERVEUR | 41 |
| A.2. ARGUMENTS DE L'UTILITAIRE ADD-USER | 45 |
| A.3. ATTRIBUTS D'INTERFACE | 46 |
| A.4. ATTRIBUTS DE LIAISONS DE SOCKETS | 48 |
| A.5. LIAISONS DE SOCKET PAR DÉFAUT | 49 |

CHAPITRE 1. INTRODUCTION

1.1. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7

Red Hat JBoss Enterprise Application Platform 7 (JBoss EAP) est une plateforme middleware créée sur des standards ouverts et conforme aux spécifications Java Enterprise Edition 7. JBoss EAP 7 intègre WildFly Application Server 10 avec une messagerie, un clustering de haute disponibilité et autres technologies.

JBoss EAP inclut une structure modulaire qui permet aux services d'être activés s'ils sont requis uniquement, améliorant ainsi la vitesse de démarrage.

La console de gestion et l'interface de ligne de commande (CLI) rendent la modification des fichiers de configuration XML inutile et ajoutent la capacité d'encoder et d'automatiser des tâches.

JBoss EAP fournit deux modes d'opération pour les instances JBoss EAP : serveur autonome ou domaine géré. Le mode d'opération serveur autonome correspond à l'exécution de JBoss EAP en tant qu'instance unique du serveur. Le mode d'opération de domaine géré permet la gestion d'instances multiples de JBoss EAP à partir d'un point de contrôle unique.

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

1.2. GUIDE DE DÉMARRAGE

Le but de ce guide est de vous expliquer l'utilisation de JBoss EAP de manière simple et rapide. Il aborde les tâches **administratives** de bases telles que l'installation, la gestion et la configuration de JBoss EAP. Ce guide aide également **les développeurs** à écrire des applications Java EE 7 à l'aide des guides de quickstart de JBoss EAP.

Pour en savoir plus, merci de consulter la [Suite de documentation JBoss EAP](#).

CHAPITRE 2. ADMINISTRATION DE JBOSS EAP

2.1. TÉLÉCHARGEMENT ET INSTALLATION DE JBOSS EAP

Ce guide fournit des instructions de base pour le téléchargement et l'installation de JBoss EAP à l'aide de l'installation ZIP, qui est indépendante de la plateforme.

Merci de consulter le [Guide d'installation](#) pour plus de détails, dont des instructions pour l'installation de JBoss EAP à l'aide du programme d'installation graphique ou des méthodes d'installation de paquet RPM.

2.1.1. Conditions préalables pour une installation

Vérifier que les conditions préalables suivantes sont bien remplies avant d'installer JBoss EAP.

Conditions préalables habituelles

- » Votre système est pris en charge en accordance avec les [configurations JBoss EAP 7 prises en charge](#).
- » Votre système est à jour avec les mises à jour et les errata les plus récentes fournies par Red Hat.

Conditions préalables d'installation de ZIP

- » L'utilisateur qui exécutera JBoss EAP possède des droits de lecture et d'écriture pour le répertoire d'installation.
- » Le kit de développement Java désiré a été installé.
- » Pour Hewlett-Packard HP-UX, un utilitaire **unzip** a été installé.
- » Pour Windows Server, les variables d'environnement **JAVA_HOME** et **PATH** ont été installés.

2.1.2. Télécharger JBoss EAP

Le fichier d'installation ZIP JBoss EAP peut être téléchargé à partir du Portail Client de Red Hat.

1. Se connecter au [Portail Clients de Red Hat](#)
2. Cliquer sur **Téléchargements**.
3. À partir de la liste **Téléchargements de produits**, cliquer sur **Red Hat JBoss Enterprise Application Platform**.
4. Sélectionner la version souhaitée à partir du menu déroulant **Version**.
5. Trouver l'entrée **Red Hat JBoss Enterprise Application Platform 7.x.x** dans le tableau et cliquer sur **Télécharger**.
6. Enregistrer le fichier ZIP sur le répertoire souhaité.

2.1.3. Installer JBoss EAP

Une fois le fichier d'installation ZIP JBoss EAP téléchargé, il peut être installé en extrayant le contenu du paquet.

1. Si nécessaire, déplacez le fichier ZIP sur le serveur et l'emplacement sur lequel JBoss EAP doit être installé.
 - » L'utilisateur qui exécutera JBoss EAP doit avoir un accès lecture et écriture à ce répertoire.
2. Veuillez extraire l'archive ZIP.

```
$ unzip jboss-eap-7.x.x.zip
```



Note

Pour le serveur Windows, effectuez un clic droit sur le fichier ZIP et sélectionnez **Tout extraire**.

Le répertoire créé grâce à l'extraction de l'archive ZIP est le répertoire du niveau le plus élevé de l'installation JBoss EAP. Il est nommée **EAP_HOME**.

2.2. DÉMARRAGE ET ARRÊT DU SERVICE JBOSS EAP 6 À PARTIR D'UN TERMINAL.

2.2.1. Démarrer JBoss EAP

JBoss EAP peut s'exécuter de deux manières : en tant que serveur autonome ou domaine géré, et est pris en charge sur plusieurs plateformes : Red Hat Enterprise Linux, Windows Server, Oracle Solaris et Hewlett-Packard HP-UX.

La commande spécifique pour démarrer JBoss EAP dépend de la plateforme sous-jacente et du mode d'exploitation souhaité.

Démarrez JBoss EAP comme un serveur autonome

```
$ EAP_HOME/bin/standalone.sh
```



Note

Pour Windows Server, utiliser le script **EAP_HOME\bin\standalone.bat**.

Ce script de démarrage utilise le fichier **EAP_HOME/bin/standalone.conf** (ou **standalone.conf.bat** pour Windows Server) pour définir des préférences par défaut, telles que les options JVM. Vous pouvez personnaliser ces paramètres dans ce fichier.

JBoss EAP utilise le fichier de configuration **standalone.xml** par défaut, mais peut être démarré à l'aide d'un autre fichier. Pour plus de détails sur les fichiers de configuration autonomes disponibles et sur comment les utiliser, merci de consulter la section [Fichiers de configuration de serveur autonome](#).

Pour obtenir une liste complète de tous les arguments de script de démarrage disponibles ainsi que leur utilité, utiliser l'argument **--help** ou consulter la section [Reference of Startup Arguments](#).

Démarrez JBoss EAP comme domaine géré

Le contrôleur de domaines doit être démarré avant les serveurs dans des groupes de serveurs du domaine. Utiliser ce script pour démarrer le contrôleur de domaine puis pour chaque contrôleur d'hôte associé.

```
$ EAP_HOME/bin/domain.sh
```



Note

Pour Windows Server, utiliser le script **EAP_HOME\bin\domain.bat**.

Ce script de démarrage utilise le fichier **EAP_HOME/bin/domain.conf** (ou **domain.conf.bat** pour Windows Server) pour définir des préférences par défaut, telles que les options JVM. Vous pouvez personnaliser ces paramètres dans ce fichier.

JBoss EAP utilise le fichier de configuration **host.xml** par défaut, mais peut être démarré à l'aide d'un autre. Pour plus de détails sur les fichiers de configuration de domaines gérés disponibles et sur la façon de les utiliser, merci de consulter la section [Fichiers de configuration du domaine géré](#).

Lors de la configuration d'un domaine géré, des arguments supplémentaires devront être passés dans le script de démarrage. Pour obtenir une liste complète de tous les arguments de script de démarrage disponibles ainsi que leur utilité, utiliser l'argument **-help** ou consulter la section [Reference of Startup Arguments](#).

2.2.2. Démarrage de JBoss EAP

La façon dont vous arrêtez JBoss EAP dépend de la façon dont il aura été démarré.

Arrêter une instance interactive de JBoss EAP

Appuyer sur **Ctrl+C** dans le terminal sur lequel JBoss EAP a été démarré.

Arrêter une instance d'arrière plan de JBoss EAP

Utiliser l'interface de ligne de commande de gestion pour se connecter à l'instance en cours et arrêter le serveur.

1. Lancer l'interface de ligne de commande de gestion.

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

2. Exécuter la commande **shutdown**.

shutdown



Note

Pour une exécution dans un domaine géré, vous devez spécifier le nom d'hôte à arrêter à l'aide de l'argument **--host** avec la commande **shutdown**.

2.3. JBOSS EAP MANAGEMENT

JBoss EAP utilise une configuration simplifiée, avec un fichier de configuration par serveur autonome ou domaine géré. La configuration par défaut pour un serveur autonome est stockée dans le fichier **EAP_HOME/standalone/configuration/standalone.xml** et la configuration par défaut pour un domaine géré est stockée dans le fichier **EAP_HOME/domain/configuration/domain.xml**. De plus, la configuration par défaut pour un contrôleur d'hôte est stockée dans un fichier **EAP_HOME/domain/configuration/host.xml**.

JBoss EAP peut être configurée à l'aide de l'interface en ligne de commande ou la console web de gestion. Les modifications effectuées à l'aide de ces interfaces de gestion durent automatiquement et les fichiers de configuration XML sont remplacés par l'API de gestion. Il n'est pas recommandé de modifier les fichiers de configuration XML manuellement.

2.3.1. Utilisateurs de gestion

Vous devez créer au moins un utilisateur administratif pour pouvoir utiliser la console de gestion. Une communication HTTP avec JBoss EAP est considérée comme un accès distant, même si le trafic débute sur localhost. Si vous tentez d'accéder à la console de gestion avant d'ajouter un utilisateur, vous recevrez un message d'erreur.

Selon la façon dont JBoss EAP a été installé, il peut n'y avoir aucun compte utilisateur disponible pour accéder aux interfaces de gestion. Si JBoss EAP est installé à l'aide du programme d'installation graphique, alors un compte utilisateur possédant les droits nécessaires sera créé lors du processus d'installation. Sinon, vous devez créer au moins un utilisateur administratif afin d'accéder aux interfaces de gestion.

Ce guide aborde la gestion d'utilisateur simple pour JBoss EAP à l'aide du script **add-user**, un utilitaire pour ajouter de nouveaux utilisateurs aux fichiers de propriété pour une authentification non courante. Pour davantage d'options d'authentification et d'autorisation avancées, telles que LDAP ou RBAC (Role-Based Access Control), voir la section *Core Management Authentication* du Guide *Security Architecture (Architecture de sécurité)* de JBoss EAP.

2.3.1.1. Ajout d'un utilisateur de gestion

- Exécuter le script utilitaire **add-user** et suivre les instructions.

\$ EAP_HOME/bin/add-user.sh

**Note**

Pour le Windows Server, utiliser le script **EAP_HOME\bin\add-user.bat**.

2. Appuyer sur la touche **ENTRÉE** pour sélectionner l'option par défaut **a** pour ajouter un utilisateur de gestion.

Cet utilisateur sera ajouté au domaine *ManagementRealm* et sera autorisé à effectuer des opérations de gestion à l'aide de la console de gestion ou de l'interface en ligne de commande. L'autre choix (**b**) ajoute un utilisateur à *ApplicationRealm* utilisé pour des applications et fournit aucun permission particulière.

3. Saisir le nom d'utilisateur et le nom de passe. On vous demandera de saisir le mot de passe une seconde fois pour confirmer.

Par défaut, JBoss EAP autorise les mots de passe faibles mais ceux-ci feront l'objet d'un avertissement. Merci de consulter la section [Définir les restrictions de mot de passe de l'utilitaire Ajout d'utilisateur](#) du *Guide de configuration JBoss EAP* pour savoir comment modifier ce comportement par défaut.

4. Saisir une liste de groupes séparés par des virgules auxquels l'utilisateur appartient. Si vous souhaitez que l'utilisateur n'appartienne à aucun groupe, appuyez sur **ENTRÉE** pour laisser ce champ vide.
5. Vérifier les informations et saisir **yes** pour confirmer.
6. Déterminer si cet utilisateur représente une instance de serveur EAP à distance. Pour un utilisateur de gestion de base, saisir **no**.

Un type d'utilisateur pouvant être ajouté au *ManagementRealm* est un utilisateur représentant une autre instance de JBoss EAP pouvant s'authentifier pour joindre en tant que membre d'un cluster. Si tel est le cas, répondre **yes** pour recevoir une valeur secrète hachée représentant le mot de passe de l'utilisateur, qui devra être ajouté à un fichier de configuration différent.

Les utilisateurs peuvent également être créés de manière non interactive en passant des paramètres au script **add-user**. Cette approche n'est pas recommandée sur des systèmes partagés, car les mots de passe seront visibles dans les journaux et fichiers d'historiques. Pour plus d'information, consulter [Exécuter l'utilitaire Ajout d'utilisateur de manière non interactive](#).

2.3.1.2. Exécuter l'utilitaire Ajout d'utilisateur de manière non interactive

Vous pouvez exécuter le script **add-user** de manière non interactive en passant des arguments en ligne de commande. Le nom d'utilisateur et mot de passe doivent être fournis.

Avertissement

Cette approche n'est pas recommandée sur des systèmes partagés, car les mots de passe seront visibles dans les fichiers de journalisation et d'historique.

Créer un utilisateur appartenant à plusieurs groupes

La commande suivante ajoute un utilisateur de gestion (**mgmtuser1**) avec les groupes **guest** et **mgmtgroup**.

```
$ EAP_HOME/bin/add-user.sh -u 'mgmtuser1' -p 'password1!' -g  
'guest,mgmtgroup'
```

Spécifier un fichier de propriétés alternatif

Par défaut, les informations de groupe et d'utilisateur créés à l'aide du script **add-user** sont stockés dans des fichiers de propriété situés dans le répertoire de configuration de serveur.

Les informations utilisateur sont stockées dans les fichiers de propriété suivants :

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

Les informations de groupe sont stockées dans les fichiers de propriété suivants :

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

Les noms de fichiers de propriété et de répertoires par défaut peuvent être remplacés. La commande suivante ajoute un nouvel utilisateur, indiquant un nom et emplacement différents pour les fichiers de propriété utilisateur.

```
$ EAP_HOME/bin/add-user.sh -u 'mgmtuser2' -p 'password1!' -sc  
'/path/to/standaloneconfig/' -dc '/path/to/domainconfig/' -up  
'newname.properties'
```

Le nouvel utilisateur a été ajouté aux fichiers de propriétés utilisateur désormais situés dans **/path/to/standaloneconfig/newname.properties** et **/path/to/domainconfig/newname.properties**. Veuillez noter que ces fichiers doivent déjà exister afin d'éviter une erreur.

Pour obtenir une liste complète de tous les arguments **add-user** disponibles ainsi que leur utilité, utiliser l'argument **--help** ou consulter la section [Add-User Utility Arguments](#).

2.3.2. Interfaces de gestion**2.3.2.1. Interface CLI**

L'interface en ligne de commande (CLI) est un outil d'administration en ligne de

commande pour JBoss EAP.

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'autres tâches administratives. Les opérations peuvent être effectuées par mode lots, ce qui permet à plusieurs tâches d'être exécutées en groupe.

Plusieurs commandes de terminal courantes sont disponibles, telles que **ls**, **cd** et **pwd**. L'interface CLI prend également en charge la saisie automatique avec la touche «Tab».

Pour obtenir des informations supplémentaires sur la façon d'utiliser l'interface de gestion CLI, y compris les commandes et opérations, la syntaxe, et comment exécuter par lots, voir le guide [Interface de commandes \(CLI\)](#) de JBoss EAP.

Lancement de l'interface CLI

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



Note

Pour Windows Server, utiliser le script **EAP_HOME\bin\jboss-cli.bat**.

Connexion à un serveur en cours d'exécution

```
connecter
```

Ou vous pouvez lancer l'interface CLI et vous connecter en une seule fois à l'aide de la commande **EAP_HOME/bin/jboss-cli.sh --connect**.

Afficher l'aide

Utiliser la commande suivante pour afficher l'aide générale.

```
aide
```

Utiliser la commande suivante pour obtenir de l'aide pour une commande spécifique.

```
deploy --help
```

Quitter l'interface CLI

```
Quitter
```

Afficher les paramètres de système

La commande suivante utilise l'opération **read-attribute** pour afficher si l'exemple de source de donnée est activé.

```
/subsystem=datasources/data-source=ExampleDS:read-  
attribute(name=enabled)  
{  
    "outcome" => "success",
```

```

    "result" => true
}

```

Pour les exécutions en domaine géré, il convient de spécifier le profil à mettre à jour en faisant précéder la commande par **/profile=PROFILE_NAME**.

```

/profile=default/subsystem=datasources/data-source=ExampleDS:read-
attribute(name=enabled)

```

Mettre à jour les paramètres système

La commande suivante utilise l'opération **write-attribute** pour désactiver l'exemple de source de donnée.

```

/subsystem=datasources/data-source=ExampleDS:write-
attribute(name=enabled,value=false)

```

Démarrer les serveurs

L'interface CLI peut également être utilisée pour démarrer et arrêter les serveurs lors d'une exécution dans un domaine géré.

```

/host=HOST_NAME/server-config=server-one:start

```

2.3.2.2. Console de gestion

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

Utilisez la console de gestion 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 gestion a également la capacité d'effectuer des tâches administratives, avec des notifications directes lorsque les modifications effectuées par l'utilisateur actuel 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 gestion du contrôleur de domaine.

Pour une instance JBoss EAP exécutée sur l'hôte local à l'aide du port de gestion par défaut, la console de gestion peut être accédée par un navigateur web sur <http://localhost:9990/console/App.html>. Vous devrez vous authentifier avec un utilisateur possédant des permissions pour accéder à la console de gestion.

La console de gestion fournit les onglets de navigation et de gestion de votre serveur autonome JBoss EAP ou de votre domaine géré suivants :

Accueil

Apprenez à effectuer quelques tâches de gestion et quelques configurations de base. Prenez le temps de visiter la console de gestion JBoss EAP.

Déploiements

Ajouter, supprimer, et activer des déploiements. Dans un domaine géré, assigner des déploiements à des groupes de serveurs.

Configuration

Configurez les sous-systèmes disponibles, qui vous donnent des capacités de services web, messagerie, ou HA. Dans un domaine géré, gérez les profils qui contiennent plusieurs configurations de sous-systèmes.

Exécution

Voir les informations de runtime (exécution), comme le statut du serveur, l'utilisation de la JVM, et les journaux des serveurs. Dans un domaine géré, gérez vos hôtes, groupes de serveurs, et serveurs.

Contrôle des accès

Assignez des rôles à des groupes et à des utilisateurs lorsque vous êtes en contrôle d'accès basé rôle.

Correctifs

Appliquez des correctifs à vos instances JBoss EAP.



Note

Pour visiter la console de gestion nouvellement mise à jour, cliquer sur le lien **Take a Tour** (Visite guidée) qui se trouve sur la page d'accueil de la console de gestion.

Pour voir les détails de champs de formulaire, cliquer sur le lien **Besoin d'aide ?**.

Pour apercevoir l'historique des actions de configuration que vous avez effectuées, cliquer sur le lien **Messages** qui se trouve en haut et à droite de la console de gestion.

2.3.3. Fichiers de configuration

2.3.3.1. Fichiers de configuration du serveur autonome

Les fichiers de configuration du serveur autonome se situent dans le répertoire **EAP_HOME/standalone/configuration**. Un fichier séparé existe pour chacun des quatre profils prédéfinis (*default*, *ha*, *full*, *full-ha*).

Tableau 2.1. Fichiers de configuration autonomes

| Fichier de configuration | Utilité |
|--------------------------|---------|
| | |

| Fichier de configuration | Utilité |
|-------------------------------|---|
| standalone.xml | Ce fichier de configuration autonome est la configuration par défaut utilisée lorsque vous démarrez votre serveur autonome. Il contient toutes les informations sur le serveur, dont les sous-systèmes, les réseaux, les déploiements, les liaisons de socket et autres détails configurables. Il ne fournit pas les sous-systèmes nécessaires à la messagerie ou la haute disponibilité. |
| standalone-ha.xml | Ce fichier de configuration autonome inclut tous les sous-systèmes par défaut et ajoute les sous-systèmes <i>mod_cluster</i> et <i>JGroups</i> pour la haute disponibilité. Il ne fournit pas les sous-systèmes nécessaires pour la messagerie. |
| standalone-full.xml | Ce fichier de configuration autonome inclut tous les sous-systèmes par défaut et ajoute les sous-systèmes <i>Messaging</i> et <i>IOP</i> . Il ne fournit pas les sous-systèmes nécessaires à la haute disponibilité. |
| standalone-full-ha.xml | Ce fichier de configuration autonome inclut la prise en charge de chaque sous-système, dont les sous-systèmes de messagerie et la haute disponibilité. |

Par défaut, le démarrage de JBoss EAP en tant que serveur autonome utilise le fichier **standalone.xml**. Pour démarrer JBoss EAP avec une configuration différente, utiliser l'argument **--server-config**. Par exemple,

```
$ EAP_HOME/bin/standalone.sh --server-config=standalone-full.xml
```

2.3.3.2. Fichiers de configuration de domaines gérés

Les fichiers de configuration du domaine géré se situent dans le répertoire **EAP_HOME/domain/configuration**.

Tableau 2.2. Fichiers de configuration de domaines gérés

| Fichier de configuration | Utilité |
|--------------------------|---|
| domain.xml | Il s'agit du fichier de configuration principal pour un domaine géré. Seul le domaine principal lit ce fichier. Ce fichier contient les configurations pour tous les profils (c'est-à-dire <i>default</i> , <i>ha</i> , <i>full</i> , <i>full-ha</i>). |

| Fichier de configuration | Utilité |
|--------------------------|--|
| 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 décrites ci-dessous. |
| host-master.xml | Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur maître de domaine géré (contrôleur de domaine). |
| host-slave.xml | Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur esclave de domaine géré (contrôleur d'hôte). |

Par défaut, le démarrage de JBoss EAP en tant que domaine géré utilise le fichier **host.xml**. Pour démarrer JBoss EAP avec une configuration différente, utiliser l'argument **--host-config**. Par exemple,

```
$ EAP_HOME/bin/domain.sh --host-config=host-master.xml
```

2.3.3.3. Sauvegarder les données de configuration

Afin de restaurer la configuration du serveur JBoss EAP, les éléments situés aux emplacements suivants doivent être sauvegardés :

- » **EAP_HOME/standalone/configuration/**
 - Sauvegarder le répertoire complet pour enregistrer les données d'utilisateur, la configuration de serveur et les paramètres de journalisation pour les serveurs autonomes.
- » **EAP_HOME/domain/configuration/**
 - Sauvegarder le répertoire complet pour enregistrer les données de profil et d'utilisateur, la configuration d'hôte et de domaine et les paramètres de journalisation pour des domaines gérés.
- » **EAP_HOME/modules/system/layers/base/**
 - Sauvegarder des modules personnalisés
- » **EAP_HOME/welcome-content/**
 - Sauvegarder tout contenu de bienvenue personnalisé.
- » **EAP_HOME/bin/**

- Sauvegarder tout script personnalisé ou fichier de configuration de démarrage

2.3.3.4. Instantané du fichier de configuration

Pour aider à la maintenance et à la gestion du serveur, JBoss EAP crée une version horodatée du fichier de configuration original au moment du démarrage. Toutes les modifications de configuration supplémentaires suite aux opérations de gestion entraîneront la sauvegarde automatique du fichier d'origine, et une copie de travail de l'instance sera alors conservée en tant que référence et pour la restauration. De plus, les instantanés de configuration peuvent être effectués, représentant des copies à un moment précis de la configuration du serveur actuel. Ces instantanés peuvent être enregistrés et chargés par un administrateur.

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

Effectuer un instantané

Utiliser l'interface CLI pour effectuer un instantané des configurations actuelles.

```
:take-snapshot
{
    "outcome" => "success",
    "result" =>
"EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/2015
1022-133109702standalone.xml"
}
```

Lister les instantanés

Utiliser l'interface CLI pour énumérer tous les instantanés effectués.

```
:list-snapshots
{
    "outcome" => "success",
    "result" => {
        "directory" =>
"EAP_HOME/standalone/configuration/standalone_xml_history/snapshot",
        "names" => [
            "20151022-133109702standalone.xml",
            "20151022-132715958standalone.xml"
        ]
    }
}
```

Supprimer un instantané

Utiliser l'interface CLI pour supprimer un instantané.

```
:delete-snapshot(name=20151022-133109702standalone.xml)
```

Démarrer le serveur avec un instantané

Le serveur peut être démarré à l'aide d'un instantané ou d'une version de configuration automatiquement enregistrée.

1. Naviguer vers le répertoire **EAP_HOME/standalone/configuration/standalone_xml_history** et identifier l'instantané ou le fichier de configuration enregistré à charger.
2. Démarrer le serveur et se diriger vers le fichier de configuration sélectionné. Passer dans le chemin d'accès du fichier relatif au répertoire de configuration (c'est-à-dire **EAP_HOME/standalone/configuration**).

```
$ EAP_HOME/bin/standalone.sh --server-config=standalone_xml_history/snapshot/20151022-133109702standalone.xml
```

Note

Pour une exécution dans un domaine géré, utiliser plutôt l'argument **--host-config** pour spécifier le fichier de configuration.

2.3.3.5. Remplacement de propriété

JBoss EAP vous autorise à utiliser des expressions pour définir des propriétés remplaçables à la place de valeurs littérales dans la configuration. Les expressions utilisent le format **\${PARAMETER:DEFAULT_VALUE}** . Si le paramètre spécifié est défini, alors la valeur du paramètre sera utilisée. Sinon, la valeur par défaut fournie sera utilisée.

Les sources prises en charge pour résoudre les expressions sont les propriétés système, variables d'environnement et de l'archivage sécurisé. Pour les déploiements uniquement, la source peut correspondre à des propriétés répertoriées dans un fichier **META-INF/jboss.properties** qui se trouve dans l'archive de déploiement. Pour les types de déploiement qui prennent en charge les sous-déploiements, la résolution est étendue à tous les sous-déploiements si le fichier de propriété se trouve dans le déploiement externe (par ex. EAR). Si le fichier de propriétés se trouve dans le sous-déploiement, alors la résolution est étendue à ce sous-déploiement.

L'exemple ci-dessous à partir du fichier de configuration **standalone.xml** définit le **inet-address** pour l'interface **public** à **127.0.0.1** à moins que le paramètre **jboss.bind.address** soit défini.

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

Le paramètre **jboss.bind.address** peut être défini au démarrage de EAP en tant que serveur autonome avec la commande suivante :

```
$ EAP_HOME/bin/standalone.sh -Djboss.bind.address=IP_ADDRESS
```

Expressions imbriquées

Les expressions peuvent être imbriquées, ce qui permet une utilisation plus avancée des expressions au lieu de valeurs fixes. Le format d'une expression imbriquée ressemble à celui d'une expression normale, mais une expression est imbriquée dans l'autre, comme par exemple :

```
 ${SYSTEM_VALUE_1}${SYSTEM_VALUE_2}
```

Les expressions imbriquées sont évaluées de façon récursive, donc l'expression *inner* est d'abord évaluée, puis *outer*. Les expressions peuvent également être récursives, lorsque l'évaluation d'une expression entraîne une autre expression, qui est à son tour évaluée. Les expressions imbriquées sont autorisées partout où les expressions sont autorisées, à l'exception des commandes CLI de gestion.

Une expression imbriquée peut être utilisée si le mot de passe utilisé dans une définition de source de donnée est masqué. La configuration pour la source de donnée peut contenir la ligne suivante :

```
<password>${VAULT::ds_ExampleDS::password::1}</password>
```

La valeur de **ds_ExampleDS** peut être remplacée avec une propriété de système (**datasource_name**) à l'aide d'une expression imbriquée. La configuration de la source de données peut contenir à la place la ligne suivante :

```
<password>${VAULT::${datasource_name}::password::1}</password>
```

JBoss EAP doit tout d'abord évaluer l'expression **\${datasource_name}**, puis l'ajouter à une plus grande expression et évaluer l'expression qui en résulte. L'avantage de cette configuration est que le nom de la source de données se soustrait de la configuration fixe

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 environnements 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 EAP 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.

Il existe plusieurs indicateurs dans le sous-système *EE* qui contrôlent si le remplacement de propriété est appliqué.

Le remplacement de descripteur spécifique à JBoss est contrôlé par l'indicateur **jboss-descriptor-property-replacement** et activé par défaut. Une fois activées, les propriétés peuvent être remplacées dans les descripteurs de déploiement suivants :

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

La commande d'interface CLI suivante peut être utilisée pour activer ou désactiver le remplacement de propriété dans les descripteurs spécifiques à JBoss :

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

Le remplacement de descripteur Java EE est contrôlé par l'indicateur **spec-descriptor-property-replacement** et désactivé par défaut. Une fois activées, les propriétés peuvent être remplacées dans les descripteurs de déploiement suivants :

- » **ejb-jar.xml**
- » **persistence.xml**
- » **application.xml**
- » **web.xml**

La commande d'interface CLI suivante peut être utilisée pour activer ou désactiver le remplacement de propriétés dans les descripteurs Java EE :

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

2.4. RÉSEAU ET CONFIGURATION DE PORT

2.4.1. Interfaces

JBoss EAP fait référence aux interfaces nommées dans la configuration. Cela permet à la configuration de faire référence aux déclarations d'interfaces individuelles par des noms logiques plutôt que de demander les informations complètes de l'interface à chaque utilisation.

Cela permet également une configuration simplifiée dans un domaine géré, où les informations d'interface de réseau peuvent varier selon les machines. Chaque instance de serveur peut correspondre à un groupe de noms logiques.

Les fichiers **standalone.xml**, **domain.xml** et **host.xml** incluent tous des déclarations d'interface. Il y a plusieurs noms d'interfaces préconfigurés, suivant la configuration utilisée par défaut. L'interface **management** peut être utilisée pour tous les composants et services nécessitant la couche de gestion, dont le point de terminaison de gestion HTTP. L'interface **public** peut être utilisée pour toutes les communications de réseau en lien avec les applications. L'interface **unsecure** est utilisée pour les sockets IIOP dans la configuration standard. L'interface **private** est utilisée pour les sockets de JGroups dans la configuration standard.

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

```

Par défaut, JBoss EAP lie ces interfaces à **127.0.0.1**, mais ces valeurs peuvent être remplacées lors de l'exécution en définissant la propriété appropriée. Par exemple, l'**inet-address** de l'interface **public** peut être définie au démarrage de JBoss EAP en tant que serveur autonome par la commande suivante.

```
$ EAP_HOME/bin/standalone.sh -Djboss.bind.address=IP_ADDRESS
```

Sinon, vous pouvez utiliser l'argument **-b** dans la ligne de commande de démarrage du serveur. Pour plus d'informations sur les options de démarrage d'un serveur, voir [Server Runtime Arguments](#).

2.4.1.2. Configuration d'interfaces

Les interfaces de réseau sont déclarées en indiquant un nom logique et des critères de sélection pour une interface physique. Les critères de sélection peuvent faire référence à une adresse wildcard ou indiquer un ensemble d'une ou plusieurs caractéristiques qu'une interface ou adresse doit avoir afin d'obtenir une correspondance valide. Pour obtenir une liste de tous les critères de sélection d'interface disponibles, consulter la section [Interface Attributes](#).

Les interfaces peuvent être configurées à l'aide de la console de gestion ou l'interface CLI. Voici plusieurs exemples d'ajout et de mise à jour d'interfaces. La commande d'interface CLI est affichée en premier, suivie par le fichier XML de configuration correspondant.

Ajouter une interface avec une valeur NIC

Ajouter une nouvelle interface avec une valeur NIC de **eth0**.

```

/interface=external:add(nic=eth0)

<interface name="external">
    <nics name="eth0"/>
</interface>

```

Ajouter une interface avec plusieurs valeurs conditionnelles

Ajouter une nouvelle interface correspondant à une interface/adresse sur le sous-réseau approprié si l'interface/adresse est active, prend en charge les multi-diffusion et n'est pas une liaison point à point.

```

/interface=default:add(subnet-
match=192.168.0.0/16,up=true,multicast=true,not={point-to-point=true})

<interface name="default">
    <subnet-match value="192.168.0.0/16"/>

```

```

<up/>
<multicast/>
<not>
    <point-to-point/>
</not>
</interface>

```

Mettre à jour un attribut d'interface

Mettre à jour la valeur **inet-address** par défaut de l'interface **publique**, en maintenant la propriété **jboss.bind.address** pour permettre à cette valeur d'être définie à l'exécution.

```

/interface=public:write-attribute(name=inet-
address,value="${jboss.bind.address:192.168.0.0}")

<interface name="public">
    <inet-address value="${jboss.bind.address:192.168.0.0}" />
</interface>

```

Ajouter une interface à un serveur dans un domaine géré

```

/host=master/server-
config=SERVER_NAME/interface=INTERFACE_NAME:add(inet-address=127.0.0.1)

<servers>
    <server name="SERVER_NAME" group="main-server-group">
        <interfaces>
            <interface name="INTERFACE_NAME">
                <inet-address value="127.0.0.1" />
            </interface>
        </interfaces>
    </server>
</servers>

```

2.4.2. 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. Une liaison de socket est une configuration nommée pour une socket. Un groupe de liaison de sockets est une collection de déclarations de liaisons de sockets regroupées sous un nom logique.

Cela permet à d'autres sections de la configuration de faire référence à des liaisons de sockets par leur nom logique plutôt que de demander des informations complètes sur la configuration des sockets à chaque utilisation.

Les déclarations de ces configurations nommées se trouvent dans les fichiers de configuration **standalone.xml** et **domain.xml**. Un serveur autonome contient uniquement un groupe de liaisons de sockets, alors qu'un domaine géré peut contenir plusieurs groupes. Vous pouvez créer un groupe de liaisons de sockets pour chaque groupe de serveurs dans le domaine géré, ou partager un groupe de liaisons de sockets entre plusieurs groupes de serveurs.

Les ports utilisés par JBoss EAP par défaut dépendent des groupes de liaisons de sockets utilisés et des conditions de vos déploiements individuels.

2.4.2.1. Ports de gestion

Les ports de gestion ont été consolidés dans JBoss EAP 7. Par défaut, JBoss EAP 7 utilise le port **9990** pour la gestion native (interface CLI) et gestion HTTP (console de gestion web). Le port **9999**, utilisé en tant que port de gestion natif dans JBoss EAP 6, n'est plus utilisé mais peut être activé si besoin.

Si HTTPS est activé pour la console de gestion, alors le port **9993** est utilisé par défaut.

2.4.2.2. Liaisons de socket par défaut

JBoss EAP comprend un groupe de liaison de socket pour chacun des quatre profils prédéfinis (*default, ha, full, full-ha*).

Pour des informations détaillées sur les liaisons de sockets par défaut, telles que les descriptions et ports par défaut, merci de consulter la section [Default Socket Bindings](#).

Serveur autonome

Pour une exécution à l'aide d'un serveur autonome, un seul groupe de liaisons de sockets est défini par fichier de configuration. Chaque fichier de configuration autonome (**standalone.xml**, **standalone-ha.xml**, **standalone-full.xml**, **standalone-full-ha.xml**) définit les liaisons de sockets pour les technologies utilisées par son profil correspondant.

Par exemple, le fichier de configuration autonome (**standalone.xml**) spécifie les liaisons de sockets ci-dessous.

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
    <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:9990}">
        <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9993}">
            <socket-binding name="ajp" port="${jboss.ajp.port:8009}" />
            <socket-binding name="http" port="${jboss.http.port:8080}" />
            <socket-binding name="https" port="${jboss.https.port:8443}" />
            <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>
```

Domaine géré

Lors d'une exécution dans un domaine géré, tous les groupes de liaisons de sockets sont définis dans le fichier **domain.xml**. Il existe quatre groupes de liaisons de sockets prédéfinis :

- » **standard-sockets**
- » **ha-sockets**

- » **full-sockets**

- » **full-ha-sockets**

Chaque groupe de liaisons de sockets indique les liaisons de sockets pour les technologies utilisées par le profil correspondant. Par exemple, le groupe de liaisons de sockets **full-ha-sockets** définit plusieurs liaisons de sockets **jgroups**, utilisées par le profil *full-ha* pour la haute disponibilité.

```
<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="${jboss.ajp.port:8009}"/>
        <socket-binding name="http" port="${jboss.http.port:8080}"/>
        <socket-binding name="https" port="${jboss.https.port:8443}"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="ha-sockets" default-interface="public">
        <!-- Needed for server groups using the 'ha' profile -->
        ...
    </socket-binding-group>
    <socket-binding-group name="full-sockets" default-interface="public">
        <!-- Needed for server groups using the 'full' profile -->
        ...
    </socket-binding-group>
    <socket-binding-group name="full-ha-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'full-ha' profile -->
        <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
        <socket-binding name="http" port="${jboss.http.port:8080}"/>
        <socket-binding name="https" port="${jboss.https.port:8443}"/>
        <socket-binding name="iiop" interface="unsecure" port="3528"/>
        <socket-binding name="iiop-ssl" interface="unsecure"
port="3529"/>
        <socket-binding name="jgroups-mping" interface="private" port="0"
multicast-address="${jboss.default.multicast.address:230.0.0.4}"
multicast-port="45700"/>
        <socket-binding name="jgroups-tcp" interface="private"
port="7600"/>
        <socket-binding name="jgroups-tcp-fd" interface="private"
port="57600"/>
        <socket-binding name="jgroups-udp" interface="private"
port="55200" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45688"/>
        <socket-binding name="jgroups-udp-fd" interface="private"
port="54200"/>
        <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
        <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>

```



Note

La configuration de socket pour les interfaces de gestion est définie dans le fichier **host.xml** du contrôleur de domaine.

2.4.2.3. Configuration de liaisons de socket

Lors de la définition d'une liaison de socket, vous pouvez configurer les attributs **port** et **interface**, ainsi que les paramètres de multidiffusion tels que **multicast-address** et **multicast-port**. Pour plus de détails sur tous les attributs de liaison de socket, voir la section [Socket Binding Attributes](#) section.

Les liaisons de socket peuvent être configurées à l'aide de la console de gestion ou l'interface CLI. Les étapes suivantes indiquent d'ajouter un groupe de liaison de socket et de configurer les paramètres de liaison de socket à l'aide de l'interface CLI.

1. Ajouter un nouveau groupe de liaisons de sockets. Merci de noter que cette étape ne peut pas être effectuée avec un serveur autonome.

```
/socket-binding-group=new-sockets:add(default-interface=public)
```

2. Ajouter une liaison de socket

```
/socket-binding-group=new-sockets/socket-binding=new-socket-
binding:add(port=1234)
```

3. Modifier les liaisons de sockets pour utiliser une interface autre que celle par défaut et définie par le groupe de liaisons de sockets.

```
/socket-binding-group=new-sockets/socket-binding=new-socket-
binding:write-attribute(name=interface,value=unsecure)
```

L'exemple suivant montre à quoi la configuration XML peut ressembler après avoir effectué les étapes ci-dessus.

```

<socket-binding-groups>
  ...
  <socket-binding-group name="new-sockets" default-interface="public">
    <socket-binding name="new-socket-binding" interface="unsecure"
      port="1234"/>
  </socket-binding-group>
</socket-binding-groups>

```

2.4.2.4. Décalages (offset) de port

Un décalage de port (Port offset) est une valeur de décalage chiffré qui vient s'ajouter aux valeurs de port données par le groupe de liaisons de sockets pour ce serveur. Cela

permet au serveur d'hériter des valeurs de port définies dans son groupe de liaisons de sockets, avec un décalage pour veiller à ce qu'il n'entre pas en conflit avec les autres serveurs du même hôte. Par exemple, si le port HTTP du groupe de liaisons de sockets est **8080** et si votre serveur utilise un port offset de **100**, son port HTTP sera **8180**.

Voici un exemple de définition de décalage de port de **250** pour un serveur dans un domaine géré à l'aide de l'interface CLI.

```
/host=master/server-config=server-two/:write-attribute(name=socket-binding-port-offset,value=250)
```

Les décalages de port peuvent être utilisés pour les serveurs dans un domaine géré et pour exécuter plusieurs serveurs autonomes sur le même hôte.

Vous pouvez indiquer un décallage de port quand vous démarrez un serveur autonome par la propriété **jboss.socket.binding.port-offset**.

```
$ EAP_HOME/bin/standalone.sh -Djboss.socket.binding.port-offset=100
```

2.4.3. Adresse IPv6

Par défaut, JBoss EAP est configuré pour exécuter avec des adresses IPv4. Les étapes ci-dessous montrent comment configurer JBoss EAP pour exécuter avec des adresses IPv6.

Configurer JVM Stack pour les adresses IPv6

Mettre à jour la configuration de démarrage pour privilégier les adresses IPv6.

1. Ouvrir le fichier de configuration de démarrage.
 - Pour une exécution en serveur autonome, modifier le fichier **EAP_HOME/bin/standalone.conf** (ou **standalone.conf.bat** pour Windows Server).
 - Pour une exécution en domaine géré, modifier le fichier **EAP_HOME/bin/domain.conf** (ou **domain.conf.bat** pour Windows Server).
2. Définir la propriété **java.net.preferIPv4Stack** sur **false**.


```
-Djava.net.preferIPv4Stack=false
```

3. Ajouter la propriété **java.net.preferIPv6Addresses** et la définir sur **true**.


```
-Djava.net.preferIPv6Addresses=true
```

L'exemple suivant montre à quoi les options JVM du fichier de configuration peuvent ressembler après avoir effectué les modifications mentionnées ci-dessus.

```
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
  JAVA_OPTS="-Xms1303m -Xmx1303m -Djava.net.preferIPv4Stack=false"
  JAVA_OPTS="$JAVA_OPTS -
```

```
Djboss.modules.system.pkgs=$JBoss_MODULES_SYSTEM_PKGS -  
Djava.awt.headless=true"  
JAVA_OPTS="$JAVA_OPTS -Djava.net.preferIPv6Addresses=true"  
else
```

Déclarations d'interface de mise à jour pour les adresses IPv6

Les valeurs d'interface par défaut dans la configuration peuvent être modifiées en adresses IPv6. Par exemple, la commande d'interface CLI ci-dessous définit l'interface **management** à l'adresse de rebouclage IPv6 (`::1`).

```
/interface=management:write-attribute(name=inet-  
address,value="${jboss.bind.address.management:[::1]}")
```

L'exemple suivant montre à quoi la configuration XML peut ressembler après avoir effectué la commande ci-dessus

```
<interfaces>  
  <interface name="management">  
    <inet-address value="${jboss.bind.address.management:[::1]}"/>  
  </interface>  
  ...  
</interfaces>
```

CHAPITRE 3. DÉVELOPPEMENT D'APPLICATIONS AVEC JBOSS EAP

3.1. VUE D'ENSEMBLE

Ce guide fournit des informations sur le démarrage d'applications de développement à l'aide de Red Hat JBoss Developer Studio et des exemples de quickstart de JBoss EAP 7.

Red Hat JBoss Developer Studio est un environnement de développement intégré (IDE) basé sur Eclipse et intégrant les plug-ins de développement d'application de JBoss. JBoss Developer Studio peut vous assister avec le développement de votre application, avec la disponibilité des assistants de JBoss et avec l'abilité de déployer des applications vers des serveurs JBoss EAP. Plusieurs exemples de codes de quickstart sont fournis avec JBoss EAP 7 pour aider les utilisateurs à écrire des applications avec les différentes technologies Java EE 7.

Ce guide fournit des instructions d'installation de base pour JBoss Developer Studio. Voir le guide [Install Red Hat JBoss Developer Studio](#) pour obtenir des instructions d'installation complètes.

3.2. CONFIGURER L'ENVIRONNEMENT DE DÉVELOPPEMENT

3.2.1. Télécharger JBoss Developer Studio

JBoss Developer Studio peut être téléchargé à partir du Portail Client de Red Hat.

1. Se connecter au [Portail Clients de Red Hat](#)
2. Cliquer sur **Téléchargements**.
3. Dans la liste **Téléchargements de produits**, cliquer sur **Red Hat JBoss Developer Studio**.
4. Sélectionner la version souhaitée à partir du menu déroulant **Version**.



Note

Il est recommandé d'utiliser JBoss Developer Studio version 9.1 ou ultérieure.

5. Trouver l'entrée **Red Hat JBoss Developer Studio 9.x.x Stand-alone Installer** dans le tableau et cliquer sur **Télécharger**.
6. Enregistrer le fichier JAR sur le répertoire souhaité.

3.2.2. Installer JBoss Developer Studio

1. Ouvrir une session de terminal et naviguer vers le répertoire qui contient le fichier JAR téléchargé.

- Exécuter la commande suivante pour lancer le GUI d'installation :

```
$ java -jar jboss-devstudio-BUILD_VERSION-installer-standalone.jar
```



Note

Sinon, ouvrez-le fichier JAR d'un double-clic pour lancer le programme d'installation.

- Cliquer sur **Next** pour démarrer le processus d'installation.
- Selectionner **I accept the terms of this license agreement** (J'accepte les conditions de ce contrat de licence), puis cliquez sur **Suite**.
- Ajuster le chemin d'accès et cliquer sur **Suite**.



Note

Si le dossier de chemin d'installation n'existe pas, vous verrez une invite. Cliquer alors sur **OK** pour créer le dossier.

- Choisir une JVM, ou bien conserver la JVM sélectionnée par défaut, et cliquer sur **Suite**.
- Cliquer sur **Suite** quand on vous demande de sélectionner des plateformes ou des serveurs.
- Revoir les informations d'installation, et cliquer sur **Suite**.
- Cliquer sur **Suite** quand le processus d'installation est terminé.
- Configurer les raccourcis de bureau de JBoss Developer Studio, et cliquer sur **Suite**.
- Cliquer sur **Terminé**.

3.2.3. Démarrer JBoss Developer Studio

Pour lancer JBoss Developer Studio, vous pouvez double-cliquer sur le raccourci du Bureau créé pendant l'installation, ou vous pouvez le lancer par ligne de commande. Effectuez les étapes ci-dessous pour démarrer JBoss Developer Studio par ligne de commande.

- Ouvrir un terminal et naviguer dans le répertoire d'installation de JBoss Developer Studio.
- Lancer la commande suivante pour démarrer JBoss Developer Studio :

```
$ ./jbdevstudio
```

**Note**

Pour Windows Server, utiliser le fichier **jbdevstudio.bat**.

3.2.4. Ajouter le serveur de JBoss EAP au JBoss Developer Studio

Ces instructions partent du principe que vous n'avez pas encore ajouté de serveur JBoss EAP à JBoss Developer Studio. Utiliser les étapes suivantes pour ajouter votre serveur JBoss EAP à l'aide de l'assistant de création **Define New Server**.

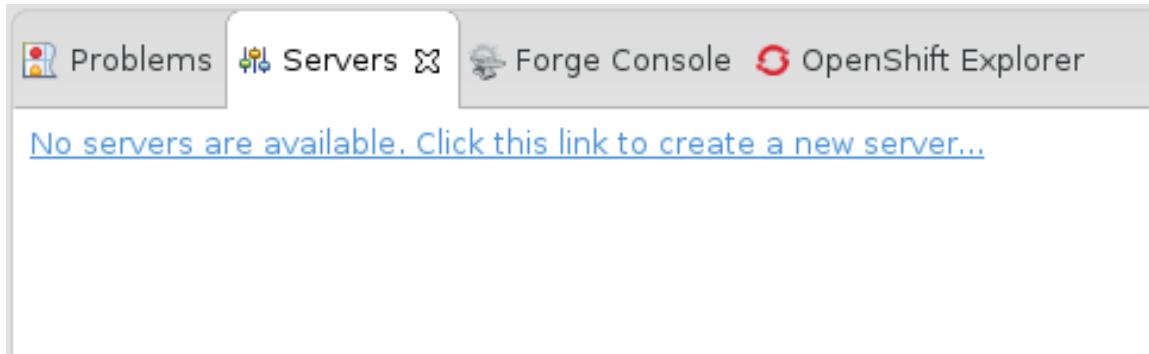
1. Ouvrir l'onglet **Servers**.

**Note**

Si l'onglet **Serveurs** ne s'affiche pas, ajoutez-le au tableau de bord en sélectionnant **Window → Afficher Vue → Serveurs**.

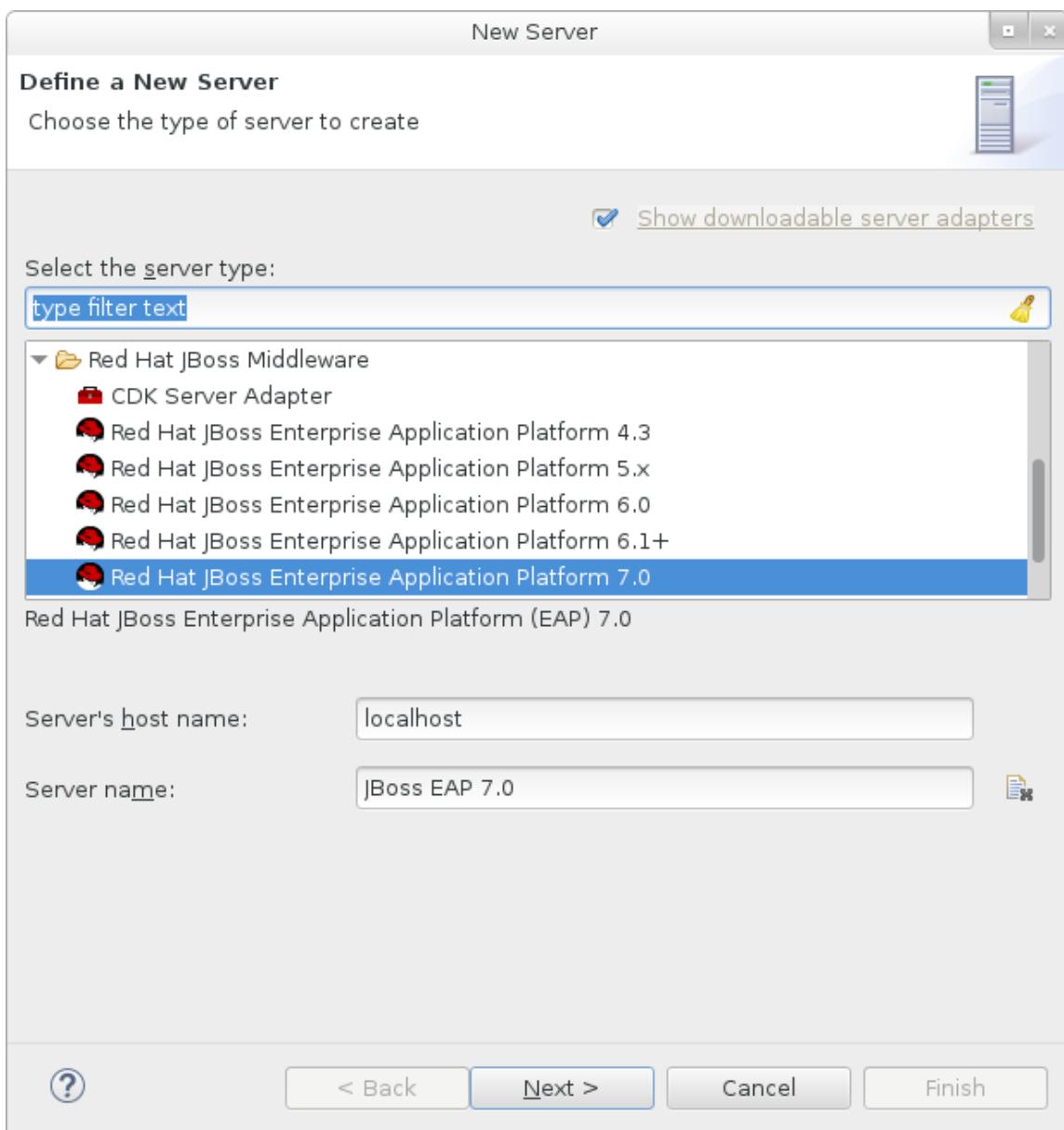
2. Cliquer sur le lien **Aucun serveur n'est disponible actuellement. Cliquer sur ce lien pour créer un nouveau serveur.**

Figure 3.1. Ajouter un nouveau serveur



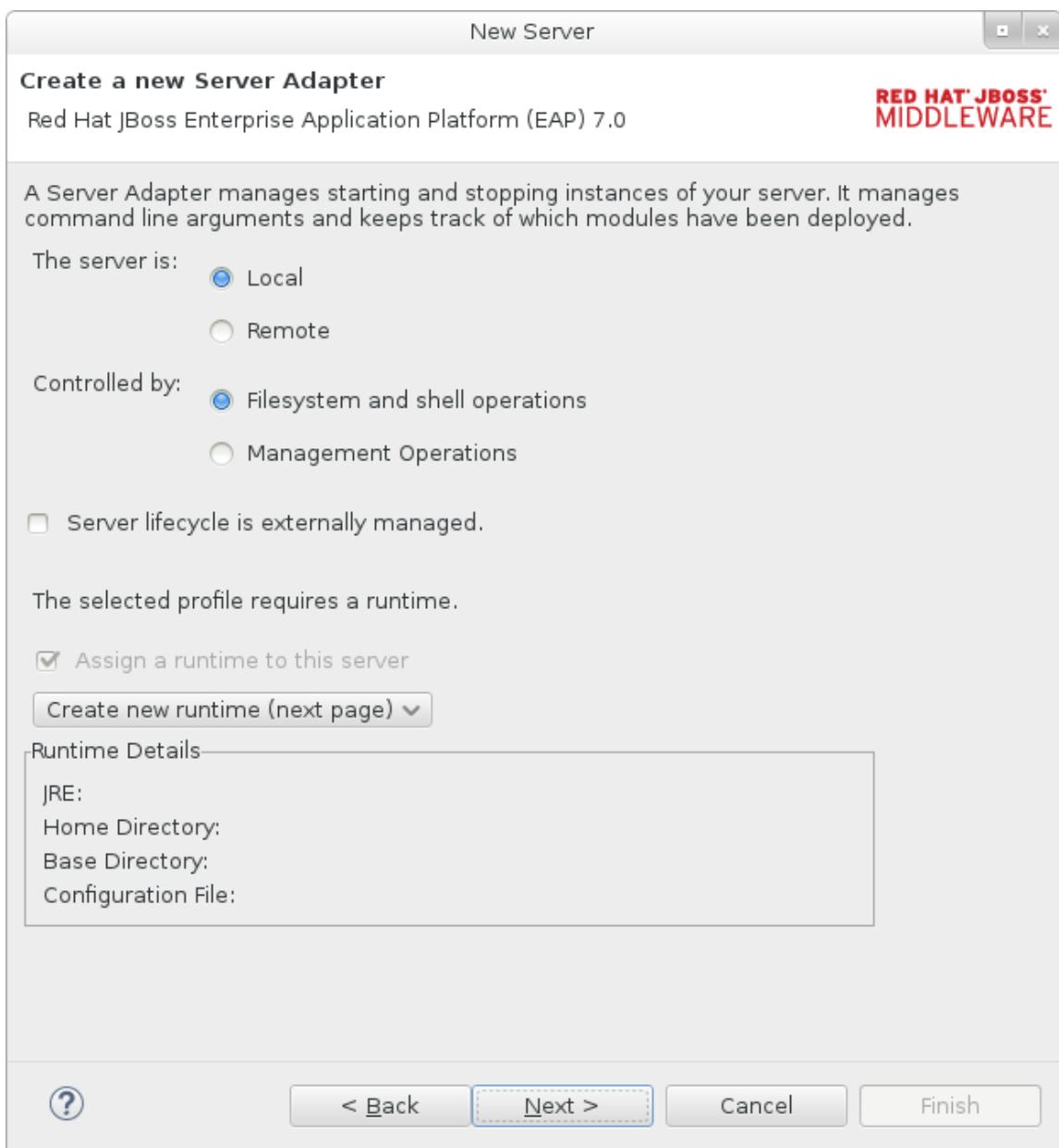
3. Agrandir **Red Hat JBoss Middleware** et choisir **JBoss Enterprise Application Platform 7.0**. Saisir un nom de serveur, par exemple **JBoss EAP 7.0**, puis cliquer sur **Suite**.

Figure 3.2. Définir un nouveau serveur



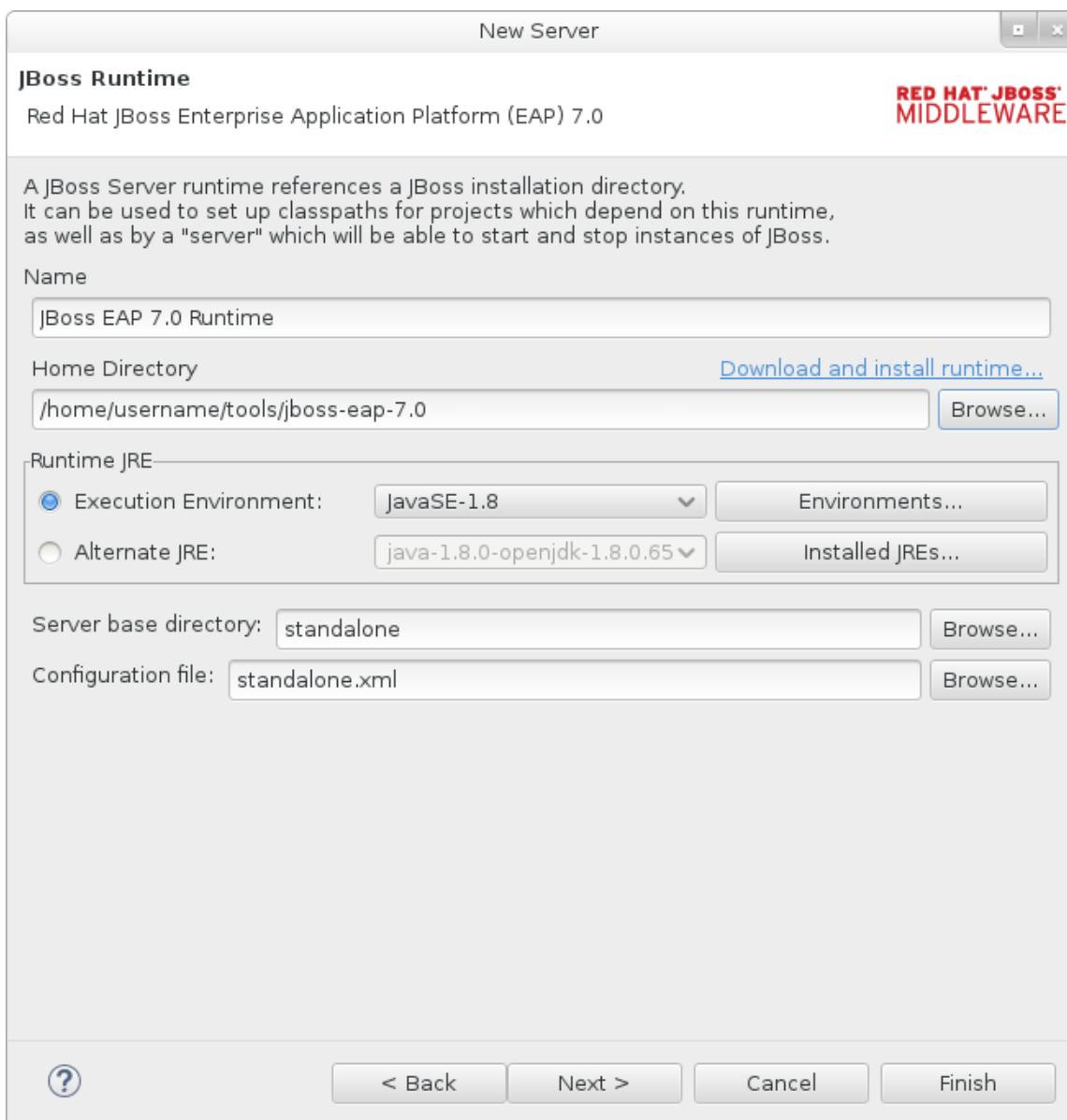
4. Créer un Adaptateur de serveur pour gérer le démarrage et l'arrêt du serveur. Conserver les valeurs par défaut, et cliquer sur **Suite**.

Figure 3.3. Créeer un nouvel Adpatateur de serveur



5. Saisir un nom, par exemple **JBoss EAP 7.0 Runtime**. Cliquer sur **Naviguer** à côté de **répertoire d'accueil** et naviguer vers votre répertoire d'installation de JBoss EAP. Cliquer ensuite sur **Suite**.

Figure 3.4. Ajouter l'environnement de runtime du nouveau serveur

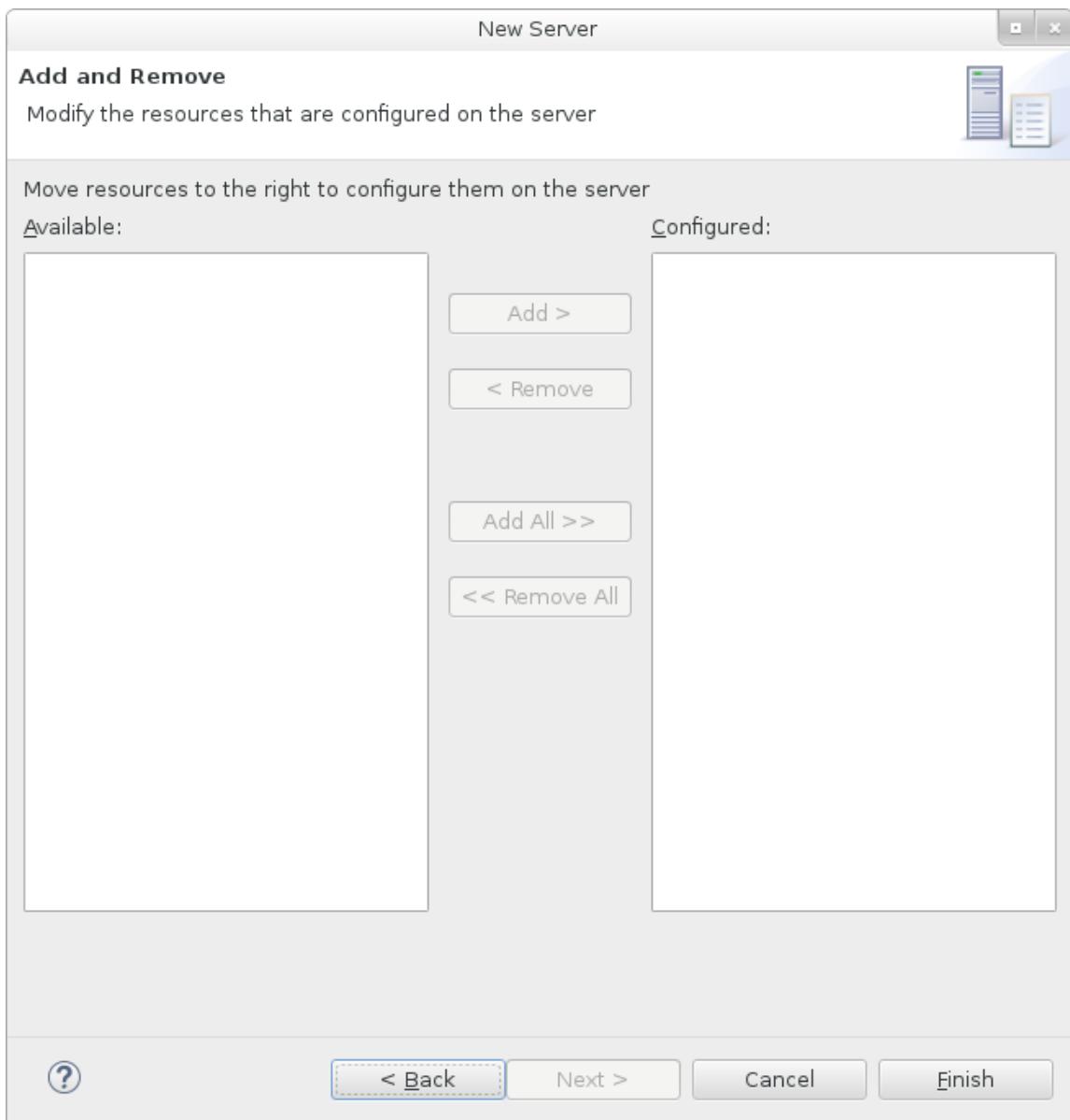


Note

Certains quickstarts exigent que vous exécutiez le serveur avec un profil différent ou avec des arguments supplémentaires. Par exemple, pour déployer un quickstart qui nécessite le profil **full**, vous devez définir un nouveau serveur et spécifier **standalone-full.xml** dans le champ **Fichier de configuration**. Veillez à donner un nom descriptif au nouveau serveur.

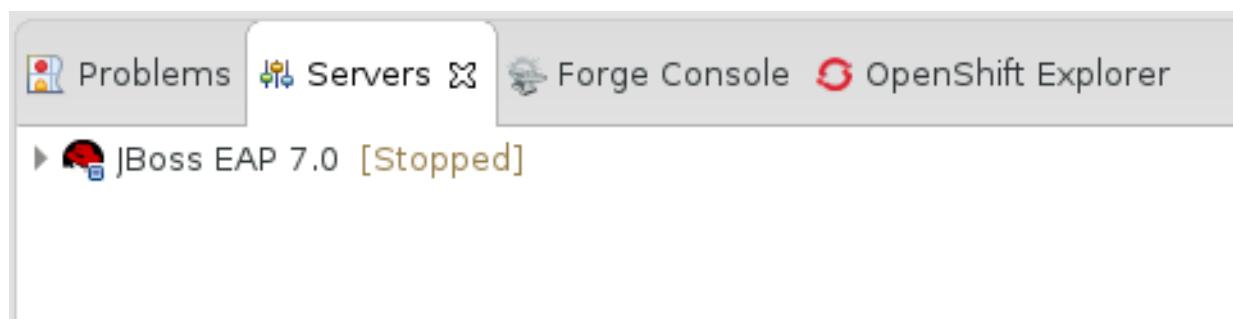
- Vous permet de configurer les projets existants pour le nouveau serveur. Puisque vous n'avez pas de projet à ce stade, cliquer sur **Terminé**.

Figure 3.5. Modifier les ressources dans le nouveau serveur



Le serveur **JBoss EAP 7.0** est désormais répertorié dans l'onglet **Serveurs**.

Figure 3.6. Liste de serveurs



3.3. UTILISATION DES EXEMPLES DE QUICKSTART

3.3.1. Maven

Apache Maven est un outil d'automation de builds distribuées, utilisé en développement Java pour créer, gérer et générer des projets de logiciels. Maven utilise des fichiers de

configuration nommés Project Object Model, ou POM, pour définir des projets et gérer le processus de build. Les POM décrivent le module et les dépendances du composant, et les sorties sont sous forme de fichier XML. Cela garantit que le projet soit construit de façon correcte et uniforme.



Note

Les exemples de Quickstart de JBoss EAP sont des projets Maven.

Maven y parvient en utilisant un référentiel. Un référentiel Maven stocke les bibliothèques, plug-ins et autres artefacts de la build. Le référentiel public par défaut est [Maven 2 Central Repository](#), mais les référentiels peuvent être privés et internes à une entreprise dans le but de partager des artefacts communs au sein d'équipes de développeurs. Les référentiels sont également disponibles auprès de tierces parties. Pour plus d'informations, voir le projet [Apache Maven](#) et le guide [Introduction aux référentiels](#).

JBoss EAP inclut un référentiel Maven contenant les prérequis que les développeurs Java EE utilisent généralement pour créer des applications sur JBoss EAP.

Pour obtenir des informations supplémentaires sur la façon d'utiliser Maven dans JBoss EAP, voir [Utiliser Maven dans JBoss EAP](#) dans le *Guide de développement* de JBoss EAP.

3.3.2. Utilisation de Maven avec des Quickstarts

Les artefacts et dépendances nécessaires pour créer et déployer des applications vers JBoss EAP 7 sont hébergés sur un référentiel public. En commençant par les quickstarts de JBoss EAP 7, il n'est plus nécessaire de configurer votre fichier `settings.xml` de Maven pour utiliser ces référentiels lors de la création des quickstarts. Les référentiels Maven sont désormais configurés dans les fichiers POM de projet de quickstart. Cette méthode de configuration est fournie pour faciliter le quickstart mais n'est généralement pas recommandée pour les projets de production car votre build peut être ralenti.

Red Hat JBoss Developer Studio inclut Maven, il est donc inutile de le télécharger et de l'installer séparément. Il est recommandé d'utiliser JBoss Developer Studio version 9.1 ou ultérieure.

Si vous souhaitez utiliser la ligne de commande Maven pour créer et déployer vos applications, vous devez tout d'abord télécharger Maven à partir du projet [Apache Maven](#) et l'installer à l'aide des instructions fournies dans la documentation Maven.

3.3.3. Télécharger et exécuter les quickstarts

3.3.3.1. Télécharger les quickstarts

JBoss EAP contient une bonne série d'exemples de code quickstart (démarrage rapide) conçus pour aider les utilisateurs à commencer à rédiger des applications en utilisant différentes technologies Java EE 7. Les quickstarts peuvent être téléchargés à partir du portail client Red Hat.

1. Se connecter au [Portail Clients de Red Hat](#)

2. Cliquer sur **Téléchargements**.
3. À partir de la liste **Téléchargements de produits**, cliquer sur **Red Hat JBoss Enterprise Application Platform**.
4. Sélectionner la version souhaitée à partir du menu déroulant **Version**.
5. Trouver l'entrée **Red Hat JBoss Enterprise Application Platform 7.0.0 Quickstarts** dans le tableau et cliquer sur **Télécharger**.
6. Enregistrer le fichier ZIP sur le répertoire souhaité.
7. Veuillez extraire le fichier ZIP.

3.3.3.2. Exécuter les quickstarts dans JBoss Developer Studio

Une fois les quickstarts téléchargés, ceux-ci peuvent être importés dans JBoss Developer Studio et déployés vers JBoss EAP.

Importer un quickstart dans JBoss Developer Studio

Chaque quickstart est fourni avec un fichier POM qui contient des informations de projet et de configuration. À l'aide de ce fichier POM, vous pouvez facilement importer le quickstart dans JBoss Developer Studio.

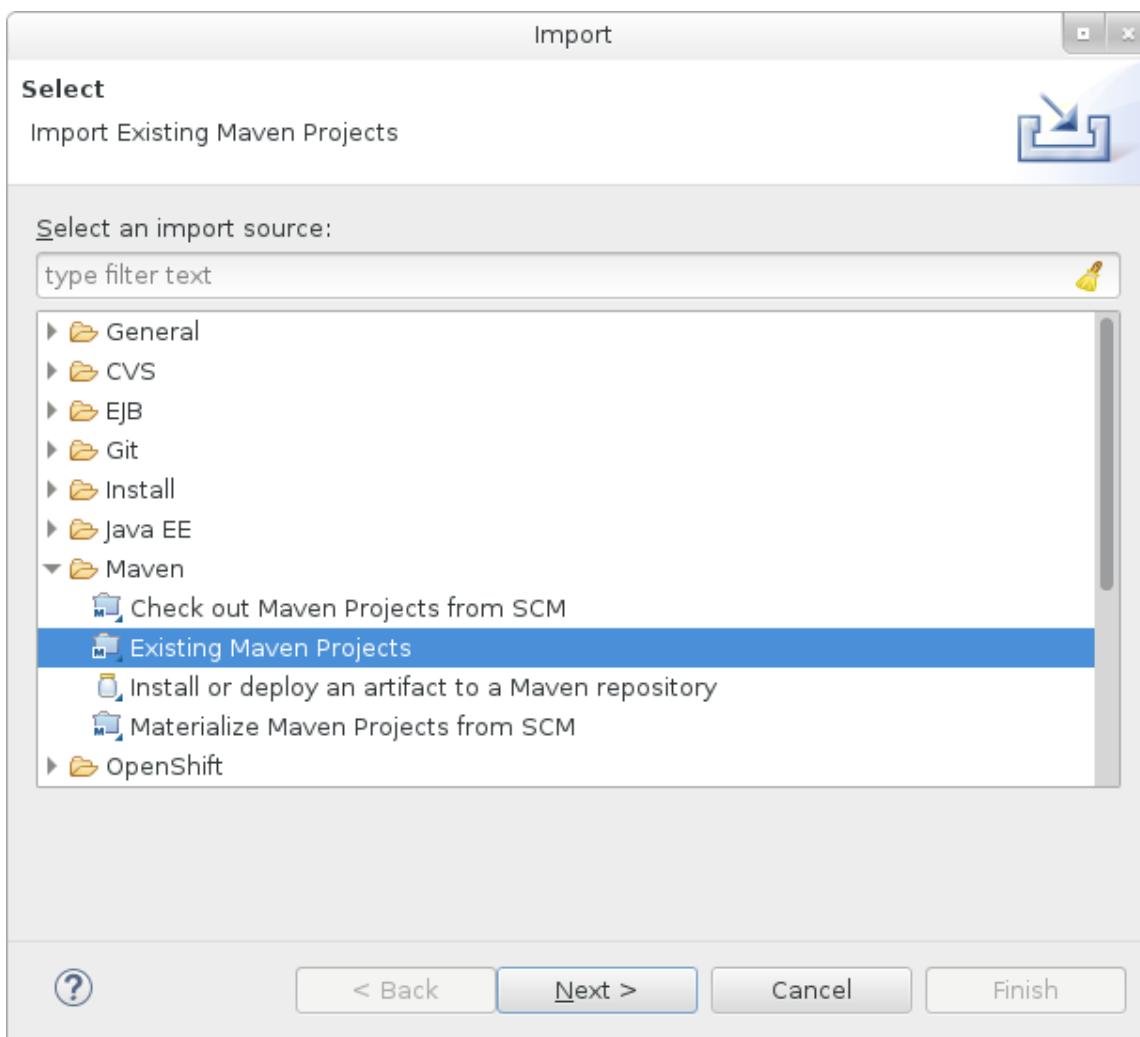


Important

Si votre dossier de projet quickstart se situe dans l'espace de travail IDE, lorsque vous l'importez dans JBoss Developer Studio, l'IDE génère un nom de projet non valide et un nom d'archive WAR. N'oubliez pas que votre dossier de projet quickstart doit être situé en dehors de l'espace de travail IDE avant de commencer.

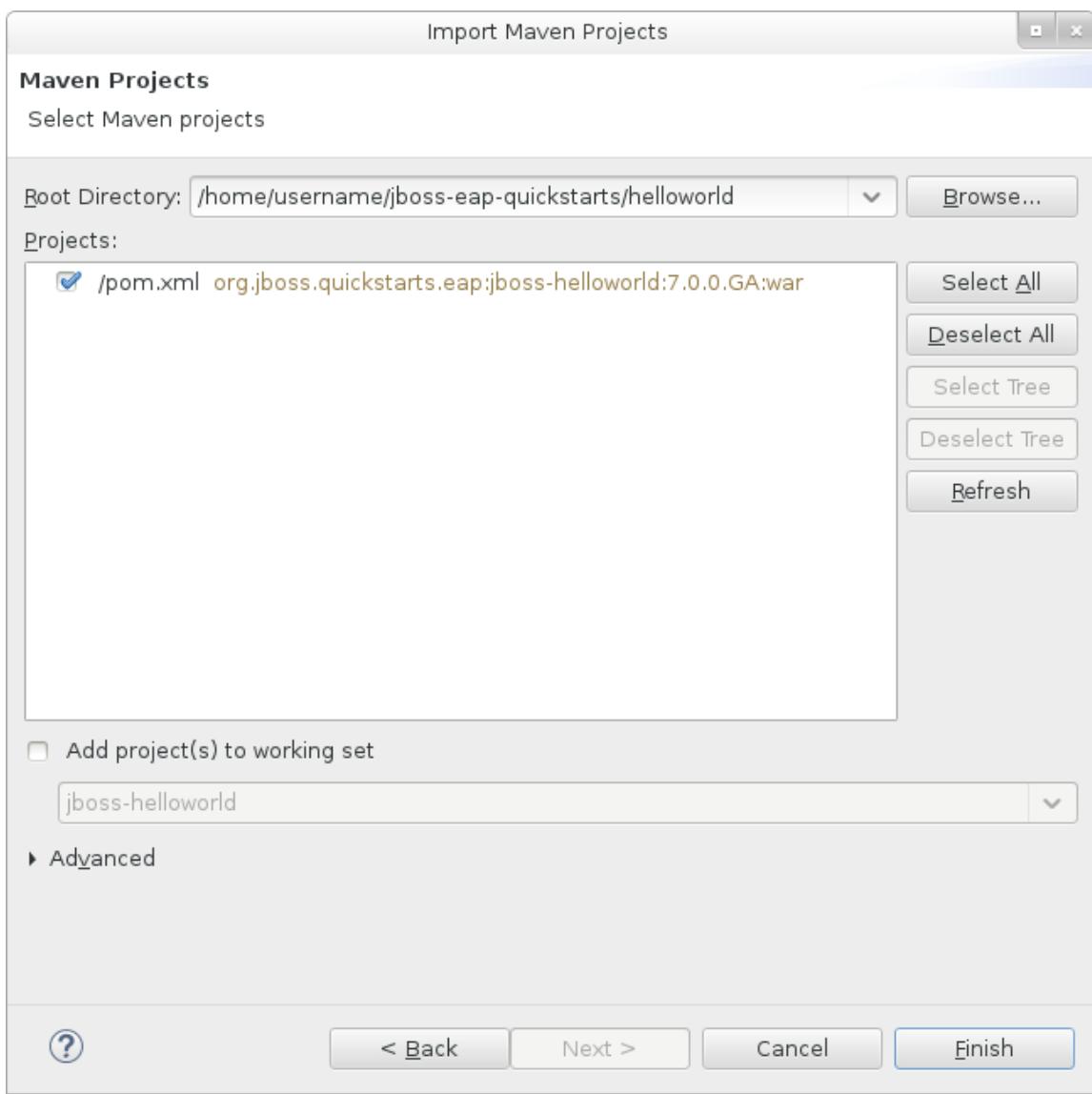
1. Démarrer JBoss Developer Studio.
2. Sélectionner **File → Import**.
3. Sélectionner **Maven → Projets Maven existants**, puis cliquer sur **Suite**.

Figure 3.7. Importer les projets Maven existants



4. Naviguer vers le répertoire de quickstart souhaité (par exemple le quickstart **helloworld**) puis cliquer sur **OK**. Le champ de la liste **Projects** se remplit avec le fichier **pom.xml** du projet de quickstart sélectionné.

Figure 3.8. Sélectionner les projets Maven



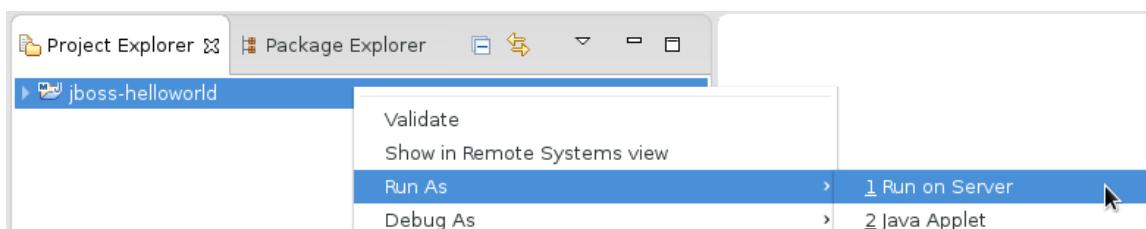
5. Cliquer sur **Finish**.

Exécuter le quickstart *helloworld*

Le quickstart **helloworld** est une manière simplifiée de vérifier que le serveur JBoss EAP est configuré et exécuté correctement.

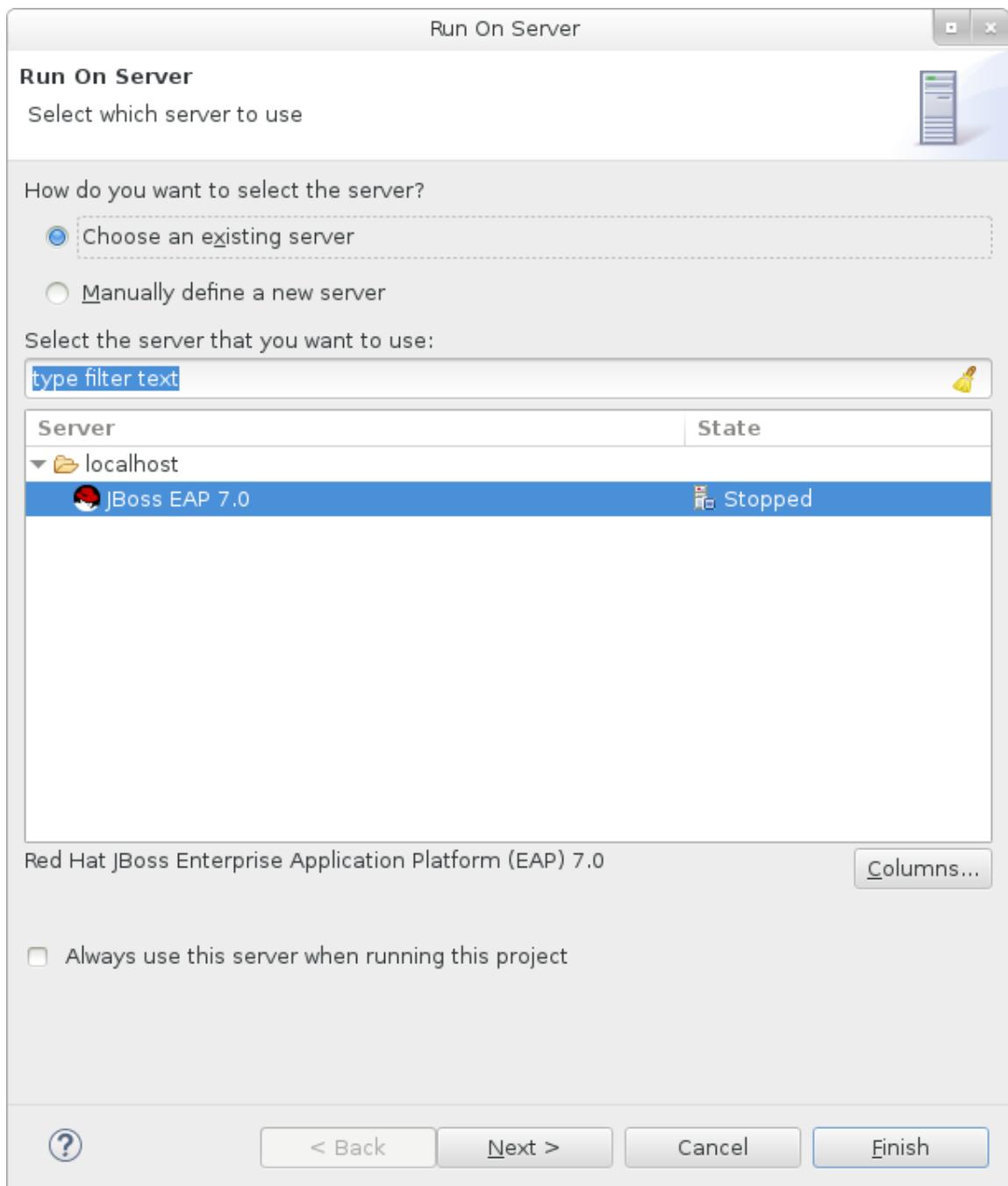
1. Si vous n'avez pas encore défini de serveur, ajouter le serveur JBoss EAP dans [JBoss Developer Studio](#)
2. Effectuer un clic droit sur le projet **jboss-helloworld** dans l'onglet **Project Explorer** et sélectionner **Run As → Run on Server**.

Figure 3.9. Run As - Run on Server



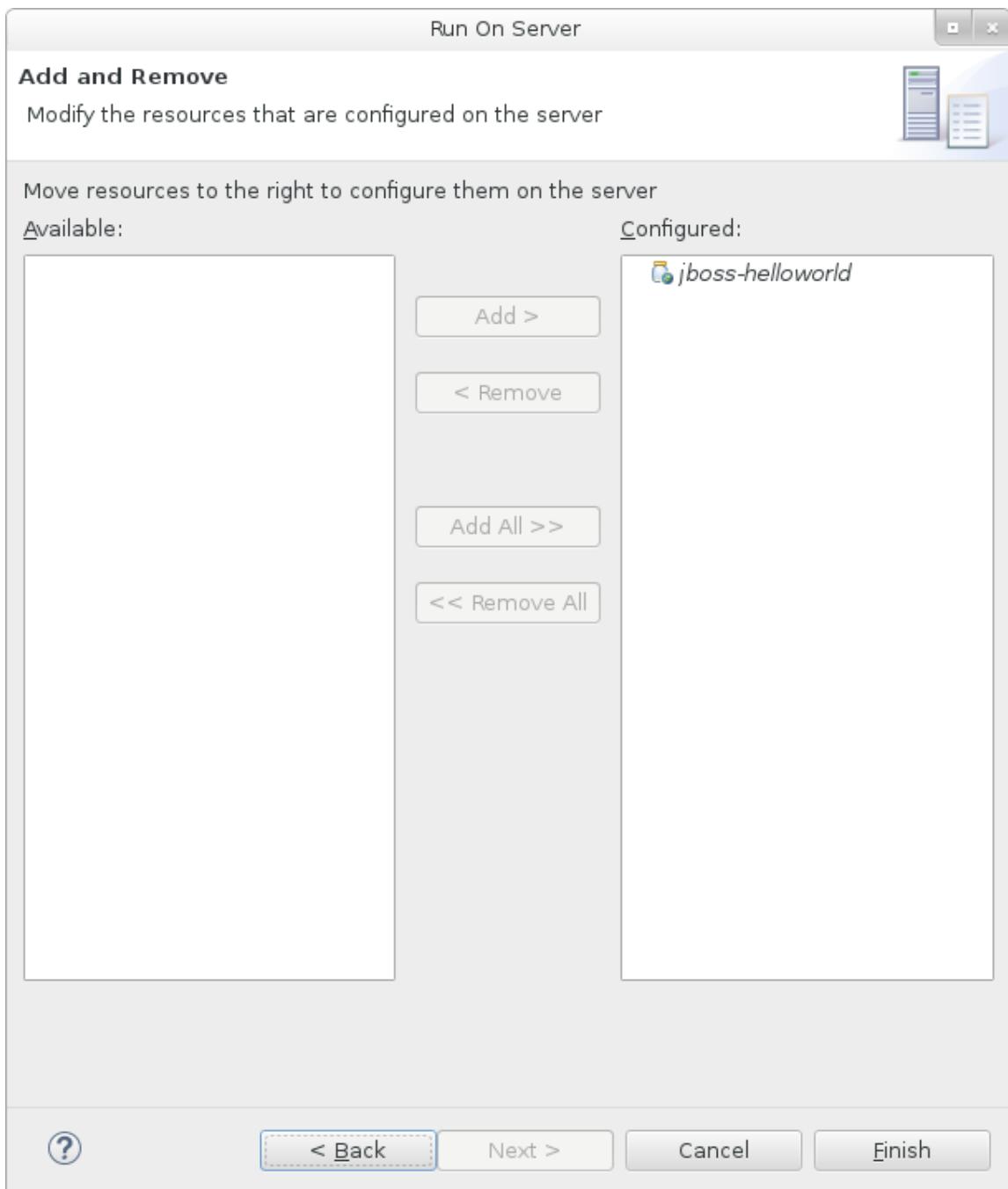
3. Sélectionner **JBoss EAP 7.0** à partir de la liste de serveurs et cliquer sur **Suite**.

Figure 3.10. Run on Server



4. Le quickstart **jboss-helloworld** apparaît déjà comme devant être configuré sur le serveur. Cliquer sur **Terminé** pour déployer le quickstart.

Figure 3.11. Modifier les ressources configurées sur le serveur



5. Vérifier les résultats.

- » Dans l'onglet **Server**, le statut du serveur **JBoss EAP 7.0** devient **Started** (Démarré).
- » L'onglet **Console** montre des messages détaillant le démarrage du serveur JBoss EAP et le déploiement du quickstart **helloworld**.

WFLYUT0021: Registered web context: /jboss-helloworld
 WFLYSRV0010: Deployed "jboss-helloworld.war" (runtime-name : "jboss-helloworld.war")
- » L'application **helloworld** est disponible sur <http://localhost:8080/jboss-helloworld> et affiche le texte **Hello World!**.

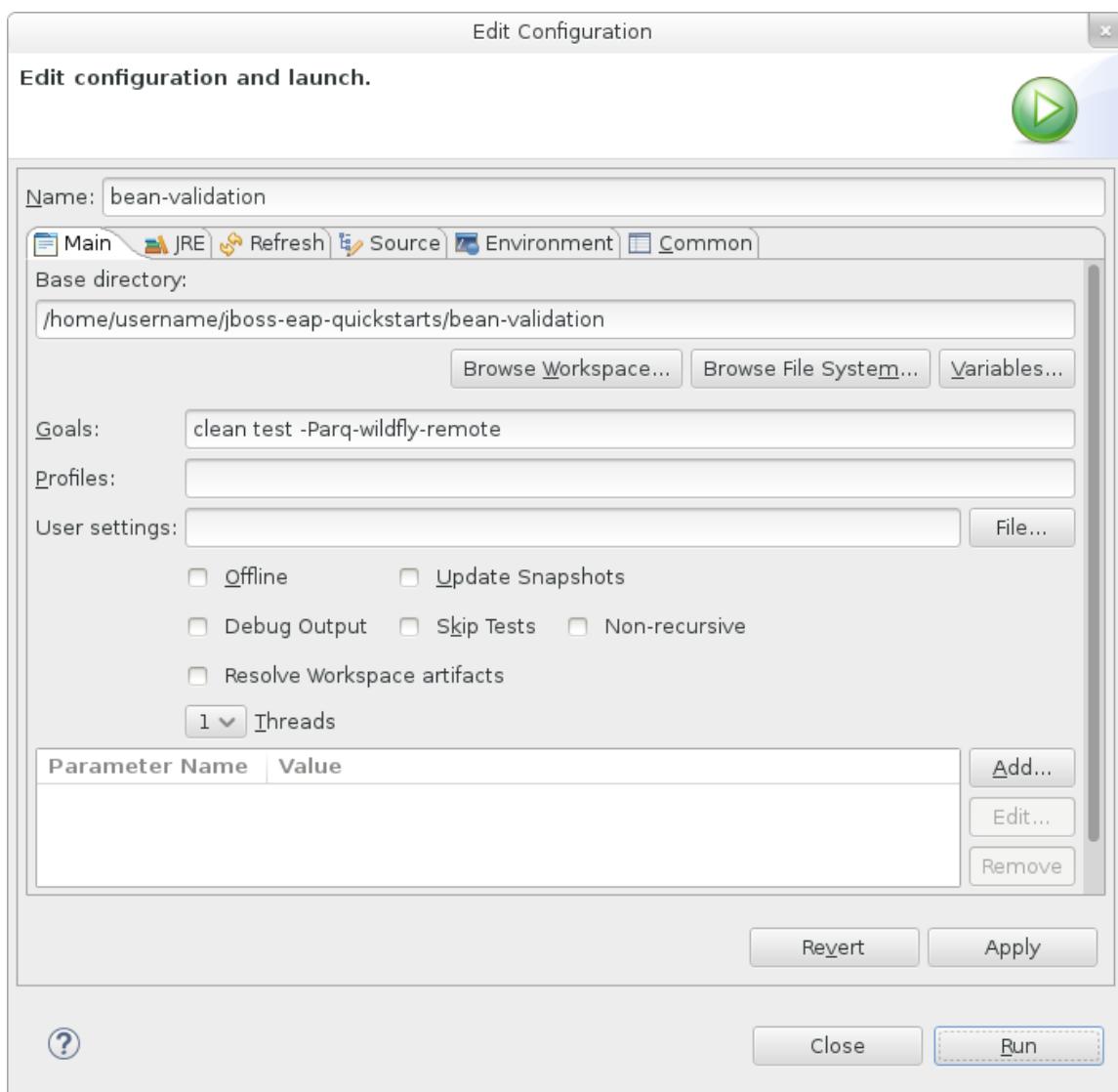
Exécuter le quickstart **bean-validation**

Certains quickstarts, comme **bean-validation** quickstart, ne fournissent pas de couche d'interface utilisateur, mais fournissent des tests Arquillian pour démontrer la fonctionnalité à la place.

1. Importer le quickstart **bean-validation** dans JBoss Developer Studio.
2. Dans l'onglet **Servers**, effectuer un clic droit sur le serveur et sélectionner **Start** pour démarrer le serveur JBoss EAP. Si vous ne voyez pas un onglet **Servers** ou n'avez pas encore défini un serveur, [ajoutez le serveur JBoss EAP au Red Hat JBoss Developer Studio](#).
3. Effectuer un clic droit sur le projet **jboss-bean-validation** dans l'onglet **Project Explorer** et sélectionner **Run As → Maven Build**.
4. Saisir ce qui suit dans le champ de saisie **Goals** puis cliquer sur **Run**.

```
clean test -Parq-wildfly-remote
```

Figure 3.12. Modifier la configuration



5. Vérifier les résultats.

L'onglet **Console** affiche les résultats des tests Arquillian**bean-validation** :

```
-----  
T E S T S  
-----  
Running  
org.jboss.as.quickstarts.bean_validation.test.MemberValidationTest  
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:  
2.189 sec  
  
Results :  
  
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0  
  
[INFO] -----  
-----  
[INFO] BUILD SUCCESS  
[INFO] -----  
-----
```

3.3.3.3. Exécuter les Quickstarts à partir de la ligne de commande

Vous pouvez facilement créer et déployer des quickstarts à partir de la ligne de commande en utilisant Maven. Si vous n'avez pas encore installé Maven, voir le projet [Apache Maven](#) pour le télécharger et l'installer.

A **README.md** file is provided at the root directory of the quickstarts that contains general information about system requirements, configuring Maven, adding users, and running the quickstarts.

Chaque quickstart contient son propre fichier **README.md** fournissant les instructions spécifiques et les commandes Maven qui servent à exécuter ce quickstart.

Exécuter le quickstart *helloworld* à partir de la ligne de commandes

1. Vérifier le fichier **README.md** dans le répertoire racine du quickstart *helloworld*.
2. Démarrer le serveur JBoss EAP.

```
$ EAP_HOME/bin/standalone.sh
```

3. Navuez dans le répertoire quickstart *helloworld*.
4. Construire et déployer le quickstart à l'aide de la commande Maven fournie dans le fichier **README.md** du quickstart.

```
mvn clean install wildfly:deploy
```

5. L'application *helloworld* se trouve maintenant à l'adresse suivante <http://localhost:8080/jboss-helloworld> et affiche le texte **Hello World!**.

ANNEXE A. MATÉRIEL DE RÉFÉRENCE

A.1. ARGUMENTS DE TEMPS D'EXÉCUTION DU SERVEUR

Le script de démarrage du serveur d'applications accepte les arguments et s'active lors du runtime. Cela permet au serveur de démarrer dans une configuration autre que celle définie dans les fichiers de configuration **standalone.xml**, **domain.xml** et **host.xml**.

Les configurations alternatives peuvent signifier de démarrer le serveur avec un ensemble de liaisons de sockets alternatif ou une configuration secondaire.

La liste des paramètres disponibles est accessible en exécutant l'aide avec l'option **-h** ou **--help** lors du démarrage.

Tableau A.1. Arguments et options de Runtime

| Argument ou Option | Mode d'opération | 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 requêtes de gestion, mais il ne pourra pas démarrer d'autres services de runtime ou accepter des requêtes de la part 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 requêtes 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 requêtes des contrôleurs hôte esclaves. |
| -b=<value>, -b <value> | Autonome, Serveur | Définir la propriété de système jboss.bind.address , utilisée dans la configuration de l'adresse de liaison pour l'interface publique. La valeur par défaut est 127.0.0.1 si aucune valeur n'est spécifiée. Voir l'entrée -b<interface>=<value> pour définir l'adresse de liaison pour les autres interfaces. |

| Argument ou Option | Mode d'opération | Description |
|--------------------------|----------------------|---|
| -b<interface>=<value> | Autonome, Serveur | Définir la propriété système jboss.bind.address.<interface> à la valeur donnée. Par exemple, -bmanagement=IP_ADDRESS |
| --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 | Nommer le fichier de configuration du 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, démarrer en utilisant 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 | Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est domain.xml . |

| Argument ou Option | Mode d'opération | Description |
|-------------------------------------|-------------------|---|
| -h, --help | Autonome, Serveur | Affiche le message d'assistance et sort. |
| --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, cette valeur est utilisée pour configurer l'adresse du contrôleur hôte maître. |
| --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, cette valeur est utilisée 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. |

| Argument ou Option | Mode d'opération | Description |
|--|-------------------|--|
| --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. |
| -secmgr | Autonome, Serveur | Exécute le serveur avec un gestionnaire de sécurité installé. |
| --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 | Définir la propriété système jboss.default.multicast.address utilisée pour configurer l'adresse multidiffusion des éléments de liaison de socket dans les fichiers de configuration. La valeur par défaut est 230.0.0.4 si aucune valeur n'est spécifiée. |
| -v, -V, --version | Autonome, Serveur | Affiche la version du serveur d'applications et sort. |

Avertissement

Les fichiers de configuration compris dans JBoss EAP sont configurés pour gérer les comportements des options (**-b**, **-u**). Si vous modifiez vos fichiers de configuration afin de ne plus utiliser la propriété système contrôlée par l'option, son ajout à la commande de lancement sera alors sans effet.

A.2. ARGUMENTS DE L'UTILITAIRE ADD-USER

Le tableau suivant décrit les arguments disponibles pour le script **add-user.sh** ou **add-user.bat**, un utilitaire pour ajouter de nouveaux utilisateurs au fichier de propriété pour une authentification non courante.

Tableau A.2. Arguments pour la commande Add-User

| Argument de ligne de commande | Description |
|---|--|
| -a | 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 <value> | 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 <value> | 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 <value> | 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 <value> | Une liste séparée par des virgules de groupes à assigner à cet utilisateur. |

| Argument de ligne de commande | Description |
|---------------------------------|--|
| -gp, --group-properties <value> | 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 <value> | Mot de passe associé à l'utilisateur. |
| -u, --user <value> | Le nom de l'utilisateur. Seuls les caractères alphanumériques et les symboles suivants sont valides : , . / = @ \ . |
| -r, --realm <value> | Le nom du domaine utilisé pour sécuriser les interfaces de gestion. S'il est omis, la valeur par défaut sera ManagementRealm . |
| -s, --silent | Exécuter le script add-user sans sortie vers la console. |
| -e, --enable | Active l'utilisateur. |
| -d, --disable | Désactive l'utilisateur. |
| -cw, --confirm-warning | Confirme les avertissements automatiquement en mode interactif. |
| -h, --help | Affiche les informations d'utilisation du script add-user . |

A.3. ATTRIBUTS D'INTERFACE

Tableau A.3. Attributs et valeurs d'interface

| Élément d'interface | Description |
|---------------------|--|
| any | Élément indiquant qu'une partie du critère de sélection d'une interface doit être qu'elle remplisse au moins un, mais pas forcément tous, les groupes de critères imbriqués. |
| 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. |
| inet-address | Soit une adresse IP en notation à points IPV6 ou IPV4, ou un nom d'hôte pouvant être résolu. |
| link-local-address | Élément vide indiquant qu'une partie du critère de sélection d'une interface devrait consister à savoir 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. |
| loopback-address | Une adresse de loopback qui ne peut pas réellement être configurée sur l'interface de loopback de la machine. Diffère du type d'adresse inet 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). |

| Élément d'interface | Description |
|---------------------|--|
| 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 indiquant qu'une partie du critère de sélection d'une interface doit être qu'elle ne remplisse aucun des ensembles de critères imbriqués. |
| 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 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 <i>slash notation</i> (e.g. 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. |

A.4. ATTRIBUTS DE LIAISONS DE SOCKETS

Tableau A.4. Attributs de liaisons de sockets

| Attribut | Description |
|-------------------|--|
| client-mappings | Spécifie les mappages de client pour cette liaison de socket. Un client qui se connecte à ce socket doit utiliser l'adresse de destination indiquée dans le mappage qui correspond à son interface de sortie désirée. Cela permet aux topologies de réseau avancées, qui utilisent une traduction d'adresses réseau ou qui ont des liaisons sur plusieurs interfaces de réseau, de fonctionner. Chaque mappage doit être évalué dans l'ordre déclaré, avec la première correspondance trouvée utilisée pour déterminer la destination. |
| fixed-port | Indique si la valeur de port doit rester fixe quand les décallages numériques sont appliqués aux autres sockets du groupe de sockets. |
| interface | Nom de l'interface à laquelle le socket doit être lié, ou, pour les sockets de multidiffusion, l'interface sur laquelle écouter. Cela doit correspondre à l'une des interfaces déclarées. Si non définie, la valeur de l'attribut d'interface par défaut du groupe de liaison de sockets englobant sera utilisée. |
| multicast-address | Adresse multidiffusion sur laquelle le socket doit recevoir le trafic multidiffusion. Si non spécifié, le socket ne sera pas configuré pour recevoir le trafic multidiffusion. |
| multicast-port | Port sur lequel le socket doit recevoir le trafic multidiffusion. Doit être configuré si l'adresse multidiffusion est configurée. |
| name | Le nom du socket. Les services ayant besoin d'accéder aux informations de configuration de socket pourront le faire par l'intermédiaire de son nom. Cet attribut est exigé. |
| port | Le numéro du port dans lequel le socket doit être lié. Notez que cette valeur peut être remplacée si les serveurs appliquent une valeur de décalage de port pour incrémenter ou décrémenter toutes les valeurs de port. |

A.5. LIAISONS DE SOCKET PAR DÉFAUT

Tableau A.5. Liaisons de socket par défaut

| Nom | Port | Port multi-diffusion | Description | Groupes de liaisons de sockets |
|----------------|-------------|-----------------------------|--|---|
| ajp | 8009 | | Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| http | 8080 | | Le port par défaut des applications déployées. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| https | 8443 | | Connexion cryptée-SSL entre les applications déployées et les clients. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| iiop | 3528 | | Services CORBA pour les transactions JTS et autres services dépendants-ORB. | full-sockets, full-ha-sockets |
| iiop-ssl | 3529 | | Services CORBA cryptés-SSL. | full-sockets, full-ha-sockets |
| jgroups-mping | | 45700 | Multidiffusion. Utilisée pour découvrir l'appartenance de groupe d'origine dans un cluster HA. | ha-sockets, full-ha-sockets |
| jgroups-tcp | 7600 | | Découverte d'homologues unicastes dans les groupements HA avec TCP. | ha-sockets, full-ha-sockets |
| jgroups-tcp-fd | 57600 | | Utilisé pour la détection des échecs en TCP. | ha-sockets, full-ha-sockets |
| jgroups-udp | 55200 | 45688 | Découverte de paires multicast dans les groupements HA avec UDP. | ha-sockets, full-ha-sockets |

| Nom | Port | Port multi-diffusion | Description | Groupes de liaisons de sockets |
|--------------------------|-------------|-----------------------------|--|---|
| jgroups-udp-fd | 54200 | | Utilisé pour la détection des échecs par UDP. | ha-sockets, full-ha-sockets |
| management-http | 9990 | | Utilisé pour les communications HTTP avec la couche de gestion. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| management-https | 9993 | | Utilisé pour les communications HTTPS avec la couche de gestion. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| modcluster | 23364 | | Port de multidiffusion pour la communication entre JBoss EAP 6 et l'équilibrEUR de charges HTTP. | ha-sockets, full-ha-sockets |
| txn-recovery-environment | 4712 | | Gestionnaire de recouvrement des transactions JTA. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |
| txn-status-manager | 4713 | | Gestionnaire des transactions JTA / JTS. | standard-sockets, ha-sockets, full-sockets, full-ha-sockets |