# XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559 Rev B

# Contents

# 1 About XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559 Rev B

*XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559 Rev B*, published 04 May 2017, supersedes *XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559 Rev A*, which was published 3 April 2017.

*XC™ Series Software Installation and Configuration Guide* (S-2559) supersedes *System Management Workstation (SMW) Software Installation Guide* (S-2480) and *CLE Installation and Configuration Guide* (S-2444).

## Scope and Audience

The *XC™ Series Software Installation and Configuration Guide* (S-2559) provides overview information and procedures to install, update, and customize System Management Workstation (SMW) and Cray Linux Environment (CLE) software and install the SMW base operating system, SUSE® Linux Enterprise Server version 12 (SLES® 12).

This publication does not include procedures for administering a Cray XC Series system; for those, see *XC™ Series System Administration Guide* (S-2393).

This publication is intended for system installers, administrators, and anyone who installs and configures software on a Cray XC™ Series system. It assumes some familiarity with standard Linux and open source tools (e.g., zypper/yum for RPMs, Ansible, YAML/JSON configuration data).

## CLE 6.0.UP03 / SMW 8.0.UP03 Release

*XC™ Series Software Installation and Configuration Guide (CLE 6.0.UP03) S-2559 Rev B* supports Cray software release CLE 6.0.UP03 / SMW 8.0.UP03 for Cray XC™ Series systems, released on 16 February 2017.

In the CLE 6.0.UP01 / SMW 8.0.UP01 release, this publication was titled *XC™ Series Software Initial Installation and Configuration Guide*. The new title complies with the standard titling convention adopted by the technical publications department. Previous versions of this publication will not be retitled.

**New in Revision B**

- The *Set the Turbo Boost Limit* on page 195 procedure was updated to reflect currently released processors.

## Command Prompt Conventions

**Host name and account in command prompts**

The host name in a command prompt indicates where the command must be run. The account that must run the command is also indicated in the prompt.

- The `root` or super-user account always has the `#` character at the end of the prompt.
- Any non-`root` account is indicated with *`account@hostname`*`>`. A user account that is neither `root` nor `crayadm` is referred to as `user`.

| | |
|---|---|
| `smw#` | Run the command on the SMW as `root`. |
| `cmc#` | Run the command on the CMC as `root`. |
| `sdb#` | Run the command on the SDB node as `root`. |
| `crayadm@boot>` | Run the command on the boot node as the `crayadm` user. |
| `user@login>` | Run the command on any login node as any non-`root` user. |
| *`hostname`*`#` | Run the command on the specified system as `root`. |
| `user@`*`hostname`*`>` | Run the command on the specified system as any non-`root` user. |
| `smw1#`<br>`smw2#` | For a system configured with the SMW failover feature there are two SMWs—one in an active role and the other in a passive role. The SMW that is active at the start of a procedure is *smw1*. The SMW that is passive is *smw2*. |
| `smwactive#`<br>`smwpassive#` | In some scenarios, the active SMW is *smw1* at the start of a procedure—then the procedure requires a failover to the other SMW. In this case, the documentation will continue to refer to the formerly active SMW as *smw1*, even though *smw2* is now the active SMW. If further clarification is needed in a procedure, the active SMW will be called *smwactive* and the passive SMW will be called *smwpassive*. |

**Command prompt inside chroot**

If the `chroot` command is used, the prompt changes to indicate that it is inside a chroot environment on the system.

```
smw# chroot /path/to/chroot
chroot-smw#
```

**Directory path in command prompt**

Example prompts do not include the directory path, because long paths can reduce the clarity of examples. Most of the time, the command can be executed from any directory. When it matters which directory the command is invoked within, the `cd` command is used to change into the directory, and the directory is referenced with a period (`.`) to indicate the current directory.

For example, here are actual prompts as they appear on the system:

```
smw:~ # cd /etc
smw:/etc# cd /var/tmp
smw:/var/tmp# ls ./file
smw:/var/tmp# su - crayadm
crayadm@smw:~> cd /usr/bin
crayadm@smw:/usr/bin> ./command
```

And here are the same prompts as they appear in this publication:

```
smw# cd /etc
smw# cd /var/tmp
smw# ls ./file
smw# su - crayadm
crayadm@smw> cd /usr/bin
crayadm@smw> ./command
```

## Typographic Conventions

`Monospace` Indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, and other software constructs.

**`Monospaced Bold`** Indicates commands that must be entered on a command line or in response to an interactive prompt.

*`Oblique`* or *`Italics`* Indicates user-supplied values in commands or syntax definitions.

**Proportional Bold** Indicates a graphical user interface window or element and key strokes (e.g., **Enter**, **Alt-Ctrl-F**).

\ (backslash) At the end of a command line, indicates the Linux® shell line continuation character (lines joined by a backslash are parsed as a single line). Do not type anything after the backslash or the continuation feature will not work correctly.

## Feedback

Your feedback is important to us. Visit the Cray Publications Portal at *http://pubs.cray.com* and make comments online using the **Contact Us** button in the upper-right corner, or email comments to pubs@cray.com.

## Trademarks

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, URIKA, and YARCDATA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYDOC, CRAYPAT, CRAYPORT, DATAWARP, ECOPHLEX, LIBSCI, NODEKARE. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

# 1.1 Related Publications

This publication supersedes *System Management Workstation (SMW) Software Installation Guide* (S-2480) and *CLE Installation and Configuration Guide* (S-2444).

Although this publication is all that is necessary for installing SMW and CLE software, the following publications contain additional information that may be helpful. The release errata and readme files are available on CrayPort, and the rest of these publications (and other Cray publications) can be found at *http://pubs.cray.com*.

- *SMW Release Errata* (includes notice of any patches) and the *SMW README*, which are provided with the SMW release software
- *CLE Release Errata* and the *CLE README*, which are provided with the CLE release software
- *XC™ Series Configurator User Guide* (S-2560)
- *XC™ Series System Administration Guide* (S-2393)
- *XC™ Series Boot Troubleshooting Guide* (S-2565)
- *XC™ Series Lustre® Administration Guide* (S-2648)
- *XC™ Series Power Management Administration Guide* (S-0043)
- *XC™ Series System Environment Data Collections (SEDC) Guide* (S-2491)
- *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564), which supersedes *DataWarp Installation Guide* (S-2547)
- *Cray Compiling Environment Release Overview and Installation Guide*
- *XC™ Series eLogin Installation Guide* (S-2566)
- *XC™ Series SEC Configuration Guide* (S-2542), which describes the Cray Simple Event Correlator
- *XC™ Series Aries™ Network Resiliency Guide* (S-0041)
- For a system that will use DVS for projecting external file systems:
  - *XC™ Series DVS Administration Guide* (S-0005)
  - *XC™ Series GPFS Software Installation Guide (CLE 6.0.UP03) S-2569*
- For a system that will be configured for SMW high availability (HA):
  - *XC™ Series SMW HA Installation Guide* (S-0044)
  - *XC™ Series SMW HA Administration Guide* (S-2551)

## 1.2 Distribution Media

The Cray CLE 6.0.UP03 / SMW 8.0.UP03 release distribution media consist of one DVD and several other pieces of media that may be DVDs or ISO files.

Configuration worksheets for CLE config sets and the global config set are also included in this distribution, so that sites can begin entering site-specific configuration data in them before and during the installation process.

For an initial installation of the SMW/CLE software, first install the DVD labeled Cray-SMWbase12-201511021655, which contains the SMW base operating system, SUSE Linux Enterprise Server version 12 (SLES 12). This table shows all installation media included with this release.

| | |
|---|---|
| bootable SMW SLES12 media | ● `Cray-SMWbase12-201511021655.iso` DVD |
| SMW release | ● `smw-8.0.3075-201701182038.iso` |
| CLE release | ● `cle-6.0.3074-201701182038.iso` |
| SLES release | ● `SLE-12-Server-DVD-x86_64-GM-DVD1.iso`<br>● `SLE-12-SDK-DVD-x86_64-GM-DVD1.iso` |

| | |
|---|---|
| | ● `SLE-12-WE-DVD-x86_64-GM-DVD1.iso`<br>● `SLE-12-Module-Legacy-DVD-x86_64-GM-DVD1.iso`<br>● `SLE-12-Module-Public-Cloud-DVD-x86_64-GM-DVD1.iso` |
| SLE update | ● `sleupdate-sle12sp0+161026-201611021158.iso` |
| CentOS software | ● `CentOS-6.5-x86_64-bin-DVD1.iso` |
| CLE and global configuration worksheets | ● cle-*MMDD*-worksheets.tar<br>● global-*MMDD*-worksheets.tar |
| **(SMW HA only)** SLE HA software | ● `SLE-12-HA-DVD-x86_64-GM-CD1.iso` |
| **(SMW HA only)** SMW HA release | ● `smwha-sleha12sp0-12.0.3079-201701241124.iso` |

# 2 Introduction to Installation and Configuration of Cray XC™ Software

This guide provides information and instructions to perform an initial installation of System Management Workstation (SMW) and Cray Linux Environment (CLE) software release packages on a Cray XC Series system, update SMW and CLE software, and customize a preinstalled system.

With the SMW 8.0 / CLE 6.0 release, Cray has changed the way software is installed, configured, and managed on XC Series systems. The changes that most affect installation and configuration are summarized here.

The new Cray Management System (CMS)

- uses a common installation process for SMW and CLE (which is why there is now a single installation guide for XC systems—this one—instead of separate guides for SMW and CLE)
- leverages standard Linux and common open source tools (e.g., zypper/yum for RPMs, Ansible, YAML/JSON configuration data)
- keeps software images and configuration separate until boot
  - prescriptive image creation using recipes
  - centralized configuration
  - configuration applied at boot time or after configuration adjusted

The core elements of this new management system are:

**IMPS** Image Management and Provisioning System (IMPS) is responsible for creating and distributing repository content and for prescriptive image creation. Note that although filepaths for configuration data and tools include `imps`, this is an artifact of an early implementation that grouped both image and configuration management under IMPS. IMPS is now image management only.

**CMF** Configuration Management Framework (CMF) comprises the configuration data (stored in config sets on the SMW), tools to manage and distribute that data (e.g., the configurator and the IMPS Distribution System (IDS)), and software to apply the configuration data to the running image (Ansible plays).

**NIMS** Node Image Mapping Service (NIMS) is responsible for keeping track of which images get booted on which nodes, what additional kernel parameters to pass to nodes at boot time, and which load file to use within a boot image.

What else is new?

- New base operating system for the SMW/CLE: SUSE® Linux Enterprise Server version 12 (SLES® 12) for x86_64
- New base operating system for HSS (Hardware Supervisory System) controllers: OpenSUSE 13.2 for 32 bit
- New modular installer

Much of the software remains unchanged, for the most part, such as Application-level Placement Scheduler (ALPS), Node Health Checker (NHC), and Resource Utilization Reporting (RUR), among others.

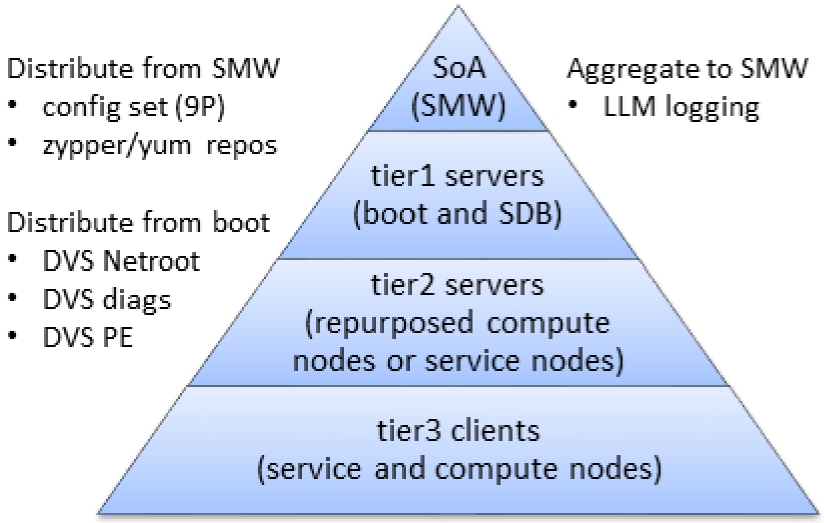
This guide includes procedures for installing the following software: the SMW base operating system, Cray SMW software, CLE software, SLES security updates, Cray Programming Environment (PE), and direct-attached Lustre (DAL), if needed.

## 2.1 About Cray Scalable Services

Cray Scalable Services is an essential part of the Cray Management System that is used to both distribute and aggregate information. Within Cray Scalable Services, nodes are designated as SoA (server of authority), tier1, tier2, or tier3. A node can be a member of only one of these groups. Tier1 nodes are clients of the SoA and servers for tier2 nodes. Tier2 nodes are clients of tier1 nodes and servers for tier3 nodes. Tier3 nodes are clients of tier2 nodes. Configuration of nodes as SoA, tier1, and tier2 is defined in the cray_scalable_services configuration service, which must be configured properly for the system to function.

As indicated in this figure, the SMW is the designated SoA in Cray XC systems. The boot and SDB nodes are designated tier1 nodes, and they must have direct network connectivity to the SMW via Ethernet. Typically, tier2 nodes are service nodes or repurposed compute nodes that have no other duties beyond being part of the Scalable Services. All other nodes are tier3 nodes.

*Figure 1. Cray Scalable Services*



This table shows what gets distributed or aggregated using Cray Scalable Services.

| | |
|---|---|
| **from SMW to rest of system** | ● config set data is shared using a 9P file system and DIOD (distributed I/O daemon)<br>● zypper software repositories can by used from any node with the Live Update feature (http forwarding from the SMW through the tiers) |
| **from boot node to rest of system** | ● PE (Programming Environment) image root<br>● diag (online diagnostics) image root<br>● Netroot image roots[1] |

| | |
|---|---|
| **from rest of system to SMW** | ● Lightweight Logging Manager (LLM) logging |

Here is an example of how Scalable Services works with Live Updates to distribute software out to nodes. Any tier3 node can run zypper to access the repositories on the SMW because it has an entry in `/etc/zypp/repos.d/liveupdates.repo` that points to the tier2 nodes by means of a baseurl, which uses http protocol listing all of the tier2 nodes. The tier2 nodes, in turn, have an entry in `/etc/zypp/repos.d/liveupdates.repo` that lists at least one tier1 node. All tier1 nodes have an entry in `/etc/zypp/repos.d/liveupdates.repo` that lists the SMW.

### Services that Depend on Cray Scalable Services

It is important to configure Cray Scalable Services correctly. The following features and services use data from the cray_scalable_services configuration service, and may they not be functional if cray_scalable_services is configured incorrectly.

**Node Image Mapping Service (NIMS) plugin**
Uses cray_scalable_services data to determine tier1 servers and adds the tier1 kernel command line parameter to each tier1 server.

**IMPS Distribution Service (IDS)**
Uses cray_scalable_services data to set the ids kernel command line parameter to the node's parent, from whom it will receive config set data.

**DVS Ansible configuration**
Uses cray_scalable_services data to determine which nodes should serve DVS file systems. This will also impact Netroot functionality, which uses DVS.

**CLE liveupdates functionality**
Configured using cray_scalable_services data to determine the parent each node should contact en route to the package repos stored on the SMW.

**LLM Ansible configuration**
Uses cray_scalable_services data to determine the next server to which a node should send its log data, which depends on the node's tier.

**NFS Ansible configuration**
Uses cray_scalable_services data to determine which nodes should act as clients and servers.

**IP forwarding Ansible configuration**
Uses cray_scalable_services data to enable IP forwarding and configure servers' routes depending on their tier.

## 2.2 About Config Sets

Users invoke the `cfgset` command to take configuration content delivered in service packages and combine it with site-specific configuration content gathered either interactively or through bulk import. The results are used by `cfgset` to create a config set, which is a central repository that stores all configuration information necessary to operate the system. Config sets reside on the management node (e.g., the SMW) in `/var/opt/cray/imps/config/sets` by default. The contents of each config set reside in the following subdirectories:

**`ansible`**
Local site-provided Ansible play content can be placed here for distribution with the config set. When the config set is created, `cfgset` copies Ansible content from service packages to this location. Whenever the config set is updated, `cfgset` copies Ansible content from service

[1] Netroot is a mechanism that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system which available to the node via a network mount.

packages again, overwriting the previous service-package Ansible content and leaving the site-provided content unchanged.

**`changelog`** YAML change logs from previous sessions with the configurator.

**`config`** Configuration templates containing configuration information. When the config set is created, the configurator copies service package templates to this location. Users can modify the content of these templates using `cfgset` to invoke the configurator. Whenever the config set is updated, the configurator merges service package templates with the templates in this location.

**`dist`** Other site-provided content, such as static files required for the configuration of a service, can be placed here for distribution with the config set. When the config set is created, `cfgset` copies dist content from service packages to this location. Whenever the config set is updated, `cfgset` copies dist content from service packages again, overwriting the previous service-package dist content and leaving the site-provided content unchanged.

**`files`** Files necessary for system configuration that are distributed with the config set. They can be placed here by:

- the `cfgset` command, which runs configuration callback scripts to generate some configuration files (e.g., `/etc/hosts`)
- the Simple Sync service
- local site administrators

**`worksheets`** Configuration worksheets generated by the configurator using data stored in the configuration templates in the `config` subdirectory of the config set. Administrators copy these worksheets to a location outside the config set, edit them with site-specific configuration data, and then import them to create a new config set or update an existing one.

## Config Set Types

All config sets have a *type* associated with them that is specified upon creation. XC systems require both a `global` config set type and a `cle` config set type. After a config set of a given type is created, its type cannot be changed. A user may create multiple config sets to support partitioned systems or alternate configurations. Typically a config set of type `cle` is created for each partition to store partition- and CLE-specific content, and another config set of type `global` is created to store configuration data that pertains to the management node domain as well as configuration data that can be easily shared among `cle` config sets. Config sets can be portable between partitions or to other systems if their partition-specific information is modified accordingly.

## Configuration Service Inheritance

When a config set is created or updated, only service package templates that match the type of the config set can be included in the config set. Cray provides several service package templates that match both types and can be included in both `cle` and `global` config sets. In such cases, the user can choose which template will be used to configure the service in question. When a `cle` config set is created, and a service that has a template of both types is ready for configuration, the configurator will inject an initial question for the user to choose between configuring the service (i.e., using the `cle` version of the template) or letting the service inherit configuration values from the `global` config set (i.e., inheriting values from the `global` version of the template). Configuration worksheets for such services also provide that choice by including an inherit field, which can be set to true or false. If the user sets it to true, the configuration data from the global config set version of the service will be used. When the Cray-provided `cray-ansible` service (part of the Cray Configuration Management Framework) is run at boot time or at the system administrator's discretion, it uses the value of the inherit field to determine which configuration template data (`global` or `cle`) to use.

Inheritance is useful for systems with multiple partitions where a subset of partitions need custom configuration of a service, but another subset of partitions can all share the same global configuration.

## 2.3 About Variable Names in the Configurator and Configuration Worksheets

In the configurator and configuration worksheets, variable names can be quite long because they are composed of a data structure hierarchy. Each variable name begins with the name of the service to which it belongs. The next part of each name is always 'settings' to indicate that what follows is a *service setting*, one of the available settings for that service. After 'settings' comes the name of the setting, which could be a simple data type (string, boolean, integer, etc.) or a more complex data type (list, multival, etc.). The next part after the name of the setting is always 'data' to indicate that what follows is one of the fields of that setting. For a full description of data types, see *XC™ Series Configurator User Guide* (S-2560).

For example, here is the variable for the IP address of the high-speed network (HSN), one of several networks.

```
cray_net.settings.networks.data.hsn.ipv4_network
```

This variable belongs to the `cray_net` service and the `networks` setting of that service. The networks setting is of type multival, which means it can have multiple entries, and each entry can have multiple fields to set. This variable targets the `ipv4_network` field of the `hsn` network entry.

This example shows the variable for the IP address of the HSN SDB node alias interface (one of several interfaces) of the SDB node (one of several hosts).

```
cray_net.settings.hosts.data.sdbnode.interfaces.hsn_sdb_alias.ipv4_address
```

This variable belongs to the `cray_net` service and the `hosts` setting of that service. The hosts setting is of type multival, and this variable belongs to the `sdbnode` host entry. The sdb_node host has a field `interfaces`, which is also of type multival. This variable targets the `ipv4_address` field of the `hsn_sdb_alias` interface entry.

## 2.4 About Snapshots and Config Set Backups

Sites can make as few or as many snapshots and config set backups as they deem useful, but Cray recommends that sites make a snapshot and back up config sets at certain milestones during the installation and configuration process. Most of these will be for archival purposes, but snapshots and config set backups can be used to stage updates/upgrades and roll back to or switch between SMW and CLE releases as well.

Snapshots are created and managed using `snaputil`, a Python utility delivered with the cray-install-support RPM that is installed by default on the SMW. However, the fresh install procedure makes the first snapshot manually, because at that point in the process, `snaputil` has not yet been installed. Config sets are created and managed using `cfgset`. Procedures for how to create snapshots and config set backups are included at each point in the process where they are needed.

**What does a snapshot contain?** Snapshots capture content in these three file systems on the SMW: root (`/`), `/var/lib/mysql`, and `/var/opt/cray/repos`. Used in conjunction with backups of config sets, they provide enough information to be able to re-create the state of the system at the time of the snapshot and config set backup.

**What does a config set contain?** See *About Config Sets* on page 14 for details about the contents of a config set.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

*Table 1. Suffixes and Corresponding Milestones for Snapshots and Config Set Backups*

| Suffix | Description | Snapshot | Config Set |
|---|---|---|---|
| preupdate | before beginning any software update activities (software updates only) | yes | yes |
| preconfig | after installing a software update and before updating the global and CLE config sets (software updates only) | no | yes |
| postinstall | after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware | yes | yes |
| postconfig | after configuring CLE and before booting the CLE system | yes | yes |
| postboot | after booting the CLE system | yes | yes |
| postpe | after installing Cray PE software | yes | yes |
| postcustomize | after customizing a preinstalled system (not for fresh installs or software updates) | yes | yes |

## 2.5 About Config Set Caching

Config sets are defined and reside on the Server of Authority, which on XC systems is the SMW. Config set content is made available to all nodes in the system by means of Cray Scalable Services.

To make the sharing of config set content both quick and reliable, the cray-cfgset-cache service was created. It caches config sets locally on nodes (compressed for a smaller footprint). On the SMW, it does the following:

- notices changes to config sets on the SMW
- refreshes the local caches dynamically
- detects failures and retries automatically

The cray-cfgset-cache service ensures that config set content gets refreshed on all nodes whenever config sets are created or updated on the SMW. It is triggered when `cray-ansible` is run on a node with the `start`, `restart`, or `link` commands.

> **ATTENTION:** If the cray-cfgset-cache service is stopped, config set content in node-local memory will not get refreshed when `cray-ansible` is run. If that happens, nodes will continue to use the most recent compressed copy of the config set data created before the service was stopped.

### What Gets Cached

The cray-cfgset-cache service does not copy an entire config set to node-local memory. Instead, it uses the config set on the SMW to create these two files in the root of the config set:

- a compressed copy of the config set using SquashFS tools, (typically < 3 MB)
- a checksum of the compressed copy of the config set

The compressed copy is made available (effectively copied) to node-local RAM, and the checksum is used to know when the config set in node-local memory no longer matches the config set on the SMW. Even though Scalable Services makes the entire config set directory structure on the SMW available to the rest of the system, only the compressed copy and its associated checksum are used by nodes. They are the key to the performance, scalability, and reliability improvements provided by config set caching.

When `cray-ansible` is run on a node, the node will do the following:

1. Check to see if the cached node-local version of the compressed config set is out of date.
2. If it is stale, replace it with a newer version available on the SMW and start using that newer version.

## 2.6 About Node Groups

The Cray Node Groups service (cray_node_groups) enables administrators to define and manage logical groupings of system nodes. Nodes can be grouped arbitrarily, though typically they are grouped by software functionality or hardware characteristics, such as login, compute, service, DVS servers, and RSIP servers.

Node groups that have been defined in a config set can be referenced by name within all CLE services in that config set, thereby eliminating the need to specify groups of nodes (often the same ones) for each service individually and greatly streamlining service configuration. Node groups are used in many Cray-provided Ansible configuration playbooks and roles and can be also used in site-local Ansible plays. Node groups are similar to but more powerful than the class specialization feature of releases prior to CLE 6.0. For example, a node can be a member of more than one node group but could belong to only one class.

Sites are encouraged to define their own node groups and specify their members. Administrators can define and manage node groups using any of these methods:

- Edit and upload the node groups configuration worksheet (`cray_node_groups_worksheet.yaml`).
- Use the `cfgset` command to view and modify node groups interactively with the configurator.
- Edit the node groups configuration template (`cray_node_groups_config.yaml`) directly. Use `cfgset` to update the config set afterwards so that pre- and post-configuration scripts are run.

After using any of these methods, remember to validate the config set.

### Characteristics of Node Groups

- Node group membership is not exclusive, that is, a node may be a member of more than one node group.
- Node group membership is specified as a list of cnames. However, if the SMW is part of a node group, it is specified with the output of the `hostid` command. Also, host names can be used for eLogin nodes that are to be included in node groups.
- All compute nodes and/or all service nodes can be added as node group members by including the keywords “platform:compute” and/or “platform:service” in a node group.
- Any CLE configuration service is able to reference any defined node group by name.
- The Configuration Management Framework (CMF) exposes node group membership of the current node through the local system "facts" provided by the Ansible runtime environment. This means that each node

knows what node groups it belongs to, and that knowledge can be used in Cray and site-local Ansible playbooks.

## Default Node Groups

Default node groups are groups of nodes that

- are likely to be customized and used by many sites
- support useful default values for many of the migrated services

Several of the default node groups require customization by a site to provide the appropriate node membership information. This table lists the Cray default groups and indicates which ones require site customization.

*Table 2. cray_node_groups*

| Default Node Group | Requires Customization? | Notes |
|---|---|---|
| compute_nodes | No | Defines all compute nodes for the given partition. The list of nodes is determined at runtime. |
| service_nodes | No | Defines all service nodes for the given partition. The list of nodes is determined at runtime. |
| smw_nodes | Yes | Add the output of the `hostid` command for the SMW. For an SMW HA system, add the host ID of the second SMW also. |
| boot_nodes | Yes | Add the cname of the boot node. If there is a failover boot node, add its cname also. |
| sdb_nodes | Yes | Add the cname of the SDB node. If there is a failover SDB node, add its cname also. |
| login_nodes | Yes | Add the names of internal login nodes on the system. |
| all_nodes | Maybe | Defines all compute nodes and service nodes on the system. Add external nodes (e.g., eLogin nodes), as needed. |
| tier2_nodes | Yes | Add the cnames of nodes that will be used as tier2 servers in the cray_scalable_services configuration. |

**Why is there no "tier1_nodes" default node group?** Cray provides a default tier2_nodes node group to support defaults in the cray_simple_shares service. Cray does not provide a tier1_nodes node group because no default data in any service requires it. Because it is likely that tier1 nodes will consist of only the boot node and the SDB node, for which node groups already exist, Cray recommends using those groups to populate the cray_scalable_services tier1_groups setting rather than defining a tier1_nodes group.

**About eLogin nodes.** To add eLogin nodes to node groups, use their 'hostname' values instead of cnames, because unlike CLE nodes, eLogin nodes do not have cname identifiers. If eLogin nodes are intended to receive configuration settings associated with the all_nodes group, add them to that group, or create a new group for eLogin nodes only (elogin_nodes), and then change the appropriate settings in other configuration services to include both all_nodes and elogin_nodes.

### Additional Platform Keywords

Cray uses these two platform keywords to create default node groups that contain all compute or all service nodes.

```
platform:compute
platform:service
```

Sites that need finer-grained groupings can use these additional platform keywords to create custom node groups that contain all compute or service nodes with a particular core type.

```
platform:compute-XXNN
platform:service-XXNN
```

For XXNN, substitute a four-character processor/core designation, such as KL64 or KL68, which designate two Intel® Xeon Phi™ "Knights Landing" (KNL) processors with different core counts.

*Table 3. Cray Supported Intel Processor/Core (XXNN) Designations*

| Processor (XX) | Core (NN) | Intel Code Name |
|---|---|---|
| BW | 12, 14, 16, 18, 20, 22, 24, 28, 32, 36, 40, 44 | "Broadwell" |
| HW | 04, 06, 08, 10, 12, 14, 16, 18, 20, 24, 28, 32, 36 | "Haswell" |
| IV | 02, 04, 06, 08, 10, 12, 16, 20, 24 | "Ivy Bridge" |
| KL | 60, 64, 66, 68, 72 | "Knights Landing" |
| SB | 04, 06, 08, 12, 16 | "Sandy Bridge" |

## 2.7 About Simple Sync

The Cray Simple Sync service (cray_simple_sync) provides a simple, easy-to-use, generic mechanism for administrators to make configuration changes to their system without resorting to writing a custom Ansible play. When enabled, the service automatically copies files found in source directories in the config set on the SMW to one or more target nodes. Simple Sync is a simple tool and not intended as the sole solution for making configuration changes to the system. Writing custom Ansible plays might provide better maintainability, flexibility and scalability in the long term.

The Simple Sync service is enabled by default and has no additional configuration options. It can be enabled or disabled during the initial installation using worksheets or with the `cfgset` command at any time.

```
smw# cfgset update --service cray_simple_sync --mode interactive <config_set_name>
```

For more information, see man cfgset(8).

### How Simple Sync Works

When enabled, Simple Sync is executed on all CLE nodes at boot time and whenever the site administrator executes `/etc/init.d/cray-ansible start` on a CLE node. When Simple Sync is executed, files placed in the following directory structure are copied onto nodes that match these criteria:

`smw:/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/`

**`./common/files/`**
Matches all nodes.

**`./hardwareid/`*`<hardwareid>`*`/files/`**
Matches a specific node with that hardware ID, which is the cname of a CLE node or the output of the `hostid` command (e.g., 1eac0b0c) on other nodes. An admin must create both the `<hardwareid>` directory and the `files` directory.

**`./hostname/`*`<hostname>`*`/files/`**
Matches a node with the specified host name. An admin must create both the `<hostname>` directory and the `files` directory. Use for eLogin nodes ONLY.

**`./nodegroups/`*`<node_group_name>`*`/files/`**
Matches all nodes in the specified node group. The directories for this `nodegroups` directory are automatically stubbed out when the config set is updated after node groups are defined and configured in the cray_node_groups service.

**`./platform/[compute, service]/files/`**
Matches all compute nodes or all service nodes, depending on whether they are placed in `platform/compute/files` or `platform/service/files`. Each time the config set is updated, the HSS data store is queried to update which nodes are service and which are compute.

**`./README`**
Provides brief guidance on using Simple Sync and a list of existing node groups in the order in which files will be copied. This ordering enables an administrator to predict behavior in cases where a file may be duplicated within the Simple Sync directory structure.

Simple Sync copies content into place prior to the standard Linux startup (`systemd`) and before `cray-ansible` runs any other services. As a result, Cray services that make small changes to files will operate on the administrator-provided file. Afterwards, the file will contain both non-conflicting administrator-provided content as well as the changes made by the Cray service. Because these changes happen prior to Linux startup, the changes will be in place when the services start up.

Note that there are some config files that are entirely managed by Cray services. Where possible, such files have a comment at the top indicating that the file is completely under the management of the Cray service. Files that have been changed by Cray services can be identified by checking the change logs on the running node in `/var/opt/cray/log/ansible`. Simple Sync does not provide a mechanism to override changes made by Cray services. To override changes made by Cray services, refer to the documentation for the specific service.

The ownership and permissions of copied directories and files are preserved when they are copied to root (`/`) on the matching target nodes. An administrator can run cray_ansible multiple times, as needed, and only the files that have changed will be copied to the target nodes.

Because of the way it works, Simple Sync can be used to configure services that have configuration parameters not currently supported by configuration templates and worksheets. An administrator can create a configuration file with the necessary settings and values, place it in the Simple Sync directory structure, and it will be distributed and applied to the specified node(s).

## Characteristics of Simple Sync

| Simple Sync is: | Simple Sync is NOT: |
|---|---|
| for simple and straightforward use cases | a comprehensive system management solution |
| for copying a moderate number of moderately sized files* | intended to transfer large objects or a large volume of files |
| | an interface to configure Cray "turnkey" services such as ALPS, Node Health or Lightweight Log Manager (LLM) |

* Bear in mind that anything in the Simple Sync directory structure is part of a config set, and a SquashFS copy of the current config set is distributed to all nodes in the system. Even though it is a reduced-size config set that is distributed, it is good practice to not add very large files to a config set, hence the use of "moderate" here.

Introduced with the CLE 6.0.UP00 / SMW 8.0.UP00 release, Simple Sync has been enhanced to:

- run as early in the Ansible execution sequence as possible (it runs BEFORE other cray-ansible plays, so it can be used to make changes to files that Cray updates, like `sshd_config`)
- run during the Netroot setup sequence, so it can be used to change LNet and DVS settings, if needed
- support Node Groups for targeting which system nodes to copy files to (see *About Node Groups* on page 18)

Simple Sync does not support:

- removing files
- appending to files
- changing file ownership and permissions (the permissions of the file in the config set are mirrored on-node)
- backing up files
- overriding Cray-set values (it cannot be used to change files that Cray completely overwrites, such as `alps.conf`, or change values in files that Cray modifies such as PermitRootLogin in `/etc/ssh/sshd_config`)

## Cautions about the Use of Simple Sync

- Simple Sync copies files from the config set, which in the case of nodes without a persistent root file-system is cached in a compressed form, locally, in memory. As a result, each file stored in the config set uses some memory on the node. Therefore, using Simple Sync to copy binary files or large numbers of files is inadvisable.
- Be aware of differences in node environments when using Simple Sync. For example, systems configured with direct-attached Lustre (DAL) have nodes running CentOS instead of SLES. Administrators would have to be very careful to avoid putting an inappropriate configuration file into place when using the Simple Sync platform/service target in such a situation.
- Storage and distribution of verbatim config files through Simple Sync creates the potential for unintentional impact to the system when config files evolve due to software changes. Making minimal necessary changes through a site-local Ansible playbook provides more flexibility and minimizes the potential for unintended consequences.

## Use Cases

### Copy a non-conflicting file to all nodes

1. Place `etc/myfile` under `./common/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/myfile` on all nodes.

### Copy a non-conflicting file to a service node

1. Place `etc/servicefile` under `./platform/service/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/servicefile` on all service nodes.

### Copy a non-conflicting file to a compute node

1. Place `etc/computefile` under `./platform/compute/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/computefile` on all compute nodes.

### Copy a non-conflicting file to a specific node

1. Place `etc/mynode` under `./hardwareid/c0-0c0s0n0/files/` in the Simple Sync directory structure.
2. Simple Sync copies it to `/etc/mynode` on c0-0c0s0n0.

### Copy a non-conflicting file to a user-defined collection of nodes

1. Create a node group called "my_nodes" containing a list of nodes.
2. Update the config set.

   ```
   smw# cfgset update p0
   ```
3. Place `etc/mynodes` under `./nodegroups/my_nodes/files/` in the Simple Sync directory structure.
4. Simple Sync copies it to `/etc/mynodes` on all nodes listed in node group my_nodes.

### Copy to a node a file that has Cray-maintained content

To reduce the number of authentication tries from the default of six,

1. Place a version of `sshd_config` with the value “MaxAuthTries 3” under `./nodegroups/login_nodes/files/etc/ssh/` in the Simple Sync directory structure.
2. The booted system will contain both:
   - “MaxAuthTries 3” (from the file copied by Simple Sync)

- “PasswordAuthentication yes” (from modification of file by Cray)

### Copy to a node a file that is exclusively maintained by Cray

Files exclusively maintained by Cray such as `alps.conf` cannot be updated using Simple Sync. Please refer to the owning service (such as ALPS) for information on how to update the contents.

### Copy to a node a file that resides on a file system that will be mounted during Linux boot

No special operational changes are necessary. However, Simple Sync will put the file in place early in the boot sequence, and then it will be over-mounted by the file system. Because Simple Sync runs again later, it will copy the file into the mounted file system. Due to the ordering of operations, the file will not be present between the time the file system was mounted until the late execution of Ansible.

### On Netroot login nodes, modify an LNet modprobe parameter

1. Generate a file `zz_lnet.conf` containing `options lnet router_ping_timeout=100`.
2. Place `zz_lnet.conf` under `./nodegroups/login/files/etc/modprobe.d/` in the Simple Sync directory structure.
3. The `lnet router_ping_timeout` value will be 100.

Note that normally Simple Sync does not allow the user to override Cray values, but this procedure takes advantage of the standard Linux mechanism to override Kernel module options.

### Copy a file with an incompatible content to a node file that has Cray-maintained content

While Simple Sync allows an administrator to make changes to the same configuration files as modified by Cray, be very careful to avoid introducing syntax errors or incompatible values that may cause the system to fail to operate correctly.

## 2.8 About Boot Automation Files

**New for the CLE 6.0.UP03 release.** With this release, the default boot behavior for Cray systems without direct-attached Lustre (DAL) nodes is to boot the boot and SDB nodes first, then boot all other service nodes and all compute nodes at the same time, thereby decreasing overall boot time. Systems with DAL must boot the computes nodes after the service nodes.

- Default for systems without DAL:
  1. Boot + SDB (if SDB image small enough to PXE boot)
  2. SDB (if SDB image too large to PXE boot)
  3. Service + Compute
- Default for systems with DAL:

1. Boot + SDB (if SDB image small enough to PXE boot)
2. SDB (if SDB image too large to PXE boot)
3. Service
4. Compute

Cray provides the following boot automation files with this release.

| | |
|---|---|
| **auto.generic** | Used to boot the entire XC system. |
| **auto.xtshutdown** | Used to shut down the entire XC system. |
| **auto.bootnode** | Used to boot only the boot node(s). |
| **auto.bootnode+sdb** | Used to boot only the boot node(s) and SDB node(s). |

During a fresh install, sites typically copy `auto.generic`, rename it with the host name of the system for which it will be used (`auto.`*`hostname`*`.start`), and customize it for that site and system. Likewise, sites typically copy `auto.xtshutdown`, rename it with the host name of the system for which it will be used (`auto.`*`hostname`*`.stop`), and customize it, as needed. The host name is included because different systems may have different software installed, resulting in different boot or shutdown requirements. For example, on a system with a workload manager (WLM) installed, extra commands may be needed in the `auto.`*`hostname`*`.stop` file to cleanly stop the WLM queues on SDB or MOM nodes before shutting down the nodes.

**When is customization of an automation file needed?**

- For systems booting tmpfs images (instead of Netroot) with no SDB node failover, no changes may be necessary.
- For systems with boot or SDB node failover, instructions for adding or enabling commands are provided at the appropriate place in the fresh install and update processes.
- For systems booting Netroot images, instructions for making Netroot-related changes after the first boot with tmpfs are provided at the appropriate place in the fresh install process.
- For systems booting direct-attached Lustre (DAL) images, instructions for making DAL-related changes are provided at the appropriate place in the fresh install process.
- For systems with added content in the recipe used for SDB nodes, if the resulting custom recipe produces a boot image too large for a PXE boot, changes to the boot automation file are necessary. If based on `auto.generic`, the system boot automation file will have an option (commented out by default) to boot the boot node via PXE boot and then boot the SDB node via the HSN.
- For systems with a workload manager (WLM) installed, WLM-related changes may be needed. Specific commands to add will vary based on the WLM.

## 2.9 About the Admin Image

**About the admin image.** The admin image can be used on boot and SDB nodes ("admin" nodes) instead of the general service node image. The admin recipe produces an image root that is smaller than that produced by the general service recipe, resulting in a boot image small enough for a PXE boot. Using the admin boot image on the boot and SDB nodes may enable them to PXE boot at the same time. And because the general service image is no longer used for nodes that are intended to PXE boot, content can be added to the general service image without regard for the PXE boot size limitation.

**Should this site use the admin recipe for both boot nodes and SDB nodes?**

**boot node(s)** Yes. This will enable a PXE boot of the boot node(s).

**SDB node(s)** It depends.

- Yes, if nothing needs to be added to the recipe for the SDB node. This will enable a PXE boot of the SDB node(s).
- Maybe, if the site needs to create a custom recipe for the SDB node (e.g., to add content for a workload manager), and the admin recipe can be used as a base. Create a custom recipe for the SDB node and add the admin recipe as a sub-recipe. A PXE boot of the SDB node(s) may be possible if the resulting boot image size does not exceed the PXE boot size limit.
- No, if the admin recipe is missing content that is needed for the custom SDB recipe. Use the service recipe as the base, instead. Create a custom recipe for the SDB node and add the service recipe as a sub-recipe. A PXE boot of the SDB node(s) may be possible if the resulting boot image size does not exceed the PXE boot size limit.

For an example of creating and extending a recipe, see *Install Third-Party Software with a Custom Image Recipe* on page 346.

# 3 Install and Configure New SMW/CLE Software

Follow the procedures in this chapter to perform a fresh install of SMW 8.0.UP03.

⚠ **WARNING:** When a fresh install is performed on a system, disks are wiped clean. To prevent loss of necessary data, before beginning any installation procedures, consider what configuration files, log files, or other files should be preserved, and save them in a location unaffected by the installation.

**SMW HA only:** For a system with two SMWs that will be configured for SMW high availability (HA), the process to install the first SMW is the same as for a system with a stand-alone SMW, with a few minor differences that are noted in this guide. However, installing the second SMW uses a completely different process. Do not use this guide for the second SMW. For more information, see *XC™ Series SMW HA Installation Guide* (S-0044).

## 3.1 Prepare for an SMW/CLE Fresh Install

In preparation for a fresh install, do the following:

- If there is any data that should be saved (configuration files, log files, etc.) before the SMW disks are wiped clean by this fresh install, use the procedure in *Back Up Site Data* on page 335 now.
- Extract the configuration worksheets in preparation for entering site data. They are provided in the Crayport CLE directory for this release, `/cray/css/release/package/release/CLE/6.0.UPxx`.
  - cle-*MMDD*-worksheets.tar
  - global-*MMDD*-worksheets.tar
- Read the *SMW Release Errata* and the *SMW README* provided with the SMW release package for any additional installation-related requirements, corrections to this guide, and other relevant information about the release package.
- Read the *CLE Release Errata* and the *CLE README* provided with the CLE release package for any additional installation-related requirements, corrections to this guide, and other relevant information about the release package.
- Read the Field Notices (FN) related to kernel security fixes to identify any changes to this release package. Apply any needed changes before installing the new software.
- Collect information about the XC system: SMW, system hardware, and service node roles and networks (see *Information to Collect Before Installation* on page 28).
- Verify that the network connections are in place (see *Network Connections* on page 30).
- Find out how much SMW internal disk space is needed (see *SMW Internal Disk Requirements* on page 30).
- Know which configuration values are site-specific and which are defaults (see *Configuration Values* on page 31).
- Be familiar with the default passwords used during the installation process (see *Passwords* on page 32).

### 3.1.1 Information to Collect Before Installation

#### SMW Information

This information will be needed to update the global config set during configuration.

- Network base IP address for SMW eth0
- Netmask for SMW eth0
- Gateway IP address for SMW eth0
- List of IP addresses to use as DNS server
- List of domains to use in the DNS search path for hosts attached to SMW eth0 network
- List of NTP servers
- Host name of the SMW: both the short name and the fully qualified domain name (FQDN)
- IP address of SMW eth0

#### Hardware Information

When `xtdiscover` is used to discover XC system hardware, it will prompt for this information.

- Maximum cabinet size in the X dimension
- Maximum cabinet size in the Y dimension
- Network topology class (0 or 2 for Cray XC Series liquid-cooled systems, 0 for Cray XC Series air-cooled systems: XC30-AC, XC40-AC)
- Primary boot node (and alternate boot node if enabling boot node failover)
- Primary SDB node (and alternate SDB node if enabling SDB node failover)

#### Service Node Roles

The XC system being installed and configured must have service nodes designated to function in some or all of the following roles. A node may have more than one role (e.g., boot and tier1). The system at this site may not require all of these roles.

boot
SDB
login
tier1 (boot node and SDB node)
tier2 (see Tier2 Node FAQ)
LNet router to external Lustre server
realm-specific IP (RSIP) nodes
DataWarp-managed nodes with SSD hardware
DataWarp API gateway nodes
nodes providing a role for a workload manager (WLM)
DVS servers to an external file system
Direct-attached Lustre (DAL) MGS, MDS, or OSS nodes

compute node repurposed to be a service node

## Tier2 Node FAQ

**Q. How many tier2 nodes are needed?**

**A.** At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400.

**Q. Will adding more tier2 nodes help performance?**

**A.** Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns.

**Q. What kind of node can be used as a tier2 node?**

**A.** Use these:

- OPTIMAL: dedicated repurposed compute nodes (RCN)
- dedicated service nodes
- nodes with uniform light to moderate load
- nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability

AVOID these (will result in sub-optimal performance):

- nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors
- direct-attached Lustre (DAL) servers
- RSIP (realm-specific IP) servers
- login nodes

**Q. Can a tier2 node have more than one role?**

**A.** Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated.

**Q. Where should tier2 nodes be placed?**

**A.** Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure.

## Service Node Network Information

For each service node with a network interface, either Ethernet or InfiniBand, collect this information.

**For each network defined:**

- unique identifier for the network (management, login, lnet)
- description or notes about the network (e.g. "Network to external Lustre")
- network base IP address
- netmask
- gateway IP address

**For each network interface added to a host**

- unique identifier for each interface (primary_ethernet, eth0, eth1, eth2, eth3, ib0, ib1, etc.) on this host
- device name for the interface (eth0, ib1, etc.)
- description or notes about the nterface (e.g., "Ethernet connecting boot node to SMW")

- any host name aliases by which this node should be known
- name of the network to which this interface belongs (see list of networks defined above)
- IPv4 network address for the interface

### 3.1.2 Network Connections

The following network connections are required.

- A stand-alone SMW with a single quad-ethernet card has these private network connections:

  `eth0` - To the customer/management network
  `eth1` - To the hardware supervisory system (HSS) network
  `eth2` - Used for SMW HA (failover) heartbeat 1 network
  `eth3` - To the boot and SDB nodes (the admin network)

- An SMW configured for SMW failover (SMW HA) has a second quad-ethernet card with these connections:

  `eth4` - Used for SMW HA heartbeat 2 network
  `eth5` - Used for SMW HA distributed replicated block device (DRBD)
  `eth6` - Reserved for future use
  `eth7` - Reserved for future use

Things to note about network connections:

- Ethernet port assignments are valid only after the SMW software installation completes.
- The SMW must have a Fibre Channel or serial attached SCSI (SAS) connection to the boot RAID.
- A boot node must have a Fibre Channel or SAS connection to the boot RAID. If boot node failover is enabled or there are multiple logical CLE partitions, then each boot node should have such a connection to the boot RAID.
- A service database (SDB) node must have a Fibre Channel or SAS connection to the boot RAID. If SDB node failover is enabled or there are multiple logical CLE partitions, then each SDB node should have such a connection to the boot RAID.

  **IMPORTANT:** The SMW must be disconnected from the boot RAID before the initial installation of the SLES software.

  **IMPORTANT:** Ensure that the Fibre Channel optic cable connectors or SAS cable connectors have protective covers when disconnected from the SMW, boot node, SDB node, or boot RAID.

### 3.1.3 SMW Internal Disk Requirements

Internal SMW disks are used for the boot disk (with optional RAID1 mirroring between two boot drives) and the power management disk (PMDISK).

The PMDISK requires a minimum of 500 GB. This may be a fresh disk or a repurposed disk on an existing SMW. The PMDISK will be allocated to `/var/lib/pgsql` in an ext4 file system.

The boot disk (or pair of boot disks in RAID1 configuration) requires a minimum of 160 GB, but may be larger. If a RAID1 mirror is enabled, the drives in the RAID1 configuration must be the same size. The boot disk has 4 GB allocated to /boot in an ext3 file system, 32 GB for swap, and the rest of the disk for the root (/) file system in a btrfs file system.

*Table 4. SMW Internal Disk Requirements*

| Mount Point | FS Type | Disk | Size | Description |
|---|---|---|---|---|
| /boot | ext3 | boot | 4 GB | Booting information |
| swap | swap | boot | 32 GB | SMW swap |
| / | btrfs | boot | 120+ GB | root file system of SMW with btrfs subvolumes |
| /var/lib/pgsql | ext4 | power management | 1000+ GB | Power Management disk |

### 3.1.4 Configuration Values

The following IP addresses are set by default and are not site dependent.

*Table 5. Default IP Addresses*

| IP Address | Description |
|---|---|
| 10.1.0.1 | Primary boot RAID controller |
| 10.1.0.2 | Secondary boot RAID controller |
| 10.1.0.15 | Storage RAID controller |
| 10.1.1.1 | SMW eth1 - HSS network |
| 10.2.1.1 | (SMW HA only) SMW eth2 - SMW HA heartbeat 1 |
| 10.3.1.1 | SMW eth3 - admin network |
| 10.3.1.253 | SDB node |
| 10.3.1.254 | boot node |
| 10.4.1.1 | (SMW HA only) SMW eth4 - SMW HA heartbeat 2 |
| 10.5.1.1 | (SMW HA only) SMW eth5 - SMW HA DRBD |
| 127.0.0.1 | localhost (loopback) |

The following configuration values are site dependent. Record the actual values for the installation site in the third column.

*Table 6. Site-dependent Configuration Values*

| Description | Example Value | Actual Value |
|---|---|---|
| SMW hostname | smw | |
| Domain | cray.com | |

| Description | Example Value | Actual Value |
|---|---|---|
| Aliases | cray-smw smw1 | |
| Customer network IP address | 192.168.78.68 | |
| Customer network netmask | 255.255.255.0 | |
| Default gateway | 192.168.78.1 | |
| Domain names to search | us.cray.com mw.cray.com | |
| Nameserver IP address | 10.0.73.30 10.0.17.16 | |
| iDRAC hostname | cray-drac | |
| iDRAC IP address | 192.168.78.69 | |
| iDRAC Subnet Mask | 255.255.255.0 | |
| iDRAC Default GW | 192.168.78.1 | |
| Timezone | US/Central | |
| NTP servers | ntphost1 ntphost2 | |
| X dimension | 1-64 | |
| Y dimension | 1-32 | |
| Topology Class | 0, 2 (see note below) | |

**NOTE:** Regardless of the number of cabinets in the system, Cray XC Series air-cooled systems must be set to topology class 0. Cray XC Series liquid-cooled systems can be topology class 0 or 2.

### 3.1.5 Passwords

The following default account names and passwords are used throughout the initial software installation process. Cray recommends changing these default passwords during the installation and configuration process at appropriate times before the SMW or network CLE nodes are connected to networks that are external to the XC system.

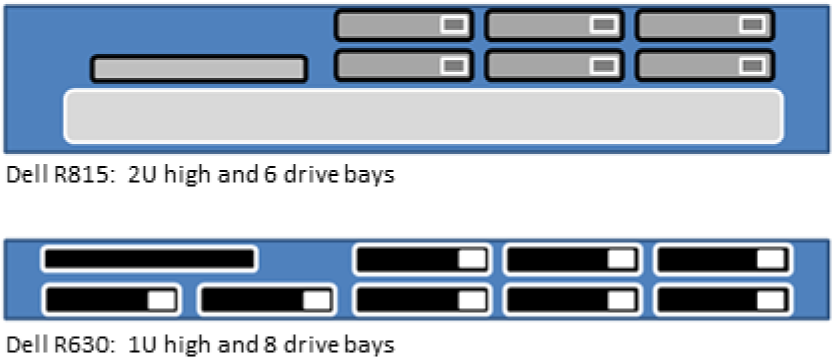
*Table 7. Default System Passwords*

| Account Name | Password |
|---|---|
| `root` | `initial0` |
| `crayadm` | `crayadm` |
| `mysql` | None; a password must be created |
| `root` (iDRAC) | `initial0` |

# 3.2 Install the Base Operating System on the SMW

The base operating system must be installed on the SMW before the Cray SMW and CLE software release packages can be installed. Cray provides two rack-mount SMW models: the Dell PowerEdge™ R815 Rack Server and the Dell PowerEdge™ R630 Rack Server. Earlier deskside SMW hardware is not supported. The figure below shows an easy way to distinguish between the two rack-mount models when viewing them from the front.

*Figure 2. Distinguishing Features of Dell R815 and R630 Servers*


Dell R815: 2U high and 6 drive bays
Dell R630: 1U high and 8 drive bays

Continue the installation process with *Prepare to Install the Base Linux Distribution* on page 33.

## 3.2.1 Prepare to Install the Base Linux Distribution

### About this task

A full initial installation begins with installing the base operating system. This procedure provides initial steps that are common to installing the base OS on both Dell R815 and R630 SMW models.

### Procedure

1. Disconnect the SMW connection to the boot RAID.

   Disconnect the data cables and place protective covers on the fibre optic cable connectors (if present).

2. Connect the SMW keyboard, monitor, and mouse.

   Connect a keyboard, monitor, and mouse to the USB and monitor connectors on the SMW, if not already connected.

   > **NOTE:** Once the iDRAC has been configured, the keyboard, monitor, and mouse can be connected to the iDRAC for remote console activities instead of being directly connected to the SMW console.

As the next step in preparing to install the base OS, do one of the following, depending on the SMW model.

- For a Dell R630 SMW, first configure the SMW RAID, then configure the BIOS and iDRAC:
  1. *Configure the Dell R630 SMW RAID Virtual Disks* on page 39
  2. *R630 SMW: Change the BIOS and iDRAC Settings* on page 43
- For a Dell R815 SMW, just configure the BIOS and iDRAC:

1. *R815 SMW: Change the BIOS and iDRAC Settings* on page 34

After the BIOS and iDRAC settings have been configured, all SMW internal disks that are not to receive the base operating system should be physically ejected from SMW internal disk drive bays.

### 3.2.1.1 R815 SMW: Change the BIOS and iDRAC Settings

## Prerequisites

This procedure assumes that the SMW is disconnected from the boot RAID and connected to a keyboard, monitor, and mouse.

## About this task

This procedure changes the system setup for a Dell R815 SMW: the network connections, remote power control, and the remote console. Depending on the server model and version of BIOS configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the Dell server used at this site. Because Cray ships systems with most of the installation and configuration completed, some of the steps may have been done already.

For a Dell R630 SMW, see *R630 SMW: Change the BIOS and iDRAC Settings* on page 43.

## Procedure

1. Remove SMW non-boot internal drives.

   Eject all the internal disk drives from the SMW except for the primary boot disk in slot 0 and the secondary boot disk in slot 1.

2. Power up the SMW. When the BIOS power-on self-test (POST) process begins, **quickly press the F2 key** after the following messages appear in the upper-right of the screen.

   ```
         F2 = System Setup
     F10 = System Services
   F11 = BIOS Boot Manager
            F12 = PXE Boot
   ```

   When the **F2** keypress is recognized, the **F2 = System Setup** line changes to **Entering System Setup**.

   After the POST process completes and all disk and network controllers have been initialized, the BIOS **System Setup** menu appears.

3. Change system time.

   The system time should be in UTC, not in the local timezone.

   a. Select **System Time** in the **System Setup** menu.

      The hours will be highlighted in blue.

   b. Set the correct time.

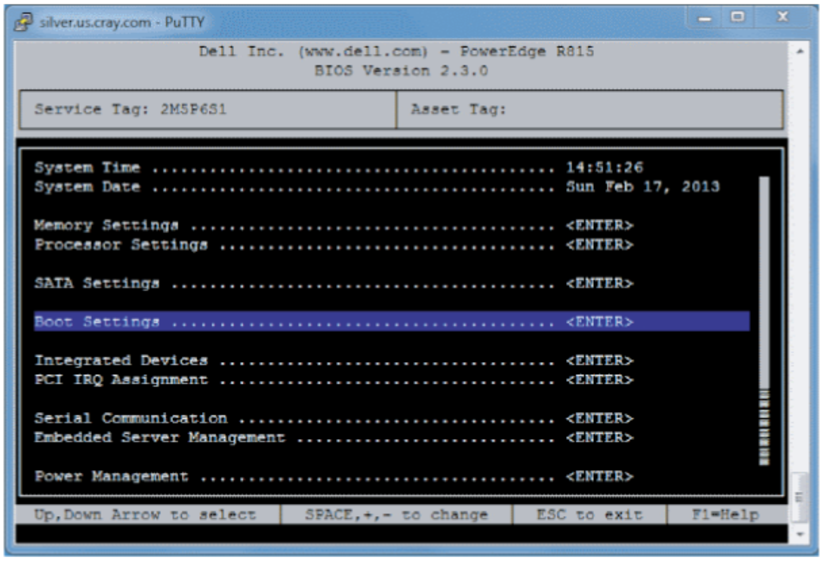
      1. Press the space key to change hours.
      2. Use the right-arrow key to select minutes, then change minutes with the space key.
      3. Use the right-arrow key to select seconds, then change seconds with the space key.

c. Press **Esc** when the correct time is set.

4. Change boot settings.

   a. Select **Boot Settings** in the **System Setup** menu, then press **Enter**.

   *Figure 3. Dell R815 SMW Boot Settings Menu*



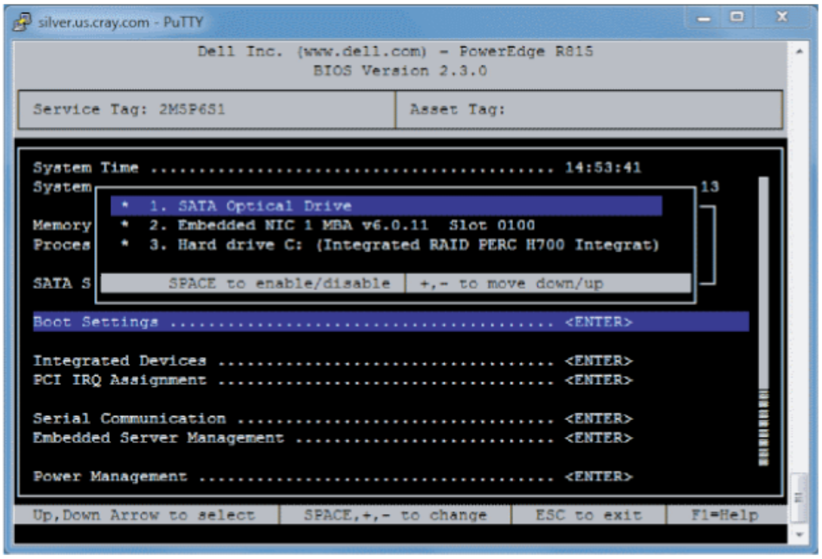
```
                  Dell Inc. (www.dell.com) - PowerEdge R815
                              BIOS Version 2.3.0

 Service Tag: 2M5P6S1                | Asset Tag:

 System Time ......................................... 14:51:26
 System Date ......................................... Sun Feb 17, 2013

 Memory Settings ..................................... <ENTER>
 Processor Settings .................................. <ENTER>

 SATA Settings ....................................... <ENTER>

 Boot Settings ....................................... <ENTER>

 Integrated Devices .................................. <ENTER>
 PCI IRQ Assignment .................................. <ENTER>

 Serial Communication ................................ <ENTER>
 Embedded Server Management .......................... <ENTER>

 Power Management .................................... <ENTER>
 Up,Down Arrow to select | SPACE,+,- to change | ESC to exit | F1=Help
```

A pop-up menu with the following list appears:

```
Boot Mode ........................................ BIOS
Boot Sequence .................................... <ENTER>
USB Flash Drive Emulation Type.................... <ENTER>
Boot Sequence Retry .............................. <Disabled>
```

   b. Select **Boot Sequence**, then press **Enter**.

   *Figure 4. Dell R815 SMW Boot Sequence Settings*



```
                  Dell Inc. (www.dell.com) - PowerEdge R815
                              BIOS Version 2.3.0

 Service Tag: 2M5P6S1                | Asset Tag:

 System Time ......................................... 14:53:41
 System                                                          13
          *  1. SATA Optical Drive
 Memory   *  2. Embedded NIC 1 MBA v6.0.11  Slot 0100
 Proces   *  3. Hard drive C: (Integrated RAID PERC H700 Integrat)

 SATA S      SPACE to enable/disable  | +,- to move down/up

 Boot Settings ....................................... <ENTER>

 Integrated Devices .................................. <ENTER>
 PCI IRQ Assignment .................................. <ENTER>

 Serial Communication ................................ <ENTER>
 Embedded Server Management .......................... <ENTER>

 Power Management .................................... <ENTER>
 Up,Down Arrow to select | SPACE,+,- to change | ESC to exit | F1=Help
```

c. Change the order of items in the **Boot Sequence** list so that the optical (DVD) drive appears first, then the hard drive. If **Embedded NIC** appears in the list, it should end up below the optical drive and hard drive in the list.

d. Disable embedded NIC.

Select `Embedded NIC` and press **Enter**, then use the space key to disable it.

e. Press **Esc** to exit the **Boot Sequence** menu.

f. Press **Esc** again to exit the **Boot Settings** menu.

**5.** Change serial communication.

a. Select **Serial Communication** in the **System Setup** menu, then press **Enter**.

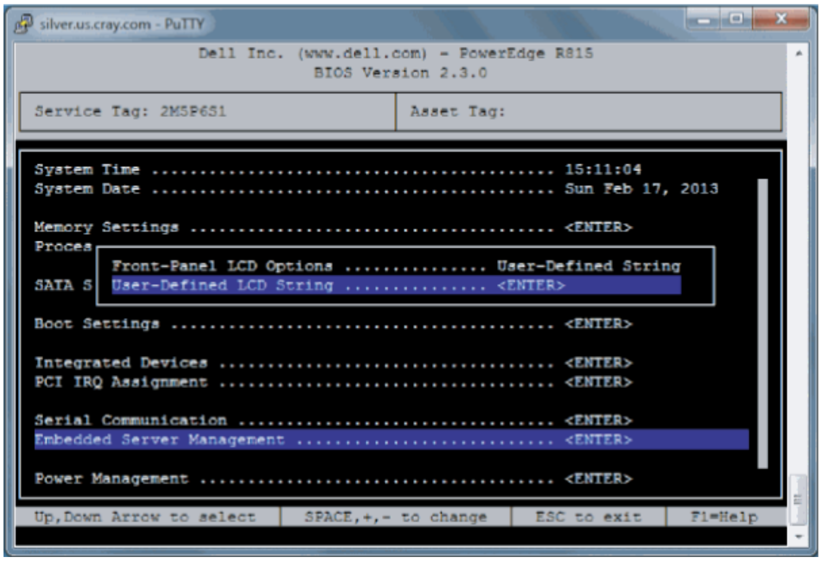
b. Confirm these settings in the **Serial Communication** menu.

- **Serial Communication** is set to **On with Console Redirection via COM2**
- **Serial Port Address** is set to **Serial Device1=COM2, Serial Device2=COM1**
- **External Serial Connector** is set to **Serial Device2**
- **Failsafe Baud Rate** is set to **115200**

c. Press **Esc** to exit the **Serial Communication** menu.

**6.** Select **Embedded Server Management** in the **System Setup** menu, then press **Enter**.

*Figure 5. Dell R815 SMW Embedded Server Management Settings*



```
              Dell Inc. (www.dell.com) - PowerEdge R815
                        BIOS Version 2.3.0

 Service Tag: 2M5P651                 Asset Tag:

 System Time ......................................... 15:11:04
 System Date ......................................... Sun Feb 17, 2013

 Memory Settings ..................................... <ENTER>
 Proces
          Front-Panel LCD Options ............... User-Defined String
 SATA S   User-Defined LCD String ............... <ENTER>

 Boot Settings ....................................... <ENTER>

 Integrated Devices .................................. <ENTER>
 PCI IRQ Assignment .................................. <ENTER>

 Serial Communication ................................ <ENTER>
 Embedded Server Management .......................... <ENTER>

 Power Management .................................... <ENTER>

 Up,Down Arrow to select | SPACE,+,- to change | ESC to exit | F1=Help
```

a. Set **Front-Panel LCD Options** to **User-Defined LCD String** in the **Embedded Server Management** menu. Use the space key to cycle through the choices, then use the down-arrow key.

b. Set **User-Defined LCD String** to the login hostname (e.g., cray-drac), then press **Enter**.

c. Press **Esc** to exit the **Embedded Server Management** menu.

**7.** Insert base operating system DVD into SMW.

Insert the base operating system DVD labeled Cray-SMWbase12-201511021655 into the DVD drive. (The DVD drive on the front of the SMW may be hidden by a removable decorative bezel.)

**8.** Save BIOS changes and exit.

a. Press **Esc** to exit the BIOS **System Setup** menu.

A menu with a list of exit options appears.

```
Save changes and exit
Discard changes and exit
Return to Setup
```

b. Ensure that **Save changes and exit** is selected, then press **Enter**.

The SMW resets automatically.

**9.** Enter BIOS boot manager.

a. When the BIOS POST process begins again, **quickly press the F11 key** within 5 seconds of when the following messages appear in the upper-right of the screen.

```
              F2 = System Setup
          F10 = System Services
        F11 = BIOS Boot Manager
                 F12 = PXE Boot
```

When the **F11** keypress is recognized, the **F11 = BIOS Boot Manager** line changes to **Entering BIOS Boot Manager**.

**10.** Change the integrated Dell Remote Access Controller (iDRAC) settings.

Watch the screen carefully as text scrolls until the **iDRAC6 Configuration Utility 1.57** line is visible. When the line **Press <Ctrl-E> for Remote Access Setup within 5 sec...** displays, press **Ctrl-E** within 5 seconds.

```
0 5 0 ATA WDC WD5000BPVT-0 1A01 465 GB
LSI Corporation MPT2 boot ROM successfully installed!
iDRAC6 Configuration Utility 1.57
Copyright 2010 Dell Inc. All Rights Reserved
iDRAC6 Firmware Revision version: 1.54.15
Primary Backplane Firmware Revision 1.07
-------------------------------
IPv6 Settings
-------------------------------
IPv6 Stack : Disabled
Address 1 : ::
Default Gateway : ::
--------------------------------
IPv4 Settings
--------------------------------
IPv4 Stack : Enabled
IP Address : 172. 31. 73.142
Subnet mask : 255.255.255. 0
Default Gateway : 172. 31. 73. 1
Press <Ctrl-E> for Remote Access Setup within 5 sec...
```

The **iDRAC6 Configuration Utility** menu appears.

**11.** Set **iDRAC LAN** to **ON**.

**12.** Configure the iDRAC LAN.

Select **LAN Parameters**, then press **Enter**.

a. Configure iDRAC6 name.

Use the arrow key to scroll down and select **iDRAC6 Name**, then press **Enter**. Enter a value for **Current DNS iDRAC6 Name** (e.g., smw-drac), then press **Enter**.

**Trouble?** If unable to set the iDRAC6 name, try this:

1. Temporarily set **Register iDRAC6 Name** to "On."
2. Set **iDRAC6 Name**.
3. Return to **Register iDRAC6 Name** and set it to "Off."

b. Configure domain name.

Use the arrow key to scroll down and select **Domain Name**, then press **Enter**. Enter a value for **Current Domain Name** (e.g., us.cray.com), then press **Enter**.

c. Configure hostname string.

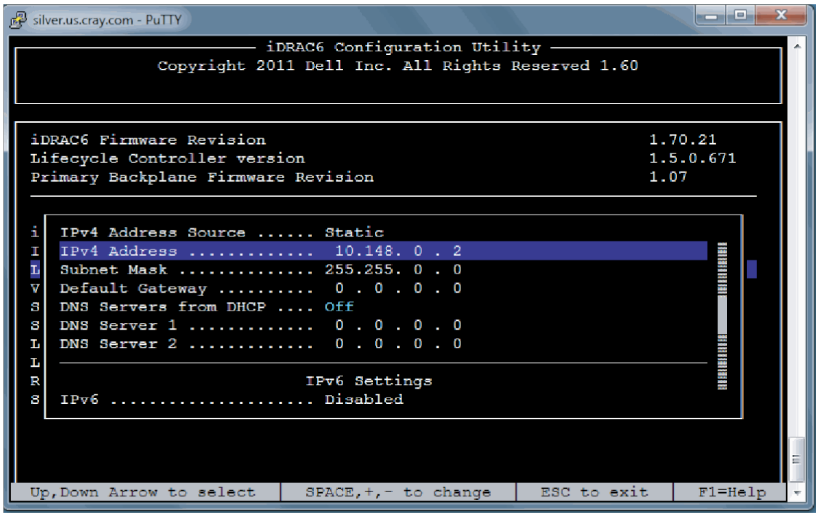
Use the arrow key to scroll down and select **Host Name String**, then press **Enter**. Enter a value for **Current Host Name String** (e.g., smw-drac), then press **Enter**.

d. Configure IPv4 settings.

Use the arrow key to scroll down into the **IPv4 Settings** group and confirm that the **IPv4 Address Source** is set to **static**. Then enter values for the following:

**IPv4 Address** (the SMW DRAC IP address)
**Subnet Mask** (the SMW iDRAC subnet mask)
**Default Gateway** (the SMW iDRAC default gateway)
**DNS Server 1** (the first site DNS server)
**DNS Server 2** (the second site DNS server)

*Figure 6. Dell R815 SMW DRAC IPv4 Parameter Settings*



```
                    iDRAC6 Configuration Utility
           Copyright 2011 Dell Inc. All Rights Reserved 1.60


 iDRAC6 Firmware Revision                                   1.70.21
 Lifecycle Controller version                               1.5.0.671
 Primary Backplane Firmware Revision                        1.07

 i  IPv4 Address Source ...... Static
 I  IPv4 Address ............   10.148. 0 . 2
 L  Subnet Mask ............. 255.255. 0 . 0
 V  Default Gateway ..........  0 . 0 . 0 . 0
 S  DNS Servers from DHCP .... Off
 S  DNS Server 1 ............   0 . 0 . 0 . 0
 L  DNS Server 2 ............   0 . 0 . 0 . 0
 L
 R                         IPv6 Settings
 S  IPv6 ..................... Disabled

 Up,Down Arrow to select | SPACE,+,- to change | ESC to exit | F1=Help
```

e. Configure IPv6 settings.

Use the arrow key to scroll down into the **IPv6 Settings** group and ensure that **IPv6** is disabled.

f. Change the IPMI settings.

Change the IPMI settings to enable the Serial Over LAN (SOL) console.

1. Use the arrow key to scroll down to the **IPMI SETTINGS** list.
2. (Stand-alone SMW only) Ensure that **Enable IPMI over LAN** is NOT selected.
3. (SMW HA only) Ensure that **Enable IPMI over LAN** is selected. This setting is used for both SMWs in an SMW HA pair.
4. Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

g. Press **Esc** to exit **LAN Parameters** and return to the **iDRAC6 Configuration Utility** menu.

**13.** Configure iDRAC virtual media.

a. Select **Domain Name**, then press **Enter**.

b. Select **Virtual Media Configuration**, then press **Enter**.

c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.

d. Press **Esc** to exit the **Virtual Media Configuration** menu.

**14.** Set the password for the iDRAC LAN root account.

Using the arrow keys, select **LAN User Configuration**, then press **Enter**. The following configuration is for both SSH and web browser access to the iDRAC.

a. Select **Account User Name** and enter the account name "root."

b. Select **Enter Password** and enter the intended password.

c. Select **Confirm Password** and enter the intended password again.

d. Press **Esc** to return to the **iDRAC6 Configuration Utility** menu.

**15.** Exit the iDRAC configuration utility.

a. Press **Esc** to exit the **iDRAC6 Configuration Utility** menu.

b. Select **Save Changes and Exit**.

The **BIOS Boot Manager** menu appears.

**16.** Choose to boot from SATA Optical Drive.

Using the arrow keys, select the **SATA Optical Drive** entry, then press **Enter**.

Dell R815 SMW system setup (changing default BIOS and iDRAC settings) is now complete. This completes the preparation for installing the base operating system. The next step in the process is *Install the SLES 12 Base Linux Distribution on the SMW* on page 53.

### 3.2.1.2 Configure the Dell R630 SMW RAID Virtual Disks

## Prerequisites

This procedure assumes that the SMW is disconnected from the boot RAID and connected to a keyboard, monitor, and mouse.

## About this task

Before installing and configuring SMW software, the base operating system needs to be installed on the SMW. And before the base operating system can be installed, the internal disk drives of the SMW must be configured as RAID virtual disks, as described in this procedure, and the default system setup for the R630 SMW node must be configured, as described in *R630 SMW: Change the BIOS and iDRAC Settings* on page 43.

A Dell R630 SMW has five physical disks. The SMW node must be reconfigured so that the internal Dell PERC RAID controller treats four of these disks as RAID 5 with a hot spare and the fifth disk as non-RAID. This procedure describes how to do that. Because Cray ships systems with most of the installation and configuration completed, some of the steps may be needed only if changes are made to the configuration.

This procedure includes detailed steps for the DELL R630 server using the PERC H330 Mini BIOS Configuration Utility `4.03-0010`. Depending on the server model and version of RAID configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the DELL PERC controller or server RAID controller software.

## Procedure

1. Connect a keyboard, monitor, and mouse to the front panel USB and monitor connectors on the SMW, if not already connected.

2. Ensure all SMW internal disk drives are inserted into the SMW drive slots.

3. Power up the SMW. As the SMW node reboots, watch for the Power Edge Expandable RAID Controller section and be ready to press **Ctrl-R** when prompted.

   Cray recommends using the RAID configuration utility (via **Ctrl-R**) to configure the RAID virtual disks instead of the **System Setup Device Settings** menu.

   **TIP:** In the RAID configuration utility:

   - Use the up-arrow or down-arrow key to highlight an item in a list.
   - Press the **Enter** key to select an item.
   - Press the **F2** key to display a menu of options for an item.
   - Use the right-arrow, left-arrow, or **Tab** key to switch between the **Yes** and **No** buttons in a confirmation window.

4. Clear existing/default disk configuration, if necessary.

   If any disk groups are currently defined:

   a. Select **Disk Group 0**, then press **F2**.

   b. Select **Delete Disk Group**, then press **Enter**.

   c. Select **Yes** in the pop-up confirmation window to confirm the changes.

5. Switch disk controller from HBA-Mode to RAID-Mode.

   Some SMW hardware might be configured for HBA-Mode. If it is, then change it to RAID-Mode using the following substeps. If it is not, then skip these substeps.

   a. Switch disk controller from HBA-Mode to RAID-Mode.

      1. Press **Ctrl-N** (multiple times) to move to the **Ctrl Mgmt** tab.

2. Press **Tab** (multiple times) to get to **Personality Mode**.
3. Press **Enter** to see choice between **RAID-Mode** and **HBA-Mode**.
4. Use the up-arrow or down-arrow key to select **RAID-Mode**, then press **Enter**.
5. Press **Tab** (multiple times) to get to **Apply**, then press **Enter**. This message appears: "The operation has been performed successfully. Reboot the system for the change to take effect."
6. Press **Enter**.

b. Exit RAID configuration utility.

1. Press **Esc** to exit the RAID configuration utility.
2. Select **OK** to confirm, then press **Enter**.

c. Reboot the SMW.

Press **Ctrl-Alt-Delete** at the prompt to reboot. The server will restart the boot process. Be prepared to press **Ctrl-R** when prompted.

d. Enter RAID configuration utility.

As the SMW node reboots, enter the RAID controller configuration utility by pressing **Ctrl-R** when prompted. This will return to the point prior to switching from HBA-Mode to RAID-Mode.

6. Configure most internal disks to appear as `/dev/sda` in a RAID-5 virtual disk.

This step configures most internal disks to appear as /dev/sda in a RAID-5 virtual disk. The R630 has four identical 500-GB drives and one 1-TB drive. The 1-TB drive will be excluded from this RAID-5 configuration. Instead, that drive will be used to hold the postgresql database with Power Management data.

a. Select **No Configuration Present**, then press the **F2** key.

b. Select **Convert to RAID capable**, then press **Enter**. The **Convert Non-RAID Disks to RAID capable** screen appears.

c. Create virtual disk /sda

7. Convert non-RAID disks to RAID-capable.

a. Press **Enter** to check the box for a physical disk, which selects it for this RAID-5 disk group. This action also advances the selection to the next disk. In this manner, select all four of the identical 500-GB drives but exclude the 1-TB drive (leave it unselected).

b. Press **Tab** to move to **OK**, then press **Enter**.

8. Verify the virtual disk changes.

To verify the virtual disk changes, compare settings with those shown in the figure.

9. Create virtual disk sda.

a. Use up-arrow to return to the **No Configuration Present!** item.

b. Press **F2** to see a pop-up menu.

c. Press **Enter** to choose **Create New VD**.

The **Convert Non - RAID Disks to RAID capable** screen appears. The only disk left on this screen should be the large (1-TB) disk which was excluded earlier. It should not be added to the RAID capable set of disks, so continue to exclude it.

d. Press **Tab** to move from the list of disks to **Cancel**, then press **Enter**.

   This cancels the conversion of non-RAID disks to RAID capable. The **Create New VD** screen appears.

**10.** Create new virtual disk (VD).

a. Press **Enter** to switch from **RAID-0** to other options.

b. Use down-arrow to select **RAID-5**, then press **Enter**.

c. Press **Tab** to move to the **Physical Disks** area.

d. Press **Enter** to select each disk except one.

   One disk should not be selected so that it can become the hot spare (configured in the next step).

e. Press **Tab** to move to **VD Name**.

f. Select name sda.

g. Press **Tab** to move to **Advanced**, then press **Enter**.

   The **Create Virtual Disk-Advanced** screen appears.

**11.** Configure one disk as the hot spare.

a. Press **Tab** multiple times to move to **Initialize**, then press **Enter** to select it.

   A pop-up window with the following text appears: "Initialization will destroy data on the virtual disk. Are you sure you want to continue?"

b. Press **Tab** or arrow keys to move to **OK**, then press **Enter** to confirm initialization.

c. Press **Tab** to move to **Configure HotSpare**, then press **Enter** to select it.

d. Press **Tab** or arrow keys to move to **OK** on the **Create Virtual Disk-Advanced** screen, then press **Enter**.

e. Press **Tab** or arrow keys to move to **OK** on the **Create New VD** screen, then press **Enter**.

   A pop-up window with the following text appears: "Virtual disk is successfully created and initialized."

f. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.

   A pop-up window with the following text appears: "Dedicated Hotspare for Disk Group 0."

g. Select the disk to be the hot spare, then press **Enter**.

h. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.

   A pop-up window with the following text appears: "Initialization complete on VD 0."

i. Press **Tab** or arrow keys to move to **OK**, then press **Enter**.

The RAID will initialize in the background.

**12.** Exit RAID configuration utility.

Exit the RAID configuration utility, reboot, and then begin installing the base operating system.

a. Press the **Esc** key to exit the RAID configuration utility.

b. Select **OK**, then press **Enter** to confirm.

**13.** Reboot the system.

A message appears that prompts to reboot.

> **ATTENTION:** Only the disk drives configured to be the RAID-5 virtual disk sda should be inserted into the SMW internal drive bays when installing SLES 12.

a. Eject the 1-TB disk (which was not added to the RAID-5 virtual disk sda) from the SMW.

   This will be re-inserted when SLES 12 installation is complete.

b. Press **Ctrl-Alt-Delete**.

The server will restart the boot process and will not interrupt RAID initialization. During the system reboot, be prepared to press **F2** when prompted, to change the system setup.

RAID configuration is now complete. The next step in preparing to install the base operating system is to configure the system setup for the R630 SMW node, as described in *R630 SMW: Change the BIOS and iDRAC Settings* on page 43.

### 3.2.1.3 R630 SMW: Change the BIOS and iDRAC Settings

#### Prerequisites

This procedure assumes that the internal disk drives of the SMW have just been configured as RAID virtual disks and the system is rebooting. If the system is not rebooting, press **Ctrl-Alt-Delete** to reboot.

#### About this task

This procedure describes how to change the system setup for the SMW: the network connections, remote power control, and the remote console. This procedure includes detailed steps for the Dell R630 server. Depending on the server model and version of BIOS configuration utility, there could be minor differences in the steps to configure the system. For more information, refer to the documentation for the Dell server used at this site. Because Cray ships systems with most of the installation and configuration completed, some of the steps may have been done already.

For a Dell R815 server, see *R815 SMW: Change the BIOS and iDRAC Settings* on page 34.

#### Procedure

Watch as the system reboots and the BIOS power-on self-test (POST) process begins. Be prepared to press **F2**, when prompted, to change the system setup.

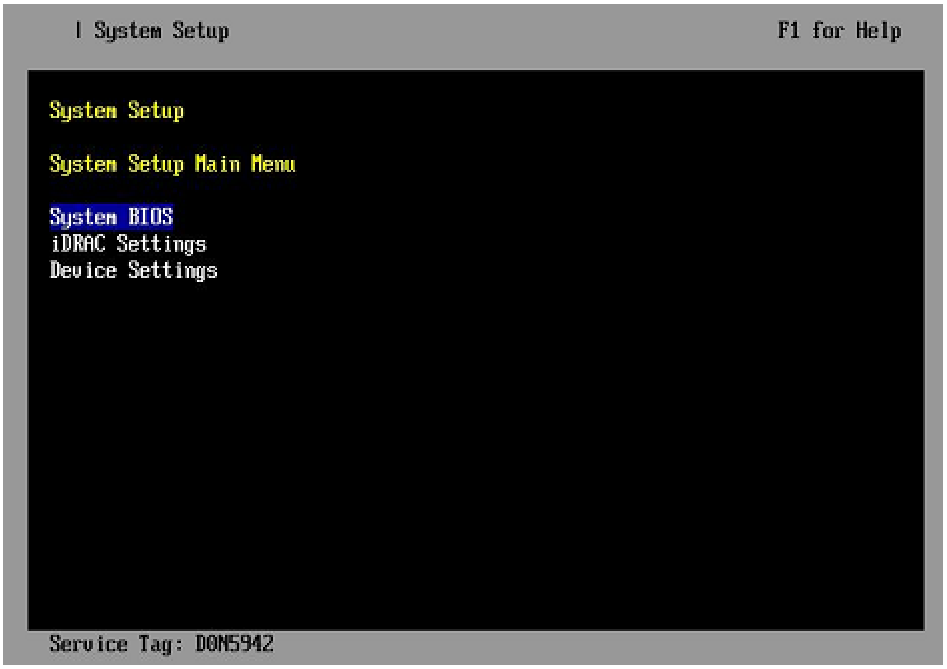
1. Press the **F2** key immediately after the following messages appear in the upper-left of the screen:

```
F2 = System Setup
F10 = System Services
F11 = BIOS Boot Manager
F12 = PXE Boot
```

   When the **F2** keypress is recognized, the **F2 = System Setup** line changes color from white-on-black to white-on-blue.

   After the POST process completes and all disk and network controllers have been initialized, the Dell **System Setup** screen appears. The following submenus are available on the **System Setup Main Menu** and will be used in subsequent steps: **System BIOS**, **iDRAC Settings**, and **Device Settings**.

*Figure 7. Dell R630 System Setup Main Menu*



**TIP:** In system setup screens,

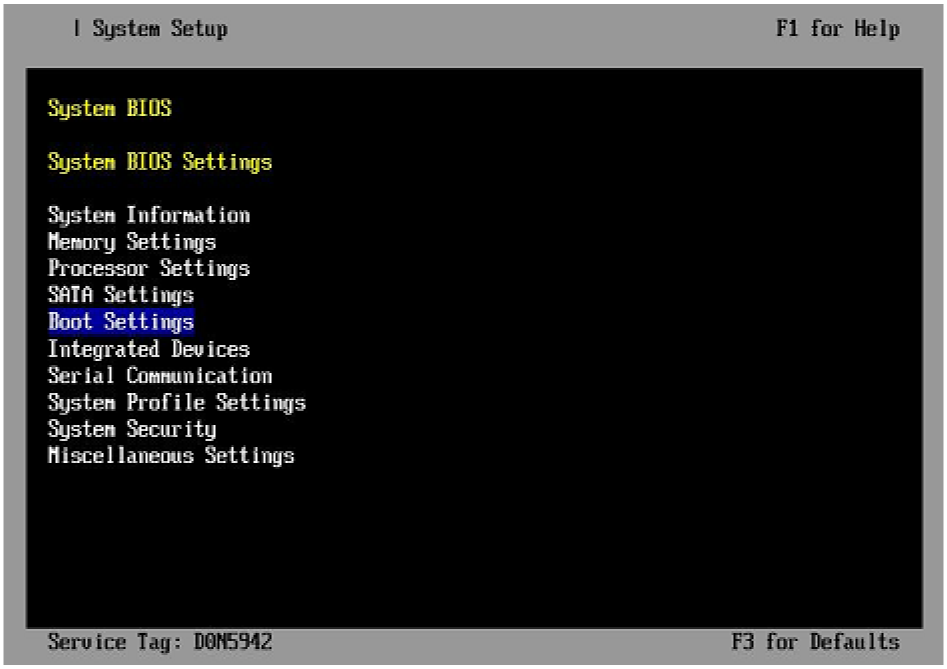
- Use the **Tab** key to move to different areas on the screen.
- Use the up-arrow and down-arrow keys to highlight or select an item in a list, then press the **Enter** key to enter or apply the item.
- Press the **Esc** key to exit a submenu and return to the previous screen.

**2.** Change the BIOS settings.

a. Select **System BIOS** on the **System Setup Main Menu**, then press **Enter**.

The **System BIOS Settings** screen appears.

*Figure 8. Dell R630 System BIOS Settings Screen*

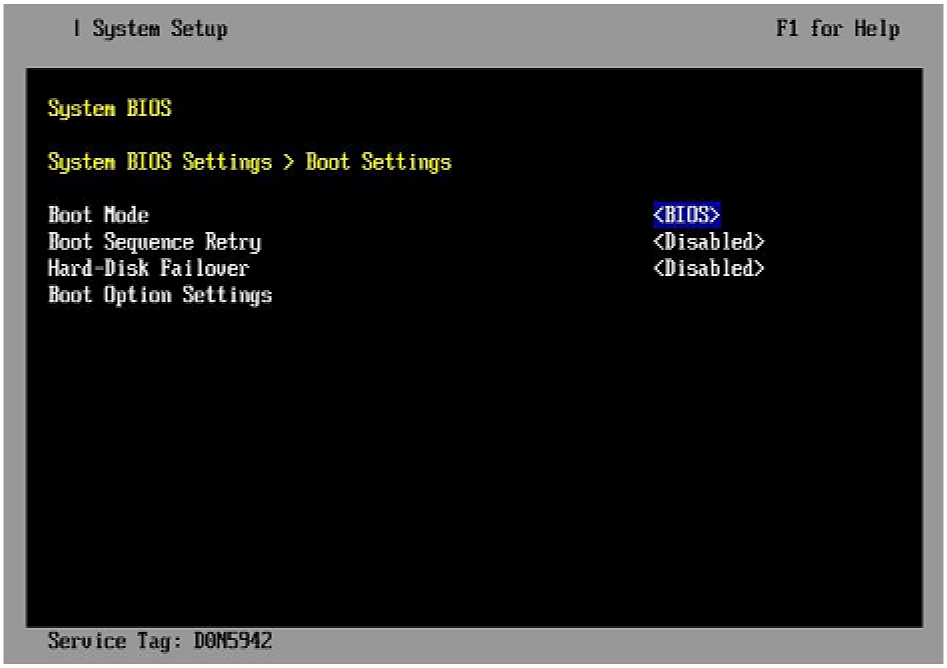


```
  | System Setup                                          F1 for Help

 System BIOS

 System BIOS Settings

 System Information
 Memory Settings
 Processor Settings
 SATA Settings
 Boot Settings
 Integrated Devices
 Serial Communication
 System Profile Settings
 System Security
 Miscellaneous Settings

 Service Tag: D0N5942                                 F3 for Defaults
```

b. Change Boot Settings.

1. Select **Boot Settings** on the **System BIOS Settings** screen, then press **Enter**. The **Boot Settings** screen appears.

   *Figure 9. Dell R630 Boot Settings Screen*

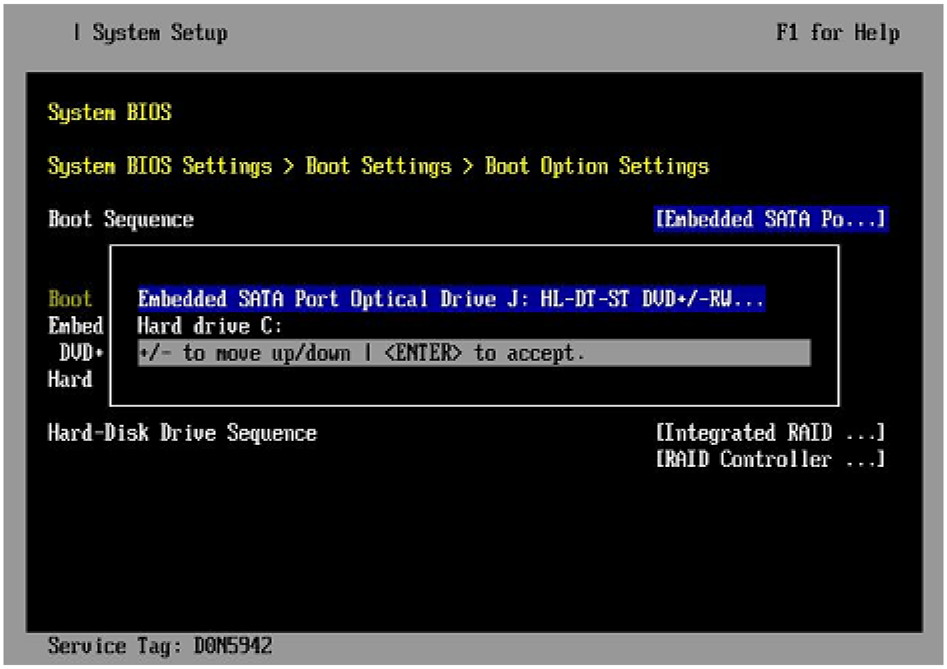
   

```
  | System Setup                                          F1 for Help

 System BIOS

 System BIOS Settings > Boot Settings

 Boot Mode                                  <BIOS>
 Boot Sequence Retry                        <Disabled>
 Hard-Disk Failover                         <Disabled>
 Boot Option Settings

 Service Tag: D0N5942
```

2. Ensure that **Boot Mode** is **BIOS** and not **UEFI**.
3. Select **Boot Option Settings**, then press **Enter**.

4. Select **Boot Sequence** on the **Boot Option Settings** screen, then press **Enter** to view a pop-up window with the boot sequence.

   *Figure 10. Dell R630 BIOS Boot Sequence*

```
  | System Setup                                        F1 for Help

System BIOS

System BIOS Settings > Boot Settings > Boot Option Settings

Boot Sequence                                  [Embedded SATA Po...]

Boot     Embedded SATA Port Optical Drive J: HL-DT-ST DVD+/-RW...
Embed    Hard drive C:
 DVD+    +/- to move up/down | <ENTER> to accept.
Hard

Hard-Disk Drive Sequence                       [Integrated RAID ...]
                                               [RAID Controller ...]

Service Tag: D0N5942
```

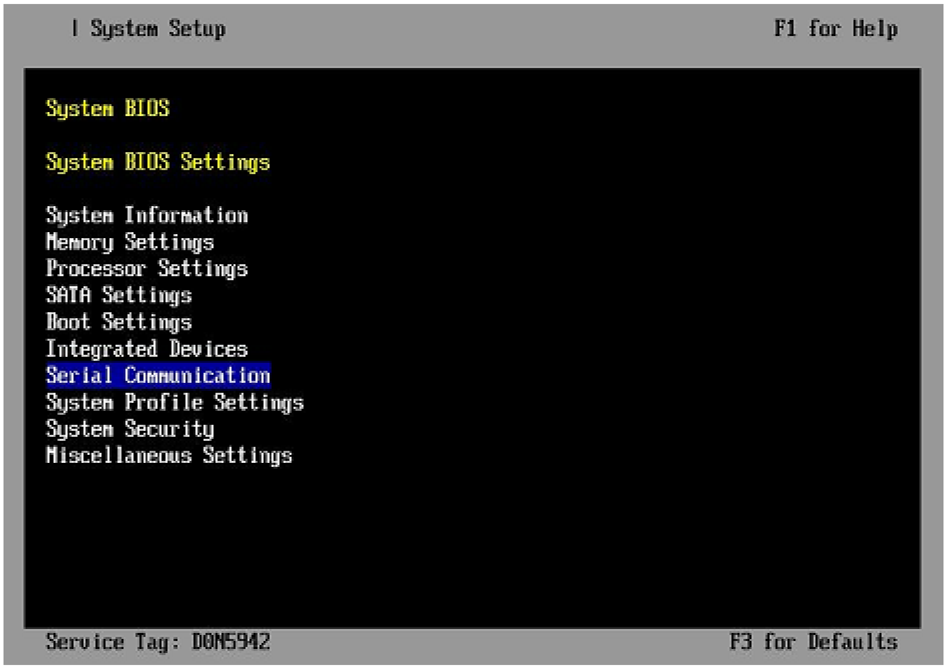
5. Change the boot order in the pop-up window so that the optical drive appears first, then the hard drive. If **Integrated NIC** appears in the list, it should end up below the optical drive and hard drive in the list.

   > **TIP:** Use the up-arrow or down-arrow key to highlight or select an item, then use the **+** and **-** keys to move the item up or down.

6. Select **OK**, then press **Enter** to accept the change.
7. Click the box next to **Hard drive C:** under the **Boot Option/Enable/Disable** section to enable it. Do the same for the optical drive, if necessary.
8. Select **integrated NIC**, then press **Enter** to disable it.
9. Press **Esc** to exit **Boot Option Settings**.
10. Press **Esc** to exit **Boot Settings** and return to the **System BIOS Settings** screen.

c. Change Serial Communication Settings.

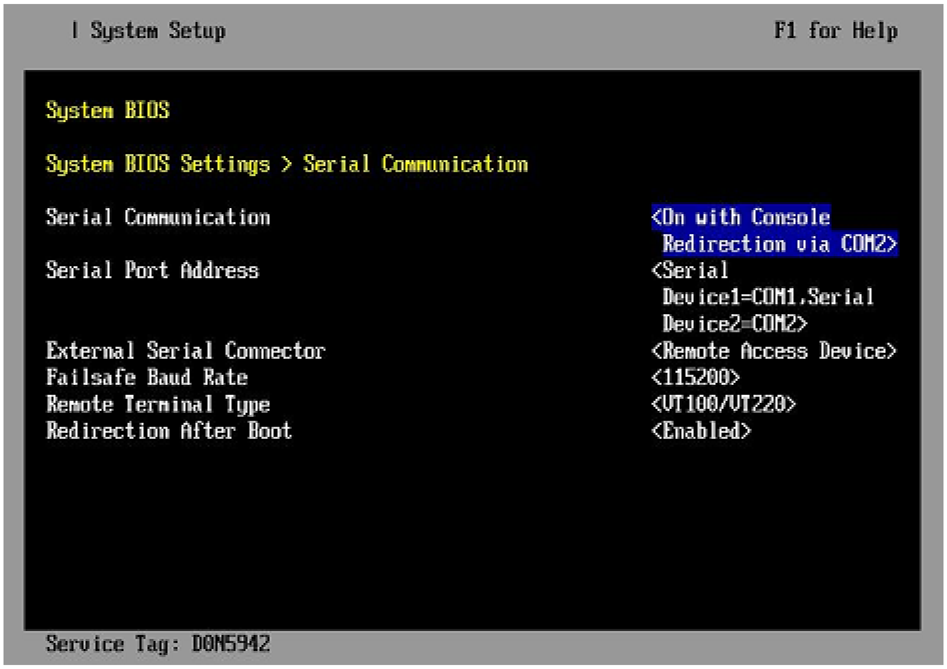
*Figure 11. Dell R630 System BIOS Settings: Serial Communication*



```
 | System Setup                                    F1 for Help

 System BIOS

 System BIOS Settings

 System Information
 Memory Settings
 Processor Settings
 SATA Settings
 Boot Settings
 Integrated Devices
 Serial Communication
 System Profile Settings
 System Security
 Miscellaneous Settings

 Service Tag: D0N5942                              F3 for Defaults
```

1. Select **Serial Communication** on the **System BIOS Settings** screen. The **Serial Communication** screen appears.

   *Figure 12. Dell R630 Serial Communication Screen*

```
 | System Setup                                    F1 for Help

 System BIOS

 System BIOS Settings > Serial Communication

 Serial Communication                 <On with Console
                                       Redirection via COM2>
 Serial Port Address                  <Serial
                                       Device1=COM1,Serial
                                       Device2=COM2>
 External Serial Connector            <Remote Access Device>
 Failsafe Baud Rate                   <115200>
 Remote Terminal Type                 <VT100/VT220>
 Redirection After Boot               <Enabled>

 Service Tag: D0N5942
```

2. Select **Serial Communication** on the **Serial Communication** screen, then press **Enter**. A pop-up window displays the available options.
3. Select **On with Console Redirection via COM2** in the pop-up window, then press **Enter** to accept the change.

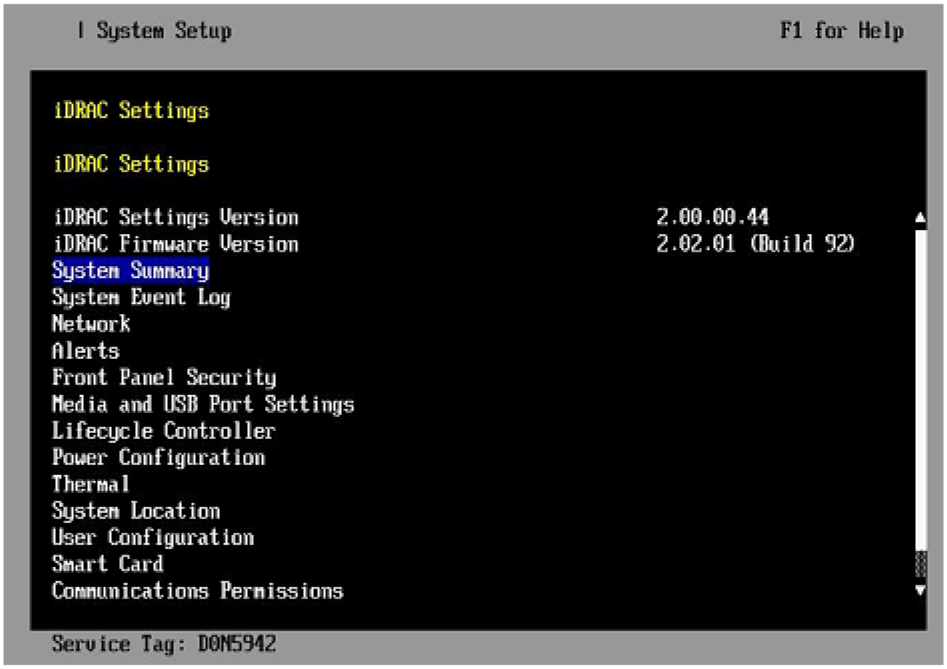
   4. Select **Serial Port Address**, then select **Serial Device1=COM1, Serial Device2=COM2**, then press **Enter**.
   5. Select **External Serial Connector**, then press **Enter**. A pop-up window displays the available options.
   6. Select **Remote Access Device** in the pop-up window, then press **Enter** to return to the previous screen.
   7. Select **Failsafe Baud Rate**, then press **Enter**. A pop-up window displays the available options.
   8. Select **115200** in the pop-up window, then press **Enter** to return to the previous screen.
   9. Press the **Esc** key to exit the **Serial Communication** screen.
   10. Press **Esc** to exit the **System BIOS Settings** screen. A "Settings have changed" message appears.
   11. Select **Yes** to save changes. A "Settings saved successfully" message appears.
   12. Select **Ok**.

3. Change the iDRAC (Integrated Dell Remote Access Controller) settings.

   Select **iDRAC Settings** on the **System Setup Main Menu**, then press **Enter**.

   The **iDRAC Settings** screen appears.

   *Figure 13. Dell R630 iDRAC Settings Screen*



```
| System Setup                                        F1 for Help

iDRAC Settings

iDRAC Settings

iDRAC Settings Version                        2.00.00.44
iDRAC Firmware Version                        2.02.01 (Build 92)
System Summary
System Event Log
Network
Alerts
Front Panel Security
Media and USB Port Settings
Lifecycle Controller
Power Configuration
Thermal
System Location
User Configuration
Smart Card
Communications Permissions

Service Tag: D0N5942
```

4. Change the iDRAC network.

   a. Select **Network** to display a long list of network settings.

   b. Change the DNS DRAC name.

      Use the arrow key to scroll down to **DNS DRAC Name**, then enter an iDRAC hostname that is similar to the SMW node hostname (e.g., cray-drac).

   c. Change the static DNS domain name.

Use the arrow key to scroll down to **Static DNS Domain Name**, then enter the DNS domain name and press **Enter**.

d. Change the IPv4 settings.

   Use the arrow key to scroll down to the **IPV4 SETTINGS** list.

   **1.** Ensure that IPv4 is enabled.

      **a.** If necessary, select **Enable IPV4**, then press **Enter**.

      **b.** Select **<Enabled>** in the pop-up window, then press **Enter** to return to the previous screen.

   **2.** Ensure that DHCP is disabled.

      **a.** If necessary, select **Enable DHCP**, then press **Enter**.

      **b.** Select **<Disabled>** in the pop-up window, then press **Enter** to return to the previous screen.

   **3.** Change the IP address.

      **a.** Select **Static IP Address**.

      **b.** Enter the IP address of the iDRAC interface (`ipmi0`) for the SMW, then press **Enter**.

   **4.** Change the gateway.

      **a.** Select **Static Gateway**.

      **b.** Enter the appropriate value for the gateway of the network to which the iDRAC is connected, then press **Enter**.

   **5.** Change the subnet mask.

      **a.** Select **Subnet Mask**.

      **b.** Enter the subnet mask for the network to which the iDRAC is connected (such as `255.255.255.0`), then press **Enter**.

   **6.** Change the DNS server settings.

      **a.** Select **Static Preferred DNS Server**, enter the IP address of the primary DNS server, then press **Enter**.

      **b.** Select **Alternate DNS Server**, enter the IP address of the alternate DNS server, then press **Enter**.

e. Change the IPMI settings.

   Change the IPMI settings to enable the Serial Over LAN (SOL) console.

   **1.** Use the arrow key to scroll down to the **IPMI SETTINGS** list.

   **2.** Ensure that **Enable IPMI over LAN** is selected.

      **TIP:** Use the left-arrow or right-arrow to switch between two settings.

   **3.** Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

f. Exit Network screen.

   Press the **Esc** key to exit the **Network** screen and return to the **iDRAC Settings** screen.

**5.** Change hostname in iDRAC LCD display.

Change front panel security to show the hostname in LCD display.

a. Use the arrow key to scroll down and highlight **Front Panel Security** on the **iDRAC Settings** screen, then press **Enter**.

b. Select **Set LCD message**, then press **Enter**.

c. Select **User-Defined String**, then press **Enter**.

d. Select **User-Defined String**, then enter the SMW hostname and press **Enter**.

e. Press the **Esc** key to exit the **Front Panel Security** screen.

**6.** (Optional) Change the iDRAC **System Location** fields.

Change the **System Location** configuration on the **iDRAC Settings** screen to set any of these fields: **Data Center Name**, **Aisle Name**, **Rack Name**, and **Rack Slot**.

**7.** Configure iDRAC virtual media.

a. Select **Domain Name**, then press **Enter**.

b. Select **Virtual Media Configuration**, then press **Enter**.

c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.

d. Press **Esc** to exit the **Virtual Media Configuration** menu.

**8.** Set the password for the iDRAC root account.

a. Use the arrow key to highlight **User Configuration** on the **iDRAC Settings** screen, then press **Enter**.

b. Confirm that User Name is root. Select **User Name**, then enter the "root" user name.

c. Select **Change Password**, then enter a new password.

d. Reenter the new password in the next pop-up window to confirm it (the default password is "calvin").

e. Press the **Esc** key to exit the **User Configuration** screen.

**9.** Exit iDRAC settings.

a. Press the **Esc** key to exit the **iDRAC Settings** screen.

A "Settings have changed" message appears.

b. Select **Yes**, then press **Enter** to save the changes.

A "Success" message appears.

c. Select **Ok**, then press **Enter**.

The main screen (**System Setup Main Menu**) appears.

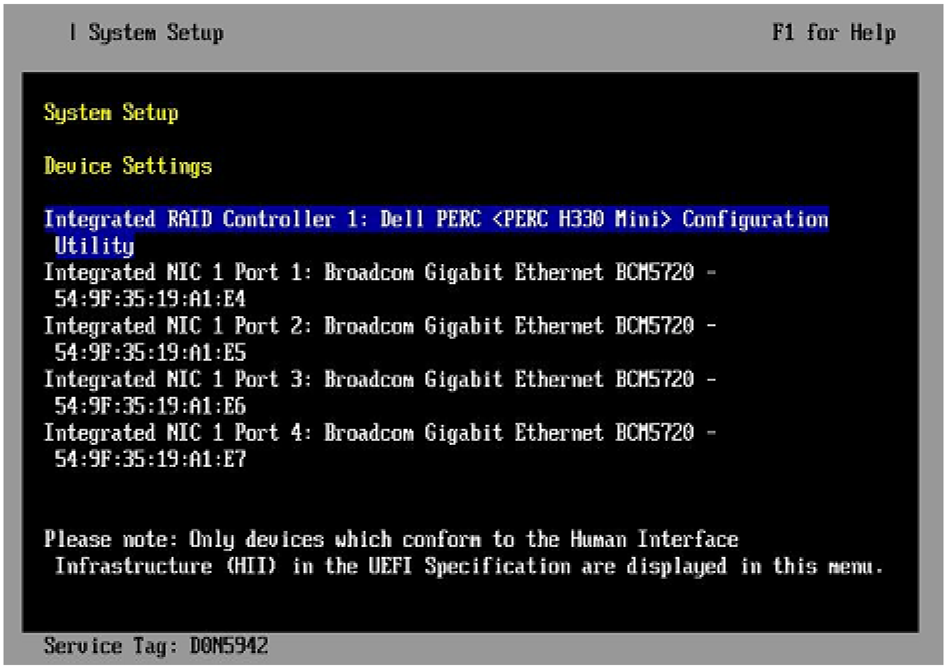
**10.** Change device settings.

These steps disable an integrated NIC device by changing the setting for the integrated NIC on a port from **PXE** to **None**.

a. Change Integrated NIC 1 Port 1

**1.** Select **Device Settings** on the **System Setup Main Menu**, then press **Enter**. The **Device Settings** screen appears.

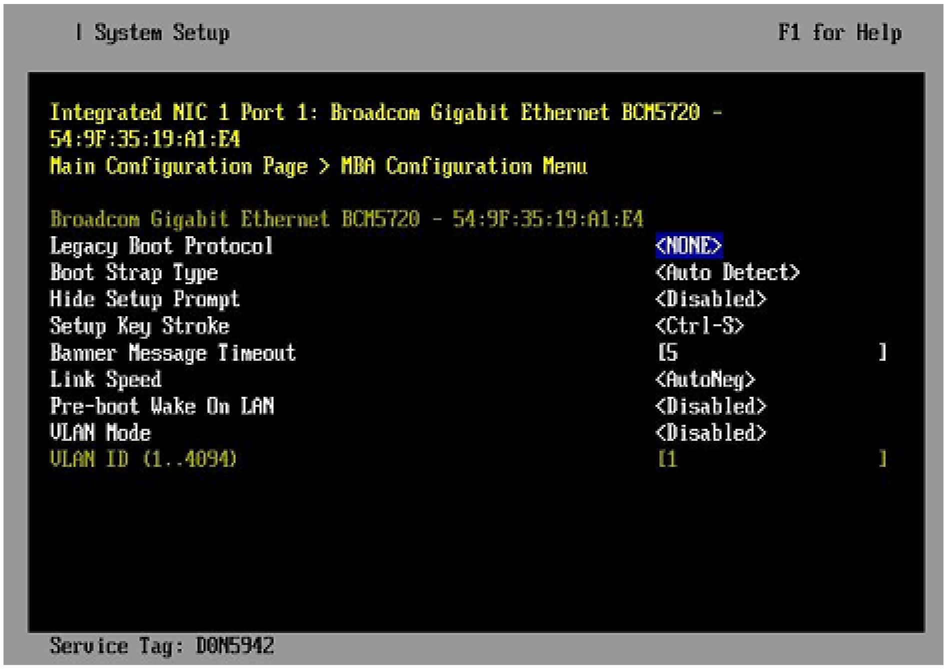
*Figure 14. Dell R630 Device Settings Screen*



```
 | System Setup                                                F1 for Help

System Setup

Device Settings

Integrated RAID Controller 1: Dell PERC <PERC H330 Mini> Configuration
 Utility
Integrated NIC 1 Port 1: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E4
Integrated NIC 1 Port 2: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E5
Integrated NIC 1 Port 3: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E6
Integrated NIC 1 Port 4: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E7


Please note: Only devices which conform to the Human Interface
 Infrastructure (HII) in the UEFI Specification are displayed in this menu.

Service Tag: D0N5942
```

2. Select **Integrated NIC 1 Port 1: ...** on the **Device Settings** screen, then press **Enter**.
3. Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

*Figure 15. Dell R630 BIOS MBA Configuration Settings*



```
 | System Setup                                                F1 for Help

Integrated NIC 1 Port 1: Broadcom Gigabit Ethernet BCM5720 -
54:9F:35:19:A1:E4
Main Configuration Page > MBA Configuration Menu

Broadcom Gigabit Ethernet BCM5720 - 54:9F:35:19:A1:E4
Legacy Boot Protocol                                   <NONE>
Boot Strap Type                                        <Auto Detect>
Hide Setup Prompt                                      <Disabled>
Setup Key Stroke                                       <Ctrl-S>
Banner Message Timeout                                 [5           ]
Link Speed                                             <AutoNeg>
Pre-boot Wake On LAN                                   <Disabled>
VLAN Mode                                              <Disabled>
VLAN ID (1..4094)                                      [1           ]

Service Tag: D0N5942
```

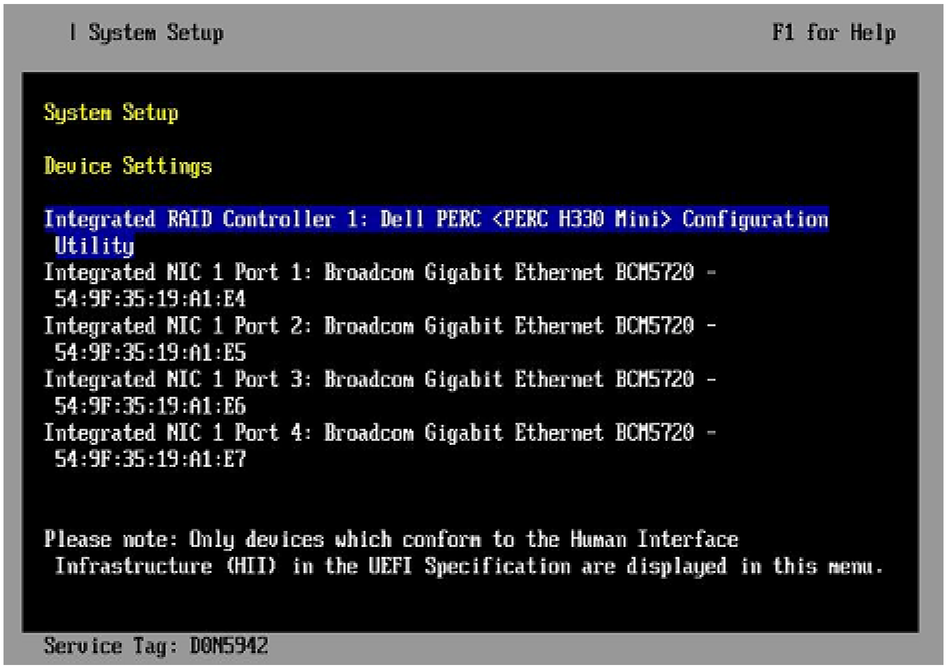
4. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, use the right-arrow or left-arrow key to highlight **None**, then press **Enter**.
5. Press the **Esc** key to exit the **MBA Configuration Menu** screen.

6. Press **Esc** to exit the **Main Configuration Page** screen. A "Warning Saving Changes" message appears.
7. Select **Yes**, then press **Enter** to save the changes. A "Success" message appears.
8. Select **OK**, then press **Enter**. The **Device Settings** screen appears.
9. Press **Esc** to exit the **Device Settings** screen. A "Settings have changed" message appears.
10. Select **Yes**, then press **Enter** to save the changes. A "Settings saved successfully" message appears.
11. Select **OK**, then press **Enter**. The main screen (**System Setup Main Menu**) appears.

b. Change Integrated NIC 1 Port 2

1. Select **Device Settings** on the **System Setup Main Menu**, then press **Enter**. The **Device Settings** screen appears.

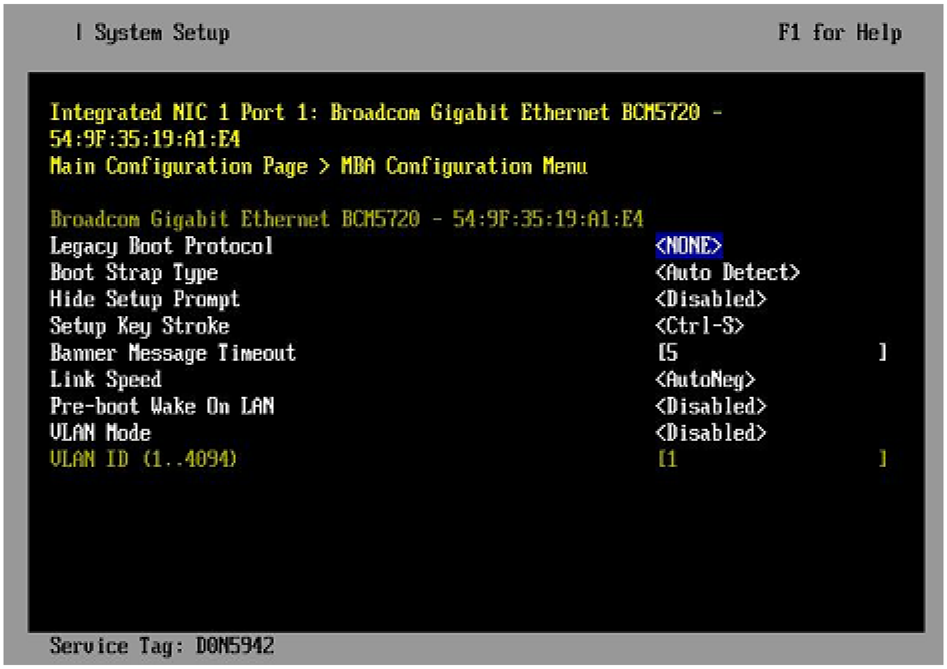
*Figure 16. Dell R630 Device Settings Screen*



```
| System Setup                                              F1 for Help

System Setup

Device Settings

Integrated RAID Controller 1: Dell PERC <PERC H330 Mini> Configuration
 Utility
Integrated NIC 1 Port 1: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E4
Integrated NIC 1 Port 2: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E5
Integrated NIC 1 Port 3: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E6
Integrated NIC 1 Port 4: Broadcom Gigabit Ethernet BCM5720 -
 54:9F:35:19:A1:E7


Please note: Only devices which conform to the Human Interface
 Infrastructure (HII) in the UEFI Specification are displayed in this menu.

Service Tag: D0N5942
```

2. Select **Integrated NIC 1 Port 2: ...** on the **Device Settings** screen, then press **Enter**.
3. Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

*Figure 17. Dell R630 BIOS MBA Configuration Settings*



```
| System Setup                                                F1 for Help

  Integrated NIC 1 Port 1: Broadcom Gigabit Ethernet BCM5720 -
  54:9F:35:19:A1:E4
  Main Configuration Page > MBA Configuration Menu

  Broadcom Gigabit Ethernet BCM5720 - 54:9F:35:19:A1:E4
  Legacy Boot Protocol                           <NONE>
  Boot Strap Type                                <Auto Detect>
  Hide Setup Prompt                              <Disabled>
  Setup Key Stroke                               <Ctrl-S>
  Banner Message Timeout                         [5          ]
  Link Speed                                     <AutoNeg>
  Pre-boot Wake On LAN                           <Disabled>
  VLAN Mode                                      <Disabled>
  VLAN ID (1..4094)                              [1          ]

  Service Tag: D0N5942
```

4. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, use the right-arrow or left-arrow key to highlight **None**, then press **Enter**.
5. Press the **Esc** key to exit the **MBA Configuration Menu** screen.
6. Press **Esc** to exit the **Main Configuration Page** screen. A "Warning Saving Changes" message appears.
7. Select **Yes**, then press **Enter** to save the changes. A "Success" message appears.
8. Select **OK**, then press **Enter**. The **Device Settings** screen appears.
9. Press **Esc** to exit the **Device Settings** screen. A "Settings have changed" message appears.
10. Select **Yes**, then press **Enter** to save the changes. A "Settings saved successfully" message appears.
11. Select **OK**, then press **Enter**. The main screen (**System Setup Main Menu**) appears.

SMW system setup (changing default BIOS and iDRAC settings) is now complete. This completes the preparation for installing the base operating system. The next step in the process is *Install the SLES 12 Base Linux Distribution on the SMW* on page 53.

## 3.2.2 Install the SLES 12 Base Linux Distribution on the SMW

### Prerequisites

This procedure assumes the following:

- The BIOS and iDRAC settings have just been changed on the SMW and it is restarting the boot process.
- All SMW internal disks that are not to receive the operating system are physically ejected from SMW internal disk drive bays.

- All connections to the boot RAID are unplugged so that no disk devices from the boot RAID will inadvertently lose existing data or receive the operating system.

## About this task

This procedure describes the base operating system installation process. It provides detailed instructions for installing SLES 12 on the SMW (both Dell R815 and R630 models); configuring the SMW; and performing final steps: reconnect cables, reinsert drives, and reboot the SMW. To install the base operating system, use the DVD labeled Cray-SMWbase12-201511021655, which contains SUSE Linux Enterprise Server version 12 (SLES 12).

## Procedure

### SLES 12 SOFTWARE PACKAGE INSTALLATION

1. Select one of the **Cray SMW Initial Install** options.

   Within 10 to 15 seconds after this **SUSE Linux Enterprise Server** boot menu displays, use the arrow key to scroll down and select one of the install options, then press **Enter**.

   ```
   - Boot from Hard Disk
   - Cray SMW Initial Install with software RAID1
   - Cray SMW Initial Install without software RAID
   - Rescue System
   - Check Installation Media
   - Firmware Test
   - Memory Test
   ```

   Select the option that is best for the SMW model:

   **For a Dell R815 SMW** Select **Cray SMW Initial Install with software RAID1**, a mirrored boot disk option, which creates a software RAID1 mirror on the first two drives. This option is best for a Dell R815 because the R815 should use two disk drives to become the software RAID1 mirror.

   **For a Dell R630 SMW** Select **Cray SMW Initial Install without software RAID**, a non-mirrored boot disk option, for servers with a single disk or virtual disk. This option is best for a Dell R630 because the R630 should have the internal RAID controller configured to present four disk drives as a virtual disk.

   ⚠ **WARNING:** If the selection is not made in time, the system will boot from the default selection, which is **Boot from Hard Disk**. If that happens, shut down the SMW, then start the power-up sequence again.

   Note: The upper left corner of the installation screen has a date/time stamp for when the bootable SLES 12 DVD was created.

   As the base installation progresses, the following phases appear on the screen:

   ```
   Starting ... Loading Linux kernel
   Initializing
   Preparing System for Automated Installation
   Initializing the Installation Environment
   System Probing
   Installation Settings
   ```

2. Review installation settings while the installation pauses on the **Installation Settings** screen.

**3.** Confirm the language for the SMW.

English (US) is the primary language by default. To change the primary language:

a. Select the **Language** heading in the **Installation Settings** screen.

The **Languages** window opens.

b. Select a language (or multiple languages) from the drop-down menu, then select **Accept** at the bottom of the window.

**4.** Begin automated install.

a. On the **Installation Settings** screen, select **Install**.

The **Confirm Installation** pop-up window appears.

b. Select **Install**.

The installation of software packages runs for approximately 20 minutes. The process automatically reboots the SMW from the hard disk, and the installation process continues with system configuration.

### SYSTEM CONFIGURATION

**5.** Log in to SMW as root.

When the login screen is displayed with the crayadm account as the account which will be logged in:

a. Select **Not listed?**, then enter `root` for the username.

b. Either press **Enter** or select **Sign In**.

c. Enter the password for root.

**6.** Change default passwords on the SMW by executing the following commands.

The SMW contains its own `/etc/passwd` file that is separate from the password file for the rest of the CLE system.

```
smw# passwd root
```

```
smw# passwd crayadm
```

```
smw# passwd mysql
```

**7.** Change the SMW local time zone, if needed.

The default time zone on the SMW is **America/Chicago**. To change it:

a. Execute this command:

```
smw# yast2 timezone
```

`yast2` opens a new window for changing the time zone, then a pop-up window appears with this message: "file `/etc/ntp.conf` has been changed manually. YaST might lose some of the changes."

b. Select the **Do not show this message anymore** checkbox, then select **Continue**.

c. Choose the time zone either by selecting a region on the map or by using the drop-down menus for **Region** and **Time Zone**.

d. Select **Other Settings** if the time is incorrect, then select the **Manually** radio button and enter **Current Time** and **Current Date**. Select **Accept** when done.

e. Select **OK** when done with time zone settings.

**8.** Configure the SMW firewall.

The SUSE firewall settings may need to be adjusted to match site firewall policy and to customize for site IP addresses. These steps enable and configure the firewall.

**TIP:** It is not necessary to shut down the system before performing this task.

a. Save the SUSE firewall configuration.

Before modifying the SUSE firewall settings, make a copy of the configuration file.

```
smw# cp -p /etc/sysconfig/SuSEfirewall2 /etc/sysconfig/SuSEfirewall2.orig
```

b. Check current firewall settings.

Check current firewall settings and change to support any site requirements. During the process of configuring Cray SMW and CLE software, some of the firewall settings may be adjusted. SSH access is one of the protocols permitted through the firewall from the external network to the SMW.

```
smw# iptables -L
smw# vi /etc/sysconfig/SuSEfirewall2
```

c. Start the firewall immediately.

Invoke the modified configuration.

```
smw# systemctl start SuSEfirewall2_init.service
smw# systemctl start SuSEfirewall2.service
```

d. Ensure that the firewall will start at next boot.

Execute the following commands to start the firewall at boot time.

```
smw# systemctl enable SuSEfirewall2_init.service
smw# systemctl enable SuSEfirewall2.service
```

e. Verify firewall changes.

Verify the changes to the `iptables`.

```
smw# iptables -nvL
```

**9.** Configure LAN on the SMW.

Set network configuration for eth0 and the hostname for the SMW.

a. Execute this command:

```
smw# yast2 lan
```

The **Network Settings** screen appears with the **Overview** tab highlighted.

b. Select the **eth0** line on the **Overview** tab, then select **Edit**.

The **Network Card Setup** screen appears with the **Address** tab highlighted.

c. Select **Statically Assigned IP address** on the **Address** tab and enter values for IP address, subnet mask, and hostname (including the domain name). Then select **Next**.

d. Select the **Hostname/DNS** tab on the **Network Settings** screen.

  **1.** For the **Hostname and Domain Name** area, enter Hostname and Domain Name.

  **2.** For the **Name Servers and Domain Search List**, enter Name Server 1, Name Server 2, Name Server 3, and Domain Search.

e. Select the **Routing** tab on the **Network Settings** screen, then enter the Default IPv4 Gateway (for the network connected to eth0) and set Device to eth0 using the dropdown menu.

f. Click **OK** after all of the **Network Settings** have been prepared.

**FINAL STEPS**

**10.** Reconnect boot RAID disk cables.

Remove the protective covers from the Fibre Channel or SAS cable connectors, clean the ends of the cable connectors, and reconnect the data cables that connect the SMW to the boot RAID.

**11.** Reinsert SMW non-boot internal drives.

Reinsert all of the SMW internal disk drives that were removed earlier.

> **TIP:** It is not necessary to turn off the power for the SMW before inserting these drives—the operating system can be in a booted state.

**12.** Eject the Cray-SMWbase12 DVD.

If the base operating system DVD (Cray-SMWbase12-201511021655) is still in the DVD drive, eject it.

```
smw# eject
```

**13.** Reboot the SMW.

Reboot the SMW to allow the SMW to discover the drives properly.

```
smw# reboot
```

If the SMW was configured with RAID1, then it may still be synchronizing the data between the two disks in the RAID1 mirror. The resync can take about 30 minutes when SLES 12 is freshly installed. If the SMW is rebooted at this point in the process, that resync will be interrupted. However, that is not a problem because as soon as the SMW is up again, the resync process will continue.

(R815 SMW only) To check the status of any RAID1 resync activities on an R815 SMW, look at /proc/mdstat.

In this example, the resync of md127 finishes in 24.3 minutes.

```
smw# cat /proc/mdstat
Personalities : [raid1]
md125 : active raid1 sdc2[1] sda2[0]
      33559424 blocks super 1.0 [2/2] [UU]
      bitmap: 0/1 pages [0KB], 65536KB chunk

md126 : active raid1 sda1[0] sdc1[1]
      4200384 blocks super 1.0 [2/2] [UU]
      bitmap: 0/1 pages [0KB], 65536KB chunk
```

```
md127 : active raid1 sda3[0] sdc3[1]
      206437248 blocks super 1.0 [2/2] [UU]
      [======>..............]  resync = 33.7% (69700352/206437248) 
finish=24.3min speed=93748K/sec
      bitmap: 2/2 pages [8KB], 65536KB chunk

unused devices: <none>
```

- For a stand-alone SMW or the first SMW in an SMW HA system, the next step in the installation and configuration of the SMW base operating system is *Configure Boot RAID Devices* on page 58.
- (SMW HA only) For the second SMW in an SMW HA system, there is no need to configure the boot RAID because it is shared with the first SMW and has already been configured. The next step in the process is *Make a Snapshot Manually* on page 77.

## 3.2.3 Configure Boot RAID Devices

In typical system installations, the RAID provides the storage for file systems used by the SMW, boot node, and SDB node. These file systems are prepared from LVM volumes in LVM volume groups using the physical volumes that are created on the RAID LUNs (logical unit numbers) or volumes. RAID units also provide user and scratch space and can be configured to support a variety of file systems. For more information about configuring RAID, see *XC™ Series Lustre® Administration Guide* (S-2648), which is provided with the CLE release package.

### Prerequisites and Assumptions for Configuring the Boot RAID

Sites that require a long distance between the SMW, XC, and the boot RAID will use Fibre Channel (FC) components, while sites that have the SMW, XC, and boot RAID in the same area (within 10 meters) will typically use SAS as the interface for the boot RAID.

- The SMW has an Ethernet connection to the Hardware Supervisory System (HSS) network.
- The SMW has a Fibre Channel (FC) or Serial Attached SCSI (SAS) connection to the boot RAID or to an FC or SAS switch.
- The boot nodes have an FC or SAS connection to the boot RAID or to an FC or SAS switch.
- The SDB nodes have an FC or SAS connection to the boot RAID or to an FC or SAS switch.

### Boot RAID Configuration Procedures

Cray provides support for system boot RAID from NetApp, Inc.

> **NOTE:** Cray ships systems with much of this software installed and configured. Performing all of the steps in these boot RAID procedures may not be necessary unless the configuration needs to be changed.

**1.** Configure the boot RAID for a NetApp, Inc. storage system using the following procedures (reference *Recommended Boot RAID LUN Values* on page 59 as needed). The first one installs the SANtricity Storage Manager Utility, which is used to perform the other procedures.

   **a.** *Install SANtricity Storage Manager for NetApp, Inc. Devices* on page 60

   **b.** *Set Up Boot RAID Space for Direct-attached Lustre* on page 61

   **c.** *Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices* on page 62

2. Zone the SAS (Serial Attached SCSI) or FC (Fibre Channel) switch. For FC storage, there will be an FC Switch to be configured. For SAS storage, there will be a SAS Switch to be configured. Use the applicable procedure(s):
   - *Zone the QLogic FC Switch* on page 64 and (recommended) *Create a Backup of the QLogic Switch Configuration* on page 66
   - *Zone the Brocade FC Switch* on page 67
   - *Zone the LSI SAS Switch* on page 73
3. *Reboot the SMW and Verify LUNs are Recognized* on page 76

### 3.2.3.1 Recommended Boot RAID LUN Values

The recommended boot RAID LUN configuration is shown in these tables for different sizes of boot RAID: 4.5 TB, 9.0 TB, and 1.5 TB.

## Boot RAID with 4.5 TB Available, Non-partitioned System

For a boot RAID with 4.5 TB available, use these values for a non-partitioned system. This is the default configuration installed in the factory.

| LUN | Label | Size | Segment Size |
|---|---|---|---|
| 0 | smw0 | 3000 GB | 256 KB |
| 1 | boot0 | 1000 GB | 256 KB |
| 2 | sdb0 | 200 GB | 256 KB |

## Boot RAID with 4.5 TB Available, Multiple Partitions

For a boot RAID with 4.5 TB available, use these values for a system with multiple CLE partitions.

- There must be one SMW LUN for the entire system with a size of at least 1000GB.
- There must be one boot LUN for each partition with a size of at least 500GB.
- There must be one SDB LUN for each partition with a size of at least 100GB.

This table shows example values for three CLE partitions.

| LUN | Label | Size | Segment Size |
|---|---|---|---|
| 0 | smw1 | 2500 GB | 256 KB |
| 1 | boot1 | 500 | 256 KB |
| 2 | sdb1 | 100 GB | 256 KB |
| 3 | boot2 | 500 GB | 256 KB |
| 4 | sdb2 | 100 GB | 256 KB |
| 5 | boot3 | 500 GB | 256 KB |
| 6 | sdb3 | 100 GB | 256 KB |

## Boot RAID with 9.0 TB Available, Non-partitioned System

For a boot RAID with 9.0 TB available, use these values for a non-partitioned system. Values for boot1 and sdb1 LUNs are shown also, because they can be added to volume groups for the boot node volume group and SDB node volume group, if needed. If added, they should be the same size as the boot0 and sdb0.

| LUN | Label | Size | Segment Size |
|---|---|---|---|
| 0 | smw0 | 4000 GB | 256 KB |
| 1 | boot0 | 1000 | 256 KB |
| 2 | sdb0 | 200 GB | 256 KB |
| 3 | boot1 | 1000 GB | 256 KB |
| 4 | sdb1 | 200 GB | 256 KB |

## Boot RAID with 1.5 TB Available, Non-partitioned System

For a boot RAID with only 1.5 TB available, use these values for a non-partitioned system.

| LUN | Label | Size | Segment Size |
|---|---|---|---|
| 0 | smw0 | 1000 GB | 256 KB |
| 1 | boot0 | 400 GB | 256 KB |
| 2 | sdb0 | 100 GB | 256 KB |

### 3.2.3.2 Install SANtricity Storage Manager for NetApp, Inc. Devices

## About this task

The SANtricity Storage Manager software is generally preinstalled and the SANtricity media is shipped with the system. If the SANtricity software is installed, then the `SMclient` executable will be found in `/opt/SMgr/client`. If this Cray system does not have the software installed on the SMW, install it using this procedure.

## Procedure

1. Prepare X Windows for NetApp SANtricity Storage Manager.

   The NetApp installation software will launch an X Windows application, so an X Windows server must be ready. There are many ways to prepare this: logging into SMW console as root, logging into SMW console as crayadm and then becoming root, or logging into SMW from a remote workstation with X Windows port forwarding enabled via ssh.

   - If already logged in to the SMW as crayadm, su to root and enable X Windows port forwarding:

     ```
     crayadm@smw> su -
     smw# ssh -X localhost
     ```

   - If not already logged on to the SMW, log in and enable X Windows port forwarding like this:

     ```
     user@host> ssh -X root@smw
     ```

**2.** Copy NetApp SANtricity Storage Manager installer to SMW.

- If installing from the SANtricity Storage Manager CD, insert it into the SMW CD drive and mount the CD.

  ```
  smw# mount /dev/cdrom /media/cdrom
  smw# mkdir -p /tmp/netapp
  smw# cp -p /media/cdrom/SMIA-LINUXX64-11.25.0A00.0016.bin /tmp/netapp
  smw# umount /media/cdrom
  smw# eject
  ```

- If installing from the `SMIA-LINUXX64-11.25.0A00.0016.bin` file, copy that file to `/tmp/netapp`.

  ```
  smw# mkdir -p /tmp/netapp
  smw# cp ./SMIA-LINUXX64-11.25.0A00.0016.bin /tmp/netapp
  ```

**3.** Run the NetAPP SANtricity Storage Manager installer.

```
smw# /tmp/netapp/SMIA-LINUXX64-11.25.0A00.0016.bin
```

The **SANtricity Storage Manager Introduction** window displays. The following substeps provide guidance through the installation, but the exact steps may differ for newer versions of the NetApp software.

a. Select **Next** in the **SANtricity Storage Manager Introduction** window.

   The **License Agreement** window displays.

b. Select **I accept the terms of the License Agreement**, then select **Next**.

   The **Select Installation Type** window displays.

c. Select **Typical (Full Installation)**, then select **Next**.

   The **Multi-Pathing Driver Warning** window displays.

d. Select **OK**.

   The **Pre-Installation Summary** window displays.

e. Select **Install**.

   The **Installing SANtricity** window displays and shows the installation progress. When the installation completes, an **Install Complete** window displays.

f. Select **Done** to acknowledge and finish.

   The SANtricity client, `SMclient`, is installed in `/opt/SMgr/client`.

**4.** Enable crayadm to run `SMclient`.

To be able to execute `SMclient` from the `crayadm` account, change the ownership and permissions for the executable files. If this step is skipped, only the `root` account will be able to run `SMclient`.

```
smw# chown crayadm /opt/SMgr
smw# chmod 775 /opt/SMgr
smw# chmod 755 /opt/SMgr/client/SMcli /opt/SMgr/client/SMclient
smw# chown -R crayadm:crayadm /var/opt/SM
smw# chmod -R ug+w /var/opt/SM
```

### 3.2.3.3 Set Up Boot RAID Space for Direct-attached Lustre

If the system will use direct-attached Lustre (DAL), create LUNs for DAL nodes to use for the MGT, MDT, and OST disk devices. This must be done before installing CLE and DAL.

If creating LUNs on the NetApp 2700 boot RAID device or external Netapp block storage device, use the SANtricity data management software installed on the SMW to create the DAL LUNs.

### 3.2.3.4 Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices

## Prerequisites

This procedure assumes the following:

- the SANtricity Storage Manager has been installed
- the user is logged on to the SMW as `crayadm`

## About this task

This procedure creates the 8+2 Volume Group and 2 Global Hot Spares for a 4.5 TB Volume Group (the amount of storage for this installation may be different). A standard new boot RAID has 2 hot spares; the number of hot spares depends on the number of available drives left over after configuration of the 8+2 RAID6.

## Procedure

1. Start the SANtricity Storage Manager.

   ```
   crayadm@smw> /usr/bin/SMclient
   ```

   The SANtricity Storage Manager window appears.

2. Select a method for adding a volume group.

   If the **Select Addition Method** window appears, choose one of the following options. Otherwise, continue with the next step.

   - **Automatic**. Select this option if a serial connection was not used to assign IP addresses to the storage array controllers. The SANtricity software automatically detects the available controllers, in-band, using the Fibre Channel link.
   - **Manual**. Select this option if IP addresses have already been assigned to the storage array controllers.

3. Create a volume group.

   The following substeps apply only if the **Select Addition Method** window did not display or if the **Manual** option was selected.

   a. Double-click the name for the storage array to be configured.

      The **Array Management** window displays.

   b. Select the **Logical/Physical** tab.

   c. Right-click **Unconfigured Capacity** and select **Create Volume**.

The **Create Volume** wizard displays.

d. Select **Next** on the **Introduction (Create Volume)** window.

e. Select the **Manual** option on the **Specify Volume Group (Create Volume)** window.

f. Select tray 99, slots 1-10, then select **Add**.

g. Verify that the RAID level is set to 6.

h. Select **Calculate Capacity**.

i. Select **Next** on the **Specify Volume Group (Create Volume)** window.

The **Array Management** window should still be displayed after performing this step.

### Create and Configure Volumes

After creating the first volume group, create the first volume when prompted. Configure the boot RAID with enough LUNs to support the various system management file systems (Cray recommends a minimum of three LUNs).

**4.** Create a volume.

a. Enter a new volume capacity. Specify units as GB or MB.

b. Enter a name for the volume.

c. Select the **Customize Settings** option.

d. Select **Next** in the **Specify Capacity/Name (Create Volume)** window.

e. Verify the settings on the **Customize Advanced Volume Parameters (Create Volume)** window.

These settings are used for the all of the LUNs.

- For **Volume I/O characteristics type**, verify that **File System** is selected.
- For **Preferred Controller Ownership**, verify that **Slot A** is selected. This places the LUN on the A Controller.

f. Select **Next** in the **Customize Advanced Volume Parameters (Create Volume)** window.

g. Select the **Default** mapping option in the **Specify Volume to LUN Mapping** window.

h. For **Host** type, select **Linux** from the drop-down menu.

i. Select **Finish** in the **Specify Volume to LUN Mapping** window.

j. Select **Yes** when prompted to create more LUNs in the **Creation Successful (Create Volume)** window, unless this is the last volume to be created. If this is the last volume, select **No** and continue with the next step (skipping the rest of these substeps).

k. Verify that **Free Capacity** is selected on **Volume Group 1 (RAID 5)** in the **Allocate Capacity (Create Volume)** window.

l. Select **Next** in the **Allocate Capacity (Create Volume)** window.

m. Repeat step 4 to create all of the volumes (applicable to this system) described in *Recommended Boot RAID LUN Values* on page 59

**5.** Indicate that volume creation and configuration is complete.

Select **OK** in the **Completed (Create Volume)** window.

6. Create a hot spare.

   The hot spare provides a ready backup if any of the drives in the volume group fail.

   a. Right-click on the last drive in the slot 12 icon on the right portion of the window and select **Hot Spare Coverage**.
   b. Select the **Manually Assign Individual Drives** option.
   c. Select **OK**.
   d. Select **Close**.

7. Exit the tool.

8. (optional) Configure remote logging of the boot RAID messages.

   The NetApp, Inc. storage system uses SNMP to provide boot RAID messages. Cray does not provide a procedure for this; see *NetApp, Inc. Storage System documentation* for information about how to configure remote logging.

The next step in the process is to zone the switches. Go to one of the following, depending on the switch type:

- *Zone the QLogic FC Switch* on page 64.
- *Zone the Brocade FC Switch* on page 67
- *Zone the LSI SAS Switch* on page 73

### 3.2.3.5 Zone the QLogic FC Switch

## Prerequisites

This procedure assumes the following:

- The QLogic SANBox™ FC (Fibre Channel) switch has been configured and is on the HSS network.
- The disk device has four host ports connected to ports 0-3 for the QLogic SANbox switch, and the following connections have been made:
  - The SMW must be connected to port 10 on the SANBox.
  - The boot node must be connected to port 4 on the SANBox.
  - The SDB node must be connected to port 5 on the SANBox.

## About this task

This procedure describes how to use the *QuickTools* utility to zone the LUNs on the QLogic SANBox FC switch. QuickTools is an application that is embedded in the QLogic switch and is accessible from a workstation browser with a compatible Java™ plug-in. It requires a Java browser plugin, version 1.4.2 or later.

Zoning is implemented by creating a *zone set*, adding one or more zones to the zone set, and selecting the ports to use in the zone.

> **NOTE:** If a LUN is to be shared between failover host pairs, each host must be given access to the LUN. The SMW host port should be given access to all LUNs.

## Procedure

1. Start a web browser.

2. Enter the IP address of the switch.

   If the configuration has a single switch, the IP address is 10.1.0.250. The IP address of each RAID controller is preconfigured by Cray and is listed on a sticker on the back of the RAID controller.

3. Enter the login name and password when the **Add a New Fabric** window pops up and prompts for them.

   The default administrative login name is `admin`, and the default password is `password`.

   The QuickTools utility displays in the browser.

4. Select **Add Fabric**.

   If a dialog box appears stating that the request failed to connect over a secured connection, select **Yes** and continue.

5. Double-click the switch icon when the switch is located and displayed in the window.

   Information about the switch displays in the right panel.

6. Select the **Configured Zonesets** tab at the bottom of the panel.

7. Select **Zoning** and then **Edit Zoning** from the toolbar menu.

   The **Edit Zoning** window displays.

8. Create a zone set.

   a. Select the **Zone Set** button.

      The **Create a Zone Set** window displays.

   b. Create a new zone set.

      In this example, assume that the zone set is named `XT0`.

9. Create a zone.

   a. Right-click the `XT0` zone and select **Create a Zone**.

   b. Create a new zone named BOOT.

10. Define the ports in the zone.

    On the right panel, select the button in front of BOOT to open a view of the domain members. Ports 0, 4, 5, and 10 are added to the BOOT zone. Define the ports in the zone to ensure that the discovery of LUNs is consistent among the SMW, the boot node, and the SDB node.

    a. Using the mouse, left-click Port # 0 and drag it to the BOOT zone.

    b. Using the mouse, left-click Port # 4 and drag it to the BOOT zone. This port is for the boot node.

    c. Using the mouse, left-click Port # 5 and drag it to the BOOT zone. This port is for the SDB node.

    d. Using the mouse, left-click Port # 10 and drag it to the BOOT zone. This port is for the SMW.

e. Select **Apply**. The error-checking window displays.

f. Select **Perform Error Check** when prompted.

g. Select **Save Zoning** after confirming that no errors were found.

**11.** Select **Yes** when prompted to activate a zone set, then select the appropriate `XT0` zone set.

At this point, Cray recommends creating a backup of the switch configuration (*Create a Backup of the QLogic Switch Configuration* on page 66) before closing and exiting the application. Otherwise, proceed to the next step in the boot RAID configuration process: *Reboot the SMW and Verify LUNs are Recognized* on page 76.

### 3.2.3.6 Create a Backup of the QLogic Switch Configuration

## About this task

Use the QuickTools utility to create a backup of the QLogic switch configuration. To use QuickTools, a Java browser plugin, version 1.4.2 or later is required.

To start a web browser and open the QuickTools utility, complete steps 1 through 4. If the QuickTools utility is already open, skip to step 5.

## Procedure

**1.** Start a web browser.

**2.** Enter the IP address of the switch.

The IP address of each RAID controller is preconfigured by Cray and is listed on a sticker on the back of the RAID controller. If the configuration has a single switch, the IP address is 10.1.0.250.

**3.** Enter the login name and password when the **Add a New Fabric** window pops up and prompts for them.

The default administrative login name is `admin`, and the default password is `password`.

The QuickTools utility displays in the browser.

**4.** Select **Add Fabric**.

If a dialog box appears stating that the request failed to connect over a secured connection, select **Yes** and continue.

The QuickTools utility opens.

**5.** Complete the configuration backup from within the QuickTools utility:

a. Select **Switch** and then **Archive** from the top bar.

A **Save** window pops up with blanks for **Save in:** and **File Name:**.

b. Enter the directory (for example, `crayadm`) and a file name (for example, `sanbox_archive`) for saving the QLogic switch configuration.

c. Select the **Save** button.

**6.** Close and exit the application.

The QLogic FC switch is now zoned and backed up. Proceed to the next step in the boot RAID configuration process: *Reboot the SMW and Verify LUNs are Recognized* on page 76.

### 3.2.3.7 Zone the Brocade FC Switch

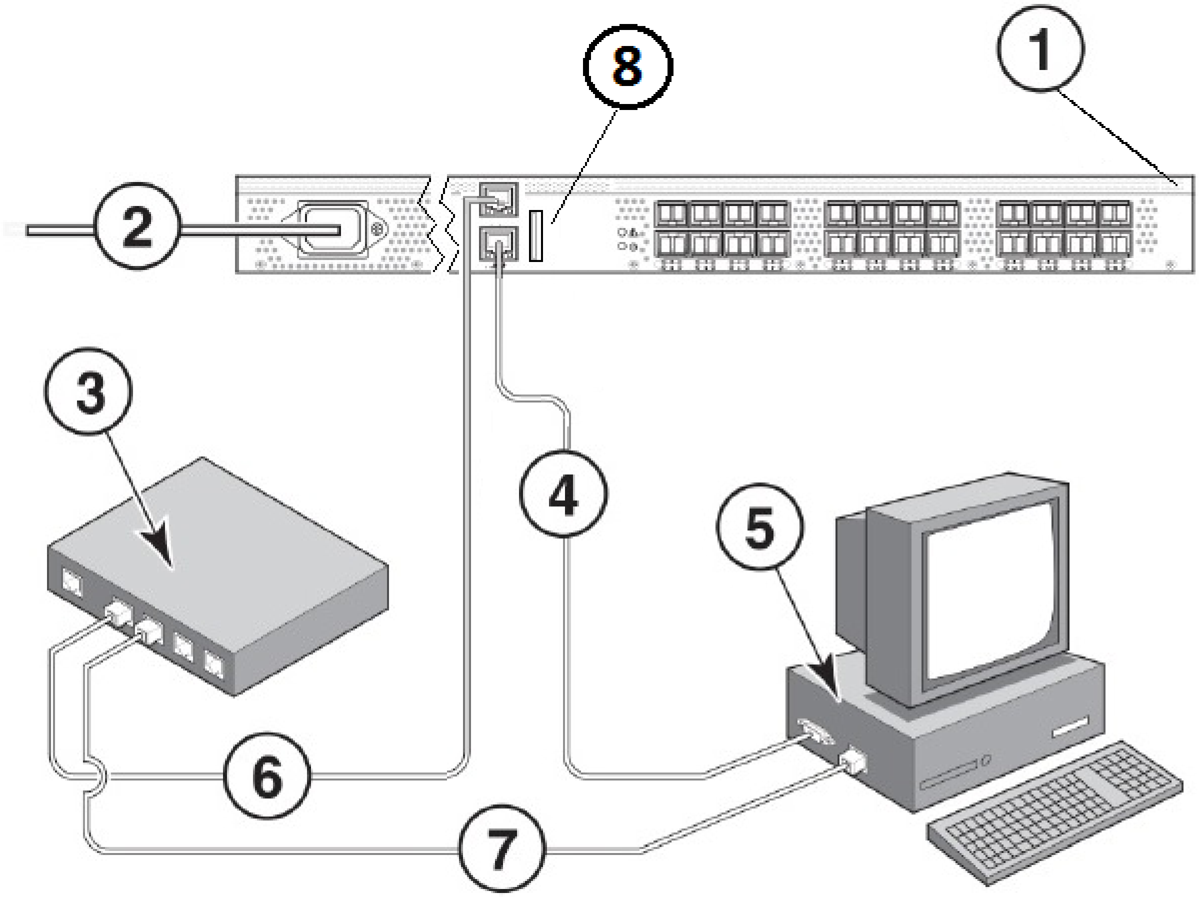
## Prerequisites

This procedure assumes the following:

- The Brocade FC (Fibre Channel) switch has been configured and is on the HSS network.
- The connections shown in the figure have been made.

**NOTE:** The SMW Ethernet port can be directly connected to the switch MGT port if no hub/switch (item 3 in figure) is available.

*Figure 18. Brocade FC Switch Connections*



| | |
|---|---|
| 1 | Brocade switch |
| 2 | Power cable |
| 3 | Ethernet hub or switch |
| 4 | Serial cable |
| 5 | Setup computer |
| 6 | Ethernet cable from hub to Brocade switch |
| 7 | Ethernet cable from setup computer to Ethernet hub or switch |
| 8 | Brocade USB -requires special USB Drive |

## About this task

This procedure describes how to set up and use Web Tools, the embedded GUI application, to configure zoning of the Brocade 6505 Fibre Channel (FC) switch for standard boot RAID usage.

## Procedure

1. Set up the GUI.

   a. Open a Firefox web browser on the SMW.

   ```
   crayadm@smw> cd
   crayadm@smw> firefox
   ```

   b. Enter the IP address of the switch (10.1.0.250) into the address bar.

   A pop up window appears.

   c. Select **Save File** to save the file in the default Downloads directory.

   The Java plugin starts the GUI and a security warning appears, stating that the certificate is not trusted and the browser will not let the application continue.

   d. Exit the browser.

   e. Run `javaws -viewer` in an SMW xterm window.

   Substitute the correct version of java in the javaws path, if different from this example.

   ```
   crayadm@smw> cd
   crayadm@smw> cd /Downloads
   crayadm@smw> /usr/lib64/jvm/java-1.7.1-ibm-1.7.1/jre/bin/javaws -viewer
   ```

   The **Java Control Panel** appears.

   f. Add the switch to the **Exception Site List** in the **Security** tab.

   Select the **Security** tab. Under **Exception Site List**, click **Edit Site List** and enter **`http://10.1.0.250`** as a trusted site, then click **Apply**.

   g. Exit the `javaws -viewer` application.

   h. Open `switchExplorer_installed.html` in an SMW xterm window.

   Change directories to the `Downloads` directory, which is where the `switchExplorer_installed.html` file should be located (type **`ls`** to verify, if desired). Run `javaws -verbose switchExplorer_installed.html`.

   Substitute the correct version of java in the javaws path, if different from this example.

   ```
   crayadm@smw> cd /Downloads
   crayadm@smw> /usr/lib64/jvm/java-1.7.1-ibm-1.7.1/jre/bin/javaws -verbose \
   switchExplorer_installed.html
   ```

   A **Verifying application** window appears. About three minutes later, a **Security Warning** window appears.

   i. Select **I accept the risk and want to run this application**, then click **Run**.

   A **Login** window appears.

j. Log in to the switch as `admin` with password `password`.

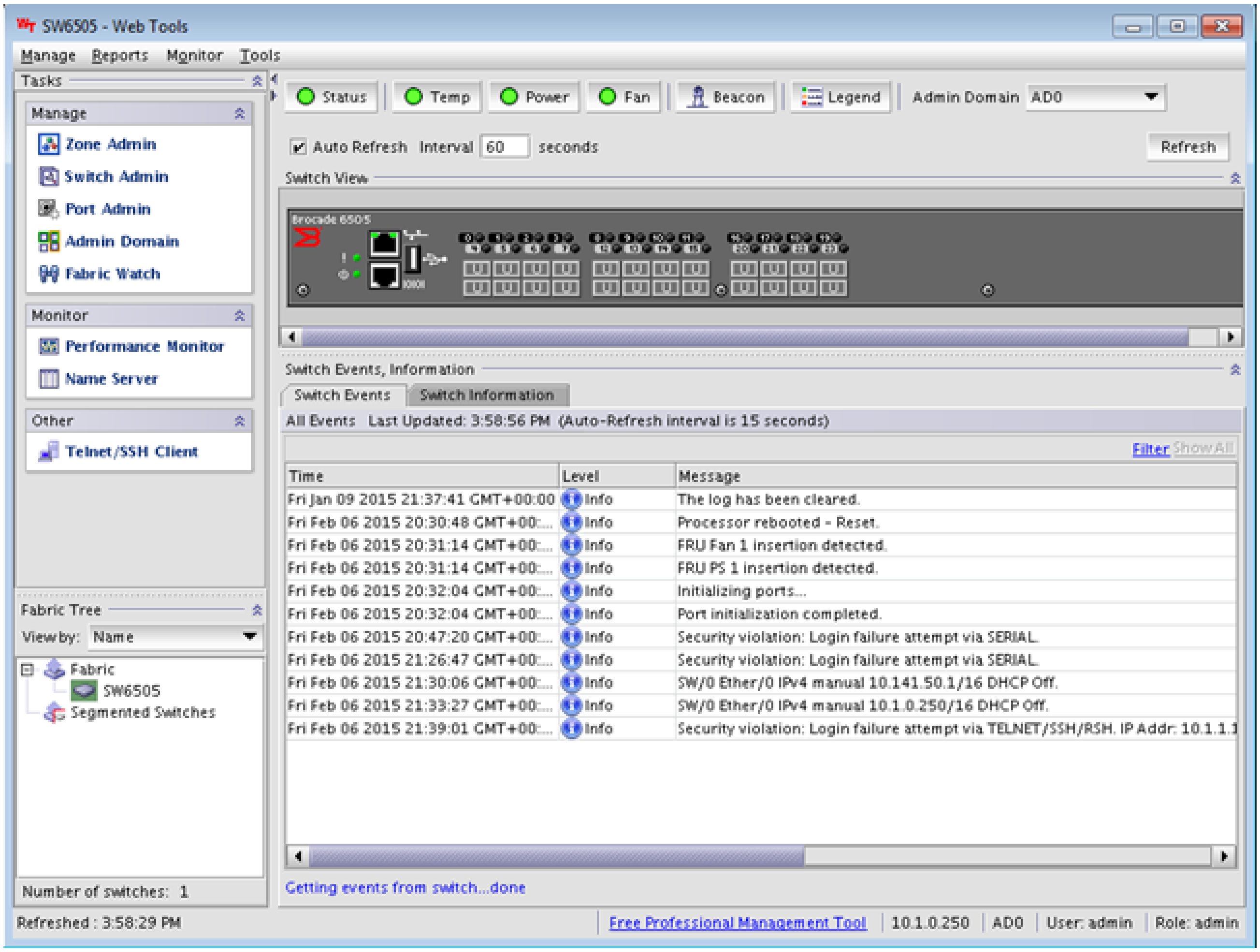
The main **Web Tools** switch window appears.

Use the Web Tools GUI to complete configuration of the Brocade FC switch.

2. Check the LED status of the tabs on the **Web Tools** window to ensure that there are no major issues.

The tab LEDs should all be green.

*Figure 19. Brocade FC Switch Web Tools Window*



☑ Auto Refresh Interval 60 seconds

All Events Last Updated: 3:58:56 PM (Auto-Refresh interval is 15 seconds)

| Time | Level | Message |
|---|---|---|
| Fri Jan 09 2015 21:37:41 GMT+00:00 | Info | The log has been cleared. |
| Fri Feb 06 2015 20:30:48 GMT+00:... | Info | Processor rebooted - Reset. |
| Fri Feb 06 2015 20:31:14 GMT+00:... | Info | FRU Fan 1 insertion detected. |
| Fri Feb 06 2015 20:31:14 GMT+00:... | Info | FRU PS 1 insertion detected. |
| Fri Feb 06 2015 20:32:04 GMT+00:... | Info | Initializing ports... |
| Fri Feb 06 2015 20:32:04 GMT+00:... | Info | Port initialization completed. |
| Fri Feb 06 2015 20:47:20 GMT+00:... | Info | Security violation: Login failure attempt via SERIAL. |
| Fri Feb 06 2015 21:26:47 GMT+00:... | Info | Security violation: Login failure attempt via SERIAL. |
| Fri Feb 06 2015 21:30:06 GMT+00:... | Info | SW/0 Ether/0 IPv4 manual 10.141.50.1/16 DHCP Off. |
| Fri Feb 06 2015 21:33:27 GMT+00:... | Info | SW/0 Ether/0 IPv4 manual 10.1.0.250/16 DHCP Off. |
| Fri Feb 06 2015 21:39:01 GMT+00:... | Info | Security violation: Login failure attempt via TELNET/SSH/RSH. IP Addr: 10.1.1.1 |

Getting events from switch...done

Refreshed : 3:58:29 PM | 10.1.0.250 | AD0 | User: admin | Role: admin

3. Change the name of the switch.

a. Click **Switch Admin** in the **Manage** pane (upper left).

The **Switch Administration** window appears.

b. Change the name of the switch to BS#, where the # is the number of the switch being configured (e.g., BS01), then click **Apply** to save the name.

A confirmation window appears.

c. Click **Yes** to confirm, then close the **Switch Administration** window to return to the main **Web Tools** window.

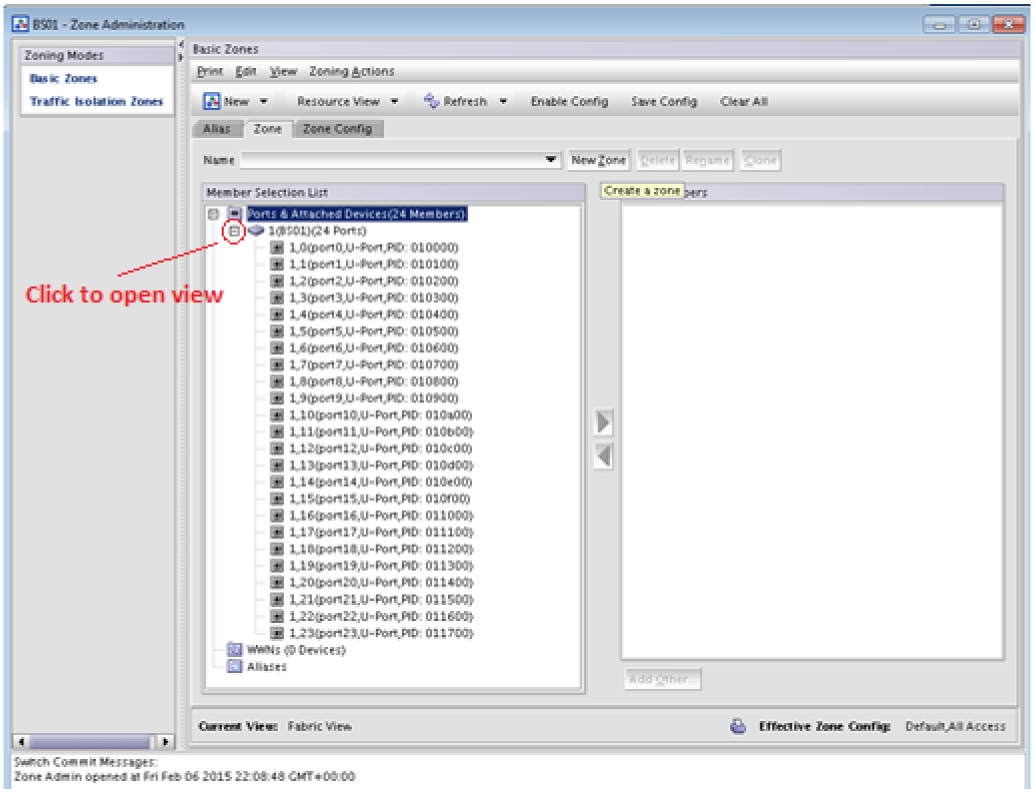
**4.** Set up a Boot zone for the switch.

a. Click **Zone Admin** in the **Manage** pane (upper left).

The **Zone Administration** window appears.

b. Click the **Zone** tab in the **Zone Administration** window, then click the **New Zone** button (located just below tabs, to the right).

*Figure 20. Brocade FC Switch Zone Administration Window*


BS01 - Zone Administration
Zoning Modes
Basic Zones
Traffic Isolation Zones
Basic Zones
Print
Edit
View
Zoning Actions
New
Resource View
Refresh
Enable Config
Save Config
Clear All
Alias
Zone
Zone Config
Name
New Zone
Delete
Rename
Clone
Member Selection List
Create a zone
Ports & Attached Devices(24 Members)
1(BS01)(24 Ports)
1,0(port0,U-Port,PID: 010000)
1,1(port1,U-Port,PID: 010100)
1,2(port2,U-Port,PID: 010200)
1,3(port3,U-Port,PID: 010300)
1,4(port4,U-Port,PID: 010400)
1,5(port5,U-Port,PID: 010500)
1,6(port6,U-Port,PID: 010600)
1,7(port7,U-Port,PID: 010700)
1,8(port8,U-Port,PID: 010800)
1,9(port9,U-Port,PID: 010900)
1,10(port10,U-Port,PID: 010a00)
1,11(port11,U-Port,PID: 010b00)
1,12(port12,U-Port,PID: 010c00)
1,13(port13,U-Port,PID: 010d00)
1,14(port14,U-Port,PID: 010e00)
1,15(port15,U-Port,PID: 010f00)
1,16(port16,U-Port,PID: 011000)
1,17(port17,U-Port,PID: 011100)
1,18(port18,U-Port,PID: 011200)
1,19(port19,U-Port,PID: 011300)
1,20(port20,U-Port,PID: 011400)
1,21(port21,U-Port,PID: 011500)
1,22(port22,U-Port,PID: 011600)
1,23(port23,U-Port,PID: 011700)
WWNs (0 Devices)
Aliases
Click to open view
Add Other...
Current View: Fabric View
Effective Zone Config: Default,All Access

Switch Commit Messages:
Zone Admin opened at Fri Feb 06 2015 22:08:48 GMT+00:00

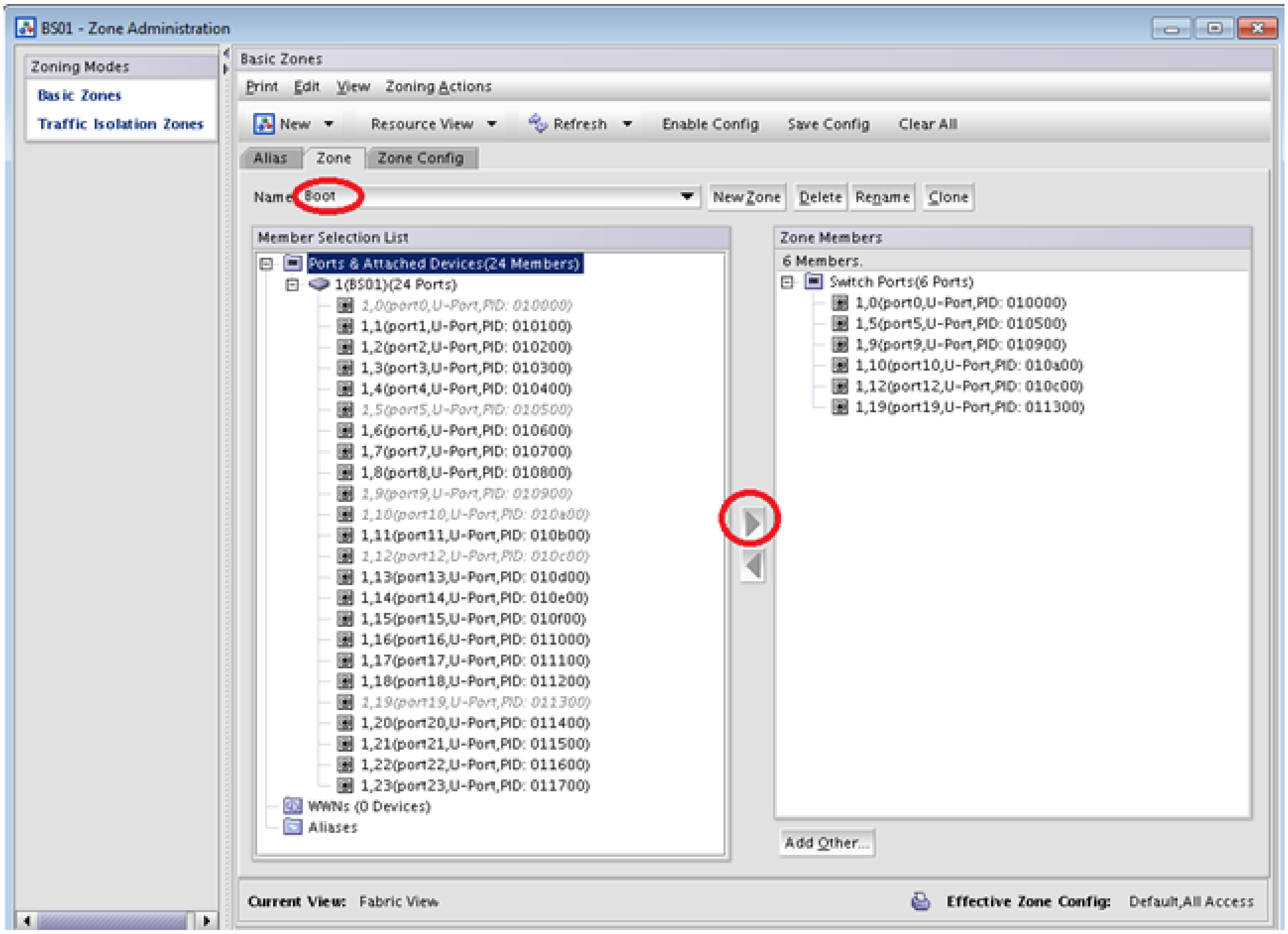
A **Create New Zone** window appears.

c. Enter the name of the new zone as "Boot."

"Boot" is displayed in the **Name** field in the **Zone** tab of the **Zone Administration** window.

d. Select port 0 in the **Member Selection List** (left pane in the **Zone** tab), then click the right arrow icon to move port 0 into **Zone Members** (right pane in the **Zone** tab). Repeat for ports 4, 5, 10, 11, 12, and 19.

*Figure 21. Brocade FC Switch Boot Zone Members*



Name Boot

Member Selection List

- Ports & Attached Devices(24 Members)
  - 1(BS01)(24 Ports)
    - 1,0(port0,U-Port,PID: 010000)
    - 1,1(port1,U-Port,PID: 010100)
    - 1,2(port2,U-Port,PID: 010200)
    - 1,3(port3,U-Port,PID: 010300)
    - 1,4(port4,U-Port,PID: 010400)
    - 1,5(port5,U-Port,PID: 010500)
    - 1,6(port6,U-Port,PID: 010600)
    - 1,7(port7,U-Port,PID: 010700)
    - 1,8(port8,U-Port,PID: 010800)
    - 1,9(port9,U-Port,PID: 010900)
    - 1,10(port10,U-Port,PID: 010a00)
    - 1,11(port11,U-Port,PID: 010b00)
    - 1,12(port12,U-Port,PID: 010c00)
    - 1,13(port13,U-Port,PID: 010d00)
    - 1,14(port14,U-Port,PID: 010e00)
    - 1,15(port15,U-Port,PID: 010f00)
    - 1,16(port16,U-Port,PID: 011000)
    - 1,17(port17,U-Port,PID: 011100)
    - 1,18(port18,U-Port,PID: 011200)
    - 1,19(port19,U-Port,PID: 011300)
    - 1,20(port20,U-Port,PID: 011400)
    - 1,21(port21,U-Port,PID: 011500)
    - 1,22(port22,U-Port,PID: 011600)
    - 1,23(port23,U-Port,PID: 011700)
- WWNs (0 Devices)
- Aliases

Zone Members

6 Members.

- Switch Ports(6 Ports)
  - 1,0(port0,U-Port,PID: 010000)
  - 1,5(port5,U-Port,PID: 010500)
  - 1,9(port9,U-Port,PID: 010900)
  - 1,10(port10,U-Port,PID: 010a00)
  - 1,12(port12,U-Port,PID: 010c00)
  - 1,19(port19,U-Port,PID: 011300)

Current View: Fabric View

Effective Zone Config: Default,All Access

Ports 0, 4, 5, 10, 11, 12, and 19 are in the **Zone Members** pane.

**5.** Configure the Boot zone.

a. Click the **Zone Config** tab in the **Zone Administration** window.

b. Click all of the **+** icons in the **Member Selection List** to expand all of the entries.

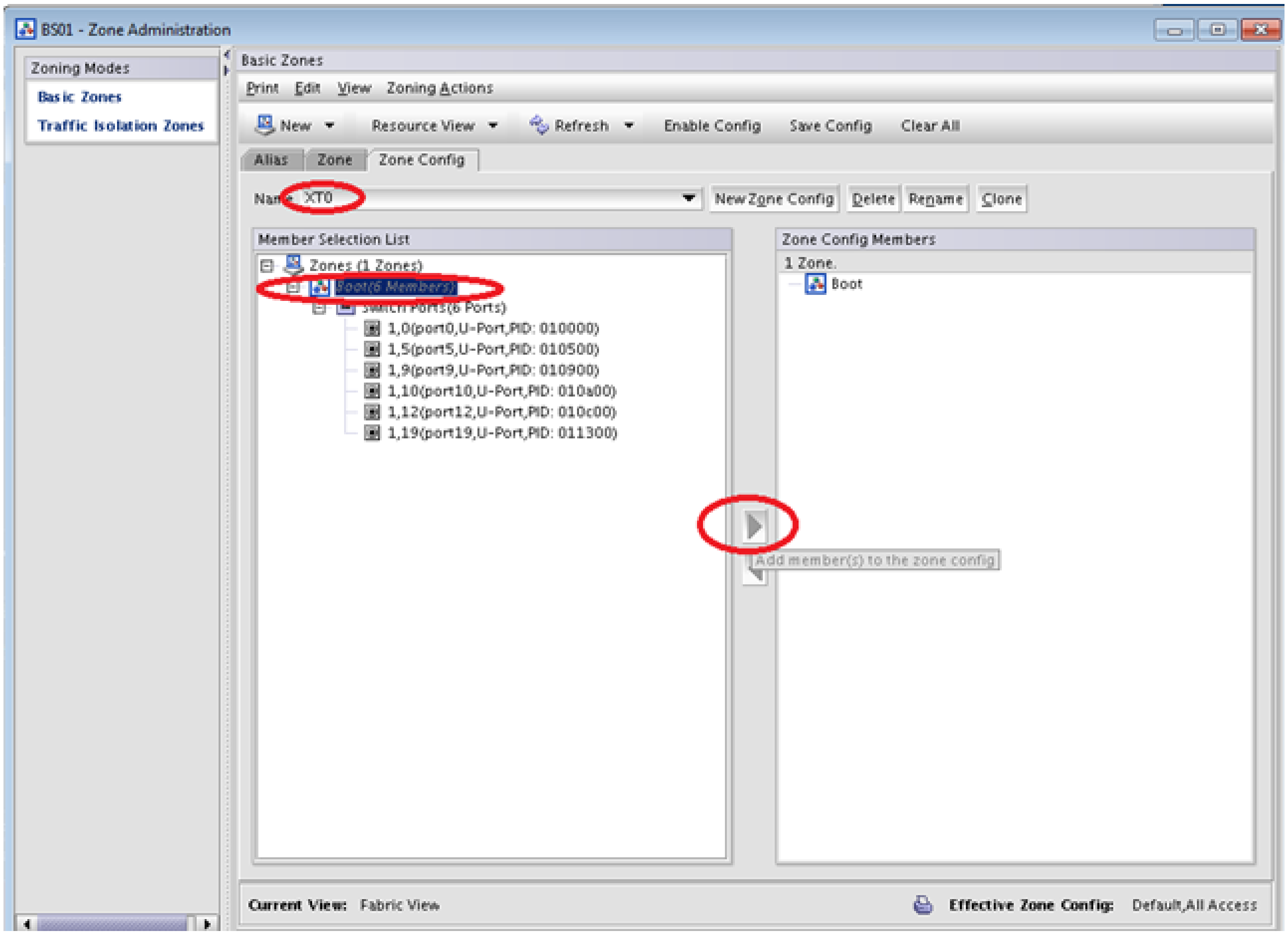
c. Click the **New Zone Config** button.

A **Create New Config** window appears.

d. Enter the name "XT0" in the **Create New Config** window, then click **OK**.

e. Select the Boot zone in the **Member Selection List** (left pane in the **Zone Config** tab), then click the right arrow icon to move the Boot zone into **Zone Config Members** (right pane in the **Zone Config** tab), which puts it in the XT0 group.

*Figure 22. Brocade FC Switch XT0 Zone Config Members*



Name XT0

Zones (1 Zones)
Boot(6 Members)
Switch Ports(6 Ports)
1,0(port0,U-Port,PID: 010000)
1,5(port5,U-Port,PID: 010500)
1,9(port9,U-Port,PID: 010900)
1,10(port10,U-Port,PID: 010a00)
1,12(port12,U-Port,PID: 010c00)
1,19(port19,U-Port,PID: 011300)

1 Zone.
Boot

Current View: Fabric View

Effective Zone Config: Default,All Access

f. Click the **Save Config** button (located just above tabs, to the right), then click **Yes** in the confirmation window that appears.

**6.** Enable the Boot zone configuration.

a. Click the **Enable Config** button (located just above tabs, to the right).

A **Choose Zone Config to be enabled** window appears.

b. Select the XT0 zone config from the menu, click **OK**, then click **Yes** in the **Enable Config XT0** confirmation window that appears.

At the bottom of the **Zone Administration** window, a status appears.

c. Click the **X** button at the top right to exit from the **Zone Administration** window when the status shows a "Commit succeeded" message, then click **Yes** in the exit confirmation window that appears.

The main **Web Tools** window appears.

**7.** Verify the configuration.

a. Remove power from the switch.

b. Re-apply power after 30 seconds, then wait for the Brocade FC switch to finish booting via the serial connection.

c. Enter the following from the serial connection:

```
BS01:> enable
BS01:> zoneshow

Defined configuration:
cfg:   XT0    boot
zone:  boot    1,0; 1,4; 1,5; 1,10; 1,11; 1,12; 1,19

Effective configuration:
cfg:   XT0
zone:  boot 1,0
            1,4
            1,5
            1,10
            1,11
            1,12
            1,19
```

d. Verify that the configuration matches the effective configuration.

The Brocade FC switch is now zoned. Proceed to the next step in the boot RAID configuration process: *Reboot the SMW and Verify LUNs are Recognized* on page 76.

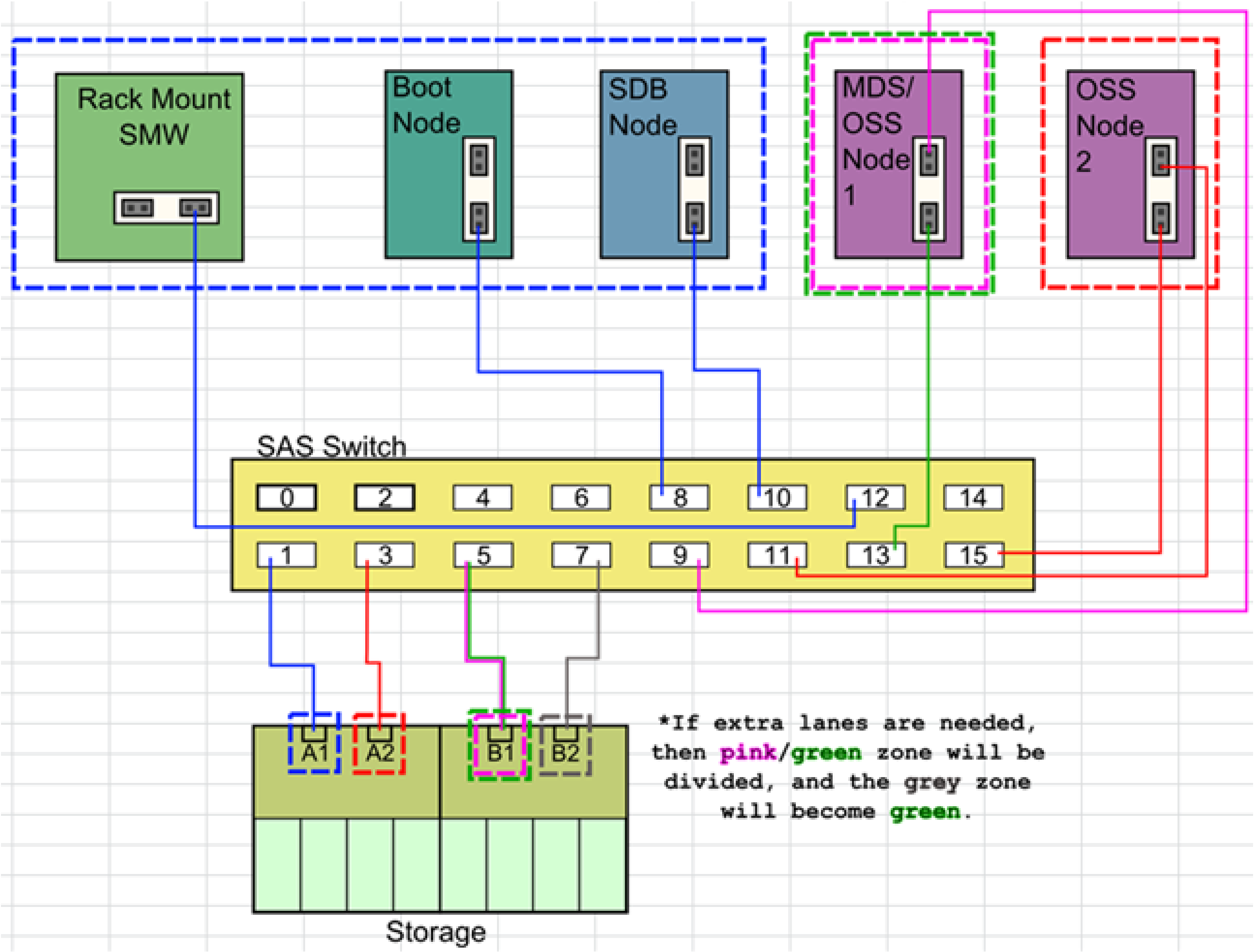
### 3.2.3.8 Zone the LSI SAS Switch

## Prerequisites

This procedure assumes the following:

- The LSI 6160 SAS (Serial Attached SCSI) switch has been configured and is on the HSS network.
- The SMW is within 10 meters of the SAS switch (otherwise an FC switch is required).
- The following connections have been made (as shown in figure):
  - The SMW SAS card port 1 is connected by SAS cable to port 12 on the SAS switch.
  - The boot node is connected by SAS cable to port 8 on the SAS switch.
  - The SDB node is connected by SAS cable to port 10 on the SAS switch.

*Figure 23. LSI SAS Switch Boot RAID Cable Connections*



## About this task

This procedure describes how to use SAS Domain Manager, the embedded GUI application, to zone the LUNs on the LSI (now Broadcom) 6160 SAS switch for standard boot RAID usage.

## Procedure

1. Open the SAS Domain Manager GUI.

   a. Open a Firefox web browser on the SMW.

   ```
   crayadm@smw> cd
   crayadm@smw> firefox
   ```

   b. Enter the IP address of the switch (10.1.0.250) into the address bar.

   c. If Firefox displays the message "JRE 1.6 or higher required," do one of the following:

   - **Select Tools > Add-ons > Plugins** from the Firefox menu bar, then click the green button marked “enable” for Java to enable Java and allow the switch GUI to work.

- Run the following command from the command line to bypass Firefox and open the GUI.

  ```
  crayadm@smw> /usr/lib64/jvm/jre/bin/javaws http://10.1.0.250/sdmgui.jnlp
  ```

  When the **Opening sdmgui.jnlp** window appears, click OK to open the file, then in the **Warning - Security** window that appears, click **Run** to run the application.

The **SAS Domain Manager GUI** login window appears.

d. Log in to the switch as `admin` with password `admin`.

The **SAS Domain Manager GUI** main window appears.

**2.** Create the zone groups.

a. Click the **Domain** tab in the **SAS Domain Manager GUI** main window, then click **Create Zone Group**.

Create the following zone groups and assign the ports as indicated. Note that it is the phys values that should be mapped to the zone group, not the port(s).

| Zone Group Name | SAS Ports | Phys |
|---|---|---|
| XT-A1-Storage | 1 | 4,5,6,7 |
| XT-A2-Storage | 3 | 12,13,14,15 |
| XT-B1-Storage | 5 | 20,21,22,23 |
| XT-B2-Storage | 7 | 28,29,30,31 |
| XT-SysRaid | 8<br>10<br>12<br>4<br>6 | 32,33,34,35<br>40,41,42,43<br>48,49,50,51<br>16,17,18,19<br>24,25,26,27 |
| XT-MDS-OSSn1(1) | 9<br>13<br>14 | 36,37,38,39<br>52,53,54,55<br>56,57,58,59 |
| XT-OSSn2(1) | 11<br>15 | 44,45,46,47<br>60,61,62,63 |

(1) If an external Lustre server will be used instead of an internal Lustre server (DAL), then the XT-MDS-OSSn1 and XT-OSSn2 zones are not necessary.

**3.** Create the zone set.

a. Click the **Domain** tab in the **SAS Domain Manager GUI** main window, then click **Create Zone Set**.

Create the zone set XT0.

b. Assign the zone groups to the zone set by clicking the empty boxes to match this layout:

| Zone Group Name | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| XT-A1-Storage | | | | | X | | |
| XT-A2-Storage | | | | | | | X |
| XT-B1-Storage | | | | | | X | |
| XT-B2-Storage | | | | | | *X*(2) | |
| XT-SysRaid | X | | | | | | |
| XT-MDS-OSSn1(1) | | | X | *X*(2) | | | |
| XT-OSSn2(1) | | X | | | | | |

(1) If an external Lustre server will be used instead of an internal Lustre server (DAL), then the XT-MDS-OSSn1 and XT-OSSn2 zones are not necessary.

(2) Setting a box marked with *X* may add more SAS lanes for performance, but do not set unless instructed to.

**4.** Activate the zone set.

a. Click the **Domain** tab, then click **Activate/Deactivate Zone Set**.

b. Select XT0 from the menu and enter the Zone Password of `lynx` to activate the zone set.

c. Click the **Views** tab under **Active zone set** to verify that the zone groups and zone set are correct and active.

The SAS switch is now zoned. Proceed to the next step in the boot RAID configuration process: *Reboot the SMW and Verify LUNs are Recognized* on page 76.

### 3.2.3.9 Reboot the SMW and Verify LUNs are Recognized

## About this task

Use this procedure to make the SMW rediscover the LUNs (logical unit numbers) and zones that were created.

## Procedure

**1.** Log on as the `root` user.

```
crayadm@smw> su - root
```

**2.** Reboot the SMW to ensure that the LUNs are recognized.

```
smw# reboot
```

⚠ **CAUTION:** Failure to reboot the SMW at this point could produce unexpected results later on.

**3.** When the SMW has finished rebooting, log on as the `root` user.

```
crayadm@smw> su - root
```

4. Execute the `lsscsi` command to verify that the LUNs (volumes) have been rediscovered.

   ```
   smw# lsscsi
   ```

5. List the disk devices by using the `fdisk` command to verify that the LUNs (volumes) are configured according to the boot LUN configuration table in *Recommended Boot RAID LUN Values* on page 59.

   ```
   smw# fdisk -l
   ```

### 3.2.4 Make a Snapshot Manually

#### Prerequisites

This procedure assumes that the SLES 12 base operating system has been installed on the SMW and boot RAID devices have been configured, but no other software has been installed yet.

#### About this task

A Btrfs snapshot of the SMW should be created immediately after SLES 12 has been installed and before any files or directories have been modified by Cray's installation software or the rest of the installation process. With this snapshot, it will be possible to revert to this point if an initial/fresh install is repeated.

Snapshots are usually made using the `snaputil` program, but that program has not been installed at this point in the installation process. `snaputil` will be installed to the SMW with other Cray RPMs for the SMW and will be used for all Btrfs snapshot manipulations after this point.

Note that snapshots capture content in these three directories: `/`, `/var/lib/mysql`, and `/var/opt/cray/repos`. Used in conjunction with backups of config sets, they provide enough information to be able to re-create the state of the system at a particular time.

#### Procedure

1. Determine the root subvolume.

   It will be the string starting with "UUID." In this example it is "UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde."

   ```
   smw# grep " / " /etc/fstab
   UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /            btrfs
   defaults        0 0
   ```

2. Mount the root subvolume.

   Substitute the correct subvolume string for the example string shown in this command.

   ```
   smw# mount -o subvol=@ UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /mnt
   ```

3. Create a subvolume for snapshots (if `/mnt/snapshots` does not already exist).

   ```
   smw# btrfs sub create /mnt/snapshots
   ```

4. Create the snapshot (if `/mnt/snapshots/SLES12` does not already exist).

```
smw# btrfs sub snap / /mnt/snapshots/SLES12
```

5. Unmount the snapshot.

```
smw# umount /mnt
```

6. Make a new `/media/root-sv` directory.

```
smw# mkdir -p /media/root-sv
```

7. Mount root subvolume under `/media/root-sv` instead of `/mnt` as was used above.

   Substitute the correct subvolume string for the example string shown in this command.

```
smw# mount -o subvol=@ UUID=ffb0b613-2033-4835-87b5-6ca8ff1bacde /media/root-sv
```

A "SLES12" snapshot has been made. Reboot to this snapshot whenever it is necessary to restart a fresh software installation from this point.

## 3.3 Install the SMW and CLE Software

To install the SMW and CLE software, use the following procedures in the order listed.

1. *Start a Typescript File* on page 78
2. *Prepare to Bootstrap the SMW Installation* on page 79
3. *Determine the Persistent Device Name for a LUN* on page 81
4. *RAID Disk Space Requirements* on page 84
5. *Bootstrap the SMW Installation* on page 85
6. *Provision SMW Storage* on page 92
7. *Run the Installer for an Initial Installation* on page 93
8. Think you know how to boot an SMW? Don't miss the extra, crucial step in this procedure: *Set Default Snapshot and Boot the SMW* on page 95

### 3.3.1 Start a Typescript File

### About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before installing a new software release
- just before configuring the newly installed software

### Procedure

1. Log in as root to the SMW.

2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

   For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

## 3.3.2 Prepare to Bootstrap the SMW Installation

### Prerequisites

This procedure assumes that the base operating system has been installed on the SMW and that the boot RAID has been set up.

### About this task

The installer requires several ISO files to be available for setting up and installing packages from SLE repositories. The names of these ISOs are hard-coded in the installer configuration, but the containing directory can be anywhere that makes sense.

> **IMPORTANT:** The default location for these ISO files is `/root/isos`. The `--iso-dir` argument must be specified for `SMWinstall` if this is not the correct location for the ISO files on this system.

The Cray release distribution media consists of one DVD and several other pieces of media that may be on DVDs or furnished as ISO files. These ISO files are available for download at CrayPort (*https://crayport.cray.com*).

### Procedure

COLLECT ISO FILES

1. Make a directory on the SMW to hold the ISO files, if one does not already exist.

   Instead of placing the ISOs directly in `/root/isos`, use these two commands to place that directory into the btrfs subvolume `/var/adm/cray`, which is exempt from snapshots. This prevents the large ISO files from unnecessarily increasing the size of snapshots.

```
smw# mkdir -p /var/adm/cray/release/isos
smw# ln -s /var/adm/cray/release/isos /root/isos
```

2. Download the SLES 12 distribution ISOs to the ISO directory on the SMW.
   - `SLE-12-Module-Legacy-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-Module-Public-Cloud-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-SDK-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-Server-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-WE-DVD-x86_64-GM-DVD1.iso`
3. Download the CentOS 6.5 distribution ISO (`CentOS-6.5-x86_64-bin-DVD1.iso`) to the ISO directory on the SMW.
4. Download CLE 6.0 and SMW 8.0 ISOs to the ISO directory on the SMW.
   - SMW release: `smw-8.0.3075-201701182038.iso`
   - CLE release: `cle-6.0.3074-201701182038.iso`
5. Download the SLES 12 security updates ISO (`sleupdate-12sp0+161026-201611021158.iso`) to the ISO directory on the SMW.
6. Make a directory on the SMW (if it does not already exist) to hold any patches that may be available on CrayPort.

   ```
   smw# mkdir -p /var/adm/cray/release/patchsets
   ```

7. Download SMW and CLE patches to the patchset directory on the SMW, as described in the release notes.

MOUNT MEDIA

8. Mount SMW media.

   a. Confirm that this is the right SMW media.

   ```
   smw# ls -l /root/isos/smw*iso
   -rw-r--r-- 1 root root 427184128 Jan 18 21:42 smw-8.0.3075-201701182038.iso
   ```

   b. Set environment variables for the SMW media.

   Use the release string (actually, the build ID) and the date-time stamp for the SMW media as the values for SMW_RELEASE and SMW_SOFTWARE, as shown in this example.

   ```
   smw# export SMW_RELEASE=8.0.3075
   smw# echo $SMW_RELEASE

   smw# export SMW_SOFTWARE=201701182038
   smw# echo $SMW_SOFTWARE
   ```

   c. Mount the SMW release media.

   ```
   smw# mkdir -p /media/SMW
   smw# mount -o loop,ro /root/isos/smw-${SMW_RELEASE}-${SMW_SOFTWARE}.iso \
   /media/SMW
   ```

**9.** Mount CLE media.

a. Confirm that this is the right CLE media.

```
smw# ls -l /root/isos/cle*iso
-rw-r--r-- 1 root root 1146388480 Jan 18 20:38 cle-6.0.3074-201701182038.iso
```

b. Set environment variables for the CLE media.

Use the release string and the date-time stamp for the CLE media as the values for CLE_RELEASE and CLE_SOFTWARE, as shown in this example.

```
smw# export CLE_RELEASE=6.0.3074
smw# echo $CLE_RELEASE

smw# export CLE_SOFTWARE=201701182038
smw# echo $CLE_SOFTWARE
```

c. Mount the CLE release media.

```
smw# mkdir -p /media/CLE
smw# mount -o loop,ro /root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
/media/CLE
```

**10.** Set an environment variable for the SLES 12 security updates media.

Use the entire name of the SLES 12 security updates media as the environment variable. This will be used when installing SMW and CLE software and SLES 12 security updates together later in the process.

```
smw# export SLE_SOFTWARE=sleupdate-12sp0+161026-201611021158
smw# echo $SLE_SOFTWARE
```

COPY THE INSTALL CONFIGURATION FILE

**11.** Copy `install.cle.conf`.

The `install.cle.conf` file contains configuration that controls the installer's image building behavior. Copy `install.cle.conf.example` from the CLE installation media to `/var/adm/cray/install.cle.conf` and modify it if necessary.

```
smw# cp -p /media/CLE/products/cle/install.cle.conf.example \
/var/adm/cray/install.cle.conf
```

At this point there is nothing in this file that should be changed for a fresh install. Later this will be changed for updates to CLE.

**12.** Unmount CLE media.

```
smw# umount /media/CLE
```

## 3.3.3 Determine the Persistent Device Name for a LUN

### About this task

After initial partitioning of the boot RAID, always address the storage using its persistent `/dev/disk/by-id/` name. Do not use the short `/dev/sdxx` name, which cannot uniquely identify the disk between reboots.

Use either step 1 (for systems not using multipath) or step 2 (for systems using multipath) of this procedure to determine the persistent (by-id) device names for the following devices:

- Disk devices on the boot RAID that can be used for boot node persistent storage
- Disk devices on the boot RAID that can be used for SDB node persistent storage
- Disk devices on the boot RAID that can be used for SMW persistent storage

## Procedure

1. (For systems not using multipath) Determine the persistent device name on a system not configured for multipath.

   a. Use `lsscsi` to show the `/dev/sd*` device name associated with a LUN or volume group.

   In the first column of the output, the LUN is the final number in the `[n:n:n:n]` value. In this example, LUN 15 is associated with `/dev/sdo`.

```
crayadm@smw> lsscsi
[0:0:0:0]    disk    ATA      TOSHIBA MK1661GS ME0D  /dev/sda
[0:0:1:0]    disk    ATA      ST91000640NS     AA03  /dev/sdb
[0:0:2:0]    disk    ATA      TOSHIBA MK1661GS ME0D  /dev/sdc
.
.
.
[5:0:0:15]   disk    LSI      INF-01-00        0786  /dev/sdo
[5:0:0:16]   disk    LSI      INF-01-00        0786  /dev/sdp
[5:0:0:17]   disk    LSI      INF-01-00        0786  /dev/sdq
[5:0:0:18]   disk    LSI      INF-01-00        0786  /dev/sdr
```

   b. Use `ls -l` to map the `/dev/sd*` device name to the persistent device name.

   To display the persistent device name for only one LUN, use `grep`. This example displays the persistent device name for `/dev/sdo` (that is, LUN 15).

```
crayadm@smw> ls -l /dev/disk/by-id | grep sdo
lrwxrwxrwx 1 root root 10 Sep  4 00:56
scsi-360080e500037667a000003a2519e3ff2 -> ../../sdo
lrwxrwxrwx 1 root root 10 Sep  4 00:56
wwn-0x60080e500037667a000003a2519e3ff2 -> ../../sdo
```

   There are two results for LUN 15. The one with prefix "scsi" is the one to use, so the persistent device name for LUN 15 is scsi-360080e500037667a000003a2519e3ff2.

2. (For systems using multipath) Determine the persistent device name on a system configured for multipath.

   On a system using multipath, when the multipath daemon is running, the persistent device name is the multipath path.

   a. Ensure that the multipath daemon is running.

   If multipathd is not running, start it now. Otherwise, the multipath path (the persistent device

```
crayadm@smw> systemctl status -l multipathd
```

   If multipathd is not running, start it now.

```
crayadm@smw> systemctl start multipathd
```

b. Use the `SMdevices` command to determine the boot RAID volume ID for each volume group.

   This command is available on systems with the SANtricity Storage Manager software installed. Note that for each volume group, there are multiple paths that reference the same volume ID (shown in bold in the following examples).

   For the boot node volume group:

```
crayadm@smw> SMdevices | grep boot0
/dev/sdj (/dev/sg10) [Storage Array RR_10000716_Boot, Volume boot0, LUN 1,
Volume ID <600a098000a9d1b9000000cb5805168d>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
/dev/sdm (/dev/sg13) [Storage Array RR_10000716_Boot, Volume boot0, LUN 1,
Volume ID <600a098000a9d1b9000000cb5805168d>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sdd (/dev/sg4) [Storage Array RR_10000716_Boot, Volume boot0, LUN 1,
Volume ID <600a098000a9d1b9000000cb5805168d>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sdg (/dev/sg7) [Storage Array RR_10000716_Boot, Volume boot0, LUN 1,
Volume ID <600a098000a9d1b9000000cb5805168d>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
```

   For the SDB node volume group:

```
crayadm@smw> SMdevices | grep sdb
/dev/sdk (/dev/sg11) [Storage Array RR_10000716_Boot, Volume sdb0, LUN 2,
Volume ID <600a098000a9d1b9000000cd5805169a>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
/dev/sdn (/dev/sg14) [Storage Array RR_10000716_Boot, Volume sdb0, LUN 2,
Volume ID <600a098000a9d1b9000000cd5805169a>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sde (/dev/sg5) [Storage Array RR_10000716_Boot, Volume sdb0, LUN 2,
Volume ID <600a098000a9d1b9000000cd5805169a>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sdh (/dev/sg8) [Storage Array RR_10000716_Boot, Volume sdb0, LUN 2,
Volume ID <600a098000a9d1b9000000cd5805169a>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
```

   For the SMW node volume group:

```
crayadm@smw> SMdevices | grep smw
/dev/sdl (/dev/sg12) [Storage Array RR_10000716_Boot, Volume smw0, LUN 0,
Volume ID <600a098000a9d1b9000000c858051681>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sdc (/dev/sg3) [Storage Array RR_10000716_Boot, Volume smw0, LUN 0,
Volume ID <600a098000a9d1b9000000c858051681>, Preferred Path (Controller-A):
Owning controller - Active/Optimized]
/dev/sdf (/dev/sg6) [Storage Array RR_10000716_Boot, Volume smw0, LUN 0,
Volume ID <600a098000a9d1b9000000c858051681>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
/dev/sdi (/dev/sg9) [Storage Array RR_10000716_Boot, Volume smw0, LUN 0,
Volume ID <600a098000a9d1b9000000c858051681>, Alternate Path (Controller-B):
Non owning controller - Active/Non-optimized]
```

c. Construct the multipath path for each volume group from its identified volume ID.

   Prepend "dm-uuid-mpath-3" to each volume ID.

- For the boot node volume group: **dm-uuid-mpath-3600a098000a9d1b9000000cb5805168d**
- For the SDB node volume group: **dm-uuid-mpath-3600a098000a9d1b9000000cd5805169a**
- For the SMW node volume group: **dm-uuid-mpath-3600a098000a9d1b9000000c858051681**

d. (optional) Verify that the constructed multipath path is correct.

For the boot node volume group (the multipath path is shown in bold):

```
crayadm@smw> ls -l /dev/disk/by-id/ | grep 600a098000a9d1b9000000cb5805168d
lrwxrwxrwx 1 root root 10 Dec 22 15:08 dm-
name-3600a098000a9d1b9000000cb5805168d -> ../../dm-2
lrwxrwxrwx 1 root root 10 Dec 22 15:08 dm-uuid-
mpath-3600a098000a9d1b9000000cb5805168d -> ../../dm-2
lrwxrwxrwx 1 root root 10 Dec 22 15:08
scsi-3600a098000a9d1b9000000cb5805168d -> ../../dm-2
lrwxrwxrwx 1 root root 10 Dec 22 15:08
wwn-0x600a098000a9d1b9000000cb5805168d -> ../../dm-2
```

## 3.3.4 RAID Disk Space Requirements

The SMW, the boot node, and the SDB node all use space on the boot RAID. Here are the recommended sizes for the RAID LUNs, or LVM volume groups, based on the file systems for each. This information will be needed to bootstrap the SMW installation, which is next in the installation process.

### SMW File Systems

On the boot RAID, the LVM volume group for the SMW will have the file systems listed in this table in the Mount Point column. The third column shows the recommended LUN size for each file system assuming a standard 4.5 TB RAID. For sites with storage constraints or extra storage, the fourth and fifth columns show suggested LUN sizes.

**IMPORTANT:** The volume for the `/var/opt/cray/imps` file system on the SMW should be significantly larger than the volume for the `/var/opt/cray/imps` file system on the boot node. This is because that file system on the SMW contains boot images, config sets, and image roots, while that file system on the boot node contains only a subset of the image roots on the SMW. The boot node does an NFS mount of the SMW boot images, so no local space is needed for those.

*Table 8. SMW RAID Requirements*

| Mount Point | FS Type | Recommended Size for 4.5 TB RAID | Minimum Size | Size for 9.0 TB RAID | Description |
|---|---|---|---|---|---|
| `/home` | xfs | 200 GB | 40 GB | 200 GB | Home directories on SMW |
| `/var/lib/mysql` | btrfs | 10 GB | 10 GB | 10 GB | HSS database |
| `/var/opt/cray/disk/1` | xfs | 1000 GB | 400 GB | 2000 GB | logs, debug, dumps |
| `/var/opt/cray/imps` | xfs | 1000 GB | 400 GB | 1000 GB | IMPS data |
| `/var/opt/cray/repos` | btrfs | 200 GB | 100 GB | 200 GB | IMPS repos |

## CLE File Systems

On the boot RAID, storage for the boot node and SDB node is defined in the CLE storage set. Within that storage set, storage for the boot node is in the boot node LVM volume group, and storage for the SDB node is in the SDB node LVM volume group. The file systems for those nodes are listed in the tables below in the Mount Point column. The fourth column shows the recommended LUN size for each file system assuming a standard 4.5 TB RAID. For sites with storage constraints, the fifth column shows suggested LUN sizes.

Note that for partitioned systems, the requirements for LUN size apply to the boot node and SDB node in each partition.

**Expanding storage space**. The LUN sizes for the `/cray_home` and `/non_volatile` file systems may need to be adjusted depending on site usage of those file systems. For example, workload managers, DataWarp, and any node that needs permanent storage can store information in `/non_volatile`, so it may need to be larger than the suggested size. If size adjustment is not made at install time, it can be made later. See *XC™ Series System Administration Guide* (S-2393) for instructions on how to expand storage in a file system, volume, or volume group.

*Table 9. Boot Node RAID Requirements*

| Owning Node | Mount Point | FS Type | Recommended Size for 4.5 TB RAID | Minimum Size | Description |
|---|---|---|---|---|---|
| boot | `/cray_home` | xfs | 50 GB | 50 GB | Home directories on CLE |
| boot | `/var/opt/cray/imps` | btrfs | 250 GB | 250 GB | IMPS data for PE image roots and for Netroot compute-large and login-large image roots |
| boot | `/non_volatile` | xfs | 200 GB | 50 GB | persistent data, including `/var` if necessary, for service nodes provided from boot node |

*Table 10. SDB Node RAID Requirements*

| Owning Node | Mount Point | FS Type | Recommended Size for 4.5 TB RAID | Minimum Size | Description |
|---|---|---|---|---|---|
| sdb | `/var/lib/mysql` | xfs | 20 GB | 10 GB | SDB database |
| sdb | `/alps_shared` | xfs | 20 GB | 10 GB | ALPS data |

## 3.3.5 Bootstrap the SMW Installation

### Prerequisites

The following information must be gathered before running the installer in bootstrap mode. To find the persistent devices names for these devices, see *Determine the Persistent Device Name for a LUN* on page 81. For typical file system sizes, see *RAID Disk Space Requirements* on page 84.

- Disk devices on the boot RAID that can be used for boot node persistent storage
- Disk devices on the boot RAID that can be used for SDB node persistent storage
- Disk devices on the boot RAID that can be used for SMW persistent storage
- Size of file systems to be created within LVM volumes within LVM volume groups

  **NOTE:** Check that these file system sizes do not exceed the total size of the volume group containing them. Adjust file system sizes, if needed.

### About this task

This procedure runs `SMWinstall` in bootstrap mode, which installs IMPS and Ansible on the SMW, along with some of the global configuration templates. The `SMWinstall` command also invokes the configurator to prepare the storage set configuration. The configurator initiates an interactive session to gather the necessary information, unless the storage configuration template is supplied as a command-line argument, in which case no interactive session is needed. This configuration can be updated later by running the configurator manually.

### Procedure

1. Start the multipath daemon.

   Start multipathd but do not enable it on the command line. The multipath service needs to be *started* so that it can display path information needed for some config set settings, but the multipath service must not be *enabled* at this point in the process. Because of a SLES bug involving multipath and swap, the multipath service must not be enabled before the first time the new snapshot is booted. After that snapshot is booted for the first time, the Ansible multipath play will enable and start multipathd and will create an `/etc/multipath.conf` file. This file will ensure that on subsequent restarts of multipath or reboots of the SMW, multipathd will do the right thing and ignore swap.

   ```
   smw# systemctl start multipathd
   ```

2. Install in bootstrap mode.

   - Method 1 (most common): Provide storage configuration information interactively.

     ```
     smw# /media/SMW/SMWinstall --mode bootstrap
     ```

   - Method 2 (less common): Provide storage configuration information using an existing storage configuration **template** (the `_config.yaml`, not the `_worksheet.yaml`). This method can be used only if the `cray_bootraid_config.yaml` file is already available—obtained from a previous installation of the same release (a reinstallation) or from a different, similarly configured SMW.

```
smw# /media/SMW/SMWinstall --mode bootstrap --storage-config \
/path/to/cray_bootraid_config.yaml
```

**Trouble?** If ERROR and WARNING messages appear shortly after running the installer with the `--storage-config` option, and they complain of template syntax and/or schema errors, first check to see if the right file was provided in the command line. It must be the template (a `_config.yaml` file, also known as the *config file*), NOT the worksheet (a `_worksheet.yaml` file). Note that this contrasts with the way the `cfgset` command works: when configuration information is provided using `cfgset` with the `-w` or `--worksheet-path` option, the file provided on the command line must be a worksheet.

If Method 1 used, continue to step *3* on page 87. If Method 2 used, skip to step *12* on page 92.

ENABLE THE STORAGE SERVICE

**3.** Ensure that `cray_bootraid.enabled` is set to `true` to enable the storage service.

Configurator navigation hints:

- To get context-sensitive command help, enter **?**.
- To add a single value, enter the data and press **Enter**.
- To add a list, enter **+**, enter each list item on its own line, and press **Ctrl-d** when finished entering the entire list of items.
- To correct an error in a previous setting, press the **<** key to go back to the previous setting, correct it, then continue forward. Use **<** to back up several settings, if needed.

CONFIGURE THE CLE DEFAULT STORAGE SET (`cledefault`) VOLUME GROUPS

The configurator now shows the settings for a `storage_set` entry named `cledefault`, within which are three `volume_groups` entries:

- `boot_node_vg`
- `sdb_node_vg`
- `compute_node_local`

The full name of settings within each volume group looks like `cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.` followed by *`<volume group name>.<field name>`*. For brevity, the next steps show only the volume group name and field name of each setting.

**4.** Configure the boot node volume group (`boot_node_vg`).

a. Set the owner of the boot node volume group.

Ensure that `boot_node_vg.owner` is set to "boot" rather than a cname. For a partitioned system, include the partition name (e.g., "boot-p2" for partition p2).

b. Add entries for the physical volumes (disk devices) that are going to be part of the boot node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for `boot_node_vg.devices` to enter list entry mode. Add persistent device names such as `/dev/disk/by-id/scsi-360080e50002f7160000014905640c0c4` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

c. For each volume of the boot node volume group, change file system size to match the recommended values in the Boot Node RAID Requirements table in *RAID Disk Space Requirements* on page 84.

The `home` volume corresponds to the `/cray_home` file system in the table, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `nvolatile` volume corresponds to `/non_volatile`.

At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $
```

| To get to this setting | enter this at prompt |
|---|---|
| `boot_node_vg.volumes.home.fs_size` | **`1d*`** |
| `boot_node_vg.volumes.imps.fs_size` | **`2d*`** |
| `boot_node_vg.volumes.nvolatile.fs_size` | **`3d*`** |

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

d. When done with the last volume, press **Enter** to set the `boot_node_vg` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

**5.** Configure the SDB node volume group (`sdb_node_vg.owner`).

a. Set the owner of the SDB node volume group.

Ensure that `sdb_node_vg.owner` is set to "sdb" rather than a cname. For a partitioned system, include the partition name (e.g., "sdb-p2" for partition p2).

b. Add entries for the physical volumes (disk devices) that are going to be part of the SDB node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for `sdb_node_vg.devices` to enter list entry mode. Add persistent device names such as `/dev/disk/by-id/scsi-360080e50002f7160000014925640c108` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

c. For each volume of the SDB node volume group, change file system size to match the recommended values in the SDB Node RAID Requirements table in *RAID Disk Space Requirements* on page 84.

The `db` volume corresponds to the `/var/lib/mysql` file system in the table, and the `alps` volume corresponds to `/alps_shared`.

At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.sdb_node_vg.volumes
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $
```

| To get to this setting | enter this at prompt |
|---|---|
| sdb_node_vg.volumes.db.fs_size | **1d*** |
| sdb_node_vg.volumes.alps.fs_size | **2d*** |

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

d. When done with the last volume, press **Enter** to set the sdb_node_vg "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.sdb_node_vg.volumes
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
```

**6.** (Only for systems using compute nodes with SSDs) Change the compute node volume group (compute_node_local), as needed.

This is the third of three predefined volume groups in the cledefault storage set. It is needed only for systems using compute nodes with on-board SSDs. For all other systems, skip this step and go to step *7* on page 89.

a. Set the owner of the compute node volume group.

Set compute_node_local.owner to "compute."

b. Add entries for the physical volumes (disk devices) that are going to be part of the compute node LVM volume group.

This setting is a list. To add list data, enter **+** at the prompt for compute_node_local.devices to enter list entry mode. Add this entry to the compute_node_local.devices list:

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_nod
e_local.devices:
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add devices (Ctrl-d to exit) $ 'select: nvme0n1'
```

Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

c. When done with the last volume, press **Enter** to set the compute_node_local "volumes" entries, or add another volume, as needed for this site.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

After completing this step, skip the next step.

**7.** (Only for systems NOT using compute nodes with SSDs) Set only the compute node volume group (compute_node_local) owner setting.

Perform this step only if this system does not use compute nodes with on-board SSDs. Set the owner field to null to ensure that this volume group is not used but is preserved in case this site decides to add compute nodes with SSDs to the system later.

Use the **>** key to skip the other compute_node_local settings when presented.

```
compute_node_local.owner: null
compute_node_local.devices: >
```

Set the compute_node_local "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.compute_node_local.volumes
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
```

8. Set the `cledefault` "volume groups" entries.

   Review the list of `cledefault` volume groups (enter ***** to see the full list if not all volume groups are displayed), then at the prompt below, enter press **Enter** to set the entries.

   ```
   cray_bootraid.settings.storage_sets.data.cledefault.volume_groups
   [<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <cr>
   ```

### CONFIGURE THE SMW DEFAULT STORAGE SET (`smwdefault`) VOLUME GROUPS

The configurator now shows the settings for a `storage_set` entry named `smwdefault`, within which is one `volume_groups` entry: `smw_node_vg`

The full name of settings within each volume group looks like `cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.` followed by *`<volume group name>.<field name>`*. For brevity, the next steps show only the volume group name and field name of each setting.

9. Configure the SMW node volume group (`smw_node_vg`).

   a. Set the owner of the SMW node volume group to `smw`.

      Ensure that `smw_node_vg.owner` is set to "smw" to maximize portability of the config set.

      ```
      cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.smw_node_vg.owner
      [<cr>=keep 'smw', <new value>, ?=help, @=less] $ <cr>
      ```

   b. Add entries for the physical volumes (disk devices) that are going to be part of the SMW node LVM volume group.

      This setting is a list. To add list data, enter **+** at the prompt for `smw_node_vg.devices` to enter list entry mode. Add persistent device names such as `/dev/disk/by-id/scsi-360080e50002f889c00000a0654e32232` for each physical volume. Do not use short names like `/dev/sdn`, which may vary from node to node (SMW, boot node, and SDB node), and may vary from boot to boot of a particular node.

      Press **Enter** after each list entry, and when done adding entries, press **Ctrl-d** to exit list entry mode. Remember to press **Enter** again to set the list entries.

   c. For each volume of the SMW node volume group, change file system size to match the recommended values in the SMW RAID Requirements table in *RAID Disk Space Requirements* on page 84.

      The `home` volume corresponds to the `/home` file system in the table, the `db` volume corresponds to `/var/lib/mysql`, the `log` volume corresponds to `/var/opt/cray/disk/1`, the `imps` volume corresponds to `/var/opt/cray/imps`, and the `repos` volume corresponds to `/var/opt/cray/repos`.

      At the following prompt, enter these values to get to the `fs_size` setting for each of the file systems.

      ```
      cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.smw_node_vg.volumes
      [<cr>=set 5 entries, +=add an entry, ?=help, @=less] $
      ```

| To get to this setting | enter this at prompt |
|---|---|
| `smw_node_vg.volumes.home.fs_size` | **1d*** |
| `smw_node_vg.volumes.db.fs_size` | **2d*** |

| To get to this setting | enter this at prompt |
|---|---|
| `smw_node_vg.volumes.log.fs_size` | **3d*** |
| `smw_node_vg.volumes.imps.fs_size` | **4d*** |
| `smw_node_vg.volumes.repos.fs_size` | **5d*** |

Then at the prompt for that setting, enter a new file system size to change the value, if needed. Accept the current or newly entered value by pressing **Enter**.

d. When done with the last volume, press **Enter** to set those `smw_node_vg` "volumes" entries.

```
cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups.smw_node_vg.volumes
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ <cr>
```

**10.** Set the `smwdefault` "volume groups" entries.

Review the list of `smwdefault` volume groups, then at the prompt below, enter press **Enter** to set the entries.

```
cray_bootraid.settings.storage_sets.data.smwdefault.volume_groups
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

**11.** Set the boot RAID "storage sets" entries.

Review the storage sets. Press **Enter** (<cr>) to set the `cledefault` and `smwdefault` storage sets, unless this system has partitions. If configuring a partitioned system, enter **+** to add another CLE storage set. A separate storage set is needed for each partition.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $
```

**Trouble?** If `SMWinstall` fails during the installation, it is because `cfgset` failed, which was invoked by `SMWinstall` to gather configuration information. That failure may be due to missing information. Do not try running `SMWinstall --mode bootstrap` again.

Try one of these options instead:

**Option** — **Description**

**Run the configurator manually**

1. Enable the `cfgset` command.

```
smw# . /opt/modules/default/etc/modules.sh
smw# module use /opt/cray/ari/modulefiles
smw# module load imps
```

2. Use `cfgset` to invoke the configurator in interactive mode to make any needed changes to the cray_bootraid configuration service in the global config set.

```
smw# cfgset update -m interactive -s cray_bootraid global
```

3. Run the installer in bootstrap mode again.

```
smw# /media/SMW/SMWinstall --mode bootstrap
```

**Run `SMWinstall` with the reconfigure option**

1. Run `SMWinstall` in bootstrap mode with the reconfigure option, which invokes the configurator in interactive mode.

```
smw# /media/SMW/SMWinstall --mode bootstrap --reconfigure
```

**12.** Display cray_bootraid information.

```
smw# . /opt/modules/default/etc/modules.sh
smw# module use /opt/cray/ari/modulefiles
smw# module load imps
smw# cfgset search -s cray_bootraid -l basic global
smw# cfgset search -s cray_bootraid -l advanced global
```

**13.** (SMW HA only) Copy the storage configuration template.

If this is the primary/first SMW installed of an SMW HA pair, save the storage configuration template to another system not on this SMW for fast and consistent system bootstrapping when installing the secondary SMW.

```
smw# scp -p /var/opt/cray/imps/config/sets/global/config/\
cray_bootraid_config.yaml user@host:~/
```

Note that it is the **template** (the _config.yaml file), not the worksheet (the _worksheet.yaml file) that must be copied.

**14.** Remove existing volume groups, as needed.

If doing a fresh install onto a system, and there is a desire to reuse the storage in any existing LVM volume groups for SMW, boot node, and SDB node, then run these commands to remove the volume groups with storage to be reused.

a. Use cfgset search to find the names of all of the volume groups defined in the storage configuration template.

```
smw# cfgset search -s cray_bootraid global |awk -F'.' '{print $7}' |sort -u
boot_node_vg
boot_test_vg
sdb_node_vg
sdb_test_vg
smw_node_vg
smw_test_vg
```

b. Display the volume groups that exist.

```
smw# vgdisplay
```

Alternative (more concise):

```
smw# vgs
```

c. Remove the volume groups with storage to be reused (in this example, the test volume groups).

```
smw# vgremove -f smw_test_vg
smw# vgremove -f boot_test_vg
smw# vgremove -f sdb_test_vg
```

The system is now ready for the provisioning of boot RAID LVM volumes.

## 3.3.6 Provision SMW Storage

### About this task

The provision-storage mode of `SMWinstall` can be run at any time. It uses the boot RAID configuration template (cray_bootraid_config.yaml) to provision persistent storage on the boot RAID by creating LVM volume groups and LVM volumes. This is a non-interactive procedure if `--mode bootstrap` was used to bootstrap the installation earlier in the process. Otherwise, it will gather the necessary site-specific configuration information interactively.

### Procedure

1. Provision storage for the default SMW storage set.

   Use this command only if using an SMW storage set called "smwdefault," which is the default.

   ```
   smw# /media/SMW/SMWinstall --mode=provision-storage
   ```

   If no errors reported, proceed to step *2* on page 93.

   **Trouble?** If errors are reported, review the boot RAID configuration settings using one of these methods. Both methods run the installer in provision-storage mode again after reviewing the settings and making changes. Note that when the installer is run again, it will ask ALL storage configuration questions, and the defaults will be prefilled with existing data.

   - Error recovery method 1: Modify using the configurator, then run installer again.

     ```
     smw# cfgset update -s cray_bootraid -m interactive global

     smw# /media/SMW/SMWinstall --mode=provision-storage
     ```

   - Error recovery method 2: Modify manually, then run installer again.

     ```
     smw# vi /var/opt/cray/imps/config/sets/global/config/\
     cray_bootraid_config.yaml

     smw# /media/SMW/SMWinstall --mode=provision-storage
     ```

2. View the new volumes.

   ```
   smw# lvs
   ```

When the provision-storage installer mode completes successfully, the system is ready for the installation of SMW and CLE software.

## 3.3.7 Run the Installer for an Initial Installation

### Prerequisites

This procedure assumes that ISOS for SLES 12 and CentOS 6.5 have been downloaded as described in *Prepare to Bootstrap the SMW Installation* on page 79 and SMW storage has been successfully configured.

## About this task

This procedure installs SMW and CLE software together to ensure that there is a matched set of software and configuration.

> **NOTE:** Do NOT run the installer from the `/root/isos` directory. Instead, run it from a directory that is not included in any snapshot, such as `/var/adm/cray/release`.

## Procedure

1. Set variable for snapshot name.

   Setting a variable here enables better command substitution in later commands dealing with snapshots. The SMW will boot from this snapshot.

   ```
   smw# ls -Llst /root/isos
   smw# export SNAPSHOT=smw-${SMW_RELEASE}_cle-${CLE_RELEASE}.${TODAY}
   smw# echo $SNAPSHOT
   ```

   (SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

2. Install SMW and CLE software and security updates together.

   It is possible to install both SMW media and CLE media with a single command to create a unified "release" that is tagged as a snapshot on the SMW system. Run the `SMWinstall` program and tell it where the CLE media is. This invocation creates the "target" snapshot, which was named in step 1, and then installs into that target snapshot (note that in the absence of an existing target snapshot, the installer creates one from the current running snapshot by default). The installer assumes that all of the SLES 12 ISOs are in `/root/isos`.

   > **IMPORTANT:** The SLE media must be specified before the CLE media on the command line so that SUSE security updates are installed before the CLE software is installed.

   ```
   smw# /media/SMW/SMWinstall \
   --plus-media=/root/isos/${SLE_SOFTWARE}.iso \
   --plus-media=/root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
   --target=${SNAPSHOT}
   ```

   It can take from 10 to 25 minutes to run a combined installation of SMW, CLE, and security updates for the first time on the SMW. The output of `SMWinstall` provides several command hints, including these three:

   **snaputil default** The first command hint (snaputil default) is used to ensure that the SMW is booted from the correct (new) snapshot, which is essential to a successful reboot.

   **snaputil chroot** The second command hint (snaputil chroot) is used in the software update process and may be used at other times to look around inside the snapshot.

   **snaputil delete** The third command hint (snaputil delete) should be used only if this site needs to remove the newly created snapshot for any reason.

   Logs will be in `/var/adm/cray/logs/install` for each invocation of `SMWinstall`.

3. Check new snapshot software versions.

   When `SMWinstall` completes, check the snapshot details for the expected SMW and CLE release versions (note that the actual output of this command will show different release versions than this example output).

```
smw# /media/SMW/snaputil show ${SNAPSHOT}
active_maps      :
    p0:/var/opt/cray/imps/config/sets/global/nims/maps/p0
boot menu        : False
booted           : True
btrfs_object_id : 365
cle_version      : 6.0.96
created          : 2016-05-23 11:28:01
default          : True
initrd           : initrd-3.12.51-52.39-default
kernel           : vmlinuz-3.12.51-52.39-default
name             : smw-8.0.96_cle-6.0.96.20160523
parent           : @
path             : /media/root-sv/snapshots/smw-8.0.96_cle-6.0.96.20160523
read-only        : False
smw_version      : 8.0.96
smwha_version    : None
storage_set      : smwdefault
subvolumes       :
    /var/lib/mysql:smw-8.0.96_cle-6.0.96.20160523
    /var/opt/cray/repos:smw-8.0.96_cle-6.0.96.20160523
total size       : 27658.24 MB
unshared size    : 1505.28 MB
updated          : 2016-05-23 11:51:01.188064
```

The SMW is now ready to reboot, which starts with setting the default snapshot to boot from. Trying to boot the SMW without first setting the default snapshot will result in an unbootable SMW.

## 3.3.8 Set Default Snapshot and Boot the SMW

### Prerequisites

This procedure assumes that the snapshot variable has been set and the SMW and CLE software has been installed.

### About this task

When the `SMWinstall` command was invoked in the previous procedure, it provided several suggested `snaputil` commands. The one used in this procedure ensures that the snapshot target is set as the default snapshot for the next boot of the SMW.

### Procedure

1. Set the release snapshot as the default.

   **IMPORTANT:** Do not skip this step. If the SMW is rebooted without first setting the default snapshot, the SMW becomes unbootable.

   ```
   smw# /media/SMW/snaputil default ${SNAPSHOT}
   ```

2. Verify that the correct snapshot is the default.

   ```
   smw# /media/SMW/snaputil list
   ```

3. Reboot the SMW to switch to the new release.

```
smw# reboot
```

> **ATTENTION:** After the SMW has completed booting, wait 10 to 15 minutes before logging in, which allows time for all startup tasks to complete. Logging in too soon can result in SSH connections being dropped.

## 3.4 Configure SMW for CLE System Hardware during a Fresh Install

In this part of the installation and configuration process it is necessary to initialize the power management database, discover hardware, create a config set, and check the status of all SMW components.

Use *Installation Checklist 3: Configure SMW for CLE Hardware during a Fresh Install* on page 363 to track progress while performing the following procedures.

1. *Set or Change the HSS Data Store (MariaDB) Root Password* on page 96

   > **NOTE:** This procedure comes before "Start a Typescript File" so that root password information is not captured in the typescript. If this site wishes to swap the order of these two procedures so that everything is captured in the typescript, consider changing the permissions on the typescript file so that only root users can access it.

2. *Start a Typescript File* on page 78
3. *Make a Post-install Snapshot using snaputil* on page 98
4. *Update install.cle.conf for Software Updates* on page 99
5. *Prepare and Update the Global Config Set* on page 99
6. *Prepare the CLE Configuration Worksheets* on page 106
7. *Bootstrap Hardware Discovery* on page 107
8. *Discover Hardware and HSN Routing, Prepare STONITH* on page 109
9. *Update Firmware* on page 111
10. *(Optional) Configure Partitions* on page 112
11. *Repurpose Compute Nodes* on page 113
12. *Finish Configuring the SMW for the CLE System Hardware* on page 114

### 3.4.1 Set or Change the HSS Data Store (MariaDB) Root Password

### About this task

The method for setting or changing the HSS data store (database) root password has changed with the release of CLE 6.0. By default, MariaDB is installed with no password set up for the root account. Cray strongly recommends adding a password as part of the fresh install procedure.

**Old** The HSS database was implemented with MySQL. After initial installation, its root password was changed from the initial default empty string to a user-defined value by the `SMWconfig` script, which was run after `SMWinstall` and the initial discovery of the system.

**New** The HSS database is now implemented with MariaDB, a MySQL work-alike database with identically named commands. As before, the initial default root password is the empty string; however, the `SMWconfig` script is no longer used to set it after installation. The administrator must use the following procedure to set the root password to a user-defined value.

After the MariaDB root password has been set, it must be placed in `/root/.my.cnf`, a file readable only by root that has the format shown in step 2. This file is the mechanism by which the installer and the `snaputil` command obtain the root password when they access MariaDB as root. If the file does not exist or it has no `password=` line, the system will attempt to access MariaDB using the default empty-string password, which will fail once the password has been changed.

- Create `/root/.my.cnf` the first time the root password is set to a user-defined value.
- Update `/root/.my.cnf` to match the MariaDB root password whenever it is changed.

**IMPORTANT:** For an SMW HA system, record the new MySQL root password. It will need to be changed on the second SMW later (by editing `/root/.my.cnf`). After the SMW HA cluster has been configured, the MySQL root password needs to be reset with `mysqladmin` on only one SMW, because the MySQL database is shared between both SMWs in the HA cluster.

## Procedure

1. Set or change the MariaDB root password.

   ```
   smw# mysqladmin -uroot password -p
   ```

   Do one of the following at the prompt:

   - To **set** the root password for fresh installs or after the database has been reinitialized, press **Enter** to enter an empty string, the default initial password.

     ```
     Enter password: <cr>
     ```

   - To **change** the root password, enter the existing password.

     ```
     Enter password: existing_password
     ```

   At these prompts, enter the new root password, and then enter it again.

   ```
   New password:
   Confirm new password:
   ```

2. Ensure that the root password in the `/root/.my.cnf` file matches the new root password.

   If this file does not yet exist, create it and add the lines shown in the example, substituting the new password for the placeholder *`MariaDB-password`*.

   ```
   smw# vi /root/.my.cnf
   [client]
   user=root
   password=MariaDB-password
   ```

3. Ensure that only root can see or write to the `/root/.my.cnf` file.

   ```
   smw# chmod 600 /root/.my.cnf
   ```

## 3.4.2 Start a Typescript File

### About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before installing a new software release
- just before configuring the newly installed software

### Procedure

1. Log in as root to the SMW.
2. (First time only) Create a release directory for the typescript file.

   ```
   smw# mkdir -p /var/adm/cray/release
   ```

3. Change to the release directory.

   ```
   smw# cd /var/adm/cray/release
   ```

4. Set a variable equal to today's date.

   ```
   smw# export TODAY=`date +%Y%m%d`
   smw# echo $TODAY
   ```

5. Start a typescript file.

   ```
   smw# script -af ${TODAY}.suffix
   ```

   For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

   ```
   smw# PS1="\u@\h:\w \t# "
   ```

## 3.4.3 Make a Post-install Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

## Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the SNAPSHOT environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

   Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

   (SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postinstall
```

# 3.4.4 Update install.cle.conf for Software Updates

## Prerequisites

This procedure assumes that the installer will not be run again at this point in the installation and configuration process.

## About this task

The /var/adm/cray/install.cle.conf file contains configuration that controls the image building behavior of the installer. Changing this file now will make later updates of CLE software easier.

## Procedure

1. Edit the configuration file.

```
smw# vi /var/adm/cray/install.cle.conf
```

2. (For all systems) Change build_images to yes to enable the CLE installer to build IMPS images as part of the install process. The remaining options determine what to do if build_images is set to yes.

```
build_images: yes
```

3. (For partitioned systems only) Uncomment the map_partition line and specify the system partitions.

```
map_partition:['p1', 'p2']
```

### 3.4.5 Prepare and Update the Global Config Set

## Prerequisites

This procedure assumes that the SMW and CLE software has been installed so that the global config set is present.

## About this task

The global config set must be updated with site-specific information about several services. This procedure describes how to add site configuration data to the configuration worksheets for each service in the global config set, update the config set with the edited configuration worksheets, and then run Ansible plays on the SMW to effect the changes.

When editing configuration worksheets, a general rule is to uncomment all settings that are marked level=basic and modify values as needed. All settings that remain commented are considered unconfigured. Some settings are already uncommented in the original worksheet; Cray recommends not modifying those preconfigured settings because they are needed for proper configuration of the system. For more information about editing configuration worksheets and updating config sets, see *XC™ Series Configurator User Guide* (S-2560).

> **NOTE:** (SMW HA only) For SMW HA systems, the following procedures are done only on the first SMW because the config sets are shared between both SMWs in the HA cluster. In contrast, Ansible plays must be run on each SMW.

## Procedure

1. Save a copy of original global worksheets.

   Copy the original configuration worksheets into a new directory to preserve them in case they are needed later for comparison.

   ```
   smw# ls -l /var/opt/cray/imps/config/sets/global/worksheets

   smw# cp -a /var/opt/cray/imps/config/sets/global/worksheets \
   /var/opt/cray/imps/config/sets/global/worksheets.orig
   ```

2. Make a work area for global worksheets.

   a. Copy the global configuration worksheets to a new work area for editing.

      The worksheets should not be edited in their original location for two reasons: (1) the configurator will not permit updating a config set from worksheets within that config set, and (2) edits would be overwritten when the config set is updated.

      ```
      smw# cp -a /var/opt/cray/imps/config/sets/global/worksheets \
      /var/adm/cray/release/global_worksheet_workarea
      ```

   b. Change to the work area directory to simplify the editing commands in the following steps.

      ```
      smw# cd /var/adm/cray/release/global_worksheet_workarea
      ```

UPDATE WORKSHEETS FOR GLOBAL SERVICES

**3.** Update cray_firewall.

a. Edit `cray_firewall_worksheet.yaml`.

```
smw# vi cray_firewall_worksheet.yaml
```

b. Uncomment `cray_firewall.enabled` and set it to `true`.

**4.** Update cray_global_net.

a. Edit `cray_global_net_worksheet.yaml`.

```
smw# vi cray_global_net_worksheet.yaml
```

b. Uncomment `cray_global_net.enabled` and ensure that it is set to `true`.

c. Search in the file for `'networks' DATA`, then uncomment all of the lines below it that begin with `cray_global_net.settings.networks` so that those settings will be applied and marked as configured. They define four networks: "admin," "SMW failover," "HSS," and "management."

> **NOTE:** Do NOT uncomment the similar lines under this heading, because they are examples only and are not configured for these four networks.

```
# ** EXAMPLE 'networks' VALUE (with current defaults) **
```

d. Enter SMW-specific or site-specific values for these management network items.

```
cray_global_net.settings.networks.data.management.ipv4_network:
cray_global_net.settings.networks.data.management.ipv4_netmask:
cray_global_net.settings.networks.data.management.ipv4_gateway:
cray_global_net.settings.networks.data.management.dns_servers:
cray_global_net.settings.networks.data.management.dns_search:
cray_global_net.settings.networks.data.management.ntp_servers:
```

Notes:

- The DNS information for `dns_servers` and `dns_search` was entered during the SLES12 installation, so those values can be found in `/etc/resolv.conf`.

  > **NOTE:** If this site does not use DNS search but does use DNS domain in `/etc/resolv.conf`, then adding a single entry to the `dns_search` setting is functionally equivalent to setting the DNS domain.

- If entering a value for a string setting that currently is set to `''` (empty string), remove the quotes before entering the new value. For example, `ipv4_network: ''` becomes `ipv4_network: 10.1.0.0`. In cases where the string value might be interpreted as a number, retain the single quotes. For example, a string setting with value '512' needs quotes.
- If entering one or more values for a list setting that is currently set to `[]` (empty list), remove the brackets and add each entry on a separate line, beginning with "`- `" (a hyphen and a space). For example, the `dns_servers`, `dns_search`, and `ntp_servers` settings are lists that can have multiple entries, and they should look like this:

```
cray_global_net.settings.networks.data.management.dns_servers:
- 172.31.84.40
- 172.30.84.40
- 172.28.84.40
```

- Do NOT change or remove the null value in lines like this that appear at the beginning of each set of network definitions. This line sets the key for that network definition, which in this example is "management."

  ```
  cray_global_net.settings.networks.data.name.management: null
  ```

e. Set the management network external firewall to true.

```
cray_global_net.settings.networks.data.management.fw_external: true
```

f. Search in the file for `'hosts' DATA`, then uncomment all of the lines that begin with `cray_global_net.settings.hosts` so that those settings will be applied and marked as configured. They define a host called "primary_smw" and two interfaces for it: one that connects to the customer management network and one that connects to admin nodes, such as the boot and SDB nodes.

g. Enter SMW-specific or site-specific values for these items.

See step 4d for information about how to change empty string and empty list values.

```
cray_global_net.settings.hosts.data.primary_smw.aliases:
cray_global_net.settings.hosts.data.primary_smw.hostid:
cray_global_net.settings.hosts.data.primary_smw.hostname:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_address:
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.module: ''
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.params: ''
```

Note that if the customer Ethernet IP address changes, the output from the `hostid` command will be different. After changing the Ethernet setting

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_addr
ess
```

ensure that this setting (the SMW host ID) is set to the output of the `hostid` command.

```
cray_global_net.settings.hosts.data.primary_smw.hostid
```

h. Set the `unmanaged_interface` field of the `customer_ethernet` and `admin_interface` interface settings to `true`.

This applies to both stand-alone SMWs and SMW HA systems. In the case of an SMW that is or will be configured for an SMW HA system, this prevents Ansible from managing eth0 and eth3 before the SMW HA cluster has been configured.

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.unmanaged_interface:
 true
...
cray_global_net.settings.hosts.data.primary_smw.interfaces.admin_interface.unmanaged_interface:
true
```

**5.** Update cray_ipforward.

a. Edit `cray_ipforward_worksheet.yaml`.

```
smw# vi cray_ipforward_worksheet.yaml
```

b. Uncomment `cray_ipforward.enabled` and ensure that it is set to `true`.

**6.** Update cray_liveupdates.

a. Edit `cray_liveupdates_worksheet.yaml`.

```
smw# vi cray_liveupdates_worksheet.yaml
```

b. Uncomment cray_liveupdates.enabled and ensure that it is set to true.

7. Update cray_logging.

a. Edit cray_logging_worksheet.yaml.

```
smw# vi cray_logging_worksheet.yaml
```

b. Uncomment cray_logging.enabled and ensure that it is set to true.

c. Uncomment cray_logging.settings.global_options.data.raid and change its value if the boot RAID has a non-standard IP address.

d. Uncomment cray_logging.settings.site_loghost.data.name and change its value if the site has a site_loghost.

8. Update cray_multipath.

Multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

a. Edit cray_multipath_worksheet.yaml.

```
smw# vi cray_multipath_worksheet.yaml
```

b. Choose one of the following options, depending on whether this site intends to use multipath.

**NOTE:** (SMW HA only) Cray recommends configuring multipath before configuring and enabling HA. If HA is configured and enabled first, then additional precautions must be taken when enabling multipath, as documented in *XC™ Series SMW HA Installation Guide*.

**Will multipath be used?**

If no, then uncomment cray_multipath.enabled and ensure that it is set to false. There is nothing else to configure in this step; proceed to step *9* on page 104.

If yes, then uncomment cray_multipath.enabled and set it to true. Continue with the following substeps.

c. Enter the list of multipath nodes.

Uncomment cray_multipath.settings.multipath.data.node_list, remove the [] (denotes empty list), and add a list of nodes (by cname or host ID) in this system that have multipath devices and need to have multipath configured. For sites with boot node failover and/or SDB node failover, Cray recommends adding both the active and passive (failover) nodes to this list.

This example shows a list of three nodes: an SMW with host ID 1eac4e0c, a boot node with cname c0-0c0s4n1, and an SDB node with cname c0-0c0s3n1.

```
cray_multipath.settings.multipath.data.node_list:
- 1eac4e0c
- c0-0c0s4n1
- c0-0c0s3n1
```

d. Configure enabled devices.

Cray has provided a number of enabled devices with pre-populated data under # ** 'enabled_devices' DATA **. These storage devices are the devices that will be whitelisted, which means they will be listed as exceptions to the blacklist. The settings for these devices have default values provided by the device vendors and do not need to be changed. If this site intends to configure a multipath device that does not appear in this group of enabled devices, contact a Cray representative for help.

e. (Optional) Configure aliases for the multipath devices.

This is the equivalent of adding aliases to the multipaths section of the `multipath.conf` file.

In the worksheet, copy the two lines below `# ** EXAMPLE 'aliases' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'aliases' setting entries here, if desired.`

```
# ** EXAMPLE 'aliases' VALUE  (with current defaults) **
#   cray_multipath.settings.aliases.data.wwid.sample_key_a: null   <-- setting a multival key
#   cray_multipath.settings.aliases.data.sample_key_a.alias: ''
#
```

Uncomment the lines, replace `sample_key_a` with the World Wide Identifier (WWID) of the device to be aliased (60080e50002e203c00002a085551b2c8 in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, add the alias for this device (smw_node_pv1 in this example). Repeat this substep for each device, as needed.

```
# NOTE: Place additional 'aliases' setting entries here, if desired.
cray_multipath.settings.aliases.data.wwid.60080e50002e203c00002a085551b2c8: null
cray_multipath.settings.aliases.data.60080e50002e203c00002a085551b2c8.alias: smw_node_pv1
#***************** END Service Setting: aliases ****************
```

f. Correct the default values for three pre-populated device settings.

The default values of the following variables are incorrect in `cray_multipath_worksheet.yaml` for this release. In the worksheet, find these variables and change their values as indicated.

```
enabled_devices.data.DDN_SFA12K_20.product: SFA12K-20

enabled_devices.data.DDN_SFA12K_40.product: SFA12K-40|SFA12KX*

enabled_devices.data.DDN_EF3015.path_grouping_policy: group_by_prio
```

**9.** Skip cray_network_boot_packages_worksheet.yaml.

The cray_network_boot_packages configuration service is enabled by default and has no variables that need to be changed.

**10.** Update cray_time.

a. Edit `cray_time_worksheet.yaml`.

```
smw# vi cray_time_worksheet.yaml
```

b. Uncomment `cray_time.enabled` and ensure that it is set to `true`.

c. Uncomment `cray_time.settings.service.data.timezone` and change its value, as needed.

UPLOAD WORKSHEETS AND UPDATE/VALIDATE GLOBAL CONFIG SET

**CAUTION: Boot failure possible if using `cfgset` under certain conditions.**

The `cfgset create` and `cfgset update` commands always call pre- and post-configuration scripts. Some of these scripts require HSS daemons and other CLE services to be running. This can cause problems under these conditions:

- If xtdiscover is running, cfgset may hang or produce incorrect data that can result in system boot failure.
- If xtbounce is in progress or if the SMW is not connected to XC hardware, cfgset will fail.

In these circumstances, use the --no-scripts option with cfgset create or cfgset update to avoid running the scripts. Because using that option results in an invalid config set, remember to run cfgset update without the --no-scripts option afterwards, when circumstances permit, to ensure that all pre- and post-configuration scripts are run.

**11.** Upload modified worksheets into global config set.

Note that the full filepath must be specified in this cfgset command, and it must be enclosed in single quotes (to prevent the shell trying to expand the file glob).

```
smw# cfgset update -w \
'/var/adm/cray/release/global_worksheet_workarea/*_worksheet.yaml' global
```

**12.** Update the global config set.

Using the configurator in interactive mode to update the global config set is a good way to check whether all required settings and basic settings have been configured for services that are enabled. If they have, then all enabled services will show OK status in the Service Configuration List Menu. If configuration of a basic setting was missed, then the menu will show how many unconfigured settings there are for each service. Set or change any settings from this menu, as needed.

```
smw# cfgset update -m interactive global
```

When the configurator session completes, it displays a message indicating the file name of the changelog file for this configuration session. The changelog is written to a file in the /var/opt/cray/imps/config/sets/global/changelog directory.

**13.** Validate the global config set.

```
smw# cfgset validate global
```

APPLY CONFIGURATION CHANGES ON THE SMW

**14.** Run Ansible plays on the SMW.

After the global config set has been updated, reapply any Ansible plays that consume global config set data.

**NOTE:** (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# /etc/init.d/cray-ansible start
```

Logs from running Ansible plays, such as cray-ansible, are stored on the SMW in /var/opt/cray/log/ansible.

CHECK TIME SETTINGS

**15.** Check for external NTP servers.

Check that external NTP servers have been set as desired in the global config set.

**NOTE:** (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# grep server /etc/ntp.conf
server ntpserver1 minpoll 4 iburst
server ntpserver2 minpoll 4 iburst
```

**16.** Put the SMW time zone setting where the cabinet and blade controllers can access it.

This SMW time zone setting will be applied to the cabinet and blade controllers when they are rebooted later in the process.

> **NOTE:** (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# cp -p /etc/localtime /opt/tftpboot/localtime
```

## 3.4.6 Prepare the CLE Configuration Worksheets

### About this task

The Cray XC system stores configuration information used to boot and customize the CLE system in the p0 config set, or if the system is partitioned, in config set p1 for partition p1 and config set p2 for partition p2, and so forth. This procedure prepares the CLE configuration worksheets, which are later edited to include site-specific configuration data.

> **NOTE:** (SMW HA only) For SMW HA systems, config set operations need to be performed on only one SMW because the config sets are shared between both SMWs in the SMW HA pair.

### Procedure

1. Obtain configuration worksheets for CLE from one of these sources.
   - Find them in the CLE 6.0.UP03 release directory available on CrayPort and extract them to `/var/opt/cray/imps/config/sets/p0_example/worksheets`.
   - Generate them by creating a CLE config set using prepare mode and the no-scripts option.

   ```
   smw# cfgset create -m prepare -t cle --no-scripts p0_example
   ```

2. Save a copy of original worksheets.

   Make a copy of the original CLE configuration worksheets directory to preserve the worksheets in case they are needed for comparison later.

   ```
   smw# ls -l /var/opt/cray/imps/config/sets/p0_example/worksheets

   smw# cp -a /var/opt/cray/imps/config/sets/p0_example/worksheets \
   /var/opt/cray/imps/config/sets/p0_example/worksheets.orig
   ```

3. Copy the CLE worksheets to a work area.

   Make a copy of the CLE configuration worksheets directory outside the config set to be used as a work area for editing. The worksheets should not be edited in their original location for two reasons: (1) the configurator will not permit updating a config set from worksheets within that config set, and (2) edits would be overwritten when the config set is updated.

> **REMEMBER:** For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Copy the CLE configuration worksheets to a separate work area for each partition.

```
smw# cp -a /var/opt/cray/imps/config/sets/p0_example/worksheets \
/var/adm/cray/release/p0_worksheet_workarea
```

These worksheets can be edited while the installation/configuration process continues with discovering hardware, updating firmware, and other hardware or HSS software activities.

- To edit the worksheets, see *Update CLE Configuration Worksheets* on page 116, but do not proceed to the task that creates the new CLE config set from the worksheets until hardware discovery and associated procedures are complete.
- To continue with hardware discovery, proceed to *Bootstrap Hardware Discovery* on page 107.

## 3.4.7 Bootstrap Hardware Discovery

### Prerequisites

This procedure assumes that the following information has been gathered. Enter this information in response to system prompts when performing this procedure.

| Information needed | Default value |
|---|---|
| maximum X cabinet size (columns) | There is no default value. Find the X and Y cabinet sizes and the network topology class from *Site-dependent Configuration Values* in *Configuration Values* on page 31. |
| maximum Y cabinet size (rows) | No default value. See above. |
| network topology class | 0 or 2 for Cray XC Series liquid-cooled systems, 0 for Cray XC Series air-cooled systems (XC30-AC, XC40-AC) |
| boot node name | c0-0c0s0n1 |
| sdb node name | c0-0c0s1n1 |

### About this task

This procedure uses the `xtdiscover --bootstrap` command to collect some basic information that will be used to bootstrap the hardware discovery process. If boot node failover or SDB node failover will be enabled, then when `xtdiscover` asks for the boot node or the SDB node, instead of entering a single node, enter a pair of nodes with a comma between them, for example "c0-0c0s0n1,c0-2c0s0n1." For more detailed information, see the `xtdiscover(8)` man page.

> **NOTE:** (SMW HA only) Hardware discovery is done only on the first SMW. Do not repeat hardware discovery on the second SMW.

**Trouble?**

- If the `xtdiscover --bootstrap` command is unable to power up the cabinets, try running `xtdiscover --testconfig` and then run `xtdiscover --bootstrap` again.

- If a step in this procedure fails because of a hardware issue, such as a cabinet failing to power up, resolve that issue and then go back to the last successful step in the procedure and continue from there. Do not skip steps or continue out of order.

## Procedure

1. Run `xtdiscover` in bootstrap mode.

   ```
   smw# xtdiscover --bootstrap
   ```

   The system prompts the user to enter the information gathered as a prerequisite to this procedure. Prior to powering on the cabinets, the system prompts the user to disable any blades that should not be powered on.

   ```
   xtdiscover is about to power on the cabinets.
   *** IF YOU NEED TO DISABLE COMPONENTS TO AVOID THEM
   *** BEING POWERED ON, PLEASE DO SO NOW USING 'xtcli disable'

   Please enter 'c' to continue, or 'a' or 'q' to abort [c]:
   ```

2. Disable any blades that should not be powered on.

   If there are any blades or other components to be disabled, open a separate window and disable them (as `crayadm`) there. In this command, replace *`cname`* with the cname of the component to be disabled.

   ```
   crayadm@smw> xtcli disable cname
   ```

3. Return to the `xtdiscover --bootstrap` window and enter **`c`** to continue the hardware discovery bootstrap.

   ```
   Please enter 'c' to continue, or 'a' or 'q' to abort [c]: c
   ```

   The `xtdiscover` command proceeds without further prompts.

   **Trouble?** If the `xtdiscover` command fails with the message, `The following cabinets were not detected by heartbeat`, power cycle the cabinet controller and retry the `xtdiscover --bootstrap` command.

4. Power down the system.

   ```
   smw# xtcli power down s0
   Turning off power to cabinet and waiting for confirmation...

              Component|        Flags:    Result
   --------------------------------------------------------------------
                   c0-0|    noflags|:    Success
               c0-0c0s0|    noflags|:    Success
               c0-0c0s1|    noflags|:    Success
               c0-0c0s2|    noflags|:    Success
               c0-0c0s3|    noflags|:    Success
   ```

5. Reboot the cabinet controllers (CC), then verify that all CCs are up.

a. Reboot the cabinet controllers.

```
smw# xtccreboot -c all
xtccreboot: reboot sent to specified CCs
smw# sleep 180
```

b. Are all cabinet controllers up now? Repeat this command until all of the cabinet controllers report in.

```
smw# xtalive -a l1sysd -l l1 s0
The expected responses were received.
```

**6.** Power up the system.

```
smw# xtcli power up s0
Turning on power to cabinet and waiting for confirmation...

           Component|       Flags:    Result
---------------------------------------------------------------------
                c0-0|    noflags|:    Success
            c0-0c0s0|    noflags|:    Success
            c0-0c0s1|    noflags|:    Success
            c0-0c0s2|    noflags|:    Success
            c0-0c0s3|    noflags|:    Success
```

Note that at this point the `xtcli status` output shows that all nodes are "off" because they have not yet been bounced.

The bootstrap process is now complete. The next task is to discover the Cray system hardware.

## 3.4.8 Discover Hardware and HSN Routing, Prepare STONITH

### Prerequisites

This procedure assumes that the `xtdiscover --bootstrap` command has been run successfully.

### About this task

**About Hardware Discovery**. This procedure uses `xtdiscover` to detect the Cray system hardware components on the system. The `xtdiscover` command confirms some basic information (entered earlier with `xtdiscover --bootstrap`) for the hardware discovery process, warns that changes will be made, and then confirms whether to abort or continue. Finally, this command creates entries in the system database to describe the hardware. To display the configuration, use the `xtcli` command after running `xtdiscover`. For more detailed information, see the `xtdiscover(8)` man page.

**About STONITH.** This procedure prepares STONITH (shoot the other node in the head), a Linux service that automatically powers down a node that is not working correctly. If either boot node failover or SDB node failover will be used, then STONITH needs to be set on the primary blade.

> **IMPORTANT:** The primary boot node and primary SDB node should not be on the same blade. Likewise the secondary boot node and secondary SDB node should not be on the same blade. Four different blades should be used if there are two boot nodes and two SDB nodes.

**Trouble?** If a step in this procedure fails because of a hardware issue, such as a cabinet failing to power up, resolve that issue and then go back to the last successful step in the procedure and continue from there. Do not skip steps or continue out of order.

## Procedure

DISCOVER CRAY SYSTEM HARDWARE

1. Log on to the SMW as `root`, if not already logged in.

2. Run the `xtdiscover` command.

   `xtdiscover` will continue until it pauses with instructions to bounce the system in a separate window.

   ```
   smw# xtdiscover
   *********** xtdiscover started ***********

   ...

   ...

   In a separate window, please bounce the system now to continue discovery.
   ```

3. If prompted, bounce the system (as `crayadm`) in a separate window.

   ```
   crayadm@smw> /opt/cray/hss/default/etc/xtdiscover-bounce-cmd
   ```

4. After the `xtbounce` command from the previous step has finished, return to the `xtdiscover` window and enter "c" to continue the hardware discovery.

   ```
   After bounce completes, enter 'c' to complete discovery
   or 'q' or 'a' to abort [c]: c
   ```

5. Commit the results of `xtbounce` to the database.

   When asked whether to commit the `xtdiscover` results to the database, enter **`y`**.

(optional) PREPARE STONITH FOR BOOT NODE AND SDB NODE FAILOVER

6. For sites using boot node failover, set STONITH for the primary boot node's blade.

   Skip this step if there will be no boot node failover at this site.

   In the example, the primary boot node is c0-0c0s0n1, so its blade is c0-0c0s0.

   ```
   smw# xtdaemonconfig c0-0c0s0 stonith=true
   ```

7. For sites using SDB failover, set STONITH for primary SDB node's blade.

   Skip this step if there will be no SDB node failover at this site.

   In the example, the primary SDB node is c0-0c2s0n1, so its blade is c0-0c2s0.

```
smw# xtdaemonconfig c0-0c2s0 stonith=true
```

DISCOVER HSN ROUTING CONFIGURATION

**8.** Discover the routing configuration of the high-speed network (HSN).

After `xtdiscover` finishes, run the `rtr` command as crayadm to determine the exact configuration of the HSN.

```
smw# su - crayadm
crayadm@smw> PS1="\u@\h:\w \t> "
crayadm@smw> rtr --discover
```

The `rtr` command may produce the following message and prompt. Answer "y" to allow `rtr` to bounce the system in diagnostic mode.

```
rtr:WARNING: No HSN discover info found, Using defaults (100% bandwidth
assumed)
System was not bounced in diagnostic mode, should I re-bounce? y
```

## 3.4.9 Update Firmware

### Prerequisites

This procedure assumes that Cray hardware discovery has been completed successfully.

### About this task

This procedure first checks whether the firmware of these components (controllers) needs to be updated, then updates the firmware only if there are Revision Mismatches.

**all cabinet-level components**

cc_mc (CC Microcontroller)
cc_bios (CC Tolapai BIOS)
cc_fpga (CC FPGA)
chia_fpga (CHIA FPGA)

**all blade-level components**

cbb_mc (CBB BC Microcontroller)
ibb_mc (IBB BC Microcontroller)
anc_mc (ANC BC Microcontroller)
bc_bios (BC Tolapai BIOS)
lod_fpga (LOD FPGA)
node_bios (Node BIOS)
loc_fpga (LOC FPGA)
qloc_fpga (QLOC FPGA)

### Procedure

**1.** Check firmware.

Check whether any firmware needs to be updated on the various controllers.

```
crayadm@smw> xtzap -r -v s0
```

If the firmware on any controllers is out of date, the output looks like this, and the firmware needs to be updated (reflashed).

```
Individual Revision Mismatches:

Type       ID                Expected   Installed
---------- ----------------- ---------- ----------------------------------------
cc_bios    c0-0              0013       0012
bc_bios    c0-0c0s0          0013       0012
bc_bios    c0-0c0s1          0013       0012
bc_bios    c0-0c0s2          0013       0012
bc_bios    c0-0c0s3          0013       0012
```

**2.** Update firmware, if any components are not current.

⚠ **CAUTION:** The xtzap command is normally intended for use by Cray Service personnel only. Improper use of this restricted command can cause serious damage to the computer system.

Run xtzap -a to update all components.

```
crayadm@smw> xtzap -a s0
```

Note that it is possible to update firmware in cabinets or blades only rather than the entire system. For more information, see *XC™ Series System Administration Guide* (S-2393).

**3.** Run xtbounce --linktune if any components were not current.

Force xtbounce to do a linktune on the full system before checking firmware again.

**full system** For a full, unpartitioned system or a fresh install:

```
crayadm@smw> xtbounce --linktune=all s0
```

**partitioned** For a partitioned system (software update process only, because partition instructions come later in the fresh install process):

```
crayadm@smw> xtbounce --linktune=all p1
crayadm@smw> xtbounce --linktune=all p2
crayadm@smw> xtbounce --linktune=all p3
```

**4.** Confirm that all components with out-of-date firmware have been updated.

Check firmware again after updating and linktuning those components.

```
crayadm@smw> xtzap -r -v s0
```

## 3.4.10 (Optional) Configure Partitions

### About this task

This procedure describes how to divide the CLE system into "logical machines" or partitions. By definition, p0 is the entire system, and p1 through p31 are smaller partitions. Each partition must have its own set of boot, sdb, and other service nodes and compute nodes to boot the partition. See the xtcli_part(8) man page for more details.

> **NOTE:** (SMW HA only) For a partitioned SMW HA system, only the first SMW requires this procedure, because the hardware configuration is stored in a shared MariaDB (formerly MySQL) database.

To add a partition, specify the boot node, SDB node, and the components that will be members of the partition. As an example, the following steps show how to add these two partitions to an unpartitioned system (p0).

```
partition: p1
boot node: c0-0c0s0n1
sdb node: c0-0c0s1n1
members:
c0-0c0s0,c0-0c0s1,c0-0c0s4,c0-0c0s5,c0-0c0s6,c0-0c0s7,c0-0c0s8,c0-0c0s9,c0-0c0s10
,c0-0c0s11,c0-0c0s12,c0-0c0s15

partition: p2
boot node: c0-0c0s3n1
sdb node: c0-0c0s3n1
members: c0-0c0s2,c0-0c0s3,c0-0c0s13,c0-0c0s14
```

### Procedure

1. Deactivate p0.

   ```
   smw# xtcli part_cfg deactivate p0
   ```

2. Add a partition.

   Note that `-b` identifies the boot node, `-d` identifies the SDB node, and `-m` identifies all members of the partition.

   ```
   smw# xtcli part_cfg add p1 -i /raw0 -b c0-0c0s0n1 -d c0-0c0s1n1 \
   -m c0-0c0s0,c0-0c0s1,c0-0c0s4,c0-0c0s5,c0-0c0s6,c0-0c0s7,\
   c0-0c0s8,c0-0c0s9,c0-0c0s10,c0-0c0s11,c0-0c0s12,c0-0c0s15
   ```

3. Activate the new partition.

   ```
   smw# xtcli part_cfg activate p1
   ```

4. Add and activate a second partition.

   ```
   smw# xtcli part_cfg add p2 -i /raw0 -b c0-0c0s3n1 -d c0-0c0s3n1 \
   -m c0-0c0s2,c0-0c0s3,c0-0c0s13,c0-0c0s14

   smw# xtcli part_cfg activate p2
   ```

## 3.4.11 Repurpose Compute Nodes

When a compute node is configured for a non-compute role, that node is a *repurposed compute node*. Compute nodes can be repurposed to become service nodes for use as tier2 servers (recommended) or in other capacities. Compute nodes should not be repurposed as service nodes for services that require external connectivity.

> **NOTE:** (SMW HA only) For SMW HA systems, perform this step only on the first SMW. This procedure is not required on the second SMW.

Use the `xtcli mark_node` command to repurpose a node in a compute blade. In this example, two compute nodes are being repurposed as service nodes and marked accordingly in the HSS database.

```
crayadm@smw> xtcli mark_node service c0-0c0s2n0,c0-0c0s2n1
```

Note that service nodes can be repurposed as compute nodes as well. In that case, the command would be **`xtcli mark_node compute`**.

## 3.4.12 Finish Configuring the SMW for the CLE System Hardware

### Prerequisites

This procedure assumes that Cray hardware has been discovered and component firmware has been updated (if needed).

### About this task

This procedure contains the final steps of configuring the SMW for the CLE system hardware. Note that a full system is referred to as "s0" here. The term "p0" could have been used, because in this context, the two terms are interchangeable. In contrast, commands that operate on config sets use only the term "p0" when referring to a full system. In the config set context, the terms are not interchangeable.

### Procedure

**1.** Check status on all components.

**full system** For a full, unpartitioned system:

```
crayadm@smw> xtcli status s0
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xtcli status p1
```

```
crayadm@smw> xtcli status p2
```

**2.** Check routing configuration of the system.

**full system** For a full, unpartitioned system:

```
crayadm@smw> rtr -R s0
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> rtr -R p1
```

```
crayadm@smw> rtr -R p2
```

Note that the `rtr -R` command produces no output unless there is a routing problem.

3. Examine the hardware inventory and verify that all nodes are visible to the SMW.

   **full system** For a full, unpartitioned system:

   ```
   crayadm@smw> xthwinv s0 > xthwinv.out
   ```

   ```
   crayadm@smw> xthwinv -x s0 > xthwinv.xml
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   crayadm@smw> xthwinv p1 > xthwinv.p1.out
   crayadm@smw> xthwinv -x p1 > xthwinv.p1.xml
   ```

   ```
   crayadm@smw> xthwinv p2 > xthwinv.p2.out
   crayadm@smw> xthwinv -x p2 > xthwinv.p2.xml
   ```

4. Check microcontroller information.

   Execute the `xtmcinfo -u` command to retrieve microcontroller information from cabinet control processors and blade control processors. Ensure that all blade controllers have output and show similar uptime values.

   ```
   crayadm@smw> xtmcinfo -u s0
   ```

5. Exit from crayadm back to root account.

   ```
   crayadm@smw> exit
   smw#
   ```

## 3.5 Configure CLE

The CLE config set stores configuration information used to boot and customize the CLE system in the p0 config set for a full, unpartitioned system, or config set p1 for partition p1 and config set p2 for partition p2. This part of the installation and configuration process has procedures to update CLE configuration worksheets, create a new CLE config set from those worksheets, update the config set to set passwords that must be encrypted and to check that all basic and required configuration settings have been set, followed by the reconfiguration of a few services, config set validation, and making a snapshot.

1. *Update CLE Configuration Worksheets* on page 116
2. *Create New CLE Config Set from Worksheets* on page 177
3. *Update CLE Config Set after a Fresh Install* on page 178
4. Perform post-configuration activities.
   a. *Check CLE Hostnames in /etc/hosts File* on page 180
   b. *Update /etc/motd for Nodes* on page 181

**c.** *Copy Files for External Lustre Fine-grained Routing* on page 181

**d.** *Configure Files for Cray Simple Sync Service* on page 182

**e.** *Display and Capture all Config Set Information* on page 183

**f.** *Validate Config Sets* on page 184

**g.** *Make a Post-config Snapshot using snaputil* on page 185

**h.** *Make a Post-config Backup of Current Global and CLE Config Sets* on page 185

> **NOTE:** (SMW HA only) For SMW HA systems, the following procedures are done only on the first SMW because the config sets are shared between both SMWs in the HA cluster. In contrast, Ansible plays must be run on each SMW.

## 3.5.1 Update CLE Configuration Worksheets

### Prerequisites

This procedure assumes that the *Prepare the CLE Configuration Worksheets* on page 106 procedure has been performed, resulting in a set of CLE configuration worksheets that reside in a work area ready to be edited with site-specific configuration information.

### About this task

The Cray XC system stores configuration information used to boot and customize the CLE system in the p0 config set, or if the system is partitioned, in config set p1 for partition p1 and config set p2 for partition p2, and so forth. Use these procedures to edit the CLE configuration worksheets to include site-specific configuration data. Afterwards, these worksheets will be uploaded to the config set to create or update it.

When editing configuration worksheets, a general rule is to uncomment all settings that are marked level=basic and modify values as needed. All settings that remain commented are considered unconfigured. Some settings are already uncommented in the original worksheet; Cray recommends not modifying those preconfigured settings because they are needed for proper configuration of the system. For more information about editing configuration worksheets and updating config sets, see *XC™ Series Configurator User Guide* (S-2560).

> **REMEMBER:** For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Assuming a work area directory was created for each partition, change to that directory and update worksheets there for each partition.
>
> **NOTE:** (SMW HA only) For SMW HA systems, the following procedures are done only on the first SMW because the config sets are shared between both SMWs in the HA cluster. In contrast, Ansible plays must be run on each SMW.

### Procedure

1. Change to the work area directory to simplify the editing commands in the following procedures.

   ```
   smw# cd /var/adm/cray/release/p0_worksheet_workarea
   ```

2. Edit and update the CLE configuration worksheets using the procedures that follow.

**TIP:** Update the node groups worksheet (`cray_node_groups_worksheet.yaml`) first. Many configuration worksheets use node groups, and it will be much easier to update those worksheets if the necessary node groups are already defined.

Use *Installation Checklist 5: Update CLE Configuration Worksheets* on page 365 to track progress updating the worksheets.

### 3.5.1.1 Update cray_alps Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Cray ALPS (Application Level Placement Scheduler) is the Cray-supported mechanism for placing and launching applications on Cray system compute nodes. ALPS provides application placement, launch, and management functions and cooperates closely with third-party workload managers (WLM) for application scheduling across Cray systems. The third-party WLMs make policy and scheduling decisions, whereas ALPS provides a mechanism to place and launch the applications contained within batch jobs. ALPS also supports interactive application placement and launch.

This procedure enables the cray_alps service and configures some settings in the cray_alps configuration worksheet to add site-specific data. For an explanation of the long variable names in configuration settings, see *About Variable Names in the Configurator and Configuration Worksheets* on page 16.

## Procedure

1. Edit `cray_alps_worksheet.yaml`.

   ```
   smw# vi cray_alps_worksheet.yaml
   ```

2. Uncomment `cray_alps.enabled` and ensure that it is set to `true`.

3. Uncomment `cray_alps.settings.common.data.xthostname` and set it to the name of this Cray system.

4. Configure ALPS node groups.

   If there are service nodes other than login nodes and the ALPS master node (the SDB node) that need to run ALPS commands, add them to a node group by editing `cray_node_groups_worksheet.yaml`. That node group should include the workload manager (WLM) server and MOM (machine-oriented miniserver) nodes.

   Uncomment `cray_alps.settings.common.data.alps_node_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`- `).

   ```
   cray_alps.settings.common.data.alps_node_groups:
   - NODE_GROUP_1
   - NODE_GROUP_2
   ```

5. (Optional) If DRC (dynamic RDMA credentials) will be used in a large system, uncomment `cray_alps.settings.apshed.data.pDomainMax` and set it to 256.

   If the maximum number of user protection domains is not increased from its default value of 10 to something like 256, DRC might exhaust all of the domains, which could cause problems for sites with larger, more complex systems.

6. Uncomment `cray_alps.settings.apsys.data.prologPath` and `cray_alps.settings.apsys.data.epilogPath`, even if they are assigned a null value.

7. (Optional) If RUR (resource utilization reporting) will be used at this site, set the `prologPath` and `epilogPath` settings (from the previous step) to these paths.

   ```
   cray_alps.settings.apsys.data.prologPath: /opt/cray/rur/default/bin/rur_prologue.py
   cray_alps.settings.apsys.data.epilogPath: /opt/cray/rur/default/bin/rur_epilogue.py
   ```

   Also, ensure that the `cray_rur` service is enabled. See *Update cray_rur Worksheet* on page 164.

### 3.5.1.2 Update cray_auth Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Authentication configuration service provides a way to list the authentication domains that should govern how users of the system are identified and authenticated. Authentication domains include LDAP, NIS, and Active Directory.

This procedure configures some settings in the cray_auth configuration worksheet to add site-specific data. For an explanation of the long variable names in configuration settings, see *About Variable Names in the Configurator and Configuration Worksheets* on page 16. For examples of modifying a config set for use with an authentication method other than the default LDAP setup, see *Modify a Config Set for use with Advanced Authentication Configurations* on page 339.

## Procedure

1. Edit `cray_auth_worksheet.yaml`.

   ```
   smw# vi cray_auth_worksheet.yaml
   ```

2. Uncomment `cray_auth.enabled` and set it to `true`.

3. Review the `nsswitch_sources` service setting of the worksheet.

   ```
   #************* START Service Setting: nsswitch_sources ************
   ```

   This service setting controls the settings in the `nsswitch.conf` file. Add, delete, or change these settings to modify the `nsswitch.conf` file, as needed.

**4.** Review the `common_ldap_options` service setting of the worksheet, especially if LDAP will NOT be used at this site.

```
#************* START Service Setting: common_ldap_options ************
```

This is an advanced level setting that has several pre-populated values for common LDAP configuration options.

- Sites NOT using LDAP for part or all of the authentication must change some settings in this section (for example, to use Kerberos or Active Directory).
- Sites using LDAP may need to add, change, or delete options in this section.

To add a common_ldap_options stanza to the worksheet, copy the two lines below `# ** EXAMPLE 'common_ldap_options' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'common_ldap_options' setting entries here, if desired.`

```
# ** EXAMPLE 'common_ldap_options' VALUE  (with current defaults) **
# cray_auth.settings.common_ldap_options.data.option.sample_key_a: null <-- setting a multival key
# cray_auth.settings.common_ldap_options.data.sample_key_a.value: ''

# ** 'common_ldap_options' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` in all lines with the LDAP option to be specified, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add the value of the option in the second line. Repeat this step for each option/value pair to be specified.

```
# NOTE: Place additional 'common_ldap_options' setting entries here, if desired.
cray_auth.settings.common_ldap_options.data.option.sample_key_a: null
cray_auth.settings.common_ldap_options.data.sample_key_a.value: ''

#************************ END Service Setting: common_ldap_options ************************
```

**5.** (If using NIS) Review the `common_nis_options` service setting of the worksheet and configure these settings if this site wishes to use NIS.

```
#************* START Service Setting: common_nis_options ************
```

This is an advanced level setting that has several pre-populated values for common NIS configuration options. Add, change, or delete options if this site has special authentication needs, such as when using Kerberos (not common).

In the worksheet, copy the two lines below `# ** EXAMPLE 'common_nis_options' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'common_nis_options' setting entries here, if desired.`

```
# ** EXAMPLE 'common_nis_options' VALUE  (with current defaults) **
# cray_auth.settings.common_nis_options.data.option.sample_key_a: null <-- setting a multival key
# cray_auth.settings.common_nis_options.data.sample_key_a.value: ''
```

Uncomment the lines, replace `sample_key_a` in all lines with the NIS option to be specified, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add the value of the option in the second line. Repeat this step for each option/value pair to be specified.

```
# NOTE: Place additional 'common_nis_options' setting entries here, if desired.
cray_auth.settings.common_nis_options.data.option.sample_key_a: null
cray_auth.settings.common_nis_options.data.sample_key_a.value: ''

#************************ END Service Setting: common_nis_options ************************
```

**6.** Review the `domain` service setting of the worksheet and configure settings, as needed.

```
#************* START Service Setting: domain ************
```

a. (If using LDAP) Configure LDAP domains to connect to LDAP servers

In the worksheet, copy the four lines below `# ** EXAMPLE 'domain' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'domain' setting entries here, if desired.`

```
# ** EXAMPLE 'domain' VALUE  (with current defaults) **
# cray_auth.settings.domain.data.reference.sample_key_a: null <-- setting a multival key
# cray_auth.settings.domain.data.sample_key_a.servers: []
# cray_auth.settings.domain.data.sample_key_a.schema: rfc2307
# cray_auth.settings.domain.data.sample_key_a.aux_settings: []
```

Uncomment the lines, replace `sample_key_a` in all lines with some unique authentication domain identifier, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add values in accordance with site requirements. For settings that are lists, remove the empty brackets and add each list element on a separate line prefixed by a hyphen and space (- ).

```
# NOTE: Place additional 'domain' setting entries here, if desired.
cray_auth.settings.domain.data.reference.<ldap_domain_name>: null
cray_auth.settings.domain.data.<ldap_domain_name>.servers: []
cray_auth.settings.domain.data.<ldap_domain_name>.schema: rfc2307
cray_auth.settings.domain.data.<ldap_domain_name>.aux_settings: []
#************************** END Service Setting: domain *************************
```

b. Configure non-LDAP domains, as needed.

As in the previous substep, copy and paste the four-line stanza for a domain setting, but instead of uncommenting all four lines, leave commented the servers variable and the schema variable, which are specific to LDAP domains.

```
# NOTE: Place additional 'domain' setting entries here, if desired.
cray_auth.settings.domain.data.reference.<domain_name>: null
#cray_auth.settings.domain.data.<domain_name>.servers: []
#cray_auth.settings.domain.data.<domain_name>.schema: rfc2307
cray_auth.settings.domain.data.<domain_name>.aux_settings: []
#************************** END Service Setting: domain *************************
```

**7.** (If using NIS) Review the `nis` service setting of the worksheet and configure these settings if this site wishes to use NIS.

```
#************* START Service Setting: nis ************
```

a. Enable NIS (the `nis.data.enabled` setting).

Uncomment `cray_auth.settings.nis.data.enabled` and set it to `true`.

b. Configure the domain name (the `nis.data.domainname` setting).

Uncomment `cray_auth.settings.nis.data.domainname` and set it to the domain name that was configured on the NIS server (must match).

c. Configure the servers (the `nis.data.servers` setting).

Uncomment `cray_auth.settings.nis.data.servers: []`, remove the empty brackets, and add a list of NIS server host names or IP addresses.

```
cray_auth.settings.nis.data.servers:
- 172.32.3.4
- 172.32.4.55
```

**8.** Review the `access` service setting of the worksheet and configure these settings, as needed.

```
#************* START Service Setting: access ************
```

a. Set the access policy (the `access.data.policy` setting).

Whether using NIS or LDAP, ensure that `cray_auth.settings.access.data.policy` is uncommented and set it to the list shown here. At a minimum, these values are recommended to ensure that root and crayadm are using the local passwd entries and not ones from the authentication service.

> **NOTICE:** The initial `- ` (hyphen and space) at the beginning of each list element is part of the YAML syntax. The access policy data, which begins with either a `-` or `+`, starts after that.

```
cray_auth.settings.access.data.policy:
- +:root:LOCAL
- +:crayadm:LOCAL
```

b. Configure access to compute nodes (the `access.data.config_computes` setting).

Uncomment `cray_auth.settings.access.data.config_computes` and set in accordance with site requirements. For most systems, set this variable to false.

Set this variable to true for any of these conditions:

- This site wants to allow compute nodes to use network lookup services to identify users (setting config_computes to true does not mean that users will be allowed to log into compute nodes directly).
- This site is using Slurm and network authentication.
- This site is using cluster compatibility mode (CCM) with LDAP accounts (`ccmrun` will work regardless, but `ccmlogin` will work only if config_computes is set to true).

> **IMPORTANT:** If cray_auth.settings.access.data.config_computes is set to true, ensure that:
>
> - RSIP is configured to enable the compute nodes to contact the LDAP server.
> - Network user lookup servers are equipped to handle the volume of requests made by the compute nodes.

c. Configure node groups to recognize user IDs provided by off-node identification services, if needed (the `access.data.config_id_service_groups` setting).

If there are any non-login nodes that may need to identify users without allowing user access, such as DAL (direct-attached Lustre) MDS nodes or MOM nodes, add their cnames to a node group by editing `cray_node_groups_worksheet.yaml`, and then add that node group to the list of `config_id_service_groups`. Nodes within these groups should be provided with a network path to the relevant servers.

Uncomment `cray_auth.settings.access.data.config_id_service_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`- `).

```
cray_auth.settings.access.data.config_id_service_groups:
- NODE_GROUP_1
- NODE_GROUP_2
```

**9.** Review the `section` service setting and the `options` embedded service setting (they go together) of the worksheet and configure these settings, as needed.

The "section" service setting refers to the section of the `sssd.conf` file, and "options" are the multiple entries within a "section" that an administrator can specify. This enables the administrator to override settings in any section of the `sssd.conf` file.

In the worksheet, copy the three lines below `# ** EXAMPLE 'section' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'section' setting entries here, if desired.`

```
# ** EXAMPLE 'section' VALUE  (with current defaults) **
# cray_auth.settings.section.data.section_name.sample_key_a: null  <-- setting a multival key
# cray_auth.settings.section.data.sample_key_a.options.option_name.sample_key_b: null  <-- setting a
multival key
# cray_auth.settings.section.data.sample_key_a.options.sample_key_b.value: ''
```

Uncomment the lines, replace `sample_key_a` in all lines with a unique section identifier and remove the `<–– setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Replace `sample_key_b` in the second and third lines with a unique option identifier. Finally, add values in accordance with site requirements.

Here is an example that adds a "debug_level = 7" line to the otherwise unnamed/unused "[pam]" section in the `sssd.conf` file.

```
# NOTE: Place additional 'section' setting entries here, if desired.
cray_auth.settings.section.data.section_name.pam: null
cray_auth.settings.section.data.pam.options.option_name.debug_level: null
cray_auth.settings.section.data.pam.options.debug_level.value: 7
#************************* END Service Setting: section *************************
```

Here is the resulting section of the `sssd.conf` file.

```
[nss]
filter_users = root, crayadm

[pam]                    <-----
debug_level = 7  <-----

[domain/crayit]
```

### 3.5.1.3 Update cray_batchlimit Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray batchlimitd daemon is used to limit the number of processes that a batch job can create, thereby avoiding a potentially harmful proliferation of processes. Linux limits the total number of processes on a per UID basis, but batchlimitd introduces process and thread creation limits on a per-cpuset basis. In CLE 5.2 / SMW 7.2, the batchlimitd daemon was supported only with workload managers (WLM) that were compiled with cpuset support, such as Moab/TORQUE 4.x or later.

This procedure enables or disables the cray_batchlimit configuration service and, if enabled, configures some settings using the cray_batchlimit configuration worksheet.

## Procedure

1. Edit cray_batchlimit_worksheet.yaml.

   ```
   smw# vi cray_batchlimit_worksheet.yaml
   ```

2. Enable cray_batchlimit, as needed.

   Uncomment cray_batchlimit.enabled and do one of the following:

   - If batchlimit NOT used at this site, set it to false to disable the service. No other settings are needed for a fresh install.
   - If batchlimit is or will be used at this site, set it to true and configure the advanced settings to values appropriate for this site.

### 3.5.1.4 Update cray_boot Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray boot configuration service provides a way to specify which nodes will act as boot nodes on the high-speed network (HSN). This service must be enabled for the system to function properly. This procedure configures some basic settings in the cray_boot configuration worksheet.

## Procedure

1. Edit cray_boot_worksheet.yaml.

   ```
   smw# vi cray_boot_worksheet.yaml
   ```

2. Uncomment cray_boot.enabled and ensure that it is set to true.

3. Configure the boot groups setting.

   This setting specifies a list of node groups whose members will act as boot nodes.

   Uncomment cray_boot.settings.node_groups.data.boot_groups and the line immediately following it. By default, the boot_nodes node group is the first node group in the list of boot groups. To use other nodes as boot nodes on the HSN, add one or more node groups to this list.

   > **IMPORTANT:** Any node group added to boot_groups must first be defined in cray_node_groups_worksheet.yaml.

   Because this is a list setting, each node group must be on a separate line prefixed by a hyphen and space (- ).

   ```
   cray_boot.settings.node_groups.data.boot_groups:
   - boot_nodes
   ```

### 3.5.1.5 Update cray_ccm Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Cray cluster compatibility mode (CCM) enables users to run independent software vendor (ISV) applications without modification. Supported workload managers (WLM) include PBS, Moab/TORQUE, Slurm, and LSF.

This procedure disables the cray_ccm service because it should be disabled until a WLM is installed.

## Procedure

1. Edit `cray_ccm_worksheet.yaml`.

   ```
   smw# vi cray_ccm_worksheet.yaml
   ```

2. Uncomment `cray_ccm.enabled` and set it to `false`.

   This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied. This worksheet contains additional configuration settings that are level advanced, but this service and its advanced settings are not needed for a fresh install.

### 3.5.1.6 Update cray_cnat Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Compute Node Administrative Tool (CNAT) is a mechanism for submitting and monitoring the execution of batch scripts; it requires a workload manager (WLM) to function. This procedure disables the Cray CNAT configuration service because CNAT is not needed for a first-time boot of CLE. It can be enabled and configured at a later time when a WLM is installed.

For one use of CNAT, see "Apply Rolling Patches to Compute Nodes" in *XC™ Series System Administration Guide* (S-2393).

## Procedure

1. Edit `cray_cnat_worksheet.yaml`.

```
smw# vi cray_cnat_worksheet.yaml
```

2. Uncomment `cray_cnat.enabled` and set it to `false`.

   This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied.

### 3.5.1.7 About Configuring Cray Dynamic RDMA Credentials (DRC)

Dynamic RDMA Credentials (DRC) is a new XC system service that enables shared network access between different user applications. DRC enables user applications to request managed network credentials, which can be shared with other users, groups, or jobs. Access to a credential is governed by the application and DRC to provide authorized and protected sharing of network access between applications. DRC extends the existing protection domain functionality provided by ALPS without exposing application data to unauthorized applications. DRC can also be used with other batch systems, such as Slurm, without any loss of functionality.

**Trouble?** Do not use DRC with VMDH (virtual memory domain handle). DRC does not use VMDH or limit its use; however, in a MAMU (multiple application multiple user) scenario, the use of VMDH by an application that is also using DRC could cause problems for other applications using VMDH on the same node, resulting in the failure of one or more of those processes.

When configuring Cray DRC, using the default values of the following settings will be sufficient for most cases. There are two required settings that must be configured with site-specific information however: server_cname and cookie_provider, which are both DRC server settings. Those must be assigned non-null values to complete the configuration process.

#### DRC Client (DRCC) Settings

None of the DRCC settings are required.

**socket_location** Location of the DRCC UNIX domain socket. This location should allow read-write access for any user, because libDRC must be able to write to the socket to make any necessary requests.

Default value: `/tmp/drcc.sock`

**logging_directory** Storage location for DRCC logs. This can be located anywhere convenient, as long as the directory is:

- (required) writeable by root
- (recommended) persistent between reboots so that the log file can be retrieved in a node-down event

Default value: `/tmp`

**logging_filename** Name of the log file for DRCC. This name can be anything except a null value.

Default value: `drcc.log`

**logging_level** Verbosity of the DRCC logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**requests_log_level** Verbosity of the python-requests logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**llm_log_enabled**
If enabled, DRCC will log messages to the lightweight log management (LLM) service.

Default value: `true`

**llm_log_level**
Verbosity of DRCC log messages to the LLM service. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

## DRC Server (DRCS) Settings

Two of the DRCS settings are required: server_cname and cookie_provider.

**server_cname (REQUIRED)**
The cname of the node where DRCS will reside (e.g., c0-0c1s4n0). DRCS can reside on a login node or any unspecialized service node, but NOT on any boot or SDB nodes. This cname is also used as the value for the clients setting when preparing a persistent mount for DRC in `cray_persistent_data_worksheet.yaml` (see the database_directory setting below). Because this is a required field and no default is provided, a value must be entered.

**logging_directory**
Storage location for DRCS logs. This can be located anywhere convenient as long as the directory is:

- (required) writeable by root
- (recommended) persistent between reboots so that the log file can be retrieved in a node-down event

Default value: `/tmp`

**logging_filename**
Name of the log file for DRCS. This name can be anything except a null value.

Default value: `drcc.log`

**logging_level**
Verbosity of the DRCS logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**port**
TCP port on which the DRC server will listen to requests. Do not assign this port to any other TCP service.

Default value: `4000`

**use_ssl**
Should the DRCS server use SSL? This additional layer of security is not necessary but is recommended.

Default value: `true`

**rpc_uri**
Remote procedure call (RPC) URI used by both client and server to correctly address DRCS services.

Default value: `json-rpc`

**werkzeug_log_level**
Verbosity of the python-werkzeug logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**jsonrpc_log_level**

Verbosity of the python-jsonrpc logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**requests_log_level**

Verbosity of the python-requests logger. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**authorized_uids**

List of UIDs that are allowed to interface directly with DRCS through DRCC, DRCCLI, and DRCJEDi (DRC job expiration director). If DRCC, DRCCLI, or DRCJEDi is run under a UID that is not in this list, any request made by that user will be rejected.

Default value: `['0']`

**admin_uids**

List of UIDs that are allowed to run DRCCLI. At present, this is limited to the values in the authorized_uids list.

Default value: `['0']`

**cookie_provider (REQUIRED)**

A string that indicates which workload manager binary DRCS should contact for cookies. Possible values:

**`apmgr`** For systems running ALPS (Application Level Placement Service)

**`ncmd`** For systems running Slurm (the native workload manager)

Because this is a required field and no default is provided, a value must be entered.

**llm_log_enabled**

If enabled, DRCS will log messages to the lightweight log management (LLM) service.

Default value: `true`

**llm_log_level**

Verbosity of DRCS log messages to the LLM service. Possible values in order of increasing verbosity: `critical`, `error`, `warning`, `info`, and `debug`.

Default value: `error`

**database_directory**

Storage location for the credential database. This needs to be persistent `/var/` so that the database can support node restart features. This is the same path name used in `cray_persistent_data_worksheet.yaml` to set up a persistent mount for DRC (see *Update cray_persistent_data Worksheet* on page 160).

Default value: `/var/opt/cray/rdma-credentials`

**database_filename**

Name of the credential database file.

Default value: `drc.db`

### 3.5.1.8 Update cray_drc Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray dynamic RDMA credentials (DRC) service configures dynamic RDMA (remote direct memory access) credentials, which are secure network credentials that can be shared between user applications to achieve intercommunication between applications running in different job reservations. This procedure configures some basic settings in the cray_drc configuration worksheet to add site-specific data.

This service is disabled by default. For additional information about Cray DRC to help decide whether to enable it and know what configuration parameters are available, see *About Configuring Cray Dynamic RDMA Credentials (DRC)* on page 125. To enable and use this service, follow these steps.

> **NOTICE:** Do not use DRC with VMDH (virtual memory domain handle).

Note that the database directory for DRC must be persistent storage to have persistent credentials and support node restart features. The server cname (step 3) and directory path (step 5) specified in this worksheet will be used in `cray_persistent_data_worksheet.yaml` to set up a client mount for persistent storage.

## Procedure

1. Edit `cray_drc_worksheet.yaml`.

   ```
   smw# vi cray_drc_worksheet.yaml
   ```

2. Uncomment `cray_drc.enabled`.
   - To disable this service, set to false and skip the rest of the procedure.
   - To enable this service, set to true and continue to the next step.

3. Uncomment `cray_drc.settings.server.data.server_cname` and set it to the cname of the service node that should be running the DRC server.

   This cname will also be used as the value for the clients setting when preparing a persistent mount for DRC in the Cray persistent data worksheet (`cray_persistent_data_worksheet.yaml`).

4. Uncomment `cray_drc.settings.server.data.cookie_provider` and set it to one of these values:
   - `apmgr` if using an ALPS (Application Level Placement Scheduler) workload manager (WLM)
   - `ncmd` if using the Slurm WLM

5. Uncomment `cray_drc.settings.server.data.database_directory` and set it to `/var/opt/cray/rdma-credentials`.

   This needs to be persistent storage so that the database can have persistent credentials and support node restart features. This is the same path name that will be used as the mount point when setting up a persistent mount for DRC in `cray_persistent_data_worksheet.yaml` (see *Update cray_persistent_data Worksheet* on page 160).

6. Go back and uncomment the following settings, and set them in accordance with site preferences.

   Cray recommends configuring these settings so that diagnostic information is available if needed. Using persistent storage for the logging directories is best; however that depends on available storage space.

   ```
   cray_drc.settings.client.data.logging_directory
   cray_drc.settings.client.data.logging_filename
   cray_drc.settings.server.data.logging_directory
   ```

```
    cray_drc.settings.server.data.logging_filename
```

If this system uses ALPS, Cray recommends increasing the maximum number of user protection domains when DRC is in use, especially for large systems. That parameter is set in the Cray ALPS service with the `pDomainMax` field. See *Update cray_alps Worksheet* on page 117.

### 3.5.1.9 Update cray_dvs Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Cray DVS (Data Virtualization Service) is a distributed network service that projects local file systems resident on I/O nodes or remote file servers to compute and service nodes within the Cray system. DVS provides a highly scalable mechanism to share file systems to a large number of client nodes using a fanout tree as configured in cray_scalable_services. This service must be enabled if Programming Environment (PE) software is to be used on compute and login nodes. It is also required if Netroot image roots will be used on compute and login nodes (Netroot is a mechanism that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system.).

This procedure enables the cray_dvs configuration service.

## Procedure

1. Edit `cray_dvs_worksheet.yaml`.

   ```
   smw# vi cray_dvs_worksheet.yaml
   ```

2. Uncomment `cray_dvs.enabled` and set it to `true`.

   Enabling the DVS configuration service is sufficient for a fresh install. Sites that plan to use DVS to project an external file system or provide access to DataWarp will need to configure other cray_dvs settings. For more information, see *XC™ Series DVS Administration Guide* (S-0005).

DVS uses the LNet (Lustre networking) networking layer, so ensure that cray_lnet is enabled as well. See *Update cray_lnet Worksheet* on page 134.

### 3.5.1.10 Update cray_dws Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray DataWarp Service (DWS) provides access to SSD (solid state device) storage for high bandwidth application I/O. For information about how to install and configure DataWarp, see *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564).

This procedure disables the cray_dws configuration service for a fresh install.

## Procedure

1. Edit `cray_dws_worksheet.yaml`.

   ```
   smw# vi cray_dws_worksheet.yaml
   ```

2. Uncomment `cray_dws.enabled` and set it to `false`.

   Save and exit the cray_dws worksheet.

### 3.5.1.11 Update cray_dw_wlm Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The cray_dw_wlm service is an interface used by workload managers (WLM) to interact with Cray DataWarp. It should be enabled on every system with Cray DataWarp. This procedure enables dw_wlm, but no other settings are changed at this point in the fresh install process. See *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564) for information about how to use this configuration service to set limits on what options users can add to DataWarp commands in their job scripts.

## Procedure

1. Edit `cray_dw_wlm_worksheet.yaml`.

   ```
   smw# vi cray_dw_wlm_worksheet.yaml
   ```

2. Uncomment `cray_dw_wlm.enabled` and set it to `true`.

   No other settings need to be changed for a fresh install.

### 3.5.1.12 Update Cray eLogin Service Worksheets

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

This procedure temporarily disables these eLogin (external login) configuration services during a fresh install of SMW/CLE. They will be enabled later during the installation and configuration process for eLogin.

**Cray eLogin LNet service** LNet (Lustre networking) is needed by any system that has external login nodes that mount Lustre file systems.

**Cray eLogin MOTD service** Generates the /etc/motd file for the eLogin nodes specified in the configuration set.

**Cray eLogin Networking service** Defines the number of eLogin nodes connected to the Cray system and their key network attributes.

**Cray eswrap service** eswrap wraps several XT, ALPS (Application Level Placement Scheduler), and WLM (workload manager) commands on eLogin nodes and executes them on the Cray login gateway.

## Procedure

1. Disable the Cray eLogin LNet service.
   a. Edit cray_elogin_lnet_worksheet.yaml.

   ```
   smw# vi cray_elogin_lnet_worksheet.yaml
   ```

   b. Uncomment cray_elogin_lnet.enabled and set it to false.

2. Disable the Cray eLogin MOTD service.
   a. Edit cray_elogin_motd_worksheet.yaml.

   ```
   smw# vi cray_motd_worksheet.yaml
   ```

   b. Uncomment cray_elogin_motd.enabled and set it to false.

3. Disable the Cray eLogin Networking service.
   a. Edit cray_elogin_networking_worksheet.yaml.

   ```
   smw# vi cray_elogin_networking_worksheet.yaml
   ```

   b. Uncomment cray_elogin_networking.enabled and set it to false.

4. Disable the Cray eswrap service.
   a. Edit cray_eswrap_worksheet.yaml.

   ```
   smw# vi cray_eswrap_worksheet.yaml
   ```

   b. Uncomment cray_eswrap.enabled and set it to false.

## 3.5.1.13 Update cray_firewall Worksheet

### Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The Cray firewall service is a mechanism for restricting packet traffic from various networks. This procedure configures the inherit and enable settings in the cray_firewall configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

### Procedure

1. Edit `cray_firewall_worksheet.yaml`.

   ```
   smw# vi cray_firewall_worksheet.yaml
   ```

2. Uncomment `cray_firewall.inherit` and ensure that it is set to `false`.

   This means that firewall settings in the CLE config set will be used instead of firewall settings in the global config set.

3. Uncomment `cray_firewall.enabled` and set it to `true`.

## 3.5.1.14 Update cray_image_binding Worksheet

### Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The Cray image binding service is a mechanism for mixing image content between booted IMPS (Image Management and Provisioning System) images and IMPS images that are projected onto a running system. This is a common scenario with the Cray Programming Environment (PE), which is installed into an IMPS image, pushed to the CLE boot node, then projected to compute nodes using DVS (Data Virtualization Service). The diagnostics (diag) image root is also pushed to the boot node and projected by DVS.

This procedure enables the cray_image_binding service but does not enable any bind mount profiles: those are enabled later in the process.

## Procedure

1. Edit `cray_image_binding_worksheet.yaml`.

   ```
   smw# vi cray_image_binding_worksheet.yaml
   ```

2. Uncomment `cray_image_binding.enabled` and set it to `true`.

   **IMPORTANT:** Do not enable any bind mount profiles now. Enabling them must wait until the associated image root has been pushed to the boot node. If the system has not been previously booted, failure to push an image root prior to enabling its profile may prevent the system from properly booting.

### 3.5.1.15 Update cray_ipforward Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray IP Forwarding service enables IP forwarding between service nodes and the SMW. This procedure configures the inherit and enable settings in the cray_ipforward configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

## Procedure

1. Edit `cray_ipforward_worksheet.yaml`.

   ```
   smw# vi cray_ipforward_worksheet.yaml
   ```

2. Uncomment `cray_ipforward.inherit` and set it to `true`.

   This means that IP forwarding settings in the global config set will be used instead of IP forwarding settings in the CLE config set.

3. Uncomment `cray_ipforward.enabled` and ensure that it is set to `true`.

### 3.5.1.16 Update cray_liveupdates Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The live updates service enables package manager (e.g., zypper, yum) actions (e.g., install, search, upgrade) on CLE nodes using repositories shared from the SMW to those nodes. This procedure sets the CLE cray_liveupdates service to inherit from the global cray_liveupdates service. There are no other settings that can be changed.

### Procedure

1. Edit `cray_liveupdates_worksheet.yaml`.

   ```
   smw# vi cray_liveupdates_worksheet.yaml
   ```

2. Uncomment `cray_liveupdates.inherit` and set it to `true`.

## 3.5.1.17 Update cray_lmt Worksheet

### Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The Cray Lustre Monitoring Tool (LMT) monitors Lustre servers using the Cerebro monitoring system. An LMT Cerebro module collects stats published in `/proc/fs/lustre` on the Lustre servers, and pushes them to the LMT server, which is co-located with the Lustre Management Server (MGS). A Cerebro module on the LMT server collects the statistics and stores them in a MySQL database. Lustre clients are not monitored.

This procedure disables the LMT configuration service, because it is not required for a fresh install. If direct-attached Lustre (DAL) is enabled and this site wishes to use LMT for DAL, enable this service later.

### Procedure

1. Edit `cray_lmt_worksheet.yaml`.

   ```
   smw# vi cray_lmt_worksheet.yaml
   ```

2. Uncomment `cray_lmt.enabled` and ensure that it is set to `false`.

   This disables this service. When a service is disabled, the values of all other settings in the service are ignored when configuration is applied. This worksheet contains additional configuration settings that are level advanced, but this service and its advanced settings are not needed for a fresh install.

### 3.5.1.18 Update cray_lnet Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

LNet (Lustre networking) is the networking layer used by Lustre and DVS. The Cray LNet configuration service must be configured on any systems that use DVS to mount external file systems or have Lustre clients and/or servers.

This procedure configures some basic settings in the cray_lnet configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_lnet_worksheet.yaml`.

   ```
   smw# vi cray_lnet_worksheet.yaml
   ```

2. Uncomment `cray_lnet.enabled` and do one of the following:

   - Set it to `true` if this system has external Lustre or DAL (direct-attached Lustre) or will use DVS to mount external file systems.
   - Set it to `false` otherwise.

   For systems with external Lustre, continue to the next step. Otherwise, Cray LNet configuration is complete for a fresh install.

   THE REMAINING STEPS ARE ONLY FOR SYSTEMS WITH EXTERNAL LUSTRE *****

   See also the *XC™ Series Lustre® Administration Guide* (S-2648).

3. Configure these settings.

   These settings are commonly configured with site-specific data when the system has external Lustre. Uncomment and set them as appropriate for this site.

   `cray_lnet.settings.ko2iblnd.data.peer_credits`
   `cray_lnet.settings.ko2iblnd.data.concurrent_sends`
   `cray_lnet.settings.local_lnet.data.lnet_name` (set to something like gni, gni1, gni2, gni3)
   `cray_lnet.settings.local_lnet.data.ip_wildcard` (change from default on a partitioned system or any system that changes the HSN (high speed network) address range)

4. Configure the following group of settings if this system uses flat routes to an external Lustre file system. Repeat this step for each external Lustre file system.

   Enter all external LNets that will be reached via flat routing. The information entered for each of these flat LNets will be used to set up ip2nets on the routers and routes to reach the external LNets through the routers on the clients.

In the worksheet, copy the six lines below `# ** EXAMPLE 'flat_routes' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'flat_routes' setting entries here, if desired.`

```
# ** EXAMPLE 'flat_routes' VALUE (with current defaults) **
# cray_lnet.settings.flat_routes.data.dest_lnet.sample_key_a: null <-- setting a multival key
# cray_lnet.settings.flat_routes.data.sample_key_a.dest_lnet_ip_wildcard: ''
# cray_lnet.settings.flat_routes.data.sample_key_a.router_groups: []
# cray_lnet.settings.flat_routes.data.sample_key_a.src_lnet: ''
# cray_lnet.settings.flat_routes.data.sample_key_a.ko2iblnd_peer_credits: 126
# cray_lnet.settings.flat_routes.data.sample_key_a.ko2iblnd_concurrent_sends: 63
#
# ** 'flat_routes' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with the name of the LNet on the external Lustre file system (***o2ib*** in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, modify the values as appropriate for this site.

This example uses "o2ib" as the name for this destination LNet and has a dest_lnet_ip_wildcard of 10.149.*.*.

> **IMPORTANT:** The settings for peer_credits and concurrent_sends must match between the external Lustre server and cray_lnet.settings.flat_routes.data.*o2ib*.ko2iblnd_peer_credits and cray_lnet.settings.flat_routes.data.*o2ib*.ko2iblnd_concurrent_sends.

Note that the cray_lnet.settings.flat_routes.data.o2ib.dest_lnet_ip_wildcard setting is the IP address wildcard that matches the IP addresses of all router interfaces for this flat route. For example, for a flat route from CLE clients to an external Lustre file system, the destination LNet might be 'o2ib', and the wildcard '10.149.*.*' would match the IB interfaces on the router nodes that are to be on the 'o2ib' LNet.

```
# NOTE: Place additional 'flat_routes' setting entries here, if desired.
cray_lnet.settings.flat_routes.data.dest_lnet.o2ib: null
cray_lnet.settings.flat_routes.data.o2ib.dest_lnet_ip_wildcard: 10.149.*.*
cray_lnet.settings.flat_routes.data.o2ib.router_groups:
- lnet_flat_routers
cray_lnet.settings.flat_routes.data.o2ib.src_lnet: gni2
cray_lnet.settings.flat_routes.data.o2ib.ko2iblnd_peer_credits: 63
cray_lnet.settings.flat_routes.data.o2ib.ko2iblnd_concurrent_sends: 63
#************* END Service Setting: flat_routes *************
```

For the flat routes router_groups setting, if there are no existing node groups that contain the router nodes for this site, create one or more node groups for this purpose (*`lnet_flat_routers`* in this example) using the procedure in *Update cray_node_groups Worksheet* on page 156 and reference the node group(s) in cray_lnet.settings.flat_routes.data.*o2ib*.router_groups.

**5.** Configure the following group of settings if this system uses fine-grained routing (FGR) to an external Lustre file system. Repeat this step for each external Lustre file system.

Enter all external LNets that will be reached via FGR. The information entered for each of these FGR routes will be used to set up ip2nets on the routers and routes to reach the external LNets through the routers on the clients.

In the worksheet, copy the six lines below `# ** EXAMPLE 'fgr_routes' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'fgr_routes' setting entries here, if desired.`

```
# ** EXAMPLE 'fgr_routes' VALUE  (with current defaults) **
# cray_lnet.settings.fgr_routes.data.dest_name.sample_key_a: null <-- setting a multival key
# cray_lnet.settings.fgr_routes.data.sample_key_a.router_groups: []
# cray_lnet.settings.fgr_routes.data.sample_key_a.ip2nets_file: ''
# cray_lnet.settings.fgr_routes.data.sample_key_a.routes_file: ''
# cray_lnet.settings.fgr_routes.data.sample_key_a.ko2iblnd_peer_credits: 126
# cray_lnet.settings.fgr_routes.data.sample_key_a.ko2iblnd_concurrent_sends: 63
#
# ** 'fgr_routes' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with the name of the external Lustre file system to which you are routing (***sonexion*** in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, modify the values as appropriate for this site.

This example uses "sonexion" as the name for this destination LNet.

> **IMPORTANT:** The settings for peer_credits and concurrent_sends must match between the external Lustre server and cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_peer_credits and cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_concurrent_sends.

```
# NOTE: Place additional 'fgr_routes' setting entries here, if desired.
cray_lnet.settings.fgr_routes.data.dest_name.sonexion: null
cray_lnet.settings.fgr_routes.data.sonexion.router_groups:
- lnet_fgr_routers
cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file: 'ip2nets.conf'
cray_lnet.settings.fgr_routes.data.sonexion.routes_file: 'routes.conf'
cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_peer_credits: 126
cray_lnet.settings.fgr_routes.data.sonexion.ko2iblnd_concurrent_sends: 63
#************* END Service Setting: fgr_routes *************
```

To use fine grained routing, the two configuration files (`ip2nets.conf` and `routes.conf`) must be generated using an external tool, such as `clcvt`, and then placed in `/var/opt/cray/imps/config/sets/`*`p0`*`/files/roles/lnet` (for p0 config set).

For the fine-grained routes router_groups setting, if there are no existing node groups that contain the router nodes for this site, create one or more node groups for this purpose (*`lnet_fgr_routers`* in this example) using the procedure in *Update cray_node_groups Worksheet* on page 156 and reference the node group(s) in cray_lnet.settings.fgr_routes.data.sonexion.router_groups.

There may be additional settings that should be set for sites with external Lustre servers. Seek advice from the site Lustre server administrator.

### 3.5.1.19 Update cray_local_users Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Local Users Configuration Service defines local user accounts such as root and crayadm. At a minimum, the root user account must be defined in order to log into the system.

Most sites use an external LDAP or NIS service for account management. The accounts listed in this configuration service are local accounts with entries in the `/etc/passwd` and `/etc/group` files on CLE nodes. Their home directories can be on the boot RAID file system `/cray_home`, such as for crayadm, or on an external file system. Most accounts using LDAP or NIS will have an external home directory mounted on a service node (or nodes) that are DVS-projected to the login and compute nodes.

This procedure configures some basic settings in the cray_local_users configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_local_users_worksheet.yaml`.

   ```
   smw# vi cray_local_users_worksheet.yaml
   ```

2. Ensure that `cray_local_users.enabled` is uncommented and set to `true` (it should be by default).

3. If using local home directories (most sites mount an external home file system instead), configure the home directory location in two places in this worksheet and one place in `cray_bootraid_worksheet.yaml`.

   a. Uncomment this line and replace `/cray_home` with the local home directory for this site.

   ```
   #cray_local_users.settings.directories.data.home: /cray_home
   ```

   b. Change the home directory for crayadm users.

   Look for this line in the worksheet:

   ```
   # ** 'users' DATA **
   ```

   Underneath, there are pre-populated 'users' settings for crayadm and root.

   Change the value of the 'crayadm' home directory. Replace `/cray_home/crayadm` with the crayadm home directory for this site.

   ```
   cray_local_users.settings.users.data.crayadm.home: /cray_home/crayadm
   ```

   c. Change the home directory in `cray_bootraid_worksheet.yaml`.

   On the boot RAID, the boot node volume group has a file system for home directories mounted by default on `/cray_home`. If this site does not use `cray_home` as the base for local account home directories, then change this setting in `cray_bootraid_worksheet.yaml` in the global config set to match what was set in substep a. The two settings should be in agreement about where local accounts have their home directory.

   ```
   cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes.home.fs_mo
   unt_point: /cray_home
   ```

4. Do nothing to the crayadm and root accounts "crypt" settings (in the pre-populated 'users' data section).

   Do not set the crayadm and root accounts "crypt" settings in the pre-populated 'users' data section, which must be an encrypted string. Later in this process, all of the configuration worksheets will be imported into the new CLE config set, and the config set will be updated. During the update, the configurator will prompt for the crayadm and root "crypt" settings. Because they are encrypted, the configurator will ask for the password, ask a second time to verify that they match, and then put an encrypted form of that password into the config set. Attempting to place an encrypted string into this worksheet manually is prone to error and could result in accounts that cannot be accessed.

5. Ensure that the root domain groups are uncommented.

   This parameter is also in the pre-populated 'users' settings under this line in the worksheet:

   ```
   # ** 'users' DATA **
   ```

   ```
   cray_local_users.settings.users.data.root.domain_groups
   - all_nodes
   ```

Make sure both lines are uncommented.

### 3.5.1.20 Update cray_logging Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

This procedure configures the inherit and enable settings in the Cray Logging service configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

## Procedure

1. Edit `cray_logging_worksheet.yaml`.

   ```
   smw# vi cray_logging_worksheet.yaml
   ```

2. Uncomment `cray_logging.inherit` and set it to `true`.

   This means that logging settings in the global config set will be used instead of logging settings in the CLE config set. If `cray_logging.inherit` is set to `false`, then other settings may need to be changed.

### 3.5.1.21 Update cray_login Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Login service provides information and listings for login nodes, which are used by users to access the Cray system. Also, the "nologin" feature is configured in this service. This procedure configures some basic settings in the Cray Login service configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_login_worksheet.yaml`.

   ```
   smw# vi cray_login_worksheet.yaml
   ```

2. Uncomment `cray_login.enabled` and set it to `true`.

3. Enter the node group (or groups) of the login nodes on this system.

   Cray has provided a pre-populated node group called "login_nodes" to contain the login nodes (by cname) for the system. If that node group has not yet been customized for this site, see *Update cray_node_groups Worksheet* on page 156.

   Uncomment `cray_login.settings.login_nodes.data.member_groups`, remove the empty list (`[]`), and add that node group (and any other node groups, as needed) on a separate line prefixed by a hyphen and space (`- `).

   ```
   cray_login.settings.login_nodes.data.member_groups:
   - login_nodes
   ```

4. Uncomment `cray_login.settings.login_nodes.data.login_prohibited_after_boot` and do one of the following:

   - Set it to `false` to have the `/etc/nologin` file removed automatically on each node in the list of login node groups (set in step 3) as it completes its boot.
   - Set it to `true` to require a system administrator to remove `/etc/nologin` on each node by running a command like the following after all of the CLE nodes have been booted and the system is ready for users to log in. This command could be added to the boot automation file.

     ```
     sdb# pcmd -r -n ALL_SERVICE "rm /etc/nologin"
     ```

### 3.5.1.22 Update cray_lustre_client Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Lustre Client configuration service is used to configure Lustre clients on an XC system. This procedure configures some basic settings in the cray_lustre_client configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_lustre_client_worksheet.yaml`.

   ```
   smw# vi cray_lustre_client_worksheet.yaml
   ```

2. Uncomment `cray_lustre_client.enabled` and do one of the following:

   - Set it to `false` for systems that are NOT a Lustre client of either an external Lustre server or direct-attached Lustre (DAL). Skip the rest of the procedure.
   - Set it to `true` for systems that are a Lustre client of either an external Lustre server or DAL. Proceed to the next step.

3. Configure a client mount for each Lustre file system that will be mounted.

Repeat this step for each client mount.

In the worksheet, copy the lines below `# ** EXAMPLE 'client_mounts' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'client_mounts' setting entries here, if desired`.

```
# ** EXAMPLE 'client_mounts' VALUE (with current defaults) **
# cray_lustre_client.settings.client_mounts.data.fs_name.sample_key_a: null <-- setting a multival
key
# cray_lustre_client.settings.client_mounts.data.sample_key_a.lustre_fs_name: ''
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_point: ''
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mgs_lnet_nids: []
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_options: rw,flock,lazystatfs
# cray_lustre_client.settings.client_mounts.data.sample_key_a.mount_at_boot: true
# cray_lustre_client.settings.client_mounts.data.sample_key_a.client_groups:
# - login_nodes
# - compute_nodes
# - elogin_nodes
#
```

Uncomment the lines, replace `sample_key_a` with a string that identifies that mount (`snx11023` in the example below), then remove the `<-- setting a multival key` text at the end of the first line in each set (note that the `null` value is required; do not remove or change it). Finally, modify the values as needed for this site.

**Example of mounting an external Lustre file system.**

This example uses a Sonexion with a file system called snx11023 that is mounted on `/lus/snx11023` by three node groups: login nodes, compute nodes, and eLogin nodes. All of them mount the file system as the node is booting.

```
# NOTE: Place additional 'client_mount' setting entries here, if desired.
cray_lustre_client.settings.client_mounts.data.fs_name.snx11023: null
cray_lustre_client.settings.client_mounts.data.snx11023.lustre_fs_name: snx11023
cray_lustre_client.settings.client_mounts.data.snx11023.mount_point: /lus/snx11023
cray_lustre_client.settings.client_mounts.data.snx11023.mgs_lnet_nids:
- 10.149.4.3@o2ib
- 10.149.4.4@o2ib
cray_lustre_client.settings.client_mounts.data.snx11023.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.snx11023.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.snx11023.client_groups:
- login_nodes
- compute_nodes
- elogin_nodes

#************ END Service Setting: client_mounts ************
```

**Example of mounting an internal Lustre file system (DAL).**

The following example shows values for two DAL client mounts: one for login nodes (first set of lines) and one for compute nodes (second set of lines).

The Lustre file system is started via lustre_control in the boot automation file after all service nodes have booted. The boot automation file then has a step to mount the Lustre file system on any service nodes that need to mount it. So those service nodes cannot have "mount_at_boot" set to true. However, the compute nodes are booted after the DAL file system has been started, so they can have "mount_at_boot" set to true. This example uses a DAL server with a file system called dal that is mounted on `/lus/dal` by three node groups: login nodes, eLogin nodes, and compute nodes. Only the compute nodes mount the file system as the node is booting (mount_at_boot=true).

```
# NOTE: Place additional 'client_mount' setting entries here, if desired.
cray_lustre_client.settings.client_mounts.data.fs_name.dal_login: null
cray_lustre_client.settings.client_mounts.data.dal_login.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_login.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_login.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_login.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_login.mount_at_boot: false
cray_lustre_client.settings.client_mounts.data.dal_login.client_groups:
- login_nodes
- elogin_nodes

cray_lustre_client.settings.client_mounts.data.fs_name.dal_compute: null
cray_lustre_client.settings.client_mounts.data.dal_compute.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.dal_compute.client_groups:
- compute_nodes

#************ END Service Setting: client_mounts *************
```

4. Verify that the node groups referenced in step *3* on page 140 have been accurately defined for this site.

   To verify, edit `cray_node_groups_worksheet.yaml` and search for these node groups:

   ```
   login_nodes
   compute_nodes
   elogin_nodes
   ```

DAL servers also need to be configured. See *Update cray_lustre_server Worksheet* on page 142. Further configuration of DAL occurs later in the installation process.

### 3.5.1.23 Update cray_lustre_server Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Lustre server configuration service should be enabled and configured only if this system uses direct-attached Lustre (DAL). It enables configuration of Lustre-server-related kernel module parameters. This procedure configures some basic settings in the cray_lustre_server configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_lustre_server_worksheet.yaml`.

   ```
   smw# vi cray_lustre_server_worksheet.yaml
   ```

2. Uncomment `cray_lustre_server.enabled` and do one of the following:

   - Set it to `false` for system that do not use DAL (direct-attached Lustre). Skip the remaining steps.
   - Set it to `true` for systems that do use DAL. Proceed to the next step.

3. (Only for systems with DAL) Enter the node group that contains the Lustre Management Server (MGS) node on this system.

   To see which node group contains the MGS node (by cname) or to create such a node group for this system (*`MGS_NODE_GROUP`* in this example), edit `cray_node_groups_worksheet.yaml`.

   Uncomment `cray_lustre_server.settings.lustre_servers.data.mgs_group`, remove the empty list (`[]`), and add that node group on a separate line prefixed by a hyphen and space (`- `).

   ```
   cray_lustre_server.settings.lustre_servers.data.mgs_group:
   - MGS_NODE_GROUP
   ```

4. (Only for systems with DAL) Enter the node group(s) that contain the Lustre MetaData Server (MDS) nodes on this system.

   To see which node group(s) contain the MDS nodes (by cname) or to create that node group(s) for this system (*`MDS_NODE_GROUP_1`* and *`MDS_NODE_GROUP_2`* in this example), edit `cray_node_groups_worksheet.yaml`.

   Uncomment `cray_lustre_server.settings.lustre_servers.data.mds_groups`, remove the empty list (`[]`), and add the node group(s) on a separate line prefixed by a hyphen and space (`- `).

   ```
   cray_lustre_server.settings.lustre_servers.data.mds_groups:
   - MDS_NODE_GROUP_1
   - MDS_NODE_GROUP_2
   ```

5. (Only for systems with DAL) Enter the node group(s) that contain the Lustre Object Storage Server (OSS) nodes on this system.

   To see which node group(s) contain the OSS nodes (by cname) or to create that node group(s) for this system (*`OSS_NODE_GROUP_1`* and *`OSS_NODE_GROUP_2`* in this example), edit `cray_node_groups_worksheet.yaml`.

   Uncomment `cray_lustre_server.settings.lustre_servers.data.oss_groups`, remove the empty list (`[]`), and add the node group(s) on a separate line prefixed by a hyphen and space (`- `).

   ```
   cray_lustre_server.settings.lustre_servers.data.oss_groups:
   - OSS_NODE_GROUP_1
   - OSS_NODE_GROUP_2
   ```

6. (Only for systems with DAL) Set Lustre kernel module parameters, as needed.

   This worksheet contains additional settings that tune the Lustre kernel modules. Seek advice from the site Lustre server administrator before changing them.

### 3.5.1.24 Update cray_multipath Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray multipath service provides a means to support redundant paths to a device for failover or performance reasons. Multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

> **NOTE:** (SMW HA only) Cray recommends configuring multipath before configuring and enabling HA. If HA is configured and enabled first, then additional precautions must be taken when enabling multipath, as documented in *XC™ Series SMW HA Installation Guide*.

The multipath configuration service has a global template as well as a CLE template, and therefore the service can be configured to inherit settings from the global config set or use settings from the CLE config set(s), if there is a need to have it configured differently in different config sets. If multipath configuration is desired on the management node (SMW) as well as CLE nodes, Cray recommends enabling this service in the global config set and configuring it there for both SMW and CLE nodes. The multipath service in the CLE config sets would then inherit the global configuration data.

This procedure configures the inherit setting and possibly some other settings in the Cray multipath service configuration worksheet in a CLE config set.

## Procedure

1. Edit `cray_multipath_worksheet.yaml`.

   ```
   smw# vi cray_multipath_worksheet.yaml
   ```

2. Uncomment `cray_multipath.inherit` and set it to one of the following values:

   - Set it to true to manage multipath settings in the global config set instead of in the CLE config set. If this option is chosen, skip the rest of the steps.
   - Set it to false to manage multipath settings in one or more CLE config sets instead of in the global config set. If this option is chosen, continue to the next step.

3. (If `inherit` set to false) Uncomment `cray_multipath.enabled`.

   Set it to `true` if this site desires to use multipath, otherwise set it to `false`. If enabling this service, continue to the next step.

4. (If `enabled` set to true) Complete the configuration of multipath.

   a. Enter the list of multipath nodes.

      Uncomment `cray_multipath.settings.multipath.data.node_list`, remove the `[]` (denotes empty list), and add a list of nodes (by cname or host ID) in this system that have multipath devices and need to have multipath configured. For sites with boot node failover and/or SDB node failover, Cray recommends adding both the active and passive (failover) nodes to this list.

      This example shows a list of three nodes: an SMW with host ID 1eac4e0c, a boot node with cname c0-0c0s4n1, and an SDB node with cname c0-0c0s3n1.

      ```
      cray_multipath.settings.multipath.data.node_list:
      - 1eac4e0c
      - c0-0c0s4n1
      - c0-0c0s3n1
      ```

   b. Configure enabled devices.

Cray has provided a number of enabled devices with pre-populated data under `# ** 'enabled_devices' DATA **`. These storage devices are the devices that will be whitelisted, which means they will be listed as exceptions to the blacklist. The settings for these devices have default values provided by the device vendors and do not need to be changed. If this site intends to configure a multipath device that does not appear in this group of enabled devices, contact a Cray representative for help.

c. (Optional) Configure aliases for the multipath devices.

This is the equivalent of adding aliases to the multipaths section of the `multipath.conf` file.

In the worksheet, copy the two lines below `# ** EXAMPLE 'aliases' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'aliases' setting entries here, if desired.`

```
# ** EXAMPLE 'aliases' VALUE  (with current defaults) **
#   cray_multipath.settings.aliases.data.wwid.sample_key_a: null    <-- setting a multival key
#   cray_multipath.settings.aliases.data.sample_key_a.alias: ''
#
```

Uncomment the lines, replace `sample_key_a` with the World Wide Identifier (WWID) of the device to be aliased (60080e50002e203c00002a085551b2c8 in this example) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, add the alias for this device (smw_node_pv1 in this example). Repeat this substep for each device, as needed.

```
# NOTE: Place additional 'aliases' setting entries here, if desired.
cray_multipath.settings.aliases.data.wwid.60080e50002e203c00002a085551b2c8: null
cray_multipath.settings.aliases.data.60080e50002e203c00002a085551b2c8.alias: smw_node_pv1
#***************** END Service Setting: aliases ****************
```

d. Correct the default values for three pre-populated device settings.

The default values of the following variables are incorrect in `cray_multipath_worksheet.yaml` for this release. In the worksheet, find these variables and change their values as indicated.

```
enabled_devices.data.DDN_SFA12K_20.product: SFA12K-20

enabled_devices.data.DDN_SFA12K_40.product: SFA12K-40|SFA12KX*

enabled_devices.data.DDN_EF3015.path_grouping_policy: group_by_prio
```

### 3.5.1.25 Update cray_munge Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Cray MUNGE is an authentication service that creates and validates credentials. It is required by the DataWarp service (cray_dws), Slurm (a workload manager), and Dynamic RDMA Credentials (cray_drc). This procedure enables/disables the cray_munge service.

## Procedure

1. Edit `cray_munge_worksheet.yaml`.

   ```
   smw# vi cray_munge_worksheet.yaml
   ```

2. Uncomment `cray_munge.enabled` and do one of the following:

   - Set it to `true` only if this site wishes to enable the DataWarp service or Slurm while doing a fresh install of SMW/CLE software or if this site is using Dynamic RDMA Credentials.
   - Set it to `false` otherwise.

   If the MUNGE service was disabled in this step, it can be enabled later when configuring DataWarp or Slurm.

### 3.5.1.26 Update cray_net Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Networking configuration service defines all network information for CLE nodes, which is necessary for a functional system. This procedure configures some basic settings in the cray_net configuration worksheet to add site-specific data.

> **REMEMBER:** For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Follow this procedure to make changes to the `cray_net_worksheet.yaml` for each partition. Some steps call out what settings should be different for different partitions.

There are two major sections to cray_net. One describes the networks to which the CLE nodes are connected, and the other describes the hosts and the network interfaces on each host that are on those networks.

**Networks.** All networks for CLE nodes must be defined here. The high speed network (HSN) will be connected to the ipogif0 inteface on each CLE node. The login network will be used by the internal login (or network gateway) nodes to an external-to-XC network. Any additional number of networks can be added or described using unique network names, such as for an InfiniBand network or a 40GigEthernet network.

**Hosts and network interfaces.** Host entries in cray_net are used to describe specific information about a host that has network interfaces or to make a host name alias in `/etc/hosts`. Every CLE node does not need to be listed here because the IP address, nid name, and cname entry in the `/etc/hosts` file will be generated based on the address range of the HSN.

Notes on editing a configuration worksheet:

- To enter a value for a string that currently is set to `''` (empty string), replace the quotes with the new value. For example, `ipv4_network: ''` becomes `ipv4_network: 10.1.0.0`. In cases where the string value might be interpreted as a number, retain the single quotes. For example, a string setting with value '512' needs quotes.

- To enter one or more values for a list that is currently set to `[]` (empty list), remove the brackets and add each entry on a separate line, preceded by a dash and a space (`- `).
- Do NOT change or remove the null value in lines like this that appear at the beginning of each set of network, host, or host interface definitions. This line sets the key, or identifier, for that definition. In this example, "hsn" is the identifier for the HSN network definition.

```
cray_net.settings.networks.data.name.hsn: null
```

## Procedure

1. Edit `cray_net_worksheet.yaml`.

```
smw# vi cray_net_worksheet.yaml
```

2. Uncomment `cray_net.enabled` and ensure that it is set to `true`.

3. Uncomment these two settings for the HSN (high speed network).

If this is a partitioned system, then enter different values for these settings. Partitions p1 and p2 will not have the same ipv4 network, but will have similar ipv4_netmask (though different from the full machine).

```
# ** 'networks' DATA **
cray_net.settings.networks.data.hsn.ipv4_network: 10.128.0.0
cray_net.settings.networks.data.hsn.ipv4_netmask: 255.252.0.0
```

DEFINE NETWORKS --------------------------------

**IMPORTANT:**

- Add values for the `dns_servers` and `dns_search` settings to the login network only, not to any other network.
- DO NOT add a value for the `ntp_servers` setting for any network used for CLE nodes, because CLE nodes must source their time/NTP settings from the SMW rather than try to contact NTP servers on the login network.

4. Configure a login network and add the information for the "Customer network" to which the login nodes connect.

Scroll down to the pre-populated network settings below the `# ** 'networks' DATA **` line and find the login network definition. Uncomment the commented lines and modify the values as needed for this site's internal systems. Note that in the first line, the `null` value is required; do not remove or change it.

**NOTE:** If this site does not use DNS search but does use DNS domain in `/etc/resolv.conf`, then adding a single entry to the `dns_search` setting is functionally equivalent to setting the DNS domain.

```
# ** 'networks' DATA **
...
cray_net.settings.networks.data.name.login: null
cray_net.settings.networks.data.login.description: Customer network
cray_net.settings.networks.data.login.ipv4_network: 172.30.48.0
cray_net.settings.networks.data.login.ipv4_netmask: 255.255.240.0
cray_net.settings.networks.data.login.ipv4_broadcast: ''
cray_net.settings.networks.data.login.ipv4_gateway: 172.30.48.1
cray_net.settings.networks.data.login.dns_servers:
- 172.30.84.40
- 172.31.84.40
- 172.28.84.40
cray_net.settings.networks.data.login.dns_search:
- us.cray.com
- americas.cray.com
- cray.com
cray_net.settings.networks.data.login.ntp_servers: []
cray_net.settings.networks.data.login.fw_external: false
```

**IMPORTANT:** If the login network should be treated as an external network for the firewall, then set `cray_net.settings.networks.data.login.fw_external` (the last line in the example) to true.

**5.** Configure additional networks, as needed for this system.

In the worksheet, copy the ten lines below `# ** EXAMPLE 'networks' VALUE (with current defaults) **` and paste one set for each network below the line `# NOTE: Place additional 'networks' setting entries here, if desired.`

```
# ** EXAMPLE 'networks' VALUE  (with current defaults) **
# cray_net.settings.networks.data.name.sample_key_a: null <-- setting a multival key
# cray_net.settings.networks.data.sample_key_a.description: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_network: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_netmask: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_broadcast: ''
# cray_net.settings.networks.data.sample_key_a.ipv4_gateway: ''
# cray_net.settings.networks.data.sample_key_a.dns_servers: []
# cray_net.settings.networks.data.sample_key_a.dns_search: []
# cray_net.settings.networks.data.sample_key_a.ntp_servers: []
# cray_net.settings.networks.data.sample_key_a.fw_external: false

# ** 'networks' FIELD SPECIFICATION -- MULTIVAL KEY FIELD **
```

Uncomment the lines, replace `sample_key_a` with an identifier for the network (lnet and ethernet40gig in the example below) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site.

The following example shows two additional networks. The first is a single Infiniband network (lnet) used for the LNet router nodes. Sites that have more than one InfiniBand network will need to add more networks like this first one. The second network has been defined for nodes that have 40GigEthernet interfaces (ethernet40gig), and for such networks, the fw_external variable must be set to true.

```
# NOTE: Place additional 'networks' setting entries here, if desired.
cray_net.settings.networks.data.name.lnet: null
cray_net.settings.networks.data.lnet.description: The InfiniBand network for
LNet router nodes to external Lustre server
cray_net.settings.networks.data.lnet.ipv4_network: 10.150.0.0
cray_net.settings.networks.data.lnet.ipv4_netmask: 255.255.0.0
cray_net.settings.networks.data.lnet.ipv4_broadcast: ''
cray_net.settings.networks.data.lnet.ipv4_gateway: ''
cray_net.settings.networks.data.lnet.dns_servers: []
cray_net.settings.networks.data.lnet.dns_search: []
cray_net.settings.networks.data.lnet.ntp_servers: []
cray_net.settings.networks.data.lnet.fw_external: false

cray_net.settings.networks.data.name.ethernet40gig: null
cray_net.settings.networks.data.ethernet40gig.description:
Network for 40GigEthernet
cray_net.settings.networks.data.ethernet40gig.ipv4_network: 138.55.19.0
cray_net.settings.networks.data.ethernet40gig.ipv4_netmask: 255.255.255.0
cray_net.settings.networks.data.ethernet40gig.ipv4_broadcast: ''
cray_net.settings.networks.data.ethernet40gig.ipv4_gateway: ''
cray_net.settings.networks.data.ethernet40gig.dns_servers: []
cray_net.settings.networks.data.ethernet40gig.dns_search: []
cray_net.settings.networks.data.ethernet40gig.ntp_servers: []
cray_net.settings.networks.data.ethernet40gig.fw_external: true
#************ END Service Setting: networks ************
```

DEFINE HOSTS AND THEIR NETWORK INTERFACES ---------------------------------------

**6.** Configure a host as the boot node.

Cray has defined a default `bootnode` host, which is located under the `# ** 'hosts' DATA **` line. Every system has this host.

> **IMPORTANT:** Never set `cray_net.settings.hosts.data.bootnode.aliases` to "boot" because that is a host name alias that belongs to the virtual IP address for the boot node in support of the boot node failover feature.

a. Configure the host ID of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.hostid` and set it to the cname of the boot node.

b. Configure the host name of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.hostname` and set it to the host name of the boot node.

Do not set the host name to "boot" because that name is reserved for the virtual IP address of the boot node, regardless of whether it is the full system or a partitioned system. Choose a name that includes the machine name and "boot" such as "boot-panda," or if this is a partitioned system, then identify the boot node as "boot-p1," "boot-p2," and so forth.

c. Configure the primary Ethernet interface of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.interfaces.primary_ethernet.ipv4_address` and set it as follows. This is on the "admin" network to the SMW.

- 10.3.1.254 for a full system (p0).
- 10.3.1.254 for p1, 10.3.1.252 for p2, and so forth for partitioned systems.

d. Configure the HSN boot alias interface of the boot node.

Uncomment `cray_net.settings.hosts.data.bootnode.interfaces.hsn_boot_alias.ipv4_address` and set it as follows. This is on the HSN and is the "'virtual IP address" for the virtual interface ipogif0:1.

- 10.131.255.254 for a full system (p0).
- The highest address possible for a partition's HSN, for partitioned systems. For example, if p1 HSN `ipv4_address`=10.128.0.0 with `ipv4_netmask` 255.255.0.0, then set `ipv4_address`=10.128.255.254 for p1. If p2 HSN `ipv4_address`=10.129.0.0 with `ipv4_netmask` 255.255.0.0, then set `ipv4_address`=10.129.255.254 for p2.

**7.** Configure a host as the SDB node.

Cray has defined a default `sdbnode` host, which is located under the `# ** 'hosts' DATA **` line. Every system has this host.

a. Configure the host ID of the SDB node.

   Uncomment `cray_net.settings.hosts.data.sdbnode.hostid` and set it to the cname of the SDB node.

b. Configure the host name of the SDB node.

   Uncomment `cray_net.settings.hosts.data.sdbnode.hostname` and set it to "sdb."

c. Configure the primary Ethernet interface of the SDB node.

   Uncomment `cray_net.settings.hosts.data.sdbnode.interfaces.primary_ethernet.ipv4_address` and set it as follows. This is on the "admin" network to the SMW.

   - 10.3.1.253 for a full system (p0).
   - 10.3.1.253 for p1, 10.3.1.251 for p2, and so forth for partitioned systems.

d. Configure the HSN SDB alias interface of the SDB node.

   Uncomment `cray_net.settings.hosts.data.sdbnode.interfaces.hsn_sdb_alias.ipv4_address` and set it as follows. This is on the HSN and is the "'virtual IP address" for the virtual interface ipogif0:1.

   - 10.131.255.253 for a full system (p0).
   - The highest address possible for a partition's HSN, for partitioned systems.

     For example, if p1 HSN `ipv4_address`=10.128.0.0 with `ipv4_netmask` 255.255.0.0, then set `ipv4_address`=10.128.255.253 for p1. If p2 HSN `ipv4_address`=10.129.0.0 with `ipv4_netmask` 255.255.0.0, then set `ipv4_address`=10.129.255.253 for p2.

**8.** Configure a host as the login node.

a. Configure the aliases of the login node.

   Uncomment `cray_net.settings.hosts.data.login_node.aliases` and set it to a list of aliases, as follows.

   - If this site wishes the login node to have a host name alias of "login:"

```
cray_net.settings.hosts.data.login_node.aliases:
- login
```

   - If this site has more than one login node, the first one could have aliases of "login" and "login1," and the others would be set to "login2," "login3," and so forth.

```
cray_net.settings.hosts.data.login_node.aliases:
- login
- login1
```

b. Configure the host ID of the login node.

   Uncomment `cray_net.settings.hosts.data.login_node.hostid` and set it to the cname of the login node. If this system has more than one login node, set this variable to the first login node.

c. Configure the host name of the login node.

   Uncomment `cray_net.settings.hosts.data.login_node.hostname` and set it to the host name.

This could be the machine name, for systems that have only one login node. For example, on a machine known as panda, this would be "panda." For systems with more than one login node, the host name could be "panda1" for the first one, "panda2" for the second one, and so forth.

d. Configure the login Ethernet interface of the login node.

Uncomment `cray_net.settings.hosts.data.login_node.interfaces.login_ethernet.ipv4_address` and set it to the IP address of the login node's eth0 interface on the "login" network.

**9.** Configure additional hosts, as needed.

If this system has additional service nodes that need to have host name or host name alias or network interface settings, then for each one add a host definition stanza like the following, placing it under `NOTE: Place additional 'hosts' setting entries here, if desired`. The first example shows the host configuration of a DVS node (dvs_node) with the host name set to "dvs1," a host name alias of "dvs," and one Ethernet interface connected to the "login" network.

```
cray_net.settings.hosts.data.common_name.dvs_node: null
cray_net.settings.hosts.data.dvs_node.description: DVS node
cray_net.settings.hosts.data.dvs_node.aliases:
- dvs
cray_net.settings.hosts.data.dvs_node.hostid: c0-0c0s0n2
cray_net.settings.hosts.data.dvs_node.host_type: ''
cray_net.settings.hosts.data.dvs_node.hostname: dvs1
cray_net.settings.hosts.data.dvs_node.standby_node: false

cray_net.settings.hosts.data.dvs_node.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.description: Ethernet
    connecting the DVS node to the customer network.
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.network: login
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.ipv4_address: 172.30.50.128
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.startmode: auto
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.bootproto: static
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.extra_attributes: []
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.module: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.params: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.unmanaged_interface: false
```

The second example shows the host configuration for an LNet router node (clfs_lnet_1) that has two different InfiniBand interfaces (ib0 and ib2) to connect to two different networks.

> **NOTICE:** In this example, the interface parameter `mtu` for both interfaces is set to a numerical value within single quotes. The quotes are important. The configurator expects a string for this setting, and without the single quotes, it could interpret this value as a number and return an error. The values provided for other parameters of type string do not need single quotes because they would not be interpreted as anything other than strings.

```
cray_net.settings.hosts.data.common_name.clfs_lnet_1: null
cray_net.settings.hosts.data.clfs_lnet_1.description: CLFS router 1 node
cray_net.settings.hosts.data.clfs_lnet_1.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.hostid: c0-0c1s0n1
cray_net.settings.hosts.data.clfs_lnet_1.host_type: ''
cray_net.settings.hosts.data.clfs_lnet_1.hostname: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.standby_node: false

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib0: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.description: InfiniBand
    ib0 connecting the CLFS router 1 node to the lnet network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.ipv4_address: 10.150.10.65
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.unmanaged_interface: false

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib2: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.name: ib2
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.description: InfiniBand
    ib2 connecting the CLFS router 1 node to the lnet1 network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.network: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.ipv4_address: 10.151.10.65
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.bootproto: static
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.unmanaged_interface: false
```

**10.** Configure a host as the second boot node for boot node failover.

If using the boot node failover feature, then define a backup boot node host with the "standby_node" variable set to true.

> **NOTE:** The host name for the primary and backup boot node should both be set to "boot." The aliases can be different so that the `/etc/hosts` entry for the cname has the host name alias.

```
cray_net.settings.hosts.data.common_name.backup_bootnode: null
cray_net.settings.hosts.data.backup_bootnode.description: backup Boot node for the system
cray_net.settings.hosts.data.backup_bootnode.aliases:
- cray-boot2
cray_net.settings.hosts.data.backup_bootnode.hostid: c0-0c0s4n1
cray_net.settings.hosts.data.backup_bootnode.host_type: admin
cray_net.settings.hosts.data.backup_bootnode.hostname: boot
cray_net.settings.hosts.data.backup_bootnode.standby_node: true

cray_net.settings.hosts.data.backup_bootnode.interfaces.common_name.hsn_boot_alias: null
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.name: ipogif0:1
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.description: Well known
address used for boot node services.
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.aliases: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.network: hsn
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.ipv4_address: 10.131.255.254
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.mac: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.startmode: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.bootproto: static
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.mtu: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.extra_attributes: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.module: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.params: ''
#cray_net.settings.hosts.data.backup_bootnode.interfaces.hsn_boot_alias.unmanaged_interface: false

cray_net.settings.hosts.data.backup_bootnode.interfaces.common_name.primary_ethernet: null
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.name: eth0
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.description: Ethernet
connecting boot node to the SMW.
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.aliases: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.network: admin
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.ipv4_address: 10.3.1.254
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.mac: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.startmode: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.bootproto: static
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.mtu: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.extra_attributes: []
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.module: ''
cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.params: ''
#cray_net.settings.hosts.data.backup_bootnode.interfaces.primary_ethernet.unmanaged_interface: false
```

**11.** Configure a host as the second SDB node for SDB node failover.

If using the SDB node failover feature, then define a backup SDB node host with the "standby_node" variable set to true.

> **NOTE:** The host name for the primary and backup SDB node should both be set to "sdb." The aliases can be different so that the `/etc/hosts` entry for the cname has the host name alias.

```
cray_net.settings.hosts.data.common_name.backup_sdbnode: null
cray_net.settings.hosts.data.backup_sdbnode.description: backup SDB node for the system
cray_net.settings.hosts.data.backup_sdbnode.aliases:
- cray-sdb2
cray_net.settings.hosts.data.backup_sdbnode.hostid: c0-0c0s3n1
cray_net.settings.hosts.data.backup_sdbnode.host_type: admin
cray_net.settings.hosts.data.backup_sdbnode.hostname: sdb
cray_net.settings.hosts.data.backup_sdbnode.standby_node: true

cray_net.settings.hosts.data.backup_sdbnode.interfaces.common_name.hsn_boot_alias: null
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.name: ipogif0:1
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.description: Well known
address used for SDB node services.
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.aliases: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.network: hsn
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.ipv4_address: 10.131.255.253
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.mac: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.startmode: auto
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.bootproto: static
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.mtu: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.extra_attributes: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.module: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.params: ''
#cray_net.settings.hosts.data.backup_sdbnode.interfaces.hsn_boot_alias.unmanaged_interface: false

cray_net.settings.hosts.data.backup_sdbnode.interfaces.common_name.primary_ethernet: null
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.name: eth0
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.description: Ethernet
connecting SDB node to the SMW.
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.aliases: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.network: admin
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.ipv4_address: 10.3.1.253
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.mac: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.startmode: auto
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.bootproto: static
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.mtu: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.extra_attributes: []
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.module: ''
cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.params: ''
#cray_net.settings.hosts.data.backup_sdbnode.interfaces.primary_ethernet.unmanaged_interface: false
```

**12.** Set the module and params settings for any hosts that are network nodes and use special network cards.

Sites that use special network cards (e.g., Mellanox ConnectX-3) must specify which kernel module is used by those cards. For each host that uses such a card, uncomment (if commented out) the following setting, then replace the empty string with the kernel module name.

This example specifies **`mlx4_en`**, the module for Mellanox ConnectX-3 cards.

```
cray_net.settings.hosts.data.network_node.interfaces.eth0.module: mlx4_en
```

Any kernel module parameters that need to be set for that module can be specified by uncommenting the following setting, then replacing the empty string with "parameter=value" pairs (pairs separated by spaces). This is not common; no parameters need to be specified for the **`mlx4_en`** module. Note that the = syntax may vary by kernel module; consult the documentation of the kernel module being used.

```
cray_net.settings.hosts.data.networknode.interfaces.eth0.params: param1=200 param2=30
```

### 3.5.1.27 About Configuring Netroot Preload

Netroot is a feature that enables nodes booted with a minimal, local in-memory file system to execute within the context of a larger, full-featured root file system. Netroot uses the Data Virtualization Service (DVS) to access the remote root content. While DVS has data and attribute caching features that minimize the impact of most remote references, files that are referenced frequently may still incur an undesirable performance penalty.

The Netroot Preload feature mitigates that performance penalty by copying specified remote files from the Netroot to a node-local in-memory file system early in the node boot process. All future references to those files will be serviced by the local file system rather than requiring remote data and/or metadata DVS operations. This improves system and application performance. However, as a consequence, the amount of memory available on the node is reduced by the cumulative size of all files copied into its memory.

Netroot Preload can be enabled, disabled, and customized using the configurator on the SMW or by editing the configuration worksheets on the SMW.

## Netroot Preload Configuration Settings

Netroot Preload configuration consists of defining one or more "loads," or sets of data to be preloaded on specified nodes. The load setting has the following fields:

**label** A convenient, descriptive label for a particular load.

**targets** A list of node groups that reference the nodes that will be preloaded with files on their local file systems. Must provide at least one node group.

**content_lists** Content lists are relative paths to files within the config set. These files contain file paths that are copied into the node-local memory by Netroot Preload. For example, content list `dist/compute-preload.cray` within config set p0 has these contents:

```
smw# cat p0/dist/compute-preload.cray
/etc/ld.so.cache
/opt/cray/rca/*/bin/rca-helper
/lib64/libc-*.so
/lib64/ld-*.so
/opt/cray/rca/*/lib*/librca.so.*
[...]
```

Pattern matching is supported.

**size_limit** The memory-consumption limit (in MB) set for this load, which limits how much can be copied to any node. As the files are copied via Netroot, Netroot Preload checks the sizes and amount of data copied so far. When it reaches the size limit, it stops, and any remaining files are not copied. Setting this to zero (0) indicates no limit.

## Cray Provides Default Loads

Cray provides two default loads: the 'compute' load, which targets all compute nodes in the system, and the 'login' load, which targets all internal login nodes in the system. The compute load has a single content list specified: `dist/compute-preload.cray`. This file contains paths that are commonly referenced during the node boot and initialization process. Similarly, the login load specifies this content list as the only entry in its content_lists setting: dist/login-preload.cray. Note that each of these is a relative path. The full path would be `/var/opt/cray/imps/config/sets/p0/dist/login-preload.cray` for the login content list entry. If a site disables or modifies these default settings, the time it takes to boot and initialize nodes may increase.

## Sites can Create Custom Loads to Optimize for Specific Workloads

Sites may define their own loads as well. This enables sites to optimize for specific workloads. For targets, sites can use existing node groups or define their own (see *Update cray_node_groups Worksheet* on page 156).

To determine which file paths to add to load content lists, use the DVS request log, which is enabled by default. That log was used to create the Cray default content lists. The `/proc/fs/dvs/request_log` file contains a log of all DVS requests that take more than a certain number of seconds to complete (the default is 15 seconds). Look for file paths that are referenced often; these are good candidates for Netroot Preload.

Use the following commands (as root) to view, disable, enable, and clear the DVS request log.

*Table 11. Commands to disable, enable, and clear the DVS request log*

| | |
|---|---|
| view | **`cat /proc/fs/dvs/request_log`** |
| disable | **`echo 0 > /proc/fs/dvs/request_log`** |
| enable | **`echo 1 > /proc/fs/dvs/request_log`** |
| clear | **`echo 2 > /proc/fs/dvs/request_log`** |

See "DVS Can Log Requests Sent to Servers" in *XC™ Series DVS Administration Guide* (S-0005) for additional information about this request log.

## The Netroot Preload Log File and a Note about Symlinks

Netroot Preload creates a log file on affected nodes at `/var/opt/cray/log/netroot_preload.log`. This log file contains details on the files preloaded, which, if any, files were not found in the Netroot, and the size of the files preloaded on the node. Any failures will also be logged to the console file on the SMW.

Note that any symlinks included in a load content list may not be copied from Netroot to the node-local RAM file system (i.e., "promoted" in the log file), which might look confusing. For example, suppose a site content list contains `/etc/alternatives/unzip`, which is a symlink to `/usr/bin/unzip-plain`. While both the link and its target are present in Netroot, neither of them appear in the node-local file system, despite the log saying `Promoted '/new_root/merge/etc/alternatives/unzip'`. This is expected and correct behavior. A site that is concerned about possible confusion for administrators can decide to exclude symlinks from content lists or simply list the target of the symlink in a content list to ensure that it is present in the node-local file system.

### 3.5.1.28 Update cray_netroot_preload Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Netroot Preload is a mechanism for populating a Cray system node's root file system early in the boot process to reduce load on the DVS (Data Virtualization Service) servers providing the data and thereby reduce boot times for

Netroot nodes. Netroot Preload also improves post-boot performance—how much improvement depends on the workloads. This service is needed if Netroot is used, and does no harm if Netroot is not used.

This procedure configures some settings in the cray_netroot_preload configuration worksheet to add site-specific "load" data. Cray provides two default load settings that define target nodes and files to be preloaded to them. Sites may define custom loads as well (optional). For more information, see *About Configuring Netroot Preload* on page 153.

## Procedure

1. Edit `cray_netroot_preload_worksheet.yaml`.

   ```
   smw# vi cray_netroot_preload_worksheet.yaml
   ```

2. Uncomment `cray_netroot_preload.enabled`. Keep it set to `true`, which is the default.

   Continue to step 3 to define a custom load (optional).

3. Define a custom load.

   In the worksheet, copy the four lines below `# ** EXAMPLE 'load' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'load' setting entries here, if desired.`

   ```
   # ** EXAMPLE 'load' VALUE  (with current defaults) **
   #   cray_netroot_preload.settings.load.data.label.sample_key_a: null   <-- setting a multival key
   #   cray_netroot_preload.settings.load.data.sample_key_a.targets: []
   #   cray_netroot_preload.settings.load.data.sample_key_a.content_lists: []
   #   cray_netroot_preload.settings.load.data.sample_key_a.size_limit: 0
   ```

   Uncomment the lines, replace `sample_key_a` with the label for this load (e.g., `my_load`) in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add site-specific values. Add each list element on a separate line prefixed by a hyphen and space (– ).

   ```
   # NOTE: Place additional 'load' setting entries here, if desired.
   cray_netroot_preload.settings.load.data.label.my_load: null
   cray_netroot_preload.settings.load.data.my_load.targets:
   - site-defined_node_group
   cray_netroot_preload.settings.load.data.my_load.content_lists:
   - relative/path/to/oft-requested/files
   cray_netroot_preload.settings.load.data.my_load.size_limit: 0
   #************ END Service Setting: load ************
   ```

### 3.5.1.29 Update cray_node_groups Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Node Groups are a mechanism for defining logical groupings of Cray system nodes to streamline node specifications for use in other Cray configuration services. The node groups defined are non-exclusive, that is, a node may belong to more than one node group. They are referenced in other configuration templates and are used in Ansible plays as well. For more information, see *About Node Groups* on page 18.

This procedure configures some basic settings in the Cray Node Groups service configuration worksheet to add site-specific data. For an explanation of the long variable names in configuration settings, see *About Variable Names in the Configurator and Configuration Worksheets* on page 16.

## Procedure

1. Edit `cray_node_groups_worksheet.yaml`.

   ```
   smw# vi cray_node_groups_worksheet.yaml
   ```

2. Uncomment `cray_node_groups.enabled` and ensure that it is set to `true`.

3. Customize pre-populated node groups.

   These pre-populated (default) node groups are provided by Cray, but sites must customize the `members` setting for most of the node groups. For example, the host ID of the SMW is 1eac199c in the first node group, "smw_nodes," but this must be replaced by the actual host ID for the SMW at this site. For more information about changing these default settings, including the use of additional platform keywords for finer-grained groupings, see *About Node Groups* on page 18.

   ```
   # ** 'groups' DATA **

   cray_node_groups.settings.groups.data.group_name.compute_nodes: null
   cray_node_groups.settings.groups.data.compute_nodes.description: Default node
       group which contains all of the compute nodes for the current partition.
   cray_node_groups.settings.groups.data.compute_nodes.members:
   - platform:compute

   cray_node_groups.settings.groups.data.group_name.service_nodes: null
   cray_node_groups.settings.groups.data.service_nodes.description: Default node
       group which contains all of the service nodes for the current partition.
   cray_node_groups.settings.groups.data.service_nodes.members:
   - platform:service

   cray_node_groups.settings.groups.data.group_name.smw_nodes: null
   cray_node_groups.settings.groups.data.smw_nodes.description: Default node
       group which contains the primary and failover (if applicable) SMW nodes.
   cray_node_groups.settings.groups.data.smw_nodes.members:
   - 1eac199c

   cray_node_groups.settings.groups.data.group_name.boot_nodes: null
   cray_node_groups.settings.groups.data.boot_nodes.description: Default node
       group which contains the primary and failover (if applicable) boot
       nodes associated with the current partition.
   cray_node_groups.settings.groups.data.boot_nodes.members:
   - c0-0c0s0n1

   cray_node_groups.settings.groups.data.group_name.sdb_nodes: null
   cray_node_groups.settings.groups.data.sdb_nodes.description: Default node
       group which contains the primary and failover (if applicable) SDB
   ```

```
    nodes associated with the current partition.
cray_node_groups.settings.groups.data.sdb_nodes.members:
- c0-0c0s1n1

cray_node_groups.settings.groups.data.group_name.login_nodes: null
cray_node_groups.settings.groups.data.login_nodes.description: Default node
    group which contains the login nodes for the configured system.
cray_node_groups.settings.groups.data.login_nodes.members:
- c0-0c0s2n2

cray_node_groups.settings.groups.data.group_name.all_nodes: null
cray_node_groups.settings.groups.data.all_nodes.description: Default node
    group which contains all of the nodes applicable to the current system.
    May also contain SMW nodes and external login nodes.
cray_node_groups.settings.groups.data.all_nodes.members:
- platform:compute
- platform:service

cray_node_groups.settings.groups.data.group_name.tier2_nodes: null
cray_node_groups.settings.groups.data.tier2_nodes.description: Default node
    group which contains the tier2 nodes in the system. See the guidance in
    the cray_scalable_services service for a detailed description of tier2
    nodes.
cray_node_groups.settings.groups.data.tier2_nodes.members:
- c0-0c0s8n0
- c0-0c0s15n0
```

To help with selecting nodes to be tier2 servers, here is a tier2 node FAQ:

| | |
|---|---|
| **Q. How many tier2 nodes are needed?** | **A.** At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400. |
| **Q. Will adding more tier2 nodes help performance?** | **A.** Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns. |
| **Q. What kind of node can be used as a tier2 node?** | **A.** Use these:<br>● OPTIMAL: dedicated repurposed compute nodes (RCN)<br>● dedicated service nodes<br>● nodes with uniform light to moderate load<br>● nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability<br>AVOID these (will result in sub-optimal performance):<br>● nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors<br>● direct-attached Lustre (DAL) servers<br>● RSIP (realm-specific IP) servers<br>● login nodes |
| **Q. Can a tier2 node have more than one role?** | **A.** Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated. |
| **Q. Where should tier2 nodes be placed?** | **A.** Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure. |

**4.** Define a custom node group, as needed.

Repeat this step for each additional node group.

Copy the three commented lines under `** EXAMPLE 'groups' VALUE (with current defaults) **` and paste them under `# NOTE: Place additional 'groups' setting entries here, if desired.`

```
** EXAMPLE 'groups' VALUE (with current defaults) **
#cray_node_groups.settings.groups.data.group_name.sample_key_a: null <--setting a multival key
#cray_node_groups.settings.groups.data.sample_key_a.decription: ''
#cray_node_groups.settings.groups.data.sample_key_a.members: []
```

Uncomment the lines, replace `sample_key_a` with the identifier chosen for the node group in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). Finally, add values for the `description` (a string) and `members` (a list) fields. For the `members` field, add each list element on a separate line prefixed by a hyphen and space (`- `).

As an example, here is the definition of a node group called lnet_nodes, which could be the list of LNet router nodes to an external Lustre file system.

```
# NOTE: Place additional 'groups' setting entries here, if desired.
cray_node_groups.settings.groups.data.group_name.lnet_nodes: null
cray_node_groups.settings.groups.data.lnet_nodes.decription: Node group that
contains all the LNet router nodes
cray_node_groups.settings.groups.data.lnet_nodes.members:
- c0-0c2s1n1
- c0-2c2s1n2
#********** END Service Setting: groups *********
```

### Other custom node groups

Other useful custom node groups might be: an rsip_nodes (for RSIP server nodes), mom_nodes (for MOM nodes with a workload manager), dvs_nodes (for a node DVS-projecting an external file system to internal nodes), datawarp_nodes (for the DataWarp SSD-endowed nodes), or postproc_nodes (for MAMU nodes in the former CLE 5.2 / SMW 7.2 postproc node_class).

The following table lists all of the CLE configuration services that require node groups for one or more variables. In some cases, a custom node group may need to be defined.

| | | |
|---|---|---|
| cray_alps | cray_lnet | cray_persistent_data |
| cray_auth | cray_local_users | cray_rsip |
| cray_boot | cray_login | cray_scalable_services |
| cray_dvs | cray_lustre_client | cray_sdb |
| cray_dws | cray_lustre_server | cray_simple_shares |

### A custom node group for use with Simple Sync

Node groups can be used in conjunction with Simple Sync to distribute some files to members of a node group. Here is an example of a custom node group called 'automount' that would have an associated 'automount' directory in the Simple Sync directory structure on the SMW (in `/var/opt/cray/imps/config/sets/p0/files/simple_sync/nodegroups`), which could be used to distribute automount maps to the nodes in that node group (for more information, about the Simple Sync directory structure, see *About Simple Sync* on page 20 or see `/var/opt/cray/imps/config/sets/p0/files/simple_sync/README`).

```
# NOTE: Place additional 'group' setting entries here, if desired.
cray_node_groups.settings.groups.data.group_name.automount: null
cray_node_groups.settings.groups.data.automount.decription: Node group that
contains all the service nodes which will get automount maps via Simple Sync
cray_node_groups.settings.groups.data.automount.members:
- c0-0c1s4n2
#********** END Service Setting: groups **********
```

### 3.5.1.30 Update cray_node_health Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Node Health service configures the Cray Node Health Checker (NHC). The Cray NHC is automatically invoked by ALPS (Application Level Placement Scheduler) upon the termination of an application. ALPS passes a list of CNL compute nodes associated with the terminated application to NHC. NHC performs specified tests to determine if compute nodes allocated to the application are healthy enough to support running subsequent applications. If not, it removes any compute nodes incapable of running an application from the resource pool.

This procedure enables the cray_node_health service. No other settings need to be changed at this point in the process. Cray recommends that sites install and configure CLE with default plugins first, and then return to the Cray Node Health service after the first system boot to configure custom plugins, if needed, using the `custom_plugins` setting.

For information about NHC configuration, see "Configure Node Health Checker Tests" under "Modify an Installed System" in *XC™ Series System Administration Guide* (S-2393).

## Procedure

1. Edit `cray_node_health_worksheet.yaml`.

   ```
   smw# vi cray_node_health_worksheet.yaml
   ```

2. Uncomment `cray_node_health.enabled` and set it to `true`.

### 3.5.1.31 Update cray_persistent_data Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Persistent Data service provides persistent storage to nodes, which can be configured on a per-node basis. This procedure configures some basic settings in the cray_persistent_data configuration worksheet to add site-specific data.

> **NOTE:** *cray_persistent_data* must be enabled when using boot node failover or SDB node failover.

## Procedure

1. Edit `cray_persistent_data_worksheet.yaml`.

   ```
   smw# vi cray_persistent_data_worksheet.yaml
   ```

2. Uncomment `cray_persistent_data.enabled` and set it to `true`.

3. Uncomment `cray_persistent_data.settings.directories.data.persistent_space_mount` and set it to match the mount point of a non-volatile volume in the CLE storage set (cledefault), which was specified previously while bootstrapping the SMW installation.

   The setting for that mount point is `cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_vg.volumes.nvolatile.fs_mount_point`, which is in the `cray_bootraid` service in the global config set. To find its value, use `cfgset search` and scan the list of matches for that setting.

   ```
   smw# cfgset search --service cray_bootraid --level advanced \
   --state all --term nvolatile global
   ```

4. Ensure that these `client_groups` settings are uncommented.

   For each setting, uncomment both the variable and its value (the line that follows it, which is a list containing one node group). They should all be set to a list containing the node group service_nodes, except for the NFS mount: nfs.client_groups should be set to a list containing boot_nodes and sdb_nodes.

   ```
   #cray_persistent_data.settings.mounts.data./var/opt/cray/alps.client_groups:
   #- service_nodes

   #cray_persistent_data.settings.mounts.data./var/opt/cray/aeld.client_groups:
   #- service_nodes

   #cray_persistent_data.settings.mounts.data./var/opt/cray/apptermd.client_groups:
   #- service_nodes

   #cray_persistent_data.settings.mounts.data./var/opt/cray/ncmd.client_groups:
   #- service_nodes

   #cray_persistent_data.settings.mounts.data./var/lib/nfs.client_groups:
   #- boot_nodes
   #- sdb_nodes
   ```

5. If the Cray DRC (dynamic RDMA credentials) service will be used with persistent storage, configure space for it by defining a cray_persistent_data mount point.

   In the worksheet, copy the five lines below `# ** EXAMPLE 'mounts' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'mounts' VALUE (with current defaults) **
# cray_persistent_data.settings.mounts.data.mount_point.sample_key_a: null <-- setting a multival key
# cray_persistent_data.settings.mounts.data.sample_key_a.alt_storage_path: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.options: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.ancestor_def_perms: '0771'
# cray_persistent_data.settings.mounts.data.sample_key_a.client_groups: []
```

Uncomment the lines, replace `sample_key_a` with `/var/opt/cray/rdma-credentials` in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). For the `client_groups` setting (last line), remove the empty list (`[]`), and add a node group (one that contains the service node that should be running the DRC service) on a separate line prefixed by a hyphen and space (– ). The cname of this node is the same as was set for the `cray_drc.settings.server.data.server_cname` setting in the Cray DRC worksheet (`cray_drc_worksheet.yaml`). To see which node group contains the node with this cname, or to create such a node group for this system (***`NODE_GROUP`*** in this example), edit `cray_node_groups_worksheet.yaml`.

Leave all other settings at the default values.

```
# NOTE: Place additional 'mounts' setting entries here, if desired.
cray_persistent_data.settings.mounts.data.mount_point./var/opt/cray/rdma-credentials: null
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.alt_storage_path: ''
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.options: ''
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.ancestor_def_perms: '0771'
cray_persistent_data.settings.mounts.data./var/opt/cray/rdma-credentials.client_groups:
- NODE_GROUP

#************************ END Service Setting: mounts ************************
```

6. If a workload manager (WLM) will be used, configure space for its spool area by defining a cray_persistent_data mount point..

   Use these spool file paths as mount points for persistent storage, depending on the WLM used at this site. Note that for Moab/TORQUE, two mount points will need to be defined.

   - Moab/TORQUE: `/var/spool/moab` and `/var/spool/torque`
   - PBS: `/var/spool/PBS`
   - Slurm: `/var/spool/slurm`

   In the worksheet, copy the five lines below `# ** EXAMPLE 'mounts' VALUE (with current defaults) **` and paste them below `# NOTE: Place additional 'mounts' setting entries here, if desired.`

```
# ** EXAMPLE 'mounts' VALUE (with current defaults) **
# cray_persistent_data.settings.mounts.data.mount_point.sample_key_a: null <-- setting a multival key
# cray_persistent_data.settings.mounts.data.sample_key_a.alt_storage_path: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.options: ''
# cray_persistent_data.settings.mounts.data.sample_key_a.ancestor_def_perms: '0771'
# cray_persistent_data.settings.mounts.data.sample_key_a.client_groups: []
```

   Uncomment the lines, replace `sample_key_a` with one the correct spool file path in all lines, and remove the `<-- setting a multival key` text at the end of the first line (note that the `null` value is required; do not remove or change it). For the `client_groups` setting (last line), remove the empty list (`[]`), and add a node group (one that contains the WLM server node) on a separate line prefixed by a hyphen and space (– ). To see which node group contains the node with this cname, or to create such a node group for this system (***`NODE_GROUP`*** in this example), edit `cray_node_groups_worksheet.yaml`.

   Leave all other settings at the default values.

   This example shows the Slurm file path as the mount point (`sample_key_a`).

```
# NOTE: Place additional 'mounts' setting entries here, if desired.
cray_persistent_data.settings.mounts.data.mount_point./var/spool/slurm: null
cray_persistent_data.settings.mounts.data./var/spool/slurm.alt_storage_path: ''
cray_persistent_data.settings.mounts.data./var/spool/slurm.options: ''
cray_persistent_data.settings.mounts.data./var/spool/slurm.ancestor_def_perms: '0771'
cray_persistent_data.settings.mounts.data./var/spool/slurm.client_groups:
- NODE_GROUP

#************************ END Service Setting: mounts ************************
```

### 3.5.1.32 Update cray_rsip Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

RSIP (realm-specific IP) helps to maintain packet integrity by allowing an RSIP host to borrow one or more IP addresses from a set of configured RSIP gateways. This procedure configures some simple settings in the Cray RSIP configuration service worksheet to add site-specific data, such as which nodes will be RSIP servers and which will be RSIP clients.

Systems with service nodes that will provide the RSIP service need to have RSIP configured. For simple RSIP configuration, enable the cray_rsip service and provide values for the settings in this worksheet. For more complex RSIP configuration, disable the cray_rsip service at this time. The service must be disabled because some of the advanced configuration can be done only after the XC system has booted, and if RSIP is enabled but not fully configured, it will cause boot errors. Therefore, for complex RSIP configurations, this service must be enabled and configured later in the process after the XC system boots successfully.

## Procedure

1. Edit `cray_rsip_worksheet.yaml`.

   ```
   smw# vi cray_rsip_worksheet.yaml
   ```

2. Uncomment `cray_rsip.enabled` and set it as follows.

   - Set it to `false` if this system will not use RSIP. Skip the rest of this procedure.
   - Set it to `false` if this system requires complex RSIP configuration (e.g., RSIP failover, RSIP pools) and the XC system has not yet booted. Skip the rest of this procedure.
   - Set it to `true` if this system requires complex RSIP configuration and the XC system has booted. Proceed to the next step.
   - Set it to `true` if this system will use RSIP and the settings in the cray_rsip configuration worksheets suffice to configure RSIP. Proceed to the next step.

3. Enter the node group (or groups) of the nodes that will be RSIP servers on this system.

   To create one or more node groups that contain the RSIP server nodes (by cname) for this system (rsip_nodes in this example), edit `cray_node_groups_worksheet.yaml`.

Uncomment `cray_rsip.settings.service.data.server_groups`, remove the empty list (`[]`), and add the node group(s) on separate lines prefixed by a hyphen and space (- ).

```
cray_rsip.settings.service.data.server_groups:
- rsip_nodes
```

4. Enter the node group (or groups) of the service nodes that will be RSIP clients on this system, such as a MOM node.

   To create one or more node groups that contain the RSIP client nodes (by cname) for this system (rsip_servicenode_clients in this example), edit `cray_node_groups_worksheet.yaml`.

   Uncomment `cray_rsip.settings.service.data.node_groups_as_client`, remove the empty list (`[]`), and add the node group(s) on separate lines prefixed by a hyphen and space (- ).

   ```
   cray_rsip.settings.service.data.node_groups_as_client:
   - rsip_servicenode_clients
   ```

5. (For complex RSIP configuration only) If this system requires complex RSIP configuration, and the XC system has booted, generate the advanced configuration files and set the use_xtrsipcfg setting.

   a. Uncomment `cray_rsip.settings.service.data.use_xtrsipcfg` and ensure that it is set to `true`.

   b. Run `xtrsipcfg_v2` as root.

      This command will generate the needed configuration files and place them in `/var/opt/cray/imps/config/sets/`*`p0`*`/files/roles/rsip/`.

      **NOTICE:** `xtrsipcfg_v2` can be run only when the CLE system is booted.

      ```
      smw# /opt/cray/xtrsipcfg/*/bin/xtrsipcfg_v2 -b
      ```

### 3.5.1.33 Update cray_rur Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

RUR (Resource Utilization Reporting) is a scalable framework for collecting utilization data from nodes within a user application. It is also a collection of plugins that report an extensible list of statistics about the hardware and software resources consumed by the application. RUR allows the creation of both data plugins for collecting statistics about the use of additional resources, and output plugins for writing the summarized usage data to additional forms of permanent storage.

This procedure enables the Cray RUR service and shows two Cray ALPS settings that must be set if RUR is used. No other settings need to be changed at this point in the process. Cray recommends that sites install and configure CLE with default plugins first, and then return to the Cray RUR service after the first system boot to configure custom plugins, if needed, using the `data_plugins` or `output_plugins` settings.

For information about RUR data collectors and how to enable them, see the procedures in "Resource Utilization Reporting" under "Monitor the System" in *XC™ Series System Administration Guide* (S-2393).

## Procedure

1. Edit `cray_rur_worksheet.yaml`.

   ```
   smw# vi cray_rur_worksheet.yaml
   ```

2. Uncomment `cray_rur.enabled` and set it to `true`.

3. Ensure that the prologPath and epilogPath variables in the Cray ALPS service have been set.

   The configuration worksheet for the Cray ALPS service has the following two settings that must be configured if RUR is used. See *Update cray_alps Worksheet* on page 117.

   ```
   cray_alps.settings.apsys.data.prologPath: /opt/cray/rur/default/bin/rur_prologue.py
   cray_alps.settings.apsys.data.epilogPath: /opt/cray/rur/default/bin/rur_epilogue.py
   ```

### 3.5.1.34 Update cray_scalable_services Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Cray Scalable Services defines a tree of servers (nodes), starting with the server of authority (SoA), that are used in the scaling of the system. Configuration of Scalable Services is required for a functioning system. For more information, see *About Cray Scalable Services* on page 13. This procedure configures some basic settings in the cray_scalable_services configuration worksheet to add site-specific data.

## Procedure

1. Edit `cray_scalable_services_worksheet.yaml`.

   ```
   smw# vi cray_scalable_services_worksheet.yaml
   ```

2. Uncomment `cray_scalable_services.enabled` and ensure that it is set to `true`.

3. Uncomment `cray_scalable_services.settings.scalable_service.data.server_of_authority` and ensure that it is set to `smw`.

4. Enter the node group (or node groups) of the nodes that will be tier1 servers on this system.

   Ensure that these node groups include the cname of the boot node and any other nodes that have an Ethernet connection to the SMW. The SDB node should also have a connection to the SMW, so it can be a tier1 server.

**IMPORTANT:** If enabling boot node failover or SDB node failover, ensure that all boot nodes and all SDB nodes are in a tier1 node group and none of them are in a tier2 node group.

Uncomment `cray_scalable_services.settings.scalable_service.data.tier1_groups`, remove the empty list (`[]`), and add these predefined node groups on separate lines prefixed by a hyphen and space (`- `).

```
cray_scalable_services.settings.scalable_service.data.tier1_groups:
- boot_nodes
- sdb_nodes
- OTHER_TIER1_NODE_GROUP
```

To verify that these node groups contain the tier1 server nodes (by cname) for this system, to add the correct tier1 nodes to them, or to add a new node group for tier1 servers, (***`OTHER_TIER1_NODE_GROUP`*** in this example), edit `cray_node_groups_worksheet.yaml` (see *Update cray_node_groups Worksheet* on page 156).

**5.** Enter the node group (or node groups) of the nodes that will be tier2 servers on this system.

Uncomment `cray_scalable_services.settings.scalable_service.data.tier2_groups` and the line below it, which is a list of one predefined node group.

```
cray_scalable_services.settings.scalable_service.data.tier2_groups:
- tier2_nodes
```

To verify that the predefined tier2 node group contains the correct tier2 server nodes (by cname) for this system or to add the correct tier2 nodes to them, edit `cray_node_groups_worksheet.yaml`.

**Q. How many tier2 nodes are needed?**

**A.** At least one server must be provided, and for resiliency, two nodes placed on different blades. The recommended ratio of tier2 nodes (servers) to tier3 nodes (clients) is 1 to 400.

**Q. Will adding more tier2 nodes help performance?**

**A.** Adding more tier2 nodes does not always yield additional performance and is subject to diminishing returns.

**Q. What kind of node can be used as a tier2 node?**

**A.** Use these:

- OPTIMAL: dedicated repurposed compute nodes (RCN)
- dedicated service nodes
- nodes with uniform light to moderate load
- nodes with relatively homogeneous single core speeds to reduce resource contention disparity during periods of partial availability

AVOID these (will result in sub-optimal performance):

- nodes with slower individual CPU cores, such as Intel® Xeon Phi™ "Knights Landing" (KNL) processors
- direct-attached Lustre (DAL) servers
- RSIP (realm-specific IP) servers
- login nodes

**Q. Can a tier2 node have more than one role?**

**A.** Small test and development systems (TDS) may use tier2 nodes that have additional roles, but generally, it is better for tier2 nodes to be dedicated.

**Q. Where should tier2 nodes be placed?** **A.** Distribute nodes throughout the system (on different blades) for resiliency in the event of hardware failure.

Check the guidance for tier2 nodes in this configuration worksheet for additional requirements or limitations.

### 3.5.1.35 Update cray_sdb Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Software Database (SDB) service configures the services and settings for the SDB node. This procedure configures some basic settings in the cray_sdb service configuration worksheet.

## Procedure

1. Edit `cray_sdb_worksheet.yaml`.

   ```
   smw# vi cray_sdb_worksheet.yaml
   ```

2. Uncomment `cray_sdb.enabled` and ensure that it is set to `true`.

3. Configure the SDB node groups setting.

   a. Uncomment the SDB node groups setting.

      Be sure to uncomment both lines.

      ```
      #cray_sdb.settings.node_groups.data.sdb_groups:
      #- sdb_nodes
      ```

   b. Verify that the sdb_nodes node group has been accurately defined for this site.

      To verify, edit `cray_node_groups_worksheet.yaml` and search for `sdb_nodes`.

4. Configure the admin and root database passwords.

   Uncomment the following two password settings and replace the default values with site-specific values.

   These passwords will be stored in clear text in the config set. Note that the values of these passwords are excluded when the config set is distributed to eLogin nodes.

   ```
   #cray_sdb.settings.database.data.db_admin_password: sys_mgt

   #cray_sdb.settings.database.data.db_current_root_password: ''
   ```

5. (Optional) Set the host for the daemon that syncs the HSS database.

   Uncomment this setting to configure it. Cray recommends keeping the default value of 'sdb'; however, if this site wishes xtdbsyncd to run on the boot node instead, change the value to 'boot.'

```
#cray_sdb.settings.database.data.synchost: sdb
```

### 3.5.1.36 Update cray_service_node Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Service Node configuration service configures the services and settings for service nodes. This procedure enables the cray_service_node service, which is sufficient for a fresh install.

## Procedure

1. Edit `cray_service_node_worksheet.yaml`.

   ```
   smw# vi cray_service_node_worksheet.yaml
   ```

2. Uncomment `cray_service_node.enabled` and set it to `true`.

   No other settings need to be changed for a fresh install.

### 3.5.1.37 Update cray_shifter Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Shifter is an HPC-focused implementation of Linux containers that was created at the Berkeley Labs NERSC supercomputing facility. It enables a large-scale HPC system to efficiently and safely allow end-users to run a docker image. The cray_shifter configuration service configures Shifter for Cray XC systems.

Shifter includes the following:

- A utility that typically runs on the compute node that creates the run-time environment for the application.
- An image gateway service that pulls images from a registry and repacks it in a format suitable for the HPC system.
- Scripts and plugins to integrate Shifter with various batch scheduler systems.

This procedure enables or disables cray_shifter, depending on whether it is needed for this site. If enabled, Cray recommends configuring the rest of the Shifter settings later after the system has been booted. To install and configure Shifter at that time, see *XC™ Series Shifter Installation Guide* (S-2572).

## Procedure

1. Edit cray_shifter_worksheet.yaml.

   ```
   smw# vi cray_shifter_worksheet.yaml
   ```

2. Uncomment cray_shifter.enabled and do one of the following:

   - Set it to false for systems that will NOT use Shifter.
   - Set it to true for systems that will use Shifter.

### 3.5.1.38 Update cray_simple_shares Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray Simple File-system Sharing service quickly shares files between compute nodes that are connected to the high speed network (HSN). This procedure configures some basic settings in the cray_simple_shares configuration worksheet.

## Procedure

1. Edit cray_simple_shares_worksheet.yaml.

   ```
   smw# vi cray_simple_shares_worksheet.yaml
   ```

2. Uncomment cray_simple_shares.enabled and ensure that it is set to true.

3. Update the NFS mount settings.

   a. Ensure that the node groups settings are configured.

   Search in the file for 'NFS' DATA, and below that line, find these server_groups and client_groups settings for several pre-populated NFS client mounts. If they are commented, uncomment them.

   ```
   # ** 'NFS' DATA **

   #cray_simple_shares.settings.NFS.data./alps_shared.server_groups:
   #- sdb_nodes
   #cray_simple_shares.settings.NFS.data./alps_shared.client_groups:
   #- service_nodes
   #cray_simple_shares.settings.NFS.data./alps_shared.client_exclude_groups:
   #- boot_nodes
   ...
   #cray_simple_shares.settings.NFS.data./cray_home.server_groups:
   ```

```
#- boot_nodes
#cray_simple_shares.settings.NFS.data./cray_home.client_groups:
#- service_nodes
...
#cray_simple_shares.settings.NFS.data./var/opt/cray/imps.server_groups:
#- boot_nodes
#cray_simple_shares.settings.NFS.data./var/opt/cray/imps.client_groups:
#- tier2_nodes
...
#cray_simple_shares.settings.NFS.data./non_volatile.server_groups:
#- boot_nodes
#cray_simple_shares.settings.NFS.data./non_volatile.client_groups:
#- service_nodes
```

b. If the home directory was changed in other configuration worksheets (e.g., `cray_local_users_worksheet.yaml`), change it here also.

   Under `'NFS' DATA`, look for settings with `cray_home` or `home` as the 'path' key. Ensure that they reflect the same home directory as used in `cray_local_users_worksheet.yaml`.

   ```
   cray_simple_shares.settings.NFS.data./cray_home.server_groups:
   - boot_nodes
   cray_simple_shares.settings.NFS.data./cray_home.fs_root: /cray_home
   cray_simple_shares.settings.NFS.data./cray_home.fs_export_opt:
       'secure,rw,no_subtree_check,no_root_squash,no_acl'
   cray_simple_shares.settings.NFS.data.path./cray_home: null
   cray_simple_shares.settings.NFS.data./cray_home.client_groups:
   - service_nodes
   cray_simple_shares.settings.NFS.data./cray_home.unconditional_mount: false
   ```

**4.** Update the DVS mount settings.

Search in the file for `'DVS' DATA`, and below that line, find these settings for a pre-populated DVS client mount. If they are commented, uncomment them.

```
# ** 'DVS' DATA **
...
#cray_simple_shares.settings.DVS.data./var/opt/cray/imps.spath: /var/opt/cray/imps
#cray_simple_shares.settings.DVS.data./var/opt/cray/imps.client_groups:
#- all_nodes
```

**Disambiguation.** Notice that the path '`/var/opt/cray/imps`' appears twice in the first setting. The first instance is the path where clients will mount the file system. It is the 'key' (*`mount_point`*) for this client mount, so it appears in all of the settings for this client mount. The second instance is the path to the file system on the server node that is to be projected. It is the default value provided for this pre-populated DVS client mount. That first setting is simply specifying that the file system will be projected from the same path on the server as it is mounted from the client.

**5.** Verify that the node groups referenced in steps 3 and 4 have been accurately defined for this site.

To verify, edit `cray_node_groups_worksheet.yaml` and search for these node groups:

```
all_nodes
boot_nodes
sdb_nodes
service_nodes
tier2_nodes
```

```
smw# vi cray_node_groups_worksheet.yaml
```

### 3.5.1.39 Update cray_simple_sync Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

Simple Sync is a mechanism for automatically distributing files to targeted locations on the Cray system. This procedure enables the cray_simple_sync service.

## Procedure

1. Edit `cray_simple_sync_worksheet.yaml`.

   ```
   smw# vi cray_simple_sync_worksheet.yaml
   ```

2. Uncomment `cray_simple_sync.enabled` and set it to `true`.

   No other settings need to be changed.

### 3.5.1.40 Update cray_ssh Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The SSH service allows the system to be accessed through a secure shell. This procedure enables the Cray SSH configuration service.

## Procedure

1. Edit `cray_ssh_worksheet.yaml`.

   ```
   smw# vi cray_ssh_worksheet.yaml
   ```

2. Uncomment `cray_ssh.enabled` and set it to `true`.

   For a fresh install, no other settings need to be changed.

### 3.5.1.41 Update cray_storage Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Storage service defines which storage set the current partition or system may use for persistent storage. Storage sets are defined in the global config set. This procedure configures some basic settings in the cray_storage configuration worksheet.

## Procedure

1. Edit `cray_storage_worksheet.yaml`.

   ```
   smw# vi cray_storage_worksheet.yaml
   ```

2. Uncomment `cray_storage.enabled` and set it to `true`.

3. Uncomment `cray_storage.settings.storage.data.active_storage_set` and set it to be the name of the CLE storage set in the `cray_bootraid` service, which is in the global config set.

   Use this command to show all storage sets defined in the global config set.

   ```
   smw# cfgset search -s cray_bootraid global |awk -F'.' '{print $5}' | sort -u
   ```

4. (For reinstall only) Uncomment `cray_storage.settings.storage.data.zero_volumes_on_create` and set it to true if this system is reinstalling to a CLE storage set that had been in use previously.

### 3.5.1.42 Update cray_sysconfig Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray System Configuration service controls configuration of files in `/etc/sysconfig`. The sysconfig service can be used to specify particular configuration file settings and values.

This procedure enables the cray_sysconfig service and provides an example of changing a configuration file in `/etc/sysconfig`.

## Procedure

1. Edit `cray_sysconfig_worksheet.yaml`.

   ```
   smw# vi cray_sysconfig_worksheet.yaml
   ```

2. Uncomment `cray_sysconfig.enabled` and set it to `true`.

3. Change configuration settings in a file in `/etc/sysconfig`, as needed.

   Repeat this step for each file with settings to be changed.

   In the worksheet, copy the six lines below

   ```
   # ** EXAMPLE 'sysconfig_files' VALUE (with current defaults) **
   ```

   and paste them below the line

   ```
   # NOTE: Place additional 'sysconfig_files' setting entries here, if desired.
   ```

   ```
   # ** EXAMPLE 'sysconfig_files' VALUE  (with current defaults) **
   #   cray_sysconfig.settings.sysconfig_files.data.name.sample_key_a: null   <-- setting a multival key
   #   cray_sysconfig.settings.sysconfig_files.data.sample_key_a.file: ''
   #   cray_sysconfig.settings.sysconfig_files.data.sample_key_a.scope:
   #   - service
   #   - compute
   #   cray_sysconfig.settings.sysconfig_files.data.sample_key_a.key_values: []
   ```

   Uncomment the lines, replace `sample_key_a` in all lines with an identifier for the file to be changed (sitekey in the example below), and remove the `<-- setting a multival key` text at the end of the first line (note that the null value is required; do not remove or change it). Finally, modify the values as needed for this site.

   There are two list settings in the sysconfig_files setting: scope and key_values. To enter a list, add each list item on a separate line prefixed by a hyphen and space (`- `). If the list was initially set to `[]`, an empty list, remove the brackets before adding list items.

   - For the scope list setting, enter a list of target node types (service, compute) and/or cnames (NOT node groups).
   - For the key_values list setting, enter a list of key=value pairs.

   The following example uses *`sitekey`* to identify the `/etc/sysconfig/`*`filename`* file and change the value of its *`MYVAR`* variable to *`newsetting`* for all service and compute nodes.

   ```
   # NOTE: Place additional 'sysconfig_files' setting entries here, if desired.
   cray_sysconfig.settings.sysconfig_files.data.name.sitekey: null
   cray_sysconfig.settings.sysconfig_files.data.sitekey.file: filename
   cray_sysconfig.settings.sysconfig_files.data.sitekey.scope:
   - service
   - compute
   cray_sysconfig.settings.sysconfig_files.data.sitekey.key_values:
   - MYVAR="newsetting"
   ```

## 3.5.1.43 Update cray_sysenv Worksheet

### Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The Cray System Environment service specifies values to be defined within the system environment. This procedure configures some basic settings in the cray_sysenv configuration worksheet.

### Procedure

1. Edit `cray_sysenv_worksheet.yaml`.

   ```
   smw# vi cray_sysenv_worksheet.yaml
   ```

2. Uncomment `cray_sysenv.enabled` and set it to `true`.

## 3.5.1.44 Update cray_time Worksheet

### Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

### About this task

The Cray Time service configures the time zone and several advanced features, such as the minimum poll interval for NTP messages. This procedure configures the inheritance setting in the Cray Time service configuration worksheet. This service has a global template as well as a CLE template, and therefore it can be configured to inherit settings from the global config set.

### Procedure

1. Edit `cray_time_worksheet.yaml`.

   ```
   smw# vi cray_time_worksheet.yaml
   ```

2. Uncomment `cray_time.inherit` and set it to `true`.

   This means that time settings in the global config set will be used instead of time settings in the CLE config set. See *Prepare and Update the Global Config Set* on page 99. No other settings need to be changed.

### 3.5.1.45 Update cray_user_settings Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray User Settings service sets the environment modules that should be loaded automatically when a user logs in to the SMW, login node, or service nodes. The SMW modules can be extended by adding to `/etc/opt/cray/modules/Base-opts.local`.

This procedure enables the cray_user_settings service.

## Procedure

1. Edit `cray_user_settings_worksheet.yaml`.

   ```
   smw# vi cray_user_settings_worksheet.yaml
   ```

2. Uncomment `cray_user_settings.enabled` and set it to `true`.

   No other settings need to be changed for a fresh install.

As other software is installed later, it might be necessary to change the set of module files loaded by default on login and service nodes.

### 3.5.1.46 Update cray_wlm_detect Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray WLM (workload manager) Detect service is a C library and command used to identify the native WLM on the system. If this service is not configured, the default ALPS will be used.

This procedure enables the cray_wlm_detect configuration service. For an explanation of the long variable names in configuration settings, see *About Variable Names in the Configurator and Configuration Worksheets* on page 16.

## Procedure

1. Edit `cray_wlm_detect_worksheet.yaml`.

```
smw# vi cray_wlm_detect_worksheet.yaml
```

2. Uncomment `cray_wlm_detect.enabled` and set it to `true`.

3. Set the active WLM (workload manager).

   This setting identifies the native WLM running on the system. For WLMs using BASIL, or to indicate no WLM, set the value to ALPS. For a native WLM, enter its name in uppercase (for example, enter SLURM for Slurm). Currently only ALPS and Slurm are supported.

   ```
   cray_wlm_detect.settings.common.data.active_wlm: ALPS
   ```

### 3.5.1.47 Update cray_wlm_trans Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The Cray WLM (workload manager) Trans service is a library that provides WLM-agnostic functions for common tasks such as setting node state and getting a list of jobs being run by a user. It is used primarily by node health checker.

This procedure enables the cray_wlm_trans configuration service.

## Procedure

1. Edit `cray_wlm_trans_worksheet.yaml`.

   ```
   smw# vi cray_wlm_trans_worksheet.yaml
   ```

2. Uncomment `cray_wlm_trans.enabled` and ensure that it is set to `true`.

   No other WLM Trans settings need to be changed for a fresh install.

### 3.5.1.48 Update cray_zonesort Worksheet

## Prerequisites

This procedure assumes that a work area has been set up for editing CLE configuration worksheets and that the current directory has been set to that work area.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

## About this task

The zonesort_module kernel module sorts free memory on the node to improve the predictability of the MCDRAM (multi-channel dynamic random-access memory) cache performance. The Cray zone sort configuration service configures the loading of the zonesort_module kernel module on compute nodes. This procedure enables that service.

## Procedure

1. Edit `cray_zonesort_worksheet.yaml`.

```
smw# vi cray_zonesort_worksheet.yaml
```

2. Uncomment `cray_zonesort.enabled` and ensure that it is set to `true`.

## 3.5.2 Create New CLE Config Set from Worksheets

### Prerequisites

This procedure assumes that worksheets have been obtained, copied to a work area outside of `/var/opt/cray/imps/config/sets/`*`CONFIG_SET_NAME`*`/worksheets`, and modified to include site-specific configuration data.

### About this task

This procedure creates a new CLE config set from existing CLE configuration worksheets. Use one or more of the following commands, depending on whether this system is partitioned or not. Note that the worksheet path provided must be enclosed in single quotes because of the file glob used. There is no need to specify the config set type because the default is type CLE.

⚠ **CAUTION: Boot failure possible if using `cfgset` under certain conditions.**

The `cfgset create` and `cfgset update` commands always call pre- and post-configuration scripts. Some of these scripts require HSS daemons and other CLE services to be running. This can cause problems under these conditions:

- If `xtdiscover` is running, `cfgset` may hang or produce incorrect data that can result in system boot failure.
- If `xtbounce` is in progress or if the SMW is not connected to XC hardware, `cfgset` will fail.

In these circumstances, use the `--no-scripts` option with `cfgset create` or `cfgset update` to avoid running the scripts. Because using that option results in an invalid config set, remember to run `cfgset update` without the `--no-scripts` option afterwards, when circumstances permit, to ensure that all pre- and post-configuration scripts are run.

**full system** Create a config set for a full (unpartitioned) system p0:

```
smw# cfgset create --worksheet-path \
'/var/adm/cray/release/p0_worksheet_workarea/*_worksheet.yaml' p0
```

**partitioned** Create a config set for each partition. For partition p1:

```
smw# cfgset create --worksheet-path \
'/var/adm/cray/release/p1_worksheet_workarea/*_worksheet.yaml' p1
```

For partition p2:

```
smw# cfgset create --worksheet-path \
'/var/adm/cray/release/p2_worksheet_workarea/*_worksheet.yaml' p2
```

Additional partitions follow the same pattern.

### 3.5.3 Update CLE Config Set after a Fresh Install

#### Prerequisites

This procedure assumes that one or more CLE config sets have been created.

#### About this task

This procedure uses the configurator in auto mode to check for any required or basic settings that were not configured earlier in the process. The configurator will prompt for values for those settings.

The crayadm and root passwords from the `cray_local_users` service were not configured earlier using worksheets because they must be encrypted, and it is difficult to enter encrypted values in a worksheet. Therefore, the configurator will prompt for those values now.

In addition, the configurator may prompt for the value of the `flat_routes` setting or the `fgr_routes` setting or both (from the `cray_lnet` service), depending on which one is not being used for external Lustre servers or whether direct-attached Lustre (DAL) is used.

#### Procedure

Invoke `cfgset` to update the config set.

**full system** Update the config set for a full (unpartitioned) system p0 (in this example, the config set is named p0):

```
smw# cfgset update p0
```

**partitioned** Update the config set for each partition. For partition p1 (in this example, the config set is named p1):

```
smw# cfgset update p1
```

For partition p2 (in this example, the config set is named p2):

```
smw# cfgset update p2
```

Additional partitions follow the same pattern.

Configurator navigation hints:

- To get context-sensitive command help, enter **?**.
- To add a single value, enter the data and press **Enter**.

- To add a list, enter **+**, enter each list item on its own line, and press **Ctrl-d** when finished entering the entire list of items.
- To correct an error in a previous setting, press the **<** key to go back to the previous setting, correct it, then continue forward. Press **<** multiple times to back up several settings, if needed.

Enter values for any settings presented by the configurator. The following steps provide instructions for specific settings.

a. Set the crayadm password when prompted for `cray_local_users.settings.users.data.crayadm.crypt`.

   This `cray_local_users` setting is for a CLE/Linux account. It is of type "protected," so it must be entered twice (the second time for confirmation), and it will not be displayed while being entered. The configurator will encrypt it before storing it in the config set.

   Enter **+**, then enter the password (NOT its encrypted form) and press **Enter**. Re-enter the password and press **Enter** again.

   This example shows the value *`crayadm_password`* entered at the prompt, but actually, the configurator will not display what is entered.

```
cray_local_users.settings.users.data.crayadm.crypt
[+=modify, ?=help, @=less] $ +
Modify crypt (Ctrl-d to cancel, <cr> to set) $ crayadm_password
Re-enter value for crypt (Ctrl-d to cancel, <cr> to set) $ crayadm_password
```

b. Set the root password when prompted for `cray_local_users.settings.users.data.root.crypt`.

   This is another `cray_local_users` setting for a CLE/Linux account, also of type "protected."

```
cray_local_users.settings.users.data.root.crypt
[+=modify, ?=help, @=less] $ +
Modify crypt (Ctrl-d to cancel, <cr> to set) $ root_password
Re-enter value for crypt (Ctrl-d to cancel, <cr> to set) $ root_password
```

c. Set the "users" entries when done setting the crayadm and root passwords.

```
cray_local_users.settings.users
[<cr>=set N entries, ?=help, @=less] $ <cr>
```

   **Not prompted for these passwords?** If the configurator did not prompt for one or both of these settings, wait until `cfgset` finishes, then run `cfgset` in interactive mode (example shows command for config set p0), and select and set these settings from the `cray_local_users` service.

```
smw# cfgset update -m interactive -s cray_local_users p0
```

   For more information about using the configurator, see *XC™ Series Configurator User Guide* (S-2560).

If no more settings are presented, it means that all required and basic settings have been set.

When the configurator is done, it displays a message indicating the file name of the changelog file for this configuration session. The changelog is written to a file in the `/var/opt/cray/imps/config/sets/`*`p0`*`/changelog` directory (for a CLE config set named p0).

## 3.5.4 Check CLE Hostnames in /etc/hosts File

### Prerequisites

This procedure assumes that the CLE config set has been created and updated.

### About this task

This procedure confirms that the post-configuration callback scripts, which were run when the CLE config set was updated, added the correct host name entries to the `/etc/hosts` file.

### Procedure

1. Confirm that host name entries exist in the CLE `/etc/hosts` file for boot, sdb, login, lnet, rsip, dvs, and any other names defined on this system.

   **full system** For a full, unpartitioned system:

   ```
   smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
   imps/config/sets/p0/files/roles/common/etc/hosts
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
   imps/config/sets/p1/files/roles/common/etc/hosts
   ```

   ```
   smw# egrep "boot|sdb|login|lnet|rsip|dvs" /var/opt/cray/\
   imps/config/sets/p2/files/roles/common/etc/hosts
   ```

   **Trouble?** If any expected host names are missing, proceed to step 2.

2. If any expected host names are missing, do one of the following:

| Option | Description |
|---|---|
| **Option 1: Update the config set (preferred)** | Use `cfgset` to update the cray_net configuration service in config set p0 and add any missing hostname, hostname alias, or network interface information.<br>`smw# cfgset update -m interactive -s cray_net p0` |
| **Option 2: Edit the `/etc/hosts` file** | Add external host names and IP addresses directly to the following file on the SMW. The additional entries and any comments will be retained every time the config set is updated. Do not add them to the `/etc/hosts` file on a CLE node.<br>`smw# vi /var/opt/cray/imps/config/sets/p0/files/roles/\`<br>`common/etc/hosts` |

   **IMPORTANT:**

   - Adding content to configuration files by editing them on nodes is ephemeral.
   - Adding content to configuration files by using `cfgset` to update the config set on the SMW (as in Option 1) or by editing them within the config set on the SMW (as in Option 2) is permanent.

Changes made to a config set on the SMW are shared with CLE nodes through config set caching. For more information, see *About Config Set Caching* on page 17.

## 3.5.5 Update /etc/motd for Nodes

### About this task

The standard `/etc/motd` on CLE nodes has this information.

```
Identity of node
Compute or service node
Boot image
Size of boot image
CLE release and build
Core and memory info
```

To append a custom message to the standard message of the day for all nodes, edit the `/etc/motd` file as shown in the example, which uses the config set common role to distribute the `/etc/motd` file to all nodes.

### Procedure

1. Create the `files/roles/common/etc` path below the config set directory.

   ```
   smw# cd /var/opt/cray/imps/config/sets/p0
   smw# mkdir -p files/roles/common/etc
   ```

2. Edit the message of the day to append the custom message.

   ```
   smw# vi files/roles/common/etc/motd
   ```

## 3.5.6 Copy Files for External Lustre Fine-grained Routing

### Prerequisites

This procedure is only for systems that use an external Lustre file system. It assumes the following:

- Fine-grained routing files have been generated by `clcvt`.
- The Cray LNet configuration service (cray_lnet) has been configured with fine-grained routing (FGR).

### About this task

This procedure places the `ip2nets.conf` and `routes.conf` files in the CLE config set for the LNet routers.

### Procedure

1. Create an `lnet` directory under roles in the CLE config set directory structure.

   This example uses a config set named p0. Substitute the correct config set name for this site.

```
smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/roles/lnet
```

2. Confirm the file names of the fine-grained routing files.

   It is possible that these two files were created with names other than `ip2nets.conf` and `routes.conf`. Check these two settings in the cray_lnet configuration service to see what file names are used (example settings are for a file system with key "sonexion").

   ```
   cray_lnet.settings.fgr_routes.data.sonexion.ip2nets_file
   cray_lnet.settings.fgr_routes.data.sonexion.routes_file
   ```

3. Copy the `ip2nets.conf` and `routes.conf` files to the `lnet` directory.

   ```
   smw# cd directory_containing_ip2nets.conf_and_routes.conf

   smw# cp -p ip2nets.conf routes.conf /var/opt/cray/imps/config/sets/p0/files/roles/lnet
   ```

## 3.5.7 Configure Files for Cray Simple Sync Service

## About this task

Cray Simple Sync provides a generic mechanism to automatically distribute files to targeted locations on the system. This mechanism can be used to override or change default system behavior through the contents of the distributed files. When enabled, the Simple Sync service is executed on all CLE nodes at boot time and whenever the administrator executes `/etc/init.d/cray-ansible start` on a CLE node. When Simple Sync is executed, files placed in the following directory structure are copied to the root file system (`/`) on the target nodes.

**About the Simple Sync Directory Structure**

The Simple Sync directory structure has this root:
`smw:/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/`

Below that root are the directories listed on the left:

| Files placed here | are copied to |
|---|---|
| `./common/files/` | all nodes |
| `./platform/[compute, service]/files/` | all compute or service nodes |
| `./hardwareid/<hardwareid>/files/` | nodes with matching hardware ID, which is the cname of a CLE node or the output of the `hostid` command (e.g., 1eac0b0c) on other nodes |
| `./hostname/<hostname>/files/` | nodes with matching host name (use this for eLogin nodes ONLY) |
| `./nodegroups/<node_group_name>/files/` | nodes in the matching node group |

> **NOTE:** The directory structure for a particular hardware ID or host name (everything below`./hardwareid/` and `./hostname/`) must be created manually as needed. This is unnecessary for node groups because their associated directories are created automatically by post-configuration callback scripts when the config set is created or updated using `cfgset`.

Anything (directory structure and files) placed below `./files/` in the Simple Sync directory structure on the SMW is replicated on the target node starting at root (`/`). For example, this path on the SMW

```
/var/opt/cray/imps/config/sets/p0/files/simple_sync/common/files/etc/myapplication.conf
```

will place the `myapplication.conf` file on all nodes in this directory:

```
/etc/myapplication.conf
```

Note that the ownership and permissions of files in the config set are preserved in the copies made to nodes.

For more information and use cases, see *About Simple Sync* on page 20.

## 3.5.8 Display and Capture all Config Set Information

### About this task

This procedure displays all of the configuration settings in a config set and captures them in a typescript file of this software update. It is not required, but it may aid in troubleshooting. Note that the `cfgset search` command does not search guidance text in the configuration templates and worksheets, so that information will not be included in the output.

### Procedure

Display all configuration settings in the CLE and global config sets, and capture them in a typescript file.

**full system** Display/capture full information (more verbose) for a full, unpartitioned system:

```
smw# cfgset search -l advanced --format full p0 | tee /var/adm/\
cray/release/p0.${TODAY}.fresh_install.advanced.conf.full

smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf.full
```

Display/capture just the settings and values for a full, unpartitioned system:

```
smw# cfgset search -l advanced p0 | tee /var/adm/cray/release/\
p0.${TODAY}.fresh_install.advanced.conf.full

smw# cfgset search -l advanced global | tee /var/adm/cray/release/\
global.${TODAY}.fresh_install.advanced.conf.full
```

**partitioned** Display/capture full information (more verbose) for a partitioned system, with partitions p1 and p2:

```
smw# cfgset search -l advanced --format full p1 | tee /var/adm/\
cray/release/p1.${TODAY}.fresh_install.advanced.conf

smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf
```

```
smw# cfgset search -l advanced --format full p2 | tee /var/adm/\
cray/release/p2.${TODAY}.fresh_install.advanced.conf
```

```
smw# cfgset search -l advanced --format full global | tee /var/adm/\
cray/release/global.${TODAY}.fresh_install.advanced.conf
```

## 3.5.9 Validate Config Sets

### About this task

It is important to validate any config set that has been modified, because there is currently no mechanism to prevent the system from trying to use an invalid config set. Validation is useful for determining if the config set is minimally viable for use with the system it is intended to configure.

> **IMPORTANT:** Validation ensures that a config set passes all rules stored on the system. A validated config set does not necessarily equate to a config set with configuration data that will result in a properly configured system.

When validating a config set, the configurator checks the following:

- Config set has the proper directory structure and permissions.
- All configuration templates have correct YAML syntax.
- All configuration templates adhere to the configurator schema.
- All fields of type `lookup` reference values and settings that exist in the available configuration services.
- All level `required` fields in enabled services are configured (i.e., their state is `set`).
- Pre-configuration and post-configuration callback scripts ran successfully during the latest config set update.
- `cfgset validate` has run all validation rules installed on the system.

For more information on how `lookup` fields work, see the "Advanced: Lookup" section in "Configurator Data Types and How to Set Them," which is in *XC™ Series Configurator User Guide* (S-2560). For more information about validation rules, see "Validate a Config Set and List Validation Rules," also in that publication.

### Procedure

Validate the CLE and global config sets.

These example commands use CLE config sets named p0, p1, and p2. Substitute the correct config set names for this site.

**full system** Validate the CLE and global config sets for a full, unpartitioned system:

```
smw# cfgset validate global

smw# cfgset validate p0
```

**partitioned** Validate the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfgset validate global

smw# cfgset validate p1
smw# cfgset validate p2
```

## 3.5.10 Make a Post-config Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after configuring CLE and before booting the CLE system.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

### Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postconfig
```

## 3.5.11 Make a Post-config Backup of Current Global and CLE Config Sets

### About this task

This procedure uses the `cfgset` command to create a post-install backup of the global and CLE config sets after configuring CLE and before booting the CLE system.

### Procedure

1. Back up the current global config set.

```
smw# cfgset create --clone global global-postconfig-${TODAY}
```

2. Back up the current CLE config set.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postconfig-${TODAY}
```

# 3.6 Prepare Boot Images and Boot the CLE System during a Fresh Install

Most system configuration is now complete, and this part of the installation and configuration process has procedures to prepare boot images, make sure those images are mapped to nodes correctly, boot the CLE system, and perform post-boot activities such as running tests and making a snapshot. However, sites must first decide where to place their root file system, because some of these procedures depend on that decision.

1. Decide where to place the root file system using *Where to Place the Root File System—tmpfs versus Netroot* on page 186.
2. Prepare boot images and NIMS maps.
   - a. *Create a NIMS Map* on page 188
   - b. *About Image Groups and How to Customize Them* on page 188
   - c. *Build Boot Images for a Fresh Install* on page 190
3. *Set the Turbo Boost Limit* on page 195
4. *Check NIMS Information during a Fresh Install* on page 196
5. *Boot the System using a Boot Automation File* on page 197
6. Perform post-boot activities.
   - a. *Run Tests after Boot is Complete* on page 200
   - b. *Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev* on page 201
   - c. *Test xtdumpsys and cdump* on page 202
   - d. *Make a Post-boot Snapshot using snaputil* on page 204
   - e. *Make a Post-boot Backup of Current Global and CLE Config Sets* on page 205

## 3.6.1 Where to Place the Root File System—tmpfs versus Netroot

The Cray XC™ Series root file system for nodes can either reside in RAM (tmpfs) or be mounted from a network source (Netroot), depending on the type of node. The boot and SDB nodes, all other service nodes (except login nodes), and all DAL (direct-attached Lustre) nodes must use tmpfs. Compute nodes and login nodes may use either tmpfs or Netroot. Use the information provided here to decide whether to use Netroot for some or all compute and login nodes at this site.

### About Netroot and Dynamic Shared Objects and Libraries (DSL)

In releases prior to CLE 6.0 / SMW 8.0, the dynamic shared objects and libraries (DSL) feature was optional. It was necessary for many sites because it enabled both dynamic shared libraries and large network-based images, which were needed for systems with NVIDIA GPUs and for most production workloads.

In the current release, DSL is supported by default. Note, however, that the DSL feature no longer includes provision for large network-based images. That capability is now provided by Netroot.

- Sites that require large network-based images and additional storage should use Netroot.

- Sites using NVIDIA GPUs must use Netroot.

## Comparison of tmpfs and Netroot

**tmpfs** The default location of the root file system on Cray XC™ Series systems is tmpfs, a type of memory-resident file system or RAM disk.

tmpfs has these characteristics and limitations:

- always used for service nodes (except login nodes) and DAL (direct-attached Lustre) nodes
- efficient and fast root file system access
- large memory footprint
- file system content needs to be restricted to reduce memory footprint
- typically used when minimal commands and libraries required
- works well for compute nodes with well defined workloads and for service nodes that are used primarily for internal services

**Netroot** Netroot is an alternative approach that mounts the root file system from a network source. It is used only for compute and login nodes. It uses overlayfs to layer tmpfs on top of a read-only network file system.

Due to the reliance on overlayfs, the decision to use Netroot should include consideration of the characteristics and limitations of overlayfs in addition to those of Netroot listed here.

Netroot has these characteristics and limitations:

- used only for compute and login nodes, never for service nodes (except login nodes)
- slower root file system access
- increased node boot time
- minimized memory footprint (mounted from network, so requires less disk space)
- no restriction on file system content
- typically used when a robust set of commands and libraries required (Netroot enables large network-based images, formerly enabled through the DSL feature)
- works well for compute nodes with diverse workloads and for compute nodes with a high memory footprint
- always used for GPUs

This comparison of tmpfs and Netroot memory footprints is based on a fresh install with nothing extra added. These numbers could be larger or smaller for a site depending on whether the Cray image recipes for tmpfs and Netroot have been extended (by adding necessary RPMs) or reduced (by removing unnecessary RPMs).

*Table 12. Comparison of tmpfs and Netroot Memory Footprints*

| Image Type | Memory Consumption | Number of RPMs |
|---|---|---|
| Admin image root - tmpfs | 1400 MB | 600 |
| Service image root – tmpfs | 1700 MB | 670 |
| Login image root – tmpfs | 3600 MB | 1100 |

| Image Type | Memory Consumption | Number of RPMs |
|---|---|---|
| Compute image root – tmpfs | 1500 MB | 745 |
| Login image root – Netroot | 125 MB | 2500 |
| Compute image root – Netroot | 150 MB | 2380 |

## 3.6.2 Create a NIMS Map

### Prerequisites

This procedure assumes that hardware is available and all previous procedures to install the operating system, discover hardware, and set up the config sets have been completed.

### About this task

For a fresh installation, a new NIMS (Node Image Mapping Service) map needs to be created. This procedure creates a NIMS map and designates it as the active map.

### Procedure

Create a NIMS map and set it as active.

These example commands use CLE config sets named p0, p1, and p2. Substitute the correct config set names for this site.

**full system**

For a full, unpartitioned system:

```
smw# cmap create p0
smw# cmap setactive p0
```

**partitioned**

For a partitioned system, with partitions p1 and p2:

```
smw# cmap create p1 --partition p1
smw# cmap setactive p1  -p p1
```

```
smw# cmap create p2 -p p2
smw# cmap setactive p2  -p p2
```

## 3.6.3 About Image Groups and How to Customize Them

Image group configuration information is used by the `imgbuilder` command to build boot images. Image groups are defined in the global config set in the cray_image_groups configuration file (`/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`). Here is an example of the contents of that file:

```
cray_image_groups:
  default:
      - recipe: "admin_cle_6.0up03_sles_12_x86-64_ari"
        dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
```

```
created{date}.cpio"
        nims_group: "admin"
      - recipe: "compute_cle_6.0up03_sles_12_x86-64_ari"
        dest: "compute{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "compute"
      - recipe: "login_cle_6.0up03_sles_12_x86-64_ari"
        dest: "login{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "login"
      - recipe: "service_cle_6.0up03_sles_12_x86-64_ari"
        dest: "service{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "service"
      - recipe: "diag-all_cle_6.0up03_sles_12_x86-64_ari"
        dest: "diag-all_cle_60up03_sles_12_x86-64_ari"
  ...
  testing:
      - recipe: "compute_cle_6.0up03_sles_12_x86-64_ari"
        dest: "{my_custom_prefix}_compute-TEST-{my_other_value}_{date}_{time}.cpio"
        nims_group: "compute-test"
```

The only way to modify this information to customize it for a site is to edit this YAML file directly.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

The following sections describe important things to know to successfully customize the cray_image_groups configuration file.

## What image groups contain

- The cray_image_groups configuration file can contain multiple *image groups* (this example shows two: default and testing). When invoked, `imgbuilder` builds one of these. It builds "default" if no image group name is passed as a parameter.
- Each image group contains a list of *image specifications* that will be built: by default, the standard admin, compute, service, and login images.
- Each image specification is a stanza containing these three fields:

  **`recipe`** An IMPS (Image Management and Provisioning System) image recipe name. This can be customized to specify which image recipe is used to build a specific boot image.

  **`dest`** The destination file name used for the IMPS image root (which may or may not be a bootable cpio file). This can be customized as described below.

  **`nims_group`** The NIMS group to which this image is mapped. The nims_group field is specified only for images that are intended as boot images, so not all specifications have this field (for example, the diag-all image in the default image group).

## How to customize an image root file name using placeholders

Placeholders like {date} can be used to customize an image root name. The `dest` values in the above example contain several such placeholders. At build time, relevant values are substituted for these placeholders. Currently, `imgbuilder` supports the following built-in placeholders for use in the cray_image_groups configuration file:

| | |
|---|---|
| **{date}** | the current system date (e.g., 20140314) |

| | |
|---|---|
| **{time}** | the current system time (e.g., 134514) |
| **{host}** | the current system host name |
| **{user}** | the current username |
| **{cle_release}** | the currently active CLE release |
| **{cle_build}** | the currently active CLE build |
| **{patch}** | the currently active patch |

**IMPORTANT:** When adding one or more placeholders to `dest`, ensure that the whole expression is enclosed by double quotes. For example,

```
dest: "login_cle_{cle_release}-build{cle_build}_sles_12-created{date}.cpio"
```

User-defined placeholders (optional) are also supported for further customization of image names. An example of a user-defined placeholder is {note}, which Cray has added to the image root name in several of the standard image specifications. {note} does not need to be defined in order for the image specifications to work; however, if a site wishes to add something more to the image root file names that contain {note}, a value for {note} can be specified on the command line when running `imgbuilder`, and substitution occurs at runtime. For example, if a site wanted to add the string "favorite" to those image root names, the following command could be used.

```
(EXAMPLE ONLY - DO NOT USE) smw# imgbuilder --map -- note=favorite
```

Other custom placeholders can be defined as well. As with {note}, the key/value pair defining the placeholder would be added to the `imgbuilder` command on the command line. The syntax is two hyphens and a space (`-- `) followed by any number of placeholder definitions as key=value pairs separated by spaces.

For example, this command would tell `imgbuilder` to build the images in the "testing" image group, map them to the NIMS groups specified in that group, and "foo" everywhere for "my_custom_prefix" and "bar" everywhere "my_other_value" appears.

```
(EXAMPLE ONLY - DO NOT USE) smw# imgbuilder --map --image-group testing \
-- my_custom_prefix=foo my_other_value=bar
```

## 3.6.4 Build Boot Images for a Fresh Install

### Prerequisites

This procedure assumes some knowledge of image groups: how they are defined and how they can be customized for a site. See *About Image Groups and How to Customize Them* on page 188 for that information.

### About this task

This procedure uses the `imgbuilder` command to build boot images. The `imgbuilder` command uses information in the cray_image_groups configuration file (`/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`) to know which images to build, how to build them, what to call the built images, and which NIMS (Node Image Mapping Service) groups to map those images to.

When invoked, the `imgbuilder` command builds all of the image specifications from one of the image groups defined in the cray_image_groups configuration file, beginning with the first image specification and working down the list of specifications within that group. Frequently used `imgbuilder` options include:

**`--bootstrap-nims`** To successfully map an image to a node, `imgbuilder` also needs to know which NIMS group that node belongs to, which means the node must have its NIMS group (i.e., its "group" field) populated. But for an initial fresh install, that field may not be populated yet. To ensure that the required node information gets populated prior to building boot images, use the `--bootstrap-nims` option. With this option, `imgbuilder` looks at the "group" field of each node, and if it is empty, `imgbuilder` adds "compute" or "service" depending on the type of that node, as reported by HSS (Hardware Supervisory System).

**`--image-group`** To specify which image group to build, use the `--image-group` option. If that option is not used, `imgbuilder` will build the group called "default."

**`--map`** When `imgbuilder` is invoked with the `--map` option, it maps the image in each image specification to the associated NIMS group (the `nims_group` field).

**`--dry-run`** To see what IMPS and NIMS commands `imgbuilder` would run, without actually running them, use the `--dry-run` option.

`imgbuilder` logs are found at `var/adm/cray/logs/imgbuilder`. For more information, see the `imgbuilder` man page or type `imgbuilder -h`.

## Procedure

**1.** Bootstrap NIMS (Node Image Mapping Service) using `imgbuilder` with the bootstrap option.

**full system** For a full, unpartitioned system:

```
smw# imgbuilder --bootstrap-nims
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# imgbuilder --bootstrap-nims -p p1
```

```
smw# imgbuilder --bootstrap-nims -p p2
```

All nodes have now been assigned to the NIMS service or compute group (i.e., have their "group" field set to either "service" or "compute"), except for the boot and SDB nodes, which need to be assigned to the "admin" group. That is done in a later step.

**2.** Install SMW and CLE patches.

Check for any available CLE 6.0 and SMW 8.0 patches in `/var/adm/cray/release/patchsets`. This directory was created and patches (if any) downloaded to it in *Prepare to Bootstrap the SMW Installation* on page 79.

The first substep prevents the patch scripts from creating images and mapping them to NIMS. Image creation and NIMS mapping are done at the end of this procedure instead, after the login and DAL nodes have been assigned and any changes to the default image group have been made.

> **NOTE:** (SMW HA only) Make a note of all patch sets that will be applied on the first SMW. The second SMW must have exactly the same patch sets.

a. Temporarily suppress building and mapping images.

```
smw# export PATCHSET_BUILD_IMAGES=false
smw# echo $PATCHSET_BUILD_IMAGES

smw# export PATCHSET_NIMS_TIMING=deferred
smw# echo $PATCHSET_NIMS_TIMING
```

b. Follow all of the instructions in the patch README files.

These instructions will include running the LOAD script and the INSTALL script for each patch, and there may be additional steps for some patches, such as running `xtzap` again to update firmware from an SMW patch.

**3.** Assign the boot and SDB nodes to the NIMS admin group.

Nodes in the NIMS admin group will be assigned the admin boot image for booting. For information about the admin image, see *About the Admin Image* on page 25.

This example uses c0-0c0s0n1 and c0-0c0s1n1 as the admin (boot and SDB) nodes. Substitute the correct cnames for this site when using these commands.

**full system** For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS admin group:

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1
```

**partitioned** For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p1
```

```
smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p2
```

**NOTE:** If a custom recipe will be created and used for the SDB node(s) instead of the admin recipe (for example, to add content for a workload manager), assign the SDB node(s) to a different NIMS group, where the name of the NIMS group may have the same name as the custom recipe.

**4.** Assign login nodes to the NIMS login group.

Nodes in the NIMS login group will be assigned the login boot image for booting. To assign more than one node, use a space-separated list of nodes. This example uses c0-0c0s1n1 and c0-0c0s3n2 as the login nodes. Substitute the correct cnames for this site when using these commands.

**full system** For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g login c0-0c0s1n1 c0-0c0s3n2
```

**partitioned** For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS login group:

```
smw# cnode update -G service -g login c0-0c0s1n1 -p p1
```

```
smw# cnode update -G service -g login c0-0c0s3n2 -p p2
```

**5.** For systems using direct-attached Lustre (DAL), assign DAL service nodes to the NIMS dal group.

Nodes in the NIMS dal group are assigned the DAL boot image for booting. To assign more than one node, use a space-separated list of nodes. This example uses c0-0c0s2n1 and c0-0c0s2n2 as the DAL nodes. Substitute the correct cnames for this site when using these commands.

**full system** For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS dal group:

```
smw# cnode update -G service -g dal c0-0c0s2n1 c0-0c0s2n2
```

**partitioned** For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS dal group:

```
smw# cnode update -G service -g dal c0-0c0s2n1 c0-0c0s2n2 -p p1
```

```
smw# cnode update -G service -g dal c0-0c0s2n1 c0-0c0s2n2 -p p2
```

PREPARE CRAY IMAGE GROUPS AND CUSTOM RECIPES

**6.** Customize the cray_image_groups configuration file, as needed, by editing `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml`. Move recipe stanzas into the default group for anything to be built by default, and modify or create other image groups as appropriate for this site.

a. Ensure that the admin image specification is in the default image group.

Fresh installs of this release will already have this stanza in the default image group, but sites with existing installations will need to add it to the end of the default image group.

```
cray_image_groups:
  default:
      ...
      - recipe: "admin_cle_6.0up03_sles_12_x86-64_ari"
        dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "admin"
```

b. For systems using DAL, ensure that this DAL stanza (image specification) are in the default image group, or customize and use the `tmpfs-w-dal` image group, which already has it.

```
cray_image_groups:
  default:
      ...
      - recipe: "dal_cle_6.0up03_centos_6.5_x86-64_ari"
        dest: "dal{note}_cle_{cle_release}-build{cle_build}
{patch}_centos_6.5-created{date}.cpio"
        nims_group: "dal"
      ...
```

c. For systems using Netroot for either compute or login nodes, those images will be created at a later step in the process. See *Configure Netroot* on page 225.

d. Ensure that any site custom recipes are added to the default image group or a site-specific stanza so that they will get built.

**7.** Build images and map them to NIMS groups.

Create a set of images as defined in `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml` and map them to the specified NIMS groups using the `--map` option.

> **NOTICE:** Building images takes approximately 5 minutes for each type of recipe in the image group. Building Netroot images takes slightly longer because there are more RPMs to be installed. If four different recipes are in the default image group, it will take about 20 minutes to build the images. If fio-service, Netroot for compute (initrd-compute-large), and Netroot for login (initrd-login-large) are added to the set it may take about 40 minutes.

**full system** To build the images in the "default" image group and map them to the NIMS groups specified in "default" for a full, unpartitioned system:

```
smw# imgbuilder --map
```

**partitioned** To build the images in the "default" image group and map them to the NIMS groups specified in "default" for a partitioned system, with partitions p1 and p2:

```
smw# imgbuilder --map -p p1 -p p2
```

**8.** (Alternative to step *7* on page 193) Build images and then manually map them to NIMS groups.

As an alternative to mapping the images using the `--map` option, that mapping can be done manually.

a. Build the images.

```
smw# imgbuilder
```

b. Map the images to specified NIMS groups.

Replace the cpio file names in these examples with the file names from the `imgbuilder` output in step a.

If any other boot images have been created for special nodes, assign them with similar `cnode update` commands filtered for the NIMS groups to which those special nodes have been assigned.

**full system** To map the images to specified NIMS groups for a full, unpartitioned system:

```
smw# cd /var/opt/cray/imps/boot_images
smw# ls -ltr

smw# cnode update -i compute_img.cpio --filter group=compute

smw# cnode update -i service_img.cpio --filter group=service

smw# cnode update -i login_img.cpio --filter group=login

smw# cnode update -i dal_img.cpio --filter group=dal
```

**partitioned** To map the images to specified NIMS groups for a partitioned system, with partitions p1 and p2:

```
smw# cd /var/opt/cray/imps/boot_images
smw# ls -ltr
```

```
smw# cnode update -i compute_img.cpio --filter group=compute -p p1

smw# cnode update -i service_img.cpio --filter group=service -p p1
```

```
smw# cnode update -i login_img.cpio --filter group=login -p p1
smw# cnode update -i dal_img.cpio --filter group=dal -p p1
```

```
smw# cnode update -i compute_img.cpio --filter group=compute -p p2
smw# cnode update -i service_img.cpio --filter group=service -p p2
smw# cnode update -i login_img.cpio --filter group=login -p p2
smw# cnode update -i dal_img.cpio --filter group=dal -p p2
```

### 3.6.5 Set the Turbo Boost Limit

Turbo boost limiting is NOT supported on Intel® Xeon Phi™ "Knights Landing" (KNL) or on Intel® Xeon® "Sandy Bridge" processors.

Because Intel processors have a high degree of variability in the amount of turbo boost each processor can supply, limiting the amount of turbo boost can reduce performance variability and reduce power consumption. Turbo boost can be limited by setting the `turbo_boost_limit` kernel parameter to one of these valid values:

| Value | Result |
|---|---|
| 0 | Disable turbo boost. |
| 100 | Limits turbo boost to 100 MHz. |
| 200 | Limits turbo boost to 200 MHz. |
| 300 | Limits turbo boost to 300 MHz. |
| 400 | Limits turbo boost to 400 MHz. |
| 999 (default) | No limit is applied. |

The limit applies only when a high number of cores are active. On an N-core processor, the limit is in effect when the active core count is N, N-1, N-2, or N-3. For example, on a 12-core processor, the limit is in effect when 12, 11, 10, or 9 cores are active.

### Set or Change the Turbo Boost Limit Parameter

To make a persistent change, use `cnode` (as crayadm or root) to change the parameter. This change will take effect later when the nodes are rebooted. Note that the following commands target all nodes or all compute nodes. To specify individual nodes, add their cnames at the end of the command line.

1. To list the current kernel parameters for a full or partitioned system:

   **full system**

   For a full, unpartitioned system:

   ```
   smw# cnode list
   ```

   **partitioned**

   For a partitioned system:

   ```
   smw# cnode list --partition p1
   ```

2. To change the `turbo_boost_limit` kernel parameter for all compute nodes in a full or partitioned system, substitute one of the values listed above for *`value`* in one of these commands:

   **full system** For a full, unpartitioned system:

   ```
   smw# cnode update --filter group=compute \
   --add-parameter turbo_boost_limit=value
   ```

   **partitioned** For a partitioned system:

   ```
   smw# cnode update --filter group=compute --partition p1 \
   --add-parameter turbo_boost_limit=value
   ```

3. To remove the change, if needed, use one of these commands:

   **full system** For a full, unpartitioned system:

   ```
   smw# cnode update --filter group=compute \
   --remove-parameter turbo_boost_limit
   ```

   **partitioned** For a partitioned system:

   ```
   smw# cnode update --filter group=compute --partition p1 \
   --remove-parameter turbo_boost_limit
   ```

### 3.6.6 Check NIMS Information during a Fresh Install

## About this task

This procedure lists NIMS (Node Image Mapping Service) information: which maps are active on the SMW and what NIMS information is stored for each node.

## Procedure

1. Check active NIMS maps.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap list
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cmap list --partition p1
   ```

   ```
   smw# cmap list -p p2
   ```

2. Check the default config set of the active NIMS map.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap list --fields default_config_set map_name
   ```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# cmap list --fields default_config_set --partition p1 p1_map_name
```

```
smw# cmap list --fields default_config_set --partition p2 p2_map_name
```

If this is not the desired default config set, use *Set Default Config Set for a NIMS Map* on page 338 to change it. If selected nodes need to use a different config set, see *Set Config Set for a Node* on page 338.

**3.** Check NIMS information for each node.

**full system** For a full, unpartitioned system:

```
smw# cnode list
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# cnode list -p p1
```

```
smw# cnode list -p p2
```

**4.** Check NIMS information for each NIMS group.

```
smw# cnode list --filter group=admin
smw# cnode list --filter group=service
smw# cnode list --filter group=login
smw# cnode list --filter group=compute
```

Check any additional NIMS groups that may have been created for Netroot compute and login nodes (typically only when Netroot is used on only a subset of compute and login nodes instead of all of them, so the NIMS compute and login groups cannot be used for that subset).

```
smw# cnode list --filter group=compute_netroot
smw# cnode list --filter group=login_netroot
```

Check any additional NIMS groups that may have been created for DataWarp with Fusion IO SSDs.

```
smw# cnode list --filter group=fio-service
```

Check any additional NIMS groups that may have been created with WLM (workload manager) or other site names.

```
smw# cnode list --filter group=wlm-admin
smw# cnode list --filter group=wlm-service
smw# cnode list --filter group=wlm-login
```

## 3.6.7 Boot the System using a Boot Automation File

### Prerequisites

This procedure assumes that configuration and image preparation are complete and the system is now ready to boot.

## About this task

This procedure describes how to customize a boot automation file and use it to boot the XC system with `xtbootsys`. For more information about boot automation files, see *About Boot Automation Files* on page 24.

## Procedure

CREATE SITE BOOT AND SHUTDOWN AUTOMATION FILES

**1.** Create a site boot automation file.

Copy the Cray generic boot automation file and rename it. Add site customizations, as needed. For sites booting tmpfs images (instead of Netroot) with no SDB node failover, no changes may be necessary. Sites that choose to boot Netroot images will make those changes later in the process after this first boot with tmpfs.

Replace *`hostname`* with the host name of the system that will use this automation file.

```
smw# cp -p /opt/cray/hss/default/etc/auto.generic \
/opt/cray/hss/default/etc/auto.hostname.start
```

**2.** Create a site automation file for shutting down the system.

Copy the Cray shutdown automation file and rename it. Add site customizations, as needed. For example, customization may be needed to cleanly shut down queues for the workload manager (WLM) on MOM or SDB nodes. The specific commands will vary based on the WLM.

Replace *`hostname`* with the host name of the system that will use this automation file.

```
smw# cp -p /opt/cray/hss/default/etc/auto.xtshutdown \
/opt/cray/hss/default/etc/auto.hostname.stop
```

CUSTOMIZE THE SITE AUTOMATION FILES

**3.** If the SDB boot image is too large for a PXE boot (often the case if a WLM is installed in that image), change `auto.`*`hostname`*`.start` to enable booting the SDB node(s) via HSN rather than PXE. See *About Boot Automation Files* on page 24 and *About the Admin Image* on page 25 for more information.

**4.** If boot or SDB node failover is used, add boot node or SDB node failover to `auto.`*`hostname`*`.start`.

If either boot node failover or SDB node failover will be used, then the boot automation file should have a setting to ensure that STONITH has been enabled on the blade that has the primary boot node and the primary SDB node. The STONITH setting does not survive a power cycle. To maintain the STONITH setting, add these lines to the boot automation file.

Use the blade that contains the primary boot node. For example, if the primary boot node is c0-0c0s0n1, then the blade to use is c0-0c0s0. Add these lines **before** the line for booting the boot node.

```
# Set STONITH for primary boot node
lappend actions {crms_exec "xtdaemonconfig c0-0c0s0 stonith=true"}
```

Use the blade that contains the primary SDB node. For example, if the primary SDB node is c0-0c1s0n1, then the blade to use is c0-0c1s0. Add these lines **before** the line for booting the SDB node.

```
# Set STONITH for primary SDB node
lappend actions {crms_exec "xtdaemonconfig c0-0c1s0 stonith=true"}
```

**5.** If boot or SDB node failover is used, enable the `xtfailover_halt` command in the `auto.`*`hostname`*`.stop` file.

Uncomment the second of these lines in `auto.`*`hostname`*`.stop`. The `xtfailover_halt` command ensures that the `xtbootsys` shutdown process sends a STOP NMI to the failover nodes.

```
# Enable the following line if boot or sdb failover is enabled:
lappend actions { crms_exec \
"/opt/cray/hss/default/bin/xtfailover_halt --partition $data(partition,given) --shutdown" }
```

**6.** If cray_login.settings.login_nodes.data.login_prohibited_after_boot is set to true, then to allow user access later, remove the `/etc/nologin` file using one of the following methods:

- Remove it manually.

  This can be done only after all of the CLE nodes have been booted and the system is ready for users to log in. To choose this option, wait until step *8* on page 200.

- Set up the boot automation file to remove it.

  Add the following lines to the boot automation file. Place them after the lines that boot all of the compute nodes and after any other special commands that prepare the system for user access.

  ```
  # Remove /etc/nologin from all service nodes as the last step in the system boot.
  lappend actions { crms_exec_on_bootnode "sdb" "pcmd -r -n ALL_SERVICE 'rm /etc/nologin'" }
  ```

BOOT THE SYSTEM

**7.** Run `xtbootsys` with `auto.`*`hostname`*`.start`.

**full system** For a full, unpartitioned system:

```
smw# su - crayadm
crayadm@smw> xtbootsys -a auto.hostname.start
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# su - crayadm

crayadm@smw> xtbootsys -p p1 -a auto.hostname.start.p1

crayadm@smw> xtbootsys -p p2 -a auto.hostname.start.p2
```

The boot automation files have different names in this example to indicate that partitions may use different automation files, if needed.

**Trouble?** If there are any problems booting CLE, see the *XC™ Series Boot Troubleshooting Guide* (S-2565) for techniques to determine what might be causing the problem.

8. Remove the `/etc/nologin` file manually after the system boots, as needed.

   If cray_login.settings.login_nodes.data.login_prohibited_after_boot is set to true, and the `/etc/nologin` file was NOT removed by means of an entry in the boot automation file in step *6* on page 199, remove it manually after all of the CLE nodes have been booted and the system is ready for users to log in.

   ```
   sdb# pcmd -r -n ALL_SERVICE "rm /etc/nologin"
   ```

**Build image roots on the SMW during system boot to save time.**

Image building can be done at any time on the SMW without negative impact to the running CLE system. To save time, the following installation tasks can be started on the SMW while the CLE nodes are booting.

- Build Netroot images on the SMW. See *Configure Netroot Images* on page 226.
- Build the PE image root on the SMW. See *Install Cray Programming Environment (PE) Software* on page 234.
- Build any WLM or other custom image roots on the SMW.

## 3.6.8 Run Tests after Boot is Complete

### Prerequisites

This procedure assumes the following:

- The system has completed booting.
- The compute nodes are “interactive," not under workload manager (WLM) control.
- ALPS is available.

If ALPS is not available and Slurm is used as the WLM, then the compute nodes can be either "interactive" or "batch," and `srun` (the equivalent Slurm command) should be used instead of the `aprun` commands in the steps that follow.

### About this task

Log in to the login node as crayadm. This can be done from the SMW to the boot node to the login node or directly from another computer to the login node without passing through the SMW and boot node. Then perform these rudimentary functionality checks.

### Procedure

1. Run `apstat` to get the number of nodes to use for the following commands.

   ```
   crayadm@login> NUMNODES=$(($(apstat -v | grep XT | awk "{print \$3}")))
   crayadm@login> echo NUMNODES is $NUMNODES
   ```

2. Verify that all nodes run (from `/tmp`).

   ```
   crayadm@login> cd /tmp
   crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
   ```

3. Verify that the home directory is working by running a job.

```
crayadm@login> cd ~
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

4. Verify that the Lustre directory is working by running a job.

```
crayadm@login> cd /lustre_file_system
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

CHECK CURRENT STATE OF COMPUTE NODE SSDs

The next step is intended only for XC systems that have compute nodes with SSDs, for example, systems with DataWarp SSDs or Intel® Xeon Phi™ "Knights Landing" processors.

5. Run `xtcheckssd` to ensure that SMW databases have the current state of compute node SSDs.

```
root@login# pcmd -r -n ALL_COMPUTE "/opt/cray/ssd/bin/xtcheckssd"
```

## 3.6.9 Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev

### Prerequisites

To run `xtgetrev`, the boot node must be booted and accessible.

### About this task

System administrators use the `xtgetrev` and `xtshowrev` commands to gather and display machine, software revision, Field Notice (FN), and patch set information. The `xtgetrev` command collects information from the administrator and from the SMW and boot node. The `xtshowrev` command displays that information, even when CLE is not running. These tools are useful for gathering information to send to Cray after installing a software upgrade, FN, or patch set and for help with troubleshooting.

This procedure describes how to use these two tools on a Cray XC Series system. These steps (except for running `xtshowrev`) must be executed as root.

> **ATTENTION:** Any information that is submitted to site_install_data@cray.com will be used only within Cray, Inc. and will not be made public. The `xtshowrev` command does not submit any information to Cray automatically.

### Procedure

1. Load the module to enable use of the tools.

```
smw# module load xtshowrev
```

2. Run `xtgetrev` to create and populate the initial files.

Only root can run this command. The first time xtgetrev is executed, when there are no files populated, the tool will prompt for site information. If the boot node does not have passwordless ssh, then the tool will prompt for the password.

This example uses *CRAY/INTERNAL* as the site name and *9999* as the serial number of the machine. Substitute the actual values for this site.

```
smw# xtgetrev
xtgetrev: No site information has been defined.

Site name: CRAY/INTERNAL
Serial Number: 9999
System Name [panda1]:
System Type [XC40]:

<snip>
```

**Trouble?** If xtgetrev does not allow entry of those values, it may be because the initial configuration files have been created already. In that case, manually edit /etc/opt/cray/release/pkginfo/site_config and modify 'site name:' and 'serial number:' values.

```
smw# vi /etc/opt/cray/release/pkginfo/site_config
```

**3.** Run xtshowrev to see the formatted information.

Note the prompt, which indicates that any user can run this command.

```
user@smw> xtshowrev
Site:                    CRAY/INTERNAL
S/N:                     9999
System Type:             XC40
Install Date:            2016-06-01

<snip>
user@smw>
```

## 3.6.10 Test xtdumpsys and cdump

### Prerequisites

This procedure assumes that the system has been booted.

### About this task

This procedure tests the xtdumpsys and cdump tools. The example output is for illustrative purposes only. Actual output may differ for the current release.

### Procedure

**1.** Start an xtdumpsys typescript.

Start a new window. Start a typescript session for xtdumpsys in that new window.

```
smw# su - crayadm
crayadm@smw> export TODAY=`date +%Y%m%d`
crayadm@smw> . /etc/opt/cray/release/cle-release
crayadm@smw> mkdir -p /home/crayadm/dump/${TODAY}_${BUILD}
crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
crayadm@smw> script -af hss.xtdumpsys
```

2. Start `xtdumpsys`.

   Start the dump, but do not press **Ctrl-d** until step *5* on page 204. When `xtdumpsys` asks for a dump reason, it will have created the dump directory.

   For a full system:

```
crayadm@smw> xtdumpsys
  INFO: Beginning dump
 INFO: Gathering system partition information
 INFO: Gathering system hardware information
 INFO: No session specified, defaulting to current.
 INFO: Moving temporary log files to the dump directory.
 INFO:
##############################################################################
 INFO: # Your dump is available in /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-
NNNNNNNNNN #
 INFO:
##############################################################################
 Enter reason for dump:
 (an EOF terminates input, usually CTRL-D)
```

   For a partitioned system:

```
crayadm@smw> xtdumpsys  -p p1
crayadm@smw> xtdumpsys  -p p2
```

3. Start a `cdump` typescript in a different window.

   Start another window. Start a typescript session for `cdump` in that window.

```
smw# su - crayadm
cdump crayadm@smw> export TODAY=`date +%Y%m%d`
cdump crayadm@smw> . /etc/opt/cray/release/cle-release
cdump crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
cdump crayadm@smw> script -af hss.cdump
```

4. Dump a node with `cdump`.

   Change to the directory created in the `xtdumpsys` window (after `INFO: # Your dump is available in`), then use `cdump` to dump a compute node that successfully booted.

```
cdump crayadm@smw> cd /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-NNNNNNNNNN
cdump crayadm@smw> mkdir cdumps; cd cdumps
```

   For a full system (example uses the c0-0c0s3n0 node):

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot c0-0c0s3n0
 Wed Mar 1 09:08:08 CDT 2017 start cdump
 ...
makedumpfile Completed.
- done
```

```
Wed Mar 1 09:08:08 CDT 2017 cdump: # of nodes 1
 success 1
 failed  0
 skipped 0
cdump crayadm@smw> exit
```

For a partitioned system, use the host name to specify which boot node. This example uses boot-p1 to `cdump` the c0-0c0s4n0 node in the p1 partition.

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot-p1 c0-0c0s4n0
cdump crayadm@smw> exit
```

5. Continue `xtdumpsys`: enter a reason.

   After `cdump` completes, return to the `xtdumpsys` window and enter a reason.

```
xtdumpsys window> testdump
```

   Then enter an end-of-file (**Ctrl-d**) to end the dump reason.

```
xtdumpsys window> <Ctrl-d>
 testdump
 INFO: Dump reason:
 ...
 INFO:
################################################################################
 INFO: # Your dump is available in /var/opt/cray/dump/
p0-20170301t081927-1304240904 #
 INFO:
################################################################################
 INFO: No post-processing plugin found at '/etc/opt/cray/dumpsys/
postprocessing.py'
 INFO: Example plugins can be found at '/opt/cray/dumpsys/
1.2.5-1.0000.35873.20.1/bin/plugins/examples/postprocessing.py.*'
 INFO: Cleaning up
```

```
xtdumpsys crayadm@smw> exit
```

6. Remove dump directory, if desired.

   If there are no errors, it is probably safe to delete the dump directory.

```
xtdumpsys crayadm@smw> rm -rf /var/opt/cray/dump/pX-YYYYMMDDtHHMMSS-NNNNNNNNNN
crayadm@smw> exit
```

## 3.6.11 Make a Post-boot Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after booting the CLE system.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

## Procedure

1. List the available snapshots on the system.

   ```
   smw# snaputil list
   ```

2. Set the SNAPSHOT environment variable using the currently booted snapshot name.

   ```
   smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
   smw# echo $SNAPSHOT
   ```

   Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

   (SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

   ```
   smw# snaputil create ${SNAPSHOT}.postboot
   ```

### 3.6.12 Make a Post-boot Backup of Current Global and CLE Config Sets

## About this task

This procedure uses the cfgset command to create a post-boot backup of the global and CLE config sets.

## Procedure

1. Back up the current global config set.

   ```
   smw# cfgset create --clone global global-postboot-${TODAY}
   ```

2. Back up the current CLE config set.

   This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

   ```
   smw# cfgset create --clone p0 p0-postboot-${TODAY}
   ```

# 3.7 Configure Other Features and Services

At this stage in the installation/configuration process, the basic SMW and CLE software has been installed, configured, and booted. These procedures complete the configuration of a functional system.

Note that the system cannot be in use during the power management configuration procedure. If this is not a fresh install, CLE must be shut down before performing that procedure. All of the other procedures require CLE to be running.

| | |
|---|---|
| REQUIRED | *Configure Power Management* on page 206 |

| | |
|---|---|
| *recommended* | *Push Diag Image to Boot Node and Update the Diags Bind Mount Profile* on page 210 |
| *recommended* | *Reduce Impact of Btrfs Periodic Maintenance on SMW Performance* on page 212 |
| optional (Required if using DAL) | *Configure Direct-attached Lustre (DAL)* on page 213 |
| optional (Optional if using DAL) | *LMT Configuration for DAL* on page 220 (Lustre Monitoring Tool for direct-attached Lustre) |
| optional (Required if using SEC) | *Configure the Simple Event Correlator (SEC)* on page 225 |
| optional (Required if using SEDC) | *Enable System Environmental Data Collections (SEDC)* on page 225 |
| optional (Required if using Netroot) | *Configure Netroot* on page 225 |
| optional | *Prevent Unintentional Re-creation of Mail Configuration Files* on page 232 |

## 3.7.1 Configure Power Management

### Prerequisites

This is a required step in bringing up a Cray XC system with releases later than CLE 6.0 UP01 / SMW 8.0 UP01. The PostgreSQL database on the SMW is needed even if a site will be using a remote (off-SMW) database node to store Power and SEDC data.

> **NOTE:** (SMW HA only) Skip this procedure if doing a fresh install of the first SMW in an SMW HA system, and the Cray SMWHA software will be installed immediately afterwards, because power management for the SMW HA system will be configured later in the HA fresh install process.

This procedure assumes that a disk drive is available for use as a dedicated drive for the PMDB. The drive should be physically located within the SMW at slot 4. On a Dell PowerEdge™ R815 Rack Server, the device for PMDISK is `/dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0`. On a Dell PowerEdge™ R630 Rack Server, the device for PMDISK is `/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:4:0`.

To determine which physical slot maps to a drive (in case the SMW at this site uses a slot or device name different than those listed above), use this command:

```
smw# smwmapdrives
List of SMW-installed disk drives
---------------------------------
Physical slot 0:
    /dev/sdbu
    /dev/disk/by-id/
    /dev/disk/by-id/scsi-SATA_ST9500620NS_9XF3BGQ5
    /dev/disk/by-path/pci-0000:05:00.0-sas-phy7-0x4433221107000000-lun-0
Physical slot 1:
    /dev/sdbx
    /dev/disk/by-id/
    /dev/disk/by-id/scsi-SATA_ST9500620NS_9XF3BGWA
    /dev/disk/by-path/pci-0000:05:00.0-sas-phy6-0x4433221106000000-lun-0
<snip>
```

The system cannot be in use during this procedure. If this is not a fresh install, CLE must be shut down.

## About this task

Power Management allows Cray® XC Series™ systems to operate more efficiently. By monitoring, profiling, and limiting power usage administrators can:

- Increase system stability by reducing heat dissipation
- Reduce system cooling requirements
- Reduce site cooling requirements
- Reduce utility costs by minimizing power usage when rates are the highest
- Respond to external environmental conditions and prevent power outages
- Calculate the actual power cost for individual users and/or jobs

⚠ **CAUTION:** Do not use this procedure in preparation for setting up an SMW HA system. As part of the HA configuration the `SMWHAconfig` copies the contents of the PMDB to a shared power management RAID disk. For more information see *XC™ Series SMW HA Installation Guide* (S-0044).

These steps are performed as `root`.

## Procedure

1. Verify that the PMDISK is inserted into the SMW by entering the correct device name. This example, and the ones that follow, are for a Dell R815.

```
smw# fdisk -l /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0: 931.5 GiB,
1000204886016 bytes, 1953525168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x10692081
```

2. Create a new primary partition for the PMDISK, and write it to the partition table. If there are any existing partitions on this disk, manually delete them first.

```
smw# fdisk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0
Welcome to fdisk (util-linux 2.25).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
  Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1953525167, default 2048): [press return]
Last sector, +sectors or +size{K,M,G,T,P} (2048-1953525167, default 1953525167): [press
return]

Created a new partition 1 of type 'Linux' and of size 931.5 GiB.

Command (m for help): w
```

```
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

3. Verify that the partition has been created. On a Dell R815 this should be device `/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1` . On a Dell R630 this should be `/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:4:0-part1`.

```
smw# fdisk -l \
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0

Disk /dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0: 931.5 GiB, 1000204886016
bytes, 1953525168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x96c1b0f0

Device                                                                    Boot Start        End
Sectors   Size Id Type
/dev/disk/by-path/pci-0000:05:00.0-sas-phy3-0x4433221103000000-lun-0-part1       2048 1953525167
1953523120 931.5G 83 Linux
```

4. Create an ext4 file system on the PMDISK partition.

```
smw# mkfs.ext4 \
/dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0-part1

Creating filesystem with 244190390 4k blocks and 61054976 inodes
Filesystem UUID: 6d791409-e327-4620-a80c-2933271b3eec
Superblock backups stored on blocks:
       32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
       4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
       102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

5. Stop the RSMS services.

```
smw# systemctl stop rsms
smw# systemctl status rsms
rsms.service - hss daemon control
  Loaded: loaded (/usr/lib/systemd/system/rsms.service; enabled)
  Active: inactive (dead) since Wed 2015-11-04 15:42:04 CST; 19s ago
 Process: 5471 ExecStop=/opt/cray/hss/default/bin/hssctl stop (code=exited, status=0/
SUCCESS)
 Process: 30305 ExecStart=/opt/cray/hss/default/bin/hssctl start (code=exited, status=0/
SUCCESS)

Nov 03 16:01:43 smw hssctl[30305]: Starting daemons: erd erdh state_mana...md
Nov 04 15:42:04 smw hssctl[5471]: Stopping daemons: sec_cmd boot_cmds ca...rd
Hint: Some lines were ellipsized, use -l to show in full.
```

6. Verify that the RSMS services are stopped. While the RSMS services are stopped, the system may continue to run applications, however the high-speed network will be throttled until RSMS is restarted.

```
smw# rsms status
PID             DAEMON                    STATE                   UPTIME
```

```
               erd                  stopped
               erdh                 stopped
               state_manager        stopped
               nid_mgr              stopped
               bootmanager          stopped
               sedc_manager         stopped
               xtpmd                stopped
               erfsd                stopped
               xtremoted            stopped
               xtpowerd             stopped
               nimsd                stopped
               xtsnmpd              stopped
               xtdiagd              stopped
```

7. Run the `xtmvpmdb` script.

```
smw# xtmvpmdb /dev/disk/by-path/pci-0000:05:00.\
0-sas-0x4433221103000000-lun-0-part1 ext4
- Checking userid
- Checking destination directory name
- Checking destination directory existence

Move database to: /dev/disk/by-path/pci-0000:05:00.0-sas-0x4433221103000000-lun-0-part1
[y/n] [y]: y
- Checking current PM database directory existence
- Checking for booted system
- Checking for rsms daemons
- Creating directory /media/temp_pgsql_data
Dir: /media/temp_pgsql_data created
- Checking status of PM database process
Checking for PostgreSQL 9.3.8: ..running
postgresql.service - LSB: Start the PostgreSQL master daemon
  Loaded: loaded (/etc/init.d/postgresql)
  Active: active (exited) since Tue 2015-11-03 15:38:45 CST; 24h ago
 Process: 16633 ExecReload=/etc/init.d/postgresql reload (code=exited, status=0/SUCCESS)
 Process: 16255 ExecStart=/etc/init.d/postgresql start (code=exited, status=0/SUCCESS)


- Stopping PM database
- Copy contents of /var/lib/pgsql to /media/temp_pgsql_data
- This may take a few minutes to complete.
- Rename previous DB directory from: /var/lib/pgsql to: /var/lib/pgsql.
11-04-2015t15:43:04
- Unmount device from temporary mount point: /media/temp_pgsql_data
- Unmount btrfs subvolume: /var/lib/pgsql
- Mount device at permanent mount point: /var/lib/pgsql
- Add mount point to /etc/fstab
- Start PM database

- Transfer of PM database complete.
```

8. Restart the RSMS services and verify that the daemons are starting.

```
smw# systemctl start rsms
smw# systemctl status rsms
rsms.service - hss daemon control
  Loaded: loaded (/usr/lib/systemd/system/rsms.service; enabled)
  Active: active (exited) since Wed 2015-11-04 15:44:24 CST; 9s ago
 Process: 5471 ExecStop=/opt/cray/hss/default/bin/hssctl stop (code=exited, status=0/
SUCCESS)
 Process: 9227 ExecStart=/opt/cray/hss/default/bin/hssctl start (code=exited, status=0/
SUCCESS)

Nov 04 15:44:24 smw hssctl[9227]: Starting daemons: erd erdh state_manag...md
Hint: Some lines were ellipsized, use -l to show in full.
```

9. Verify that the RSMS services are running.

```
smw# rsms status
PID             DAEMON              STATE            UPTIME
9306            erd                 running          Wed 2015-11-04 15:43:27 CST
9435            erdh                running          Wed 2015-11-04 15:43:30 CST
9560            state_manager       running          Wed 2015-11-04 15:43:31 CST
9691            nid_mgr             running          Wed 2015-11-04 15:43:32 CST
9827            bootmanager         running          Wed 2015-11-04 15:43:32 CST
9953            sedc_manager        running          Wed 2015-11-04 15:43:33 CST
10703           xtpmd               running          Wed 2015-11-04 15:43:44 CST
11487           erfsd               running          Wed 2015-11-04 15:43:50 CST
12247           xtremoted           running          Wed 2015-11-04 15:43:56 CST
12521           xtpowerd            running          Wed 2015-11-04 15:44:00 CST
12688           nimsd               running          Wed 2015-11-04 15:44:04 CST
12855           xtsnmpd             running          Wed 2015-11-04 15:44:08 CST
13019           xtdiagd             running          Wed 2015-11-04 15:44:12 CST
```

## 3.7.2 Push Diag Image to Boot Node and Update the Diags Bind Mount Profile

### Prerequisites

This procedure assumes that the system has been booted after a fresh install.

### About this task

The online diagnostics image provides some useful tools that are made available on CLE nodes through the Cray Image Binding service using the profile for the diag image root. This procedure describes how to push the diag image root to the boot node. It also enables that service and configures it to reference the correct diag image and enable the diag profile.

### Procedure

1. Determine the name of the image root used in the diag profile in cray_image_binding.

   In this example, p0 is the name of the CLE config set.

   ```
   smw# cfgset search -t image -s cray_image_binding p0 | grep diag
   cray_image_binding.settings.profiles.data.diags.image: diags_cle_6.0up03_sles_12_x86-64
   ```

2. Check for an existing diag image root.

   ```
   smw# image list | grep diag
   diag-all_cle_6.0up03_sles_12_x86-64
   ```

3. Push the diag image root to the boot node.

   In this example, the diag image root is diags_cle_6.0up03_sles_12_x86-64.

   ```
   smw# image push -d boot diags_cle_6.0up03_sles_12_x86-64
   ```

   **Trouble?** If passwordless `ssh` has not been prepared between root@smw and root@boot, then the system will prompt for the password for root@boot twice.

4. Update cray_image_binding, which is in the CLE config set (p0 in this example).

```
smw# cfgset update -s cray_image_binding -m interactive p0
```

The configurator displays the **Service Configuration Menu**. The service name and status appear at the top of the menu. That menu also includes a list of settings. The Cray Image Binding service has a single setting: `profiles`. Under it is a list of bind mount profile entries.

**5.** If this service is not yet enabled, enable it now.

This example shows the service as disabled. Enter **`E`** to enable it.

```
Service Configuration Menu (Config Set: p0, type: cle)
  cray_image_binding       [status: disabled] [validation: skipped]
...
IMPS Image Binding Service Menu [default: save & exit - Q] $ E
```

**6.** Select the `profiles` setting to configure it.

Enter **`1`** to select the `profiles` setting, and then enter **`C`** to configure that setting.

```
IMPS Image Binding Service Menu [default: save & exit - Q] $ 1
...
IMPS Image Binding Service Menu [default: save & exit - Q] $ C
```

The configurator displays guidance about the `profiles` setting and a numbered list of profile entries that have already been added. A 'PE' profile and a 'diags' profile should be in that list.

**7.** Change the value of the 'diag' profile image field.

a. Enter the number for the 'diag' profile followed by 'a' and '*' to select and edit the field for the diag profile image name.

In this example, the number of the 'diag' profile is 2.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 2a*
```

b. Enter the name of the diag image.

In this example, the diag image root is diags_cle_6.0up03_sles_12_x86-64. This must be the same as the image root that was pushed to the boot node.

```
cray_image_binding.settings.profiles.data.diags.image
[<cr>=keep 'diags_cle_6.0up03_sles_12_x86-64', <new value>, ?=help, @=less]
$ <cr>
```

**8.** Enable the diags profile.

Enable the profile only after the diag image root has been pushed to the boot node.

a. Enter the number for the diags profile followed by 'd' and '*' to select the field for enabling the diags profile.

In this example, the number of the diags profile is 2.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 2d*
```

b. Enter `true`, then press **Enter**.

```
cray_image_binding.settings.profiles.data.diags.enabled
[<cr>=keep 'false', <new value>, ?=help, @=less] $ true
```

**9.** Set the profile entries, and then save changes and exit the configurator.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
...

IMPS Image Binding Service Menu [default: save & exit - Q] $ Q
```

**10.** Validate the config set.

```
smw# cfgset validate p0
```

**11.** To use diags on the system, reboot the system.

## 3.7.3 Reduce Impact of Btrfs Periodic Maintenance on SMW Performance

### About this task

Btrfs (B-tree file system) runs periodic maintenance. The weekly and monthly maintenance scripts, which include balance, trim, and scrub actions, can consume large amounts of compute resource. This can impact a site's ability to use the SMW for normal operations, even using SSH to log into nodes. This procedure describes how to reduce the impact to SMW performance by controlling when these scripts are run.

### Procedure

**1.** Create a file `/etc/cron.d/cray_btrfs.cron`.

The new cron file needs to be in `/etc/cron.d` because the btrfs RPM installs links to maintenance scripts into the `/etc/cron.{weekly,monthly}` directories.

```
smw# vi /etc/cron.d/cray_btrfs.cron
```

Add these lines to the new file. Adjust as needed for this site.

```
# Control when btrfs maintenance scripts run by deleting the corresponding
# 'lastrun' files at a predetermined time.  Caveat, this affects all of the
# scripts in the corresponding cron directories (/etc/cron.{weekly,monthy})

# Run weekly on Saturday at 2 AM as root
0 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
# Run monthly on the first Sunday of the month at 2 AM as root
0 2 * * 0 root [ $(date +%d) -le 07 ] && rm -f /var/spool/cron/lastrun/cron.monthly
```

**2.** Set ownership of the new cron file to root,root with permissions 644.

```
smw# chown root:root /etc/cron.d/cray_btrfs.cron
smw# chmod 644 /etc/cron.d/cray_btrfs.cron
```

## 3.7.4 Configure Direct-attached Lustre (DAL)

### Prerequisites

This procedure assumes the following:

- Service nodes to support direct-attached Lustre® (DAL) have been identified with `xtdiscover` as management server (MGS), metadata server (MDS), or object storage server (OSS) nodes.
- Configuration worksheets for Cray Linux environment (CLE) have been created and updated for DAL:
  - The `cray_lnet` worksheet is updated and the `cray_lnet.enabled` setting is uncommented and set to `true`. See *Update cray_lnet Worksheet* on page 134.
  - The `cray_lustre_client` worksheet is updated and the `cray_lustre_client.enabled` setting is uncommented and set to `true`. See *Update cray_lustre_client Worksheet* on page 140.
  - `cray_lustre_server` worksheet is updated and the `cray_lustre_server.enabled` setting is uncommented and set to `true`. See *Update cray_lustre_server Worksheet* on page 142

    **NOTE:** There are additional settings which tune the Lustre kernel modules.
  - If using the Lustre Monitoring Tool (LMT), a MySQL database, storage space for that database, Cerebro, and the LMT GUI must be configured on the MGS node. The cray_lmt worksheet inlcudes settings for configuring LMT.
- All DAL service nodes are assigned to the DAL group so that they are assigned the DAL boot image for booting.
- The `imgbuilder` configuration for DAL has the DAL stanza added to the to the default image group.

### About this task

This procedure configures direct-attached Lustre (DAL) nodes that provide a Lustre file system.

### Procedure

**Identify Logical Unit Numbers (LUNs) for DAL**

**1.** Identify the LUNs used for DAL.

Log in to the DAL service nodes to identify the persistent storage device names to be used for the Lustre file system. Identify all disk device names that will be used for the metadata target (MDT) / management target (MGT) and object storage target (OST) devices.

```
smw# ssh boot
boot# ssh dal-mds
```

**2.** If the LUN number is known, then use the `lsscsi` command to map the LUN to the short disk name.

This example shows that LUN 17 is `/dev/sdr`.

```
dal-mds# lsscsi | grep 17
[0:0:0:17]   disk    LSI      INF-01-00        0786  /dev/sdr
```

**3.** Use the short disk name from the previous step to determine the long persistent disk device name.

This example shows that `sdr` has two different persistent device names that could be used.

⚠ **CAUTION:** Use persistent device names in the Lustre file system definition. Non-persistent device names (for example, `/dev/sdc`) can change when the system reboots. If non-persistent names are specified in the *`fs_name`*`.fs_defs` file, then Lustre may try to mount the wrong devices and fail to start when the system reboots.

For more information about Lustre control utilities, see the `lustre_control(8)` and `lustre.fs_defs(5) man` pages.

```
dal-mds# ls -l /dev/disk/by-id | grep sdr
lrwxrwxrwx 1 root root  9 Aug  4 13:23 scsi-360080e500036ae3e000002e6524a8369 -
> ../../sdr
lrwxrwxrwx 1 root root  9 Aug  4 13:23 wwn-0x60080e500036ae3e000002e6524a8369 -
> ../../sdr
```

### Create and Install the Lustre fs_defs File

**4.** Prepare the Lustre `fs_defs` file on the system management workstation (SMW).

This file is used by `lustre_control` to format, reformat, start, and stop the file system. When creating the Lustre `fs_defs` file in this example, use `/dev/disk/by-id/scsi-0x60080e500036ae3e000002e6524a8369` for LUN 17. Refer to the *XC™ Series Lustre® Administration Guide* (S-2648) for detailed information about how to create an `fs_defs` file for a Lustre file system.

**5.** Create a variable called `FS_NAME` to be the name of the file system using 8 characters or less ("dal" in this example). The file name of the `fs_defs` file should be similar to the file system it defines.

```
smw# export FS_NAME=dal
smw# echo $FS_NAME
dal
```

**6.** Copy the `example.fs_defs` file to the one named after the DAL file system.

```
smw# cp -p /opt/cray-xt-lustre-utils/default/etc/example.fs_defs \
/home/crayadm/$FS_NAME.fs_defs
```

**7.** Edit the `$FS_NAME.fs_defs` file. This is a simple example for the p0 partition, which calls the file system "dal" and has the MGT on nid00027, MDT on nid00027 and nid00029, first OST on nid00028, and second OST on nid00031. Substitute site-specific values in this site's `fs_defs` file.

```
smw# vi /home/crayadm/$FS_NAME.fs_defs
```

**8.** Locate `fs_name: example` and change `example` to the name defined by `$FS_NAME` ("dal" in this example).

```
fs_name: dal
```

9. Set the Lustre server hosts to LNet NIDs mapping.

```
# Lustre server hosts to LNET NIDs mapping.
# Multiple lines are additive.
# Use multiple lines with the same nodes if you have more than one nid for each
# node.
# Nodes and nids can be specified using range expressions. See the
# lustre.fs_defs man page for more information on range expressions.
# Each line should have a one-to-one mapping between the nodes and nids.
nid_map: nodes=nid000[27-29,31] nids=[27-29,31]@gni
```

10. Update the `fs_defs` file with these settings (substituting appropriate site-specific values). Identify which nodes and devices are being used for MGT, MDT, and OSTs. There are other settings in the `fs_defs` file that can be changed, but they are probably acceptable for most sites.

```
## MGT
## Management Target
mgt: node=nid00027
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170000

## MDT
## MetaData Target(s)
mdt: node=nid00027
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170100
     index=0
mdt: node=nid00029
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170200
     index=1

## OST
## Object Storage Target(s)
ost: node=nid00028
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad79111170300
     index=0
ost: node=nid00031
     dev=/dev/disk/by-id/scsi-360001ff020021101061ad7a811170400
     index=1
```

11. Install the `fs_defs` file into the appropriate CLE config_set (p0 in the example).

```
smw# lustre_control install -c p0 /home/crayadm/$FS_NAME.fs_defs
```

The `lustre_control install` command copies the `fs_defs` file into a directory in the config set, makes a `lustre_control` readable version of it with the suffix `.config.data`, and updates the list of installed file systems.

12. Verify that the `fs_defs` file is installed in the config set by listing the files in the `lustre/.lctrl/` directory of the config set.

```
smw# ls /var/opt/cray/imps/config/sets/p0/lustre/.lctrl/
dal.config.data    dal.filesys.data                   dal.service.data
dal.failover.data  dal.fs_defs.20160421.1461256838  installed_filesystems
```

### Modify the Config Set to Load the `lustre-utils` Module

**13.** Modify `cray_user_settings.settings.default_modules.data.service` to add `lustre-utils`.

a. Update the cray_user_settings service in config set p0.

```
smw# cfgset update -s cray_user_settings -m interactive -l advanced p0
```

b. Select the default modules `service` setting (a list of autoloaded modules for non-login service nodes) to configure it.

Enter **2** and press **Enter** to select `service`, then enter **C** and press **Enter** to configure it.

```
Cray User Settings Menu [default: save & exit - Q] $ 2
...
Cray User Settings Menu [default: configure - C] $ C
```

c. Add the lustre-utils module to the list.

Enter **+** to add an entry, then enter "lustre-utils" and press **Enter**. Press **Ctrl-d** to finish adding entries, then press **Enter** to set the entries for this setting.

```
cray_user_settings.settings.default_modules.data.service
[<cr>=set 7 entries, +=add an entry, ?=help, @=less] $ +
Add service (Ctrl-d to exit) $ lustre-utils
Add service (Ctrl-d to exit) $ <Ctrl-d>
...
cray_user_settings.settings.default_modules.data.service
[<cr>=set 8 entries, +=add an entry, ?=help, @=less] $ <cr>
```

d. Save the changes and exit the configurator.

```
Cray User Settings Menu [default: save & exit - Q] $ Q
```

**14.** Validate the config set.

- Entire system:

```
smw# cfgset validate p0
```

- Partitioned system:

```
smw# cfgset validate p1
smw# cfgset validate p2
```

### Boot the System and Reformat the DAL File System

The DAL file system must be formatted using `lustre_control` from the boot node after initial set up, and before automating the start up and mounting of the DAL file system.

**15.** Boot the system.

- If CLE is not booted, proceed to step *16* on page 216.
- If CLE is booted, proceed to step *19* on page 217.

**16.** If CLE is not booted:

```
crayadm@smw> xtbootsys -a auto.hostname.start
```

**17.** Reformat the DAL file system after a full system boot.

```
smw# ssh boot
boot# export FS_NAME=dal
boot# lustre_control reformat -f $FS_NAME
```

**18.** Proceed to step *21* on page 217

**19.** If CLE is booted, run `cray-ansible`, then reboot only the DAL nodes.

Restarting `/etc/init.d/cray-ansible` refreshes the config set cache on the boot node. This example specifies a comma-separated list of `cnames` (for example `c0-0c0s0n0`) for all DAL nodes (MGS, MDS, and OSS) to create a $DALNODES variable.

```
boot# /etc/init.d/cray-ansible start
```

Note that the following commands are run as crayadm, not root.

```
crayadm@smw> export DALNODES=mgsnode,mdsnode,ossnode1,ossnode2
crayadm@smw> xtbounce -s $DALNODES
crayadm@smw> xtcli boot DEFAULT $DALNODES
```

**20.** Reformat the DAL file system after a reboot of only the DAL nodes.

```
smw# ssh boot
```

```
boot# module load lustre-utils
boot# export FS_NAME=dal
boot# lustre_control reformat -f $FS_NAME
Continue? (y|n|q) y
```

### Start and Mount the DAL File System

**21.** Start the DAL file system using `lustre_control` on the boot node.

```
boot# lustre_control start -p -f $FS_NAME
```

**22.** Verify that the Lustre targets are mounted on each DAL node.

```
boot# lustre_control status -f $FS_NAME
```

**23.** Test mount the DAL file system on a login node.

```
boot# ssh login
login# export FS_NAME=dal
login# mkdir -p /lus/$FS_NAME
```

```
login# mount -t lustre 27@gni:/$FS_NAME /lus/$FS_NAME
```

In the above mount command, substitute the site-specific value for *27@gni*, which is a combination of the nid of the MGT node and the LNet name by which the external Lustre server is accessed (will be something like gni or gni1). The MGT node nid was defined in the `fs_defs` file in step 10, and the LNet name can be found by searching for "gni" the CLE config set (p0 in this example) on the SMW.

```
smw# cfgset search -t gni -l advanced -s cray_lnet p0
# 2 matches for 'gni' from cray_lnet_config.yaml
#--------------------------------------------------------------------------------
```

```
-
cray_lnet.settings.local_lnet.data.lnet_name: gni4
cray_lnet.settings.flat_routes.data.o2ib.src_lnet: gni4
```

### Add DAL file system to cray_lustre_client Configuration

**24.** Add the DAL file system to `cray_lustre_client` configuration so that Lustre clients can mount the file system from the Lustre server.

Note that the `cray_lustre_client` service must be enabled in addition to setting information like the settings below (substitute appropriate site-specific values).

```
smw# cfgset update -s cray_lustre_client -l advanced -m interactive p0
```

In the `client_mounts` setting, add two new entries for the DAL file system. One will be for the compute nodes, which can mount the file system at boot time. The other will be for the login node(s). These cannot currently mount the file system at boot time since they are booted before the DAL file system is started. Follow the guidance for the `client_mounts` settings. Set the `mgs_lnet_nids` to the NID number of the MGS (and failover MGS if applicable) followed by `@gni`. Set `mount_at_boot` to `false` for the login node entry and set it to `true` for the compute node entry.

```
cray_lustre_client.settings.client_mounts.data.fs_name.dal_login: null
cray_lustre_client.settings.client_mounts.data.dal_login.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_login.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_login.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_login.mount_options:
rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_login.mount_at_boot: false
cray_lustre_client.settings.client_mounts.data.dal_login.client_groups:
- login_nodes

cray_lustre_client.settings.client_mounts.data.fs_name.dal_compute: null
cray_lustre_client.settings.client_mounts.data.dal_compute.lustre_fs_name: dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_point: /lus/dal
cray_lustre_client.settings.client_mounts.data.dal_compute.mgs_lnet_nids:
- 27@gni
- 29@gni
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_options:
rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.dal_compute.mount_at_boot: true
cray_lustre_client.settings.client_mounts.data.dal_compute.client_groups:
- compute_nodes
```

### Add DAL File System to cray_lustre_server Configuration

**25.** Add the DAL file system node groups to the cray_lustre_server service.

```
smw# cfgset update -s cray_lustre_server -l advanced -m interactive p0
```

```
cray_lustre_server.settings.lustre_servers.data.mgs_group: MGS_NODE_GROUP
cray_lustre_server.settings.lustre_servers.data.mds_groups:
- MDS_NODE_GROUP_1
- MDS_NODE_GROUP_2
cray_lustre_server.settings.lustre_servers.data.oss_groups:
- OSS_NODE_GROUP_1
- OSS_NODE_GROUP_2
```

### Configure LMT to Monitor DAL

**26.** (Optional) If using LMT to enable monitoring of DAL, see *LMT_Configuration_for_DAL.ditamap#C320691*.

### Enable Realm-Specific Internet Protocol (RSIP) on DAL Nodes

**27.** Enable RSIP on DAL nodes so they can communicate with an external LDAP or NIS server.

DAL nodes do not have external network connections, but require access to LDAP or NIS servers external to the system for uid/gid information associated with the Lustre file system.

a. Add DAL node groups to the list of `cray_rsip.settings.service.data.node_groups_as_client`.

```
smw# cfgset update -s cray_rsip -l advanced -m interactive p0
```

Add the DAL MDS node group(s).

```
cray_rsip.settings.service.data.node_groups_as_client:
- MDS_NODE_GROUP_1
- MDS_NODE_GROUP_2
```

**28.** Validate the config set.

- Entire system:

```
smw# cfgset validate p0
```

- Partitioned system:

```
smw# cfgset validate p1
smw# cfgset validate p2
```

### Update the Boot Automation File for DAL

**29.** Edit the site boot automation file (in `/opt/cray/hss/default/etc/`) so that the DAL file system is started during the CLE boot.

Because the config set modifications made in an earlier step set it up so that login and elogin nodes do not attempt to mount DAL at boot time, but the compute nodes do, add these DAL lines to the site boot automation file **after** the boot of the service nodes but **before** the boot of the compute nodes.

```
#Boot all the service nodes
lappend actions {crms_boot_all_serv}

# start Lustre server on DAL nodes & mount Lustre filesystem on login nodes
 lappend actions { crms_exec_on_bootnode "root" "lustre_control start -f dal" }
 lappend actions { crms_exec_on_bootnode "root" "lustre_control mount_clients -
f dal -w login[1-2]" }

#Boot specific compute nodes
#lappend actions [list crms_boot_loadfile DEFAULT compute "c0-0c0s7n0
c0-0c0s7n1" linux]

#Boot compute nodes
lappend actions {crms_boot_all_comp}
```

This uses a pdsh style list of nodes as an argument for the mount_clients command. For example, lustre_control will interpret login[1-8] as nodes login1 through login8. Replace dal in the command with the name of the DAL file system for this site.

With client_mounts.data.dal_compute.mount_at_boot set to true in the cray_lustre_clients service, the compute nodes automatically mount the DAL file system when they boot. This also ensures that they mount the DAL file system even when rebooted individually, outside the control of the auto boot file.

## 3.7.5 LMT Configuration for DAL

The Lustre® monitoring tool (LMT) for direct-attached Lustre (DAL) on Cray Linux environment (CLE 6.0) requires some manual configuration during the software installation process.

| | |
|---|---|
| **Configure Storage for the LMT Database** | At least 40GB of storage space must be made available to the MGS node. See *LMT Disk Usage* on page 223. |
| **Configure the LMT MySQL Database** | The IMPS configuration does not set up this database, so this must be configured manually for CLE 6.0 UP01 and later releases. See *Configure LMT MySQL Database for DAL* on page 220. |
| **Configure the LMT GUI (Optional)** | See *Configure the LMT GUI* on page 222. |

Use the configurator to configure the LMT for DAL on CLE 6.0. Guidance is provided for each LMT configuration setting in the cfgset utility.

The cray_lmt configurator template configures LMT settings for specific nodes when they are booted. The default system configuration value for the LMT service is disabled (false). Log in to the SMW as root and use the cfgset command to modify the cray_lmt configuration settings to configure LMT.

```
smw# cfgset update -s cray_lmt -m interactive CONFIG_SET
```

### 3.7.5.1 Configure LMT MySQL Database for DAL

## Prerequisites

A MySQL server instance must be configured on the management server (MGS) node. All commands described below should be executed on the MGS for the direct-attached Lustre (DAL) file system.

## About this task

A MySQL server instance on the management server (MGS) node stores real-time and historical Lustre monitoring tool (LMT) data. The configurator does not handle the initial setup of the LMT MySQL users and database. It must, therefore, be done manually. All commands described below should be executed on the MGS for the DAL file system.

## Procedure

1. Log on to the MGS as root.

   (Where *nidMGS* is the node ID (NID) of the MGS node.)

   ```
   boot# ssh nidMGS
   ```

**2.** Start the MySQL server daemon (if not already running).

```
mgs# /sbin/service mysqld start
```

**3.** Run the `mysql_secure_installation` script to improve MySQL server instance security.

This sets the password for the `root` MySQL user, disallows remote `root` access to the database, removes anonymous users, removes the test database, and reloads privileges. If this is the first time configuring LMT, create a symlink before running `mysql_secure_installation` to ensure that MySQL uses the correct socket.

a. Create a symbolic link.

```
mgs# ln -s /var/run/mysql/mysql.sock /var/lib/mysql/mysql.sock
```

b. Run `mysql_secure_installation` utility.

```
mgs# mysql_secure_installation
```

c. Respond to script prompts.

Prompts and recommended responses generated by the script.

```
Enter current password for root (enter for none): <Enter>

Set root password? [Y/n] Y
New password: Enter a secure password
Re-enter new password: Enter the secure password again

Remove anonymous users? [Y/n] Y

Disallow root login remotely? [Y/n] Y

Remove test database and access to it? [Y/n] Y

Reload privilege tables now? [Y/n] Y
```

**4.** Ensure `root` only access to the LMT user configuration file, `/usr/share/lmt/mkusers.sql`.

```
mgs# chmod 600 /usr/share/lmt/mkusers.sql
```

**5.** Edit the LMT user configuration file `/usr/share/lmt/mkusers.sql`.

This file is not used at run time by LMT or MySQL processes. This script creates the MySQL users on the persistent storage configured for the MySQL databases. After it is run through MySQL, it is no longer needed.

This file contains MySQL statements that create users named `lwatchclient` and `lwatchadmin`. It gives them privileges only on databases that start with `filesystem_`. Cray recommends making the following changes to `mkusers.sql`.

**Edit the GRANT Statement** Edit the GRANT statements to grant privileges on only `filesystem_`*`fsname`*`.*` where *`fsname`* is the name of the file system. This will only grant permissions on the database for the file system being monitored.

**Edit the Password** Edit the password for `lwatchadmin` by changing `mypass` to the desired password. Also add a password for the `lwatchclient` user.

```
CREATE USER 'lwatchclient'@'localhost' IDENTIFIED BY 'foo';
GRANT SELECT ON filesystem_scratch.* TO 'lwatchclient'@'localhost';

CREATE USER 'lwatchadmin'@'localhost' IDENTIFIED BY 'bar';
GRANT SELECT,INSERT,DELETE  ON filesystem_scratch.* TO 'lwatchadmin'@'localhost';
GRANT CREATE,DROP           ON filesystem_scratch.* TO 'lwatchadmin'@'localhost';

FLUSH PRIVILEGES;
```

6. Save the changes and execute the following command. (This prompts for the MySQL `root` user password, which was set when `mysql_secure_installation` was executed.)

   ```
   mgs# mysql -u root -p < /usr/share/lmt/mkusers.sql
   ```

7. Create the database for the file system to be monitored.

   (Where *`fsname`* is the name of the DAL file system.)

   ```
   mgs# lmtinit -a fsname
   ```

   LMT data will be inserted into the LMT MySQL database the next time the Cerebro service is restarted on the MGS.

8. Restart Cerebro.

   ```
   mgs# service cerebrod restart
   ```

9. Verify that LMT is adding data to the MySQL database.

   a. Initiate the LMT shell.

   ```
   mgs# lmtsh -f fsname
   ```

   b. List tables.

   ```
   fsname> t
   ```

   c. List tables again after several seconds to verify that `Row Count` is increasing.

### 3.7.5.2 Configure the LMT GUI

## About this task

The Lustre monitoring tool (LMT) graphical user interface (GUI) package is installed on login nodes. It contains a GUI called `lwatch` and a command-line tool for viewing live data called `lstat`. The configuration file `~/.lmtrc` must be set up prior to using either tool.

## Procedure

1. Login to the MGS node as `root`.

2. Edit the sample configuration file `/usr/share/doc/packages/lmt-gui/sample.lmtrc` to reflect the site specific LMT configuration—where *`db_name`* is set to the name of the MySQL database used by LMT, that is, filesystem_*`fsname`*.

   ```
   # LMT Configuration File - place in $HOME/.lmtrc

   filesys.1.name=<insert_fsname_here>
   filesys.1.mountname=<insert_/path/to/mountpoint_here>
   filesys.1.dbhost=<insert_db_host_ip_here>
   filesys.1.dbport=<insert_db_port_here>
   filesys.1.dbuser=<insert_db_client_username_here>
   # Leave dbauth blank if the given client has no password
   filesys.1.dbauth=<insert_db_client_password_here>
   filesys.1.dbname=<insert_db_name_here>
   ```

3. Save the updated `.lmtrc` as `~/.lmtrc`.

Here is an example for configuring access to the LMT database for the file system named *`scratch_1`*, which was set up so that the user *`lwatchclient`* has no password. In this example, access is being configured on the LMT server node, so the database is local. Thus, the `db_host` is *`localhost`*.

```
filesys.1.name=scratch_1
filesys.1.mountname=/lus/scratch_1
filesys.1.dbhost=localhost
filesys.1.dbport=3306
filesys.1.dbuser=lwatchclient
filesys.1.dbauth=
filesys.1.dbname=filesystem_scratch_1
```

After setting up `~/.lmtrc`, `lwatch` and `lstat` can be run on this node. To run the GUI from a remote node, the MySQL database must be configured to allow remote access for the read-only user, `lwatchclient`. See *Configure LMT MySQL for Remote Access* on page 223.

### 3.7.5.3 Configure LMT MySQL for Remote Access

In order to run the Lustre monitoring tool (LMT) graphical user interface (GUI) on a separate node from the LMT server, the MySQL server instance (running on the LMT server) must be configured to enable remote access for the LMT read-only user, `lwatchclient`. These MySQL statements can be added to `/usr/share/lmt/mkusers.sql` prior to executing the statements in that file. They can also be executed directly. In these examples, *`FSNAME`* is the name of the file system being monitored.

```
CREATE USER 'lwatchclient'@'%' IDENTIFIED BY 'foo';
GRANT SELECT ON filesystem_FSNAME.* TO 'lwatchclient'@'%';
```

To execute these statements directly, log on to the DAL MGS node, open a mysql shell as the root MySQL user, and run the statements as follows.

1. Connect to the database as `root`.

   ```
   mgs# mysql -u root -p
   Enter password:
   Welcome to the MySQL monitor.  Commands end with ; or \g.
   ...
   ```

2. Create `lwatchclient` user.

   ```
   mysql> CREATE USER 'lwatchclient'@'%';
   Query OK, 0 rows affected (0.00 sec)
   ...
   ```

3. Grant privileges to `lwatchclient` user.

```
mysql> GRANT SELECT ON filesystem_FSNAME.* TO 'lwatchclient'@'%';
Query OK, 0 rows affected (0.00 sec)
```

This enables the user named `lwatchclient` to connect from any hostname.

To allow connections from a certain IP address, replace the '%' with an IP address in single quotes.

```
CREATE USER 'lwatchclient'@'10.11.255.252' IDENTIFIED BY 'foo';
GRANT SELECT ON filesystem_FSNAME.* TO 'lwatchclient'@'10.11.255.252';
```

### 3.7.5.4 LMT Disk Usage

LMT requires at least 40GB persistent storage attached to the LMT server (i.e., the management server (MGS)) to store historical data. If the storage becomes full, data can be deleted from the database using MySQL delete statements.

## MySQL Tables

Five tables store general file system statistics. These tables are populated by lmt_agg.cron script.

*Table 13. General File System Tables*

| Table Name | On-Disk Growth Rate |
|---|---|
| FILESYSTEM_AGGREGATE_HOUR | 0.8 KB/hour |
| FILESYSTEM_AGGREGATE_DAY | 0.8 KB/day |
| FILESYSTEM_AGGREGATE_WEEK | 0.8 KB/week |
| FILESYSTEM_AGGREGATE_MONTH | 0.8 KB/month |
| FILESYSTEM_AGGREGATE_YEAR | 0.8 KB/year |

*Table 14. MDS Aggregate Tables and Growth Rates*

| Table Name | Approximate On-Disk Growth Rate |
|---|---|
| MDS_AGGREGATE_HOUR | 0.5 KB/hour/MDS |
| MDS_AGGREGATE_DAY | 0.5 KB/day/MDS |
| MDS_AGGREGATE_WEEK | 0.5 KB/week/MDS |
| MDS_AGGREGATE_MONTH | 0.5 KB/month/MDS |
| MDS_AGGREGATE_YEAR | 0.5 KB/year/MDS |

*Table 15. OST Aggregate Tables and Growth Rates*

| Table Name | On-Disk Growth Rate |
|---|---|
| OST_AGGREGATE_HOUR | 0.7 KB/hour/OST |
| OST_AGGREGATE_DAY | 0.7 KB/day/OST |
| OST_AGGREGATE_WEEK | 0.7 KB/week/OST |
| OST_AGGREGATE_MONTH | 0.7 KB/month/OST |
| OST_AGGREGATE_YEAR | 0.7 KB/year/OST |

## Calculate Expected Disk Usage for a File System

Use this formula to calculate the approximate rate of disk space usage for a file system. Disregard the AGGREGATE tables as they grow so much more slowly than the raw data tables.

```
(56 KB/hour/filesystem) * (# of filesystems) + (1000 KB/hour/MDS) * (# of MDSs)
    + (44 KB/hour/OSS) * (# of OSSs) + (70 KB/hour/OST) * (# of OSTs) = Total KB/hour
```

### Calculate the Disk Usage for a File System for 1 Year

In this example, LMT is monitoring one file system with one MDS, four object storage servers (OSS), and eight object storage targets (OST). The amount of disk space used by the LMT database to is expected to grow at this hourly rate.

```
56 KB/hour/filesystem * 1 filesystem + 1000 KB/hour/MDS * 1 MDS
    + 44 KB/hour/OSS * 4 OSSs + 70 KB/hour/OST * 8 OSTs = 1792 KB/hour
```

Which translates to this yearly rate.

```
1792 KB/hour * 24 hours/day * 365 days/year * 1 MB/1024KB
       * 1 GB/1024MB = 15 GB / year
```

## 3.7.6 Configure the Simple Event Correlator (SEC)

The Simple Event Correlator (SEC) is an SMW utility that parses every line being appended to system log files, watching for specific strings that represent the occurrence of significant system events. When a specified string is detected, SEC sends notification that this has happened, either by email, IRC, writing to a file, or some user-configurable combination of all three.

SEC is enabled by default, and by default is configured to generate email notifications to `crayadm`. The types of notifications generated and the recipients to whom notifications are sent are defined in the SEC configuration file, `/etc/opt/cray/cray_sec_actions_config`.

The System Management Workstation (SMW) release includes `sec-2.7.6` and an SEC support package, `cray-sec-8.0.0`. The SEC support package contains control scripts to manage the starting and stopping of SEC around a Cray mainframe boot session, in addition to other utilities and a rule set designed for Cray systems.

For configuration procedures, see *XC™ Series SEC Configuration Guide* (S-2542) for release CLE 6.0.UP03.

## 3.7.7 Enable System Environmental Data Collections (SEDC)

SEDC is a tool that collects and reports in real time the environmental data on all Cray systems. Data includes information from sensors located on significant hardware components at the cabinet and blade level, such as power supplies, processors, memory and fans. SEDC refers to these sensors as *scan IDs*. Examples of collected data include cabinet and blade/node temperatures, voltage, current, power, cooling system air pressure, humidity, and statuses. At the node level, data is collected only from the nodes that are powered on.

Use this command to enable SEDC data collection after a fresh install:

```
crayadm@smw> sedc_enable_default
```

For information about how SEDC data is stored in the power management database (PMDB) and querying the PMDB for that data, see *System Environmental Data Collections Guide* (S-2491) for release SMW 8.0.UP03.

## 3.7.8 Configure Netroot

This part of the installation and configuration process is optional unless this site has decided to use Netroot. See *Where to Place the Root File System—tmpfs versus Netroot* on page 186 for more information.

Netroot needs a matched pair of images for compute nodes and login nodes.

**compute nodes**

- initrd-compute-large: the NIMS boot image is set to this image
- compute-large: the NIMS kernel parameter "netroot" is set to this image, and this image is pushed to the boot node

**login nodes**

- initrd-login-large: the NIMS boot image is set to this image
- login-large: the NIMS kernel parameter "netroot" is set to this image, and this image is pushed to the boot node

The following procedures describe how to configure the Netroot images, push Netroot images to the boot node, and reboot nodes with Netroot.

### 3.7.8.1 Configure Netroot Images

## Prerequisites

This procedure assumes the following:

- Basic configuration is complete and the system has been booted.
- No Netroot images have been built yet. If that is not the case, and the Netroot specifications shown below are already in the "default" image group of `cray_image_groups.yaml`, then skip this procedure.

## About this task

This procedure assigns Netroot-specific NIMS groups (if needed), adds entries to the "default" image group (if they are not already there), and then builds the Netroot images. Going forward, with the Netroot image specifications added to the "default" image group, both compute and login Netroot images will be built every time `imgbuilder` is run.

## Procedure

**1.** Prepare Netroot-specific NIMS groups.

- If this site plans to use Netroot for ALL compute and login nodes, skip this step and proceed to step *3* on page 227. Netroot-specific NIMS groups are not needed because the NIMS login group will be used for all login nodes and the NIMS compute group will be used for all compute nodes.
- If this site plans to use Netroot on only a subset of compute and login nodes instead of all of them, then continue with this step.

a. Create and assign Netroot-specific NIMS groups.

In the following example, the new NIMS groups are called *`login_netroot`* and *`compute_netroot`*, and each subset of nodes (*`SUBSET_LOGIN_NODES`* and *`SUBSET_COMPUTE_NODES`*) is a space-separated list of nodes.

```
smw# cnode update -G login -g login_netroot SUBSET_LOGIN_NODES
smw# cnode update -G compute -g compute_netroot SUBSET_COMPUTE_NODES
```

b. Confirm that the intended nodes were added to the NIMS Netroot groups.

```
smw# cnode list --filter group=login_netroot
```

```
smw# cnode list --filter group=compute_netroot
```

2. Prepare Netroot compute and login image stanza for use by a subset of compute and login nodes.

   - If this site plans to use Netroot for ALL compute and login nodes, skip this step and proceed to step *3* on page 227.
   - If this site plans to use Netroot on only a subset of compute and login nodes instead of all of them, then continue with this step.

   Add these two Netroot image specifications to the default image group, if they are not already there. The safest way is to find these two image specifications elsewhere in the file, then copy and paste them from there to the default image group and substitute the correct NIMS group names, as shown in this example.

```
cray_image_groups:
  default:
      ...
      - recipe: "initrd-compute-large_cle_6.0.up03_sles_12_x86-64_ari"
        dest: "initrd-compute-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "compute_netroot"
      - recipe: "initrd-login-large_cle_6.0.up03_sles_12_x86-64_ari"
        dest: "login-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "login_netroot"
      ...
```

   Each of these Netroot image recipes builds two image roots and only one boot image (the .cpio file). For example, the first builds an initrd-compute-large image root, a compute-large image root, and an initrd-compute-large boot image.

   > **NOTE:** The value for 'dest' in the login Netroot image specification begins with "login-large" but it should begin with "initrd-login-large" to be similar to the value of 'dest' in the compute Netroot stanza. The omission of 'initrd-' does not affect the behavior of `imgbuilder`: the correct image roots and boot image are created.

3. Prepare Netroot compute and login image stanza for use by all compute and login nodes.

   - If this site plans to use Netroot on only a subset of compute and login nodes instead of all of them, then skip this step and proceed to step *4* on page 228.
   - If this site plans to use Netroot for ALL compute and login nodes, then continue with this step.

   a. Add these two Netroot image specifications to the default image group, if they are not already there.

      The safest way to do this is to find these two image specifications elsewhere in the file, then copy and paste them from there to the default image group.

```
cray_image_groups:
  default:
      ...
      - recipe: "initrd-compute-large_cle_6.0.up03_sles_12_x86-64_ari"
        dest: "initrd-compute-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "compute"
      - recipe: "initrd-login-large_cle_6.0.up03_sles_12_x86-64_ari"
        dest: "login-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "login"
      ...
```

      Each of these Netroot image recipes builds two image roots and only one boot image (the .cpio file). For example, the first builds an initrd-compute-large image root, a compute-large image root, and an initrd-compute-large boot image.

**NOTE:** The value for 'dest' in the login Netroot image specification begins with "login-large" but it should begin with "initrd-login-large" to be similar to the value of 'dest' in the compute Netroot stanza. The omission of 'initrd-' does not affect the behavior of `imgbuilder`: the correct image roots and boot image are created.

b. Comment out image specifications with redundant NIMS group assignments.

Because this site is using Netroot for all compute and login nodes, comment out any other image specifications in the "default" image group that have these NIMS group assignments: `nims_group: "compute"` or `nims_group: "login"`. This will avoid building unnecessary image roots.

**4.** Create new images using the default image group.

**full system** For a full, unpartitioned system:

```
smw# imgbuilder --map
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# imgbuilder --map -p p1
```

```
smw# imgbuilder --map -p p2
```

At the end of the output from `imgbuilder`, there will be a command hint for how to push the resulting Netroot images to the boot node. Note the `image push` command with the specific image name that needs to be pushed to the boot node.

```
IMPORTANT: The netroot image for initrd-compute-large_cle_6.0.UP03-
build201508120201_sles_12-created20150813.cpio
must be pushed to the boot node:
smw:# image push -d boot compute-large_cle_6.0.UP03-build201508120201_sles_12-
created20150813

IMPORTANT: The netroot image for initrd-login-large_cle_6.0.UP03-
build201508120201_sles_12-created20150813.cpio
must be pushed to the boot node:
smw:# image push -d boot login-large_cle_6.0.UP03-build201508120201_sles_12-
created20150813
```

### 3.7.8.2 Push Netroot Images to Boot Node

## Prerequisites

This procedure assumes the following:

- The boot node is booted.
- Netroot images have been built using `imgbuilder`, and the output of that command provided the specific image name that needs to be pushed to the boot node.

## Procedure

**1.** Check for existing Netroot image roots for both compute-large and login-large.

```
smw# image list | grep "^compute-large"
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image list | grep "^login-large"
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

**2.** (Optional) If both image names have a common string, set an environment variable for it.

Using the output generated by one of the image list commands in step 1, set an environment variable to represent the common string appearing in both image names. This example assumes that the common string is everything that follows "-large." If this is not the case (for example, if the date-time stamp is different), creating an environment variable may not be worthwhile.

```
smw# export BASEIMAGE=cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

**3.** Push the Netroot images to the boot node.

Note that these commands may take 10 minutes or more to complete.

**full system** For a full, unpartitioned system:

```
smw# image push -d boot \
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image push -d boot \
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

If an environment variable was defined that applies to both image names, use these commands instead:

```
smw# image push -d boot compute-large_$BASEIMAGE
smw# image push -d boot login-large_$BASEIMAGE
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# image push -d boot-p1 \
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image push -d boot-p1 \
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

```
smw# image push -d boot-p2 \
compute-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222

smw# image push -d boot-p2 \
login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

If an environment variable was defined that applies to both image names, use these commands instead:

```
smw# image push -d boot-p1 compute-large_$BASEIMAGE
smw# image push -d boot-p1 login-large_$BASEIMAGE
```

```
smw# image push -d boot-p2 compute-large_$BASEIMAGE
smw# image push -d boot-p2 login-large_$BASEIMAGE
```

**Trouble?** If passwordless `ssh` has not been prepared between root@smw and root@boot, then the system will prompt for the password for root@boot twice.

**4.** Push custom Netroot image roots to boot node.

If any custom image roots were created with Netroot content or something that will be used by a profile in cray_image_binding, push that image root to the boot node. Installed workload managers (WLM) will have such custom images if login Netroot is included in the WLM recipe for the login nodes. For example, if a WLM ('wlm') is installed, there will be wlm-login and wlm-admin or wlm-service image roots created. However, only if wlm-login-large was created as a Netroot image root will it need to be pushed to the boot node.

a. Check for existing custom Netroot image roots.

This example shows checking for 'wlm' image roots. Substitute the name for the particular WLM used in this system.

```
smw# image list | grep wlm
wlm-login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

b. Push custom Netroot image roots to the boot node, if any were found.

This example uses the output of the previous substep as the image name. Substitute the image name(s) displayed in the output of the `image list` command for this system. If this image name contains the same base string as the images from step 1, and an environment variable was defined, that can be substituted for everything after 'wlm-login-large' in this push command.

```
smw# image push -d boot \
wlm-login-large_cle_6.0.UP03-build6.0.3074_sles_12-created20170222
```

The nodes that will use the Netroot images can be warm booted, or the entire system can be rebooted.

### 3.7.8.3 Reboot Nodes with Netroot

## Prerequisites

This procedure assumes that Netroot images have been pushed out to the boot node.

## About this task

This procedure reboots nodes with the new Netroot images, either by shutting down the entire system and rebooting it or by warm booting only the nodes that need the new Netroot images.

When a node is booted using a Netroot image, during the early stages of the boot, cray-ansible runs only Ansible plays of type *netroot_setup*, and it logs to these three files in `/var/opt/cray/log/ansible`.

`sitelog-init-netroot_setup` (has Ansible play output)

`file-changelog-init-netroot_setup` (shows each file changed by an Ansible play)

`file-changelog-init-netroot_setup.yaml` (YAML version of the previous log file)

## Procedure

**1.** Reboot entire system with new Netroot images.

In these examples, replace `auto.`*`hostname`*`.stop` and `auto.`*`hostname`*`.start` with boot automation files used at this site.

**full system** For a full, unpartitioned system:

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.hostname.stop
crayadm@smw> xtbootsys -a auto.hostname.start
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# su - crayadm

crayadm@smw> xtbootsys -s last -p p1 -a auto.hostname.stop
crayadm@smw> xtbootsys -p p1 -a auto.hostname.start.p1
```

```
crayadm@smw> xtbootsys -s last -p p2 -a auto.hostname.stop
crayadm@smw> xtbootsys -p p2 -a auto.hostname.start.p2
```

**2.** (Alternative) Warm boot only nodes needing new Netroot images.

a. Warm boot all login nodes.

Provide the same list of cnames for the login nodes to both `xtnmi` and `xtbootsys --reboot`.

**full system** For a full, unpartitioned system:

```
smw# su - crayadm

crayadm@smw> export LOGINNODES=c0-0c0s7n3,c0-0c0s8n0
crayadm@smw> xtnmi $LOGINNODES
crayadm@smw> sleep 60

crayadm@smw> xtbootsys --reboot -r "warmboot for login netroot"
$LOGINNODES
```

**partitioned** For a partitioned system, with partitions p1 and p2 (example for p1):

```
smw# su - crayadm

crayadm@smw> export
LOGINNODES=c0-0c0s9n0,c0-0c0s10n0,c0-0c0s11n0,c0-0c0s13n0
crayadm@smw> xtnmi -p p1 $LOGINNODES
crayadm@smw> sleep 60

crayadm@smw> xtbootsys --reboot -r "warmboot for login netroot" -p
p1 $LOGINNODES
```

b. Warm boot all compute nodes.

**full system** For a full, unpartitioned system (example for p0 with cabinet c0-0):

```
smw# su - crayadm

crayadm@smw> export COMPUTENODES=$(xtcli status p0 | \
egrep -v "empty|service|disabled" | grep c0-0 | \
awk '{ FS=":"; print $1 }' | tr ':' ' ' | \
awk '{ printf "%s,", $1 }' | sed s'/.$//')
crayadm@smw> echo -e "Compute nodes are:\n$COMPUTENODES"

crayadm@smw> xtcli status  $COMPUTENODES
crayadm@smw> xtcli shutdown $COMPUTENODES
crayadm@smw> xtcli status $COMPUTENODES
crayadm@smw> xtnmi $COMPUTENODES
crayadm@smw> sleep 60
crayadm@smw> xtcli status $COMPUTENODES

crayadm@smw> xtbootsys --reboot -r \
"warmboot for compute netroot" $COMPUTENODES

crayadm@smw> xtcli status $COMPUTENODES
```

**partitioned** For a partitioned system, with partitions p1 and p2 (example for p1 with cabinet c0-0):

```
smw# su - crayadm

crayadm@smw> export COMPUTENODES=$(xtcli status p1 | \
egrep -v "empty|service|disabled" | grep c0-0 | \
awk '{ FS=":"; print $1 }' | tr ':' ' ' | \
awk '{ printf "%s,", $1 }' | sed s'/.$//') \
echo -e "Compute nodes are:\n$COMPUTENODES"

crayadm@smw> xtcli status  $COMPUTENODES
crayadm@smw> xtcli shutdown $COMPUTENODES
crayadm@smw> xtcli status $COMPUTENODES
crayadm@smw> xtnmi -p p1 $COMPUTENODES
crayadm@smw> sleep 60
crayadm@smw> xtcli status $COMPUTENODES

crayadm@smw> xtbootsys --reboot -p p1 -r \
"warmboot for compute netroot" $COMPUTENODES

crayadm@smw> xtcli status $COMPUTENODES
```

### 3.7.9 Prevent Unintentional Re-creation of Mail Configuration Files

This procedure is optional. It applies to systems where postfix or sendmail are configured on the SMW.

To prevent the `master.cf` and `main.cf` postfix configuration files from being re-created during software updates or fixes, edit the `/etc/sysconfig/mail` file on the SMW and ensure that the `MAIL_CREATE_CONFIG` setting is set to "no."

```
smw# vi /etc/sysconfig/mail

MAIL_CREATE_CONFIG="no"
```

## 3.8 Install Additional Software

This is the final stage in the installation/configuration process. These procedures provide additional functionality for a Cray XC Series system. The features are optional, but these procedures are required for any feature in use at this site.

| | |
|---|---|
| optional | *Install the Dell Systems Management Tools and Documentation DVD* on page 233 |
| optional | *Install and Configure DataWarp* on page 234 |
| optional | *Install Cray Programming Environment (PE) Software* on page 234 (PE) |
| optional | *Install and Configure a Workload Manager (WLM)* on page 240 |
| optional | *Install and Configure CMC/eLogin* on page 241 |

### 3.8.1 Install the Dell Systems Management Tools and Documentation DVD

### About this task

This procedure installs the OpenManage Server Administrator (OMSA) software from the Dell Systems Management Tools and Documentation DVD, which is shipped with the SMW. This software enables advanced control over the Integrated Dell Remote Access Controller (iDRAC) and provides features such as Automatic Recovery (automatic system boot after a power event).

Visit the Dell OpenManage Linux Repository to view the Dell OpenManage Server Administrator documentation: *http://linux.dell.com/wiki/index.php/Repository/OMSA*

### Procedure

1. Obtain the Dell System Management Tools and Documentation DVD.
2. Log on to the SMW as `root`.
3. Mount the DVD.

   ```
   smw# mount /dev/cdrom /media/cdrom
   ```

4. Go to the location of the installation scripts.

   ```
   smw# cd /media/cdrom/SYSMGMT/srvadmin/linux/supportscripts
   ```

5. Execute the script to install the software.

   ```
   smw# sh srvadmin-install.sh --express
   ```

6. Start the Server Administrator services.

   ```
   smw# sh srvadmin-services.sh start
   ```

7. Double-click the icon named **Launch Server Administrator** on the SMW screen.
8. Enter the SMW user name `root`.

9. Enter the SMW `root` account password.

The system can now be managed for Properties, Shutdown, Logs, Alert Management, and Session Management.

## 3.8.2 Install and Configure DataWarp

Cray DataWarp provides an intermediate layer of high bandwidth, file-based storage to applications running on compute nodes. It is comprised of commercial SSD hardware and software, Linux community software, and Cray system hardware and software. DataWarp storage is located on server nodes connected to the Cray system's high speed network (HSN). I/O operations to this storage completes faster than I/O to the attached parallel file system (PFS), allowing the application to resume computation more quickly and resulting in improved application performance. DataWarp storage is transparently available to applications via standard POSIX I/O operations and can be configured in multiple ways for different purposes. DataWarp capacity and bandwidth are dynamically allocated to jobs on request and can be scaled up by adding DataWarp server nodes to the system.

For installation and configuration procedures, see *XC™ Series DataWarp™ Installation and Administration Guide* (S-2564) for this release.

## 3.8.3 Install Cray Programming Environment (PE) Software

### About this task

The Cray Developers Toolkit (CDT) for Cray XC Series systems is a package that consists of the basic libraries and components needed to develop and compile code on Cray systems, including the GNU Fortran, C, and C++ compilers. The CDT also includes the Cray Compiling Environment (CCE), but a valid license key is required before CCE can be installed. All other compilers are sold, installed, and licensed separately.

This procedure installs and configures the Cray Programming Environment (PE) software to make its content available on Cray XC Series compute nodes. A typical PE installation takes about 30 minutes.

### Procedure

1. Create the PE image root.

   Use a PE image for several of the monthly releases of PE software, and use a fresh image with each new CLE release.

   a. Set an environment variable for the PE image name.

   ```
   smw# export PEIMAGE=pe_compute_cle_6.0up03_sles_12
   smw# echo $PEIMAGE
   ```

   If this site wishes to use a different name for the PE image when setting the `$PEIMAGE` environment variable, update the name in the PE profile of the cray_image_binding service for the CLE configuration set to match (a later step in this procedure).

   Note that although the PE image name has 'compute' in it, the same image is also used for login nodes.

   Check for an existing pe_compute image.

   ```
   smw# image list | egrep "^[ ]*$PEIMAGE"
   ```

b. Create PE image on the SMW.

  **1.** Get the name of the PE image recipe on the system.

```
smw# recipe list | grep ^pe
pe_image_cle_6.0up03_sles_12
```

  **2.** Create the PE image ($PEIMAGE) using the recipe name discovered by the command in the previous step.

```
smw# image create -r pe_image_cle_6.0up03_sles_12 $PEIMAGE
```

**2.** Install the compiler license RPMs.

The Cray Compiling Environment (CCE), Intel, and PGI compilers all require licenses. These licenses must be installed at this point before installing any of the PE software. For instructions, see

**CCE** *Cray Compiling Environment Release Overview and Installation Guide*, available at *http://pubs.cray.com*

**Intel compilers** *http://software.intel.com/en-us/articles/intel-software-technical-documentation*

**PGI compilers** *http://www.pgroup.com*

**3.** Copy the most recent PE ISO to the SMW and mount the ISO.

```
smw# mkdir -p /var/adm/cray/release/pe
smw# cd /var/adm/cray/release/pe
smw# mkdir -p /var/adm/cray/release/pe/mount_iso
smw# mount -o loop,ro <downloaded PE ISO> /var/adm/cray/release/pe/mount_iso
```

**4.** Install the `craype-installer` rpm from the PE ISO on the SMW.

```
smw# rpm -ivh /var/adm/cray/release/pe/mount_iso/installer/\
craype-installer-*.x86_64.rpm
```

**5.** Configure the installer configuration file.

a. Copy the install configuration file from the `craype-installer` installation directory.

```
smw# cp -p /opt/cray/craype-installer/default/conf/install-cdt.yaml .
```

b. Create logs directory that will be used by the installer.

```
smw# mkdir ./logs
```

c. Update the configuration file, `install-cdt.yaml`.

When `install-cdt.yaml` is opened, there are comment blocks before every keyword listed below describing the valid values for each.

  **1.** For `IMAGE_DIRECTORIES` specify the directory (or directories) for the installer to install into. This example uses an `image_root` of `pe_compute_cle_6.0up03_sles_12`. This parameter must have data on the next line. The data must have four space characters and then a dash character and then a space character and the path to the directory.

2. Specify **`YES`** in each of the `INSTALL_*_LIBRARIES` for the compiler specific PE libraries to be installed. The Pathscale compiler is no longer supported by PE.
3. If the system includes an `ACCELERATOR`, change **`NONE`** to a comma separated list of one or more of the supported accelerators - **`FERMI or KEPLER`**. See the comments in `install-cdt.yaml` for examples and more information.
4. If the system has more than one type of processor installed, then specify the lowest common denominator for the processor for `CRAY_CPU_TARGET`.

Because this file supports older releases as well, some of the items are not applicable for this release. Those that are applicable for this release are shown in bold.

```
smw# vi install-cdt.yaml
```

```
---
HAS_MAMU_NODES : NO
ACCELERATORS                        : NONE
NETWORK_TYPE                        : NONE
CRAY_CPU_TARGET                     : sandybridge
BOOTNODE_HOSTNAME                   : NONE
BOOTNODE_ROOT_DIRS                  :
    - /rr/current
ESMS_HOSTNAME                       : NONE
ESMS_IMAGE_DIRS                     :
    - /cm/images/<your image name>
UNMANAGED_ESLOGINS  : NONE
IMAGE_DIRECTORIES :
    - /var/opt/cray/imps/image_roots/pe_compute_cle_6.0up03_sles_12
LOGS_DIR                            : ./logs
ISO_MOUNT_DIR                       : ./mount_iso
INSTALL_CCE_LIBRARIES               : YES
INSTALL_GNU_LIBRARIES               : YES
INSTALL_INTEL_LIBRARIES             : YES
INSTALL_PATHSCALE_LIBRARIES         : NO
INSTALL_PGI_LIBRARIES               : YES
```

6. Install PE software from the most recent PE installation media and installer.

   a. Link `/opt/cray/pe/bin` to `/opt/cray`.

   ```
   smw# chroot /var/opt/cray/imps/image_roots/$PEIMAGE ln \
   -s /opt/cray/pe/bin /opt/cray/bin
   ```

   b. Run the PE installer.

   This step can take about 30 minutes.

   ```
   smw# module load craype-installer
   smw# craype-installer.pl --install --install-yaml-path ./install-cdt.yaml
   ```

   When the installation completes, output such as the following will be displayed, summarizing the installed packages.

```
1) atp-1.7.5-0_3605.x86_64      (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
 2) cray-ccdb-1.0.3-0_3575.x86_64      (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
 3) cray-dwarf-14.2.0-0.x86_64        (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
 <snip>
71) perftools-clients-6.2.2-1.x86_64  (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
```

c. Set the default versions for PE (if the install succeeds) by running `set_default` scripts.

```
smw# craype-installer.pl --set-default --install-yaml-path ./install-cdt.yaml
```

Note that at the monthly PE update, this step would not be done until after the image is pushed to the boot node and tested.

d. Unmount the ISO.

```
smw# umount ./mount_iso
```

e. Clean up the PE ISO and PE RPMs.

These can be removed since they are large and use up disk space.

```
smw# rm *.iso *.rpm *.tar.gz
```

**7.** Push the PE image root to the boot node.

This step can take about 10 minutes.

For p0:

```
smw# image push -d boot $PEIMAGE
INFO - Remotely cloning Image '<name of image>' to 'boot'...
INFO - Checking remote destination...
INFO - Transferring Image '<name of image>' to 'root@boot:/var/opt/cray/imps/
image_roots/<name of image>'...
INFO - Cloned Image '<name of image>' to remote host 'root@boot:/var/opt/cray/
imps/image_roots/<name of image>'.
```

For partitioned systems, push to the boot node for that partition, `boot-p1`:

```
smw# image push -d boot-p1 $PEIMAGE
```

**8.** Enable PE.

For a fresh install, configure and enable the PE bind mount profile in the Cray Image Binding service, and then validate the config set.

a. Update cray_image_binding, which is in the CLE config set.

Full system:

```
smw# cfgset update -s cray_image_binding -m interactive p0
```

Partitioned system:

```
smw# cfgset update -s cray_image_binding -m interactive pN
```

b. Select the `profiles` setting to configure it.

Enter **1** and press **Enter** to select the `profiles` setting, then enter **C** and press **Enter** to configure it.

c. Change the value of the 'PE' profile image field to match the name of the image used in earlier steps (`$PEIMAGE`).

In this example, the number of the 'PE' profile is 1, so enter **1a*** to edit the PE `image` setting.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 1a*
```

Enter the name of the PE image.

```
cray_image_binding.settings.profiles.data.PE.image
[<cr>=keep 'pe_compute_cle_6.0up03_sles_12', <new value>, ?=help, @=less] $ 
PE_image_name
```

d. Ensure that the 'PE' profile callbacks field is set.

In this example, the number of the 'PE' profile is 1, so enter **1c*** to edit the PE `callbacks` setting.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 1c*
```

Press **Enter** (`<cr>`) to set the existing entry, enter **1*** to edit that entry, or **+** to add a new entry. Note that the default callbacks entry (`opt/cray/pe/bin/pe_postmount_callback.sh`) is a relative path. If a new path is entered, it also must be a relative path and should not start with a forward slash (`/`) character.

```
cray_image_binding.settings.profiles.data.PE.callbacks
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

e. Enable the PE profile.

Has the PE image root been pushed to the boot node? If not, do step 7 first, and then return to this step.

In this example, the number of the 'PE' profile is 1, so enter **1d*** to edit the PE profile `enabled` setting.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 1d*
```

```
cray_image_binding.settings.profiles.data.PE.enabled
[<cr>=keep 'false', <new value>, ?=help, @=less] $ true
```

f. Set the profile entries, and then save changes and exit the configurator.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
...
```

```
IMPS Image Binding Service Menu [default: save & exit - Q] $ Q
```

g. Validate the config set.

   Full system:

```
smw# cfgset validate p0
```

   Partitioned system:

```
smw# cfgset validate pN
```

**9.** Reboot the system with PE.

A reboot is done only for an initial installation because of the initial setup. It is not required for the monthly PE updates.

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.hostname.stop
crayadm@smw> xtbootsys -a auto.hostname.start
```

**10.** Build a sample MPI job that leverages the PE product by compiling and executing the application.

a. Test basic `aprun` functionality.

   **1.** Log in to the login node.

```
crayadm@smw> ssh boot
crayadm@boot> ssh login
```

   **2.** Run `apstat` to get the number of nodes to use for the following commands:

```
crayadm@login> NUMNODES=$(($(apstat -v | grep XT | awk "{print \$3}")); \
echo NUMNODES is $NUMNODES
```

```
crayadm@login> aprun -n $NUMNODES -N2 python -c "print 'hello world.'"
```

b. Compile a sample MPI program.

   **1.** Unload the default `PrgEnv` module.

```
crayadm@login> module unload PrgEnv-cray
```

   **2.** Load modules.

```
crayadm@login> module load PrgEnv-gnu cray-mpich
crayadm@login> cd /tmp
crayadm@login> export CRAY_CPU_TARGET=x86-64
```

   **3.** Obtain sample MPI code for compile.

   **4.** Compile sample MPI code.

   **5.** Execute sample MPI code.

c. Log out of the login node and boot node and su session to return to being root on the SMW.

```
crayadm@login> exit
crayadm@boot> exit
crayadm@smw> exit
smw#
```

**11.** Make a snapshot post PE installation.

Cray recommends saving a snapshot of the system immediately after the PE software installation is complete. If any root users make bad changes after the software install is complete, revert to this snapshot to avoid a redo of the entire software install.

```
smw# snaputil list

smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

```
smw# snaputil create ${SNAPSHOT}.postpe
```

**12.** Back up the CLE and global config sets post PE installation.

```
smw# cfgset create --clone global global-postpe-${TODAY}
```

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postpe-${TODAY}
```

### 3.8.4 Install and Configure a Workload Manager (WLM)

Cray XC Series systems support the use of workload manager (WLM) software products. The SMW 8.0.UP03 release supports these three WLM products: PBS, Moab/TORQUE, and Slurm. Each product requires installation and configuration prior to use.

**PBS Professional™**

PBS Professional is a commercial product licensed by Altair Engineering, Inc.

- For general product information: *http://www.altair.com*
- For PBS Professional documentation: *http://www.pbsworks.com/PBSProductGT.aspx?n=PBS-Professional&c=Overview-and-Capabilities&d=PBS-Professional,-Documentation*
- Note that PBS Professional uses a license manager, which requires a network connection between the license server and the SDB node on a Cray system.

**Moab™ and TORQUE**

Moab and TORQUE are commercial products licensed by Adaptive Computing.

- For product information: *http://www.adaptivecomputing.com*
- For a CLE 5.2 to CLE 6.0 migration,

**Slurm**

Slurm (Simple Linux Utility for Resource Management) is an open source application that is commercially supported by SchedMD, among others.

- For more product information: *http://www.schedmd.com/*

- For Cray-specific installation/configuration instructions: *XC™ Series Slurm Installation Guide* (S-2538)

For the most up-to-date information regarding workload manager software compatibility with CLE releases, look on the CrayPort website at *http://crayport.cray.com*.

### 3.8.5 Install and Configure CMC/eLogin

External services formerly provided by esLogin are now supported by eLogin. eLogin uses Cray System Management Software (CSMS) installed on a separate Cray Management Controller (CMC) to deploy eLogin images to external Cray Development and Login (CDL) nodes. The CMC and the CDL nodes are each deployed to Dell 720s.

The CMC connects to the SMW, which provides shared image and configuration services. A recipe for eLogin nodes exists on the SMW so that an image root for eLogin can be created and packaged into the proper boot image format thus enabling the SMW to deliver it to the eLogin node.

> **NOTE:** In previous software versions, the CMC was called the CIMS or esMS node, and the eLogin nodes were called the CDL or esLogin nodes.

For installation and upgrade instructions, see *XC™ Series eLogin Installation Guide* (S-2566) for release CLE 6.0.UP03.

## 3.9 Back Up the Newly Installed and Configured SMW/CLE Software

After installing and configuring the new SMW/CLE software, create a backup of it, if needed.

**Dell R815 SMW with software RAID** If an R815 SMW was configured using the recommended software RAID1 configuration for the boot disk, there is no need to make a backup of the SMW boot disk because the boot disks are mirrored automatically by the RAID software.

**Dell R630 SMW with software RAID** If an R630 SMW was configured using the recommended hardware RAID5 virtual disk, there is no need to make a backup of the SMW boot disk. Any one of the four disks in the RAID5 configuration can fail without losing any data.

# 4 Update SMW/CLE Software

Cray provides periodic updates and upgrades to each SMW and CLE release. In an update release, only the minor version numbers (following *UP*) change, for example, from CLE 6.0.UP01 to CLE 6.0.UP02. In an upgrade release, the major and possibly the minor version numbers change, for example, from SMW 8.0.UP01 to SMW 8.1.UP00.

Follow the procedures in this chapter to update to CLE 6.0.UP03 / SMW 8.0.UP03. The procedures provided here do not change the base operating system version running on the SMW.

**update path** To use these procedures, this system must be running CLE 6.0.UP01 / SMW 8.0.UP01 or CLE 6.0.UP02 / SMW 8.0.UP02 software, and the SMW must be running the initial release of SUSE Linux Enterprise Server (SLES) version 12.

If this system is running CLE 5.2.UP04 / SMW 7.2.UP04, a migration to to CLE 6.0.UP03 / SMW 8.0.UP03 is required. See *XC™ Series CLE 5.2 to CLE 6.0 Software Migration Overview (CLE 6.0.UP03) S-2574*.

The installers for CLE 6.0 / SMW 8.0 are a rewrite from the previous generation, and they use some newer technology to make the update installation process faster and more flexible and to minimize system downtime. These improvements include using a btrfs file system for staging upgrades, zypper repositories for managing packages, and a flexible installer task processor.

**SMW HA only:** For a system that has been configured for SMW high availability (HA), the active SMW must be updated first and then powered down to fail over to the passive SMW, which then becomes the active SMW and can be updated. Do not use this guide for updating an SMW HA system. Instead, use *XC™ Series SMW HA Installation Guide (SLEHA12.SP0.UP03) S-0044*.

## 4.1 Prepare for an SMW/CLE Software Update

To prepare for an update of the SMW/CLE software, do the following:

- Read the *SMW Release Errata* and *SMW README* provided with the SMW release package for any additional installation-related requirements, corrections to this installation guide, and other relevant information about the release package.
- Read the *CLE Release Errata* and the *CLE README* provided with the CLE release package for any additional update-related requirements, corrections to this installation guide, and other relevant information about the release package.
- Read the Field Notices (FN) related to kernel security fixes to identify any changes to this release package. Apply any needed changes before installing the new software.
- If local changes have been made to any automation files, such as `/opt/cray/hss/default/etc/auto.xtshutdown`, back them up before beginning the SMW/CLE update.

- If using the Cray simple event correlator (SEC) and the `/opt/cray/default/SEC_VARIABLES` file has local changes, make a backup copy of this file before beginning the SMW/CLE update. For more information, see *XC™ Series SEC Configuration Guide* (S-2542).

When those preparation activities are done, use these procedures prior to beginning the software update.

1. *Start a Typescript File*
2. *Show Current HSS Partition Configuration*
3. *Set Variable for Release Snapshot Name*
4. *Make a Pre-update Release Snapshot using snaputil*
5. *Make a Pre-update Backup of Current Global and CLE Config Sets*
6. *Prepare to Migrate Node Groups Configuration Data*
7. *Collect Software Media*

## 4.1.1 Start a Typescript File

### About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before installing a new software release
- just before configuring the newly installed software

### Procedure

1. Log in as root to the SMW.

2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

For *`suffix`*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

## 4.1.2 Show Current HSS Partition Configuration

### Procedure

Check the current HSS partition configuration.

```
smw# xtcli part_cfg show
```

## 4.1.3 Set Variable for Release Snapshot Name

### About this task

This procedure sets a variable for the name of the snapshot that will be used to install and configure the software update. Setting a variable now enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

### Procedure

Set the SNAPSHOT environment variable for the release snapshot to the name of the release to be installed and today's date.

```
smw# export SNAPSHOT=SMW-8.0UP03_CLE-6.0UP03.${TODAY}
smw# echo $SNAPSHOT
```

## 4.1.4 Make a Pre-update Release Snapshot using snaputil

### Prerequisites

This procedure assumes that the variable for the release snapshot name was set in *Set Variable for Release Snapshot Name* on page 244.

### About this task

This procedure uses snaputil to make an archival release snapshot prior to any update activities.

**How many snapshots are needed?** Sites can make as few or as many snapshots as they deem useful. Cray recommends making an archival snapshot of the system at these software update milestones.

| | |
|---|---|
| **preupdate** | before beginning any software update activities (software update only) |
| **postinstall** | after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware |
| **postpe** | after installing Cray PE software |

**postboot** after booting the CLE system

For more information, see *About Snapshots and Config Set Backups* on page 16.

## Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
Status    Name                                      Size (MB unshared)  Created
-------  -----------------------------------------  ----------------  ------------------
         @                                                    5089.28  2015-05-12 07:27:18
         SLES12                                               16445.4  2015-05-12 08:28:44
         SMW-8.0.UP01_CLE-6.0.UP01.20160330                   1010.67  2016-03-30 14:40:54
         SMW-8.0.UP01_CLE-6.0.UP01.20160330.preconfig           3.79  2016-03-30 16:44:50
cur,def  SMW-8.0.UP01_CLE-6.0.UP01.20160331                      9.04  2016-03-31 07:02:31
         SMW-8.0.UP01_CLE-6.0.UP01.20160331.postconfig        683.04  2016-03-31 09:20:27
         SMW-8.0.UP01_CLE-6.0.UP01.20160331.postboot          104.57  2016-03-31 09:20:27
         SMW-8.0.UP01_CLE-6.0.UP01.20160331.postpe            102.08  2016-03-31 09:20:27
```

2. Create the pre-update archival release snapshot.

   If the running system is what will be updated, create a snapshot from the currently booted system (denoted by "cur"), which is what `snaputil` uses by default. (Note that the default snapshot, denoted by "def," is what the system will boot from by default.)

```
smw# snaputil create ${SNAPSHOT}.preupdate
```

   If a different snapshot will be used for the software update, specify it using the `--from` argument with the `snaputil` command. This example uses a snapshot from March 30, which was a pre-config snapshot for that day.

```
smw# create ${SNAPSHOT}.preupdate \
--from SMW-8.0.UP01_CLE-6.0.UP01.20160330.preconfig
```

## 4.1.5 Make a Pre-update Backup of Current Global and CLE Config Sets

## About this task

Sites can back up the current global and CLE config sets as few or as many times as they deem useful. Cray recommends backing up the config sets at these software installation/configuration milestones, which correspond to the suggested milestones for making a snapshot.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

**preupdate** before beginning any software update activities (software update only)

**preconfig** after installing a software update and before updating the global and CLE config sets (software update only)

**postinstall** after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware

**postconfig** after configuring CLE and before booting the CLE system

**postboot** after booting the CLE system

**postpe** after installing Cray PE software

For more information, see *About Snapshots and Config Set Backups* on page 16.

## Procedure

1. Back up the current global config set.

   ```
   smw# cfgset create --clone global global-preupdate-${TODAY}
   ```

2. Back up the current CLE config set.

   This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

   ```
   smw# cfgset create --clone p0 p0-preupdate-${TODAY}
   ```

## 4.1.6 Prepare to Migrate Node Groups Configuration Data

## Prerequisites

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02.

This procedure assumes the following:

- This site is updating from SMW 8.0.UP01 / CLE 6.0.UP01 to SMW 8.0.UP03 / CLE 6.0.UP03.
- The SMW 8.0.UP03 / CLE 6.0.UP03 software has NOT been installed yet.

## About this task

This procedure prepares current CLE config sets to migrate site-specific node group configuration data from the SMW 8.0.UP01 / CLE 6.0.UP01 release to the SMW 8.0.UP03 / CLE 6.0.UP03 release. Repeat this procedure for each CLE config set that will be used to boot the system. Examples show commands for CLE config set p0; substitute the correct config set name(s) for this site.

Node Groups are a mechanism for defining logical groupings of Cray system nodes to streamline node specifications for use in other Cray configuration services. The node groups defined are non-exclusive, that is, a node may belong to more than one node group. They are referenced in other configuration templates and are used in Ansible plays as well. For more information, see *About Node Groups* on page 18.

## Procedure

1. Determine whether node groups were enabled and defined in this system in the previous release.

   The results of this search will show whether the cray_node_groups service is enabled and any node groups that might have been defined.

   ```
   smw# cfgset search -s cray_node_groups p0
   ```

2. (Optional) Save node groups worksheet.

If node groups have been defined for use with Simple Sync or other custom purposes, and this site plans to keep using them after the software update, save the current node groups worksheet now for reference after the new software is installed.

a. Generate a current set of configuration worksheets.

```
smw# cfgset update --mode prepare p0
```

b. Copy the node groups worksheet (cray_node_groups_**worksheet**.yaml) to a safe place for later reference.

The "safe" location could be off the SMW, or if on the SMW, a location that is accessible from any SMW snapshot, such as `~crayadm` or `/var/tmp`.

```
smw# cp /var/opt/cray/imps/config/sets/p0/worksheets/\
cray_node_groups_worksheet.yaml \
/some/safe/location/cray_node_groups_worksheet.yaml
```

**3.** Search current CLE config set(s) for data that will be needed for the new node groups settings.

This step can be done now or during the update of the new config set(s) after the new release software has been installed into a snapshot.

- To do it now, see *Update All Config Sets after a Software Update* on page 259, which lists the affected configuration services and provides links to migration procedures for each one. At this point in the process, only the `cfgset search` step of each migration procedure should be done. Any other steps must wait until the update of the new config set(s) later in the process.
- To do it later during the update of the new config set(s), a second terminal window will be needed to run the `cfgset search` command on the old config set, while the `cfgset update` command is running on the new config set in the main window.

## 4.1.7 Collect Software Media

### Prerequisites

This procedure assumes that the base operating system is installed on the SMW and the boot RAID is set up.

### About this task

The Cray release distribution media consists of one DVD and several other pieces of media that may be on DVDs or furnished as ISO files. These ISO files are available for download at CrayPort (*https://crayport.cray.com*).

The installer requires several ISO files to be available for setting up and installing packages from SLE repositories. The names of these ISOs are hard-coded in the installer configuration, but the containing directory can be anywhere that makes sense for this site.

**IMPORTANT:** The default location for these ISO files is `/root/isos`. If that default location is not used for this system, specify the correct location for the ISO files by using the `--iso-dir` argument with the `SMWinstall` command.

## Procedure

1. Make a directory on the SMW to hold the ISO files, if one does not already exist.

   Instead of placing the ISOs directly in `/root/isos`, use these two commands to place that directory into the btrfs subvolume `/var/adm/cray`, which is exempt from snapshots. This prevents the large ISO files from unnecessarily increasing the size of snapshots.

   ```
   smw# mkdir -p /var/adm/cray/release/isos
   smw# ln -s /var/adm/cray/release/isos /root/isos
   ```

2. Download the SLES 12 distribution ISOs to the ISO directory on the SMW.
   - `SLE-12-Module-Legacy-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-Module-Public-Cloud-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-SDK-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-Server-DVD-x86_64-GM-DVD1.iso`
   - `SLE-12-WE-DVD-x86_64-GM-DVD1.iso`

3. Download the CentOS 6.5 distribution ISO (`CentOS-6.5-x86_64-bin-DVD1.iso`) to the ISO directory on the SMW.

4. Download CLE 6.0 and SMW 8.0 ISOs to the ISO directory on the SMW.
   - SMW release: `smw-8.0.3075-201701182038.iso`
   - CLE release: `cle-6.0.3074-201701182038.iso`

5. Download the SLES 12 security updates ISO (`sleupdate-12sp0+161026-201611021158.iso`) to the ISO directory on the SMW.

6. Make a directory on the SMW to hold any patches that may be available on CrayPort, if one does not already exist.

   See the release Errata documents for information about patches.

   ```
   smw# mkdir -p /var/adm/cray/release/patchsets
   ```

7. Download patches to the patchset directory on the SMW.

   Note that patches are RPMs, not binaries.

## 4.2 Install the SMW and CLE Software Update

These procedures mount the software release media for the update, make a release snapshot, prepare boot images, and then install the software update into the newly created release snapshot. These changes all occur while the system is running.

1. *Mount Software Media and Prepare install.cle.conf* on page 249
2. *Make a Release Snapshot using snaputil* on page 250
3. *Prepare Boot Images and Recipes during a Software Update* on page 251

4. *Run the Installer for a Software Update* on page 254

## 4.2.1 Mount Software Media and Prepare install.cle.conf

### Prerequisites

This procedure assumes that the release software media have been collected and placed in the appropriate directories on the SMW.

### About this task

This procedure describes how to mount the SMW and CLE media and update the current `install.cle.conf` so that the installer will not build images and any new configuration options from this release are added.

### Procedure

MOUNT MEDIA

1. Mount SMW media.

   a. Confirm that this is the right SMW media.

   ```
   smw# ls -l /root/isos/smw*iso
   -rw-r--r-- 1 root root  427184128 Nov  9 10:41 smw-8.0.3075-201701182038.iso
   ```

   b. Set environment variables for the SMW media.

   Use the release string (actually, the build ID) and the date-time stamp for the SMW media as the values for SMW_RELEASE and SMW_SOFTWARE, as shown in this example.

   ```
   smw# export SMW_RELEASE=8.0.3075
   smw# echo $SMW_RELEASE

   smw# export SMW_SOFTWARE=201701182038
   smw# echo $SMW_SOFTWARE
   ```

   c. Mount the SMW release media.

   ```
   smw# mkdir -p /media/SMW
   smw# mount -o loop,ro /root/isos/smw-${SMW_RELEASE}-${SMW_SOFTWARE}.iso \
   /media/SMW
   ```

2. Mount CLE media.

   a. Confirm that this is the right CLE media.

   ```
   smw# ls -l /root/isos/cle*iso
   -rw-r--r-- 1 root root 1146388480 Nov  9 09:22 cle-6.0.3074-201701182038.iso
   ```

   b. Set environment variables for the CLE media.

   Use the release string and the date-time stamp for the CLE media as the values for CLE_RELEASE and CLE_SOFTWARE, as shown in this example.

```
smw# export CLE_RELEASE=6.0.3074
smw# echo $CLE_RELEASE

smw# export CLE_SOFTWARE=201701182038
smw# echo $CLE_SOFTWARE
```

c. Mount the CLE release media.

```
smw# mkdir -p /media/CLE
smw# mount -o loop,ro /root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
/media/CLE
```

**3.** Set an environment variable for the SLES12 security updates media.

Use the entire name of the SLES12 security updates media as the environment variable. This will be used when installing SMW and CLE software and SLES12 security updates together later in the process.

```
smw# export SLE_SOFTWARE=sleupdate-sle12sp0+161026-201611021158
smw# echo $SLE_SOFTWARE
```

PREPARE THE install.cle.conf FILE

**4.** Compare the current `install.cle.conf` on the SMW with `install.cle.conf.example` from the CLE media.

The `install.cle.conf` file contains configuration that controls the installer's image building behavior. Compare `install.cle.conf.example` from the CLE media to `install.cle.conf` on the SMW to see if there are additional configuration options available with this release of CLE software.

```
smw# diff /media/CLE/products/cle/install.cle.conf.example \
/var/adm/cray/install.cle.conf
```

The diff output should show at least the following, indicating that the current `install.cle.conf` will automatically generate boot images when the installer (`SMWinstall`) is run. It may show other differences as well, if new configuration options are available with this release of CLE software.

```
13c13
< build_images: no
---
> build_images: yes
```

**5.** Update the current `install.cle.conf`, if needed.

If the output of the diff command in step *4* on page 250 shows `build_images: no`, edit `/var/adm/cray/install.cle.conf` and set build_images to yes (if this site wants the installer to build images).

If there are other configuration options in `install.cle.conf.example` that appear in the diff output, add those to `/var/adm/cray/install.cle.conf` as well.

UNMOUNT MEDIA

**6.** Unmount CLE media.

```
smw# umount /media/CLE
```

## 4.2.2 Make a Release Snapshot using snaputil

### Prerequisites

This procedure assumes the variable for the release snapshot name was set in *Set Variable for Release Snapshot Name* on page 244.

### About this task

This procedure creates the release snapshot, which is the snapshot into which the software update will be installed while the system is running. The new software will be configured within the release snapshot, and then the system will be booted from it. See the snaputil(8) man page for more information about using the `snaputil` program.

### Procedure

1. List the available snapshots on the system.

   This example shows snapshots from the CLE 6.0.UP01 release. The actual list at this site will be different.

   ```
   smw# snaputil list
   Status    Name                                      Size (MB unshared)  Created
   -------  -----------------------------------------  ----------------  ------------------
            @                                                    5089.28  2015-05-12 07:27:18
            SLES12                                               16445.4  2015-05-12 08:28:44
            SMW-8.0.UP01_CLE-6.0.UP01.20160330                   1010.67  2016-03-30 14:40:54
            SMW-8.0.UP01_CLE-6.0.UP01.20160330.preconfig           3.79  2016-03-30 16:44:50
   cur,def  SMW-8.0.UP01_CLE-6.0.UP01.20160331                      9.04  2016-03-31 07:02:31
            SMW-8.0.UP01_CLE-6.0.UP01.20160331.postconfig        683.04  2016-03-31 09:20:27
            SMW-8.0.UP01_CLE-6.0.UP01.20160331.postboot          104.57  2016-03-31 09:20:27
            SMW-8.0.UP01_CLE-6.0.UP01.20160331.postpe            102.08  2016-03-31 09:20:27
            SMW-8.0.UP01_CLE-6.0.UP01.20160401.preupdate         510.21  2016-04-01 12:17:12
   ```

2. Create the release snapshot.

   If the running system is what will be updated, create a snapshot from the currently booted system (denoted by "cur"), which is what `snaputil` uses by default. (Note that the default snapshot, denoted by "def," is what the system will boot from by default.)

   ```
   smw# snaputil create ${SNAPSHOT}
   ```

   If a different snapshot will be used for the software update, specify it using the `--from` argument with the `snaputil` command. This example uses an UP01 snapshot from March 30, which was a pre-config snapshot for that day.

   ```
   smw# snaputil create ${SNAPSHOT} \
   --from SMW-8.0.UP01_CLE-6.0.UP01.20160330.pre-config
   ```

## 4.2.3 Prepare Boot Images and Recipes during a Software Update

### About this task

This procedure makes changes in node NIMS group assignments (if needed), updates the cray_image_groups configuration file, and ensures that the Cray image groups file and custom recipes have recipe names that indicate 'up03' instead of an earlier release. All of this is done within the newly created release snapshot.

### Procedure

1. Change into the newly created release snapshot.

   ```
   smw# snaputil chroot $SNAPSHOT
   chroot-smw#
   ```

   (Although steps 2 and 3 do not need to be done within the snapshot, step 4 does, so for simplicity, the entire procedure takes place there.)

   CHANGE NIMS GROUP ASSIGNMENTS

2. Assign the boot and SDB nodes to the admin NIMS group.

   If updating from CLE 6.0.UP02, skip this step and proceed to step *3* on page 253. This step is needed only for sites updating from UP01 because the admin image recipe was introduced with the UP02 release. For more information, see *About the Admin Image* on page 25.

   If a custom recipe was created for the SDB node(s), it may be necessary to assign a different NIMS group, where the name of the NIMS group may have the same name as the custom recipe. This example uses c0-0c0s0n1 and c0-0c0s1n1 as the admin (boot and SDB) nodes. Substitute the correct cnames for this site when using these commands.

   **full system** For a full, unpartitioned system, remove from the NIMS service group and add to the NIMS admin group. Note that this change to NIMS will affect the running system.

   ```
   smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1
   ```

   If this site wishes to minimize the impact to the running system, add the nodes to the admin group without removing them from the service group.

   ```
   smw# cnode update -g admin c0-0c0s0n1 c0-0c0s1n1
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2, remove from the NIMS service group and add to the NIMS login group:

   ```
   smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p1
   smw# cnode update -G service -g admin c0-0c0s0n1 c0-0c0s1n1 -p p2
   ```

   If this site wishes to minimize the impact to the running system, add the nodes to the admin group without removing them from the service group.

   ```
   smw# cnode update -g admin c0-0c0s0n1 c0-0c0s1n1 -p p1
   smw# cnode update -g admin c0-0c0s0n1 c0-0c0s1n1 -p p2
   ```

   PREPARE CRAY IMAGE GROUPS AND CUSTOM RECIPES

**3.** Update the Cray image groups file.

Make a copy of the default image group so that if this site needs to run `imgbuilder` while still booted from the previous release, the copy can be used. Then update the default image group in preparation for building images for the UP03 release.

```
chroot-smw# vi /
/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

a. Save a copy of the default image group.

Copy the current default image group to another section in `cray_image_groups.yaml` and name it 'default_up02' (if upgrading from the UP02 release) or 'default_up01' (if upgrading from the UP01 release).

b. Ensure that the admin image specification is in the default image group.

Sites updating from UP02 will already have this stanza in the default image group, but sites updating from UP01 will need to add it. Place the admin stanza at the end of the default group, so that if the boot and SDB nodes have been assigned to other NIMS groups as well, their images will be assigned correctly by `imgbuilder --map`.

```
cray_image_groups:
  default:
...
      - recipe: "admin_cle_6.0up03_sles_12_x86-64_ari"
        dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "admin"
```

c. Update recipe names in the default image group.

In the default image group, find all instances of the old version and change them to the new version. For example, the 'up01' or 'up02' in these recipe names would be changed to 'up03.'

Also, look for any instances of 'lus25' and change them to 'lus27.' The current release uses Lustre 2.7, while CLE 6.0.UP02 and earlier releases used Lustre 2.5.

```
cray_image_groups:
  default:
      - recipe: "compute_cle_6.0up03_sles_12_x86-64_ari"
        dest: "compute{note}_cle_{cle_release}-build{cle_build}
{patch}_sles_12-created{date}.cpio"
        nims_group: "compute"
      - recipe: "login_cle_6.0up03_sles_12_x86-64_ari"
        dest: "login{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "login"
      - recipe: "service_cle_6.0up03_sles_12_x86-64_ari"
        dest: "service{note}_cle_{cle_release}-build{cle_build}
{patch}_sles_12-created{date}.cpio"
        nims_group: "service"
...
      - recipe: "admin_cle_6.0up03_sles_12_x86-64_ari"
        dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12-
created{date}.cpio"
        nims_group: "admin"
```

**4.** Update recipe and package collection names in existing custom recipes.

If this site has custom recipes, such as for the installation of workload manager (WLM) software or local site repositories, package collections, or RPMs, then clone the custom recipes and update the clones to reference the UP03 recipes and package collections. Note that WLM recipes may reference Lustre 2.5. If so, change all instances of 'lus25' to 'lus27' because the current release uses Lustre 2.7 instead of Lustre 2.5.

Cray recommends using the `recipe` command to update a recipe rather than editing the recipe JSON file directly.

a. View the contents of the custom recipe.

```
chroot-smw# recipe show my_old_recipe
```

b. Create a new custom recipe.

```
chroot-smw# recipe create --clone my_old_recipe my_up03_recipe
```

c. Remove UP02 or UP01 recipes, package collections, and repositories contained by the new custom recipe.

```
chroot-smw# recipe update --remove-recipe old_subrecipe my_up03_recipe
chroot-smw# recipe update --remove-coll old_packagecollection my_up03_recipe
chroot-smw# recipe update --remove-repo old_repo my_up03_recipe
```

d. Add the names of UP03 recipes, package collections, and repositories to the new custom recipe.

   The actual UP03 recipes, package collections, and repositories are not available in this snapshot until the UP03 release is installed, which occurs in the next procedure. For this substep, simply take the names of the recipes, package collections, and repositories that were removed in substep 4c, replace 'up01' or 'up02' with 'up03,' and add them to the UP03 custom recipe using these commands.

```
chroot-smw# recipe update --add-recipe new_subrecipe my_up03_recipe
chroot-smw# recipe update --add-coll new_packagecollection my_up03_recipe
chroot-smw# recipe update --add-repo new_repo my_up03_recipe
```

e. Ensure that any site custom recipes are in the default image group or a site-specific stanza in `/var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml` so that they will get built later.

**5.** Exit from the release snapshot.

```
chroot-smw# exit
smw#
```

## 4.2.4 Run the Installer for a Software Update

### Prerequisites

This procedure assumes the following:

- All of the ISOS for SLES 12 and CentOS 6.5 have been downloaded as described in *Collect Software Media* on page 247.

- The release snapshot, which will be used as the target snapshot, was created in *Make a Release Snapshot using snaputil* on page 250.

## About this task

This procedure installs SMW and CLE software together to ensure that there is a matched set of software and configuration.

> **NOTE:** Do NOT run the installer from the `/root/isos` directory. Instead, run it from a directory that is not included in any snapshot, such as `/var/adm/cray/release`.

## Procedure

**1.** Install SMW and CLE software and security updates together into the release snapshot.

The `SMWinstall` program installs SMW media, CLE media, and SLES security updates with a single command to create a unified "release" that is tagged as a snapshot on the SMW system.

> **IMPORTANT:** Specify the SLE media before the CLE media on the command line so that SUSE security updates are installed before the CLE software is installed.

```
smw# /media/SMW/SMWinstall \
--plus-media=/root/isos/${SLE_SOFTWARE}.iso \
--plus-media=/root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
--target=${SNAPSHOT}
```

Note: If running the installer on a system that has more than one SMW storage set, or the SMW storage set does not have the name "smwdefault," then invoke `SMWinstall` with this additional argument: `--storage-set=`*`alternatesmwstorageset`*.

It will take about 15 minutes to run a combined installation of SMW, CLE, and security updates for the first time on the SMW. The time required for the images to build depends on how many images are to be built. To build the admin, tmpfs compute, tmpfs login, Netroot compute, Netroot login, service, and DAL images could take another 90 minutes after the installation has completed. Here is example output of a completed installation.

The output of `SMWinstall` provides several command hints, including these three:

**snaputil default**
The first command hint (snaputil default) is used to ensure that the SMW is booted from the correct (new) snapshot, which is essential to a successful reboot.

**snaputil chroot**
The second command hint (snaputil chroot) is used in the next section to switch to a chroot'd environment. Within the new snapshot, the configurator will be run to make any needed changes in configuration due to new or changed features and also to create new boot images containing the new software.

**snaputil delete**
The third command hint (snaputil delete) should be used only if this site needs to remove the newly created snapshot for any reason.

Logs will be in `/var/adm/cray/logs/install` for each invocation of `SMWinstall`.

**Trouble?** If the update needs to be uninstalled, or the system must be downgraded to an earlier release, use the `--forceupdate` flag. This flag bypasses the version validation check and passes the `--force` argument to zypper, which performs a full uninstall+reinstall of every specified package.

```
smw# /media/SMW/SMWinstall \
--plus-media=/root/isos/${SLE_SOFTWARE}.iso \
--plus-media=/root/isos/cle-${CLE_RELEASE}-${CLE_SOFTWARE}.iso \
--target=${SNAPSHOT} --forceupdate
```

2. Check new snapshot software versions.

   When `SMWinstall` completes, check the snapshot details for the expected SMW and CLE release versions.

   ```
   smw# /media/SMW/snaputil show ${SNAPSHOT}
   ```

## 4.3 Configure in Release Snapshot

These procedures make configuration changes to the config set within the release snapshot, not the running system. When they are complete, the SMW can be rebooted to the new release snapshot to update the system. This approach minimizes downtime on the CLE system.

1. *Begin Configuration in Release Snapshot* on page 256
2. *Update Node Groups with Cray Defaults* on page 256
3. *Update All Config Sets after a Software Update* on page 259
4. *Display All Config Set Information* on page 283
5. *End Configuration in Release Snapshot* on page 283

### 4.3.1 Begin Configuration in Release Snapshot

#### Prerequisites

This procedure assumes that a new release of SMW and CLE software has been installed into a release snapshot using *Run the Installer for a Software Update* on page 254.

#### About this task

This procedure uses snaputil to chroot into the release snapshot so that the software newly installed into that snapshot can be configured.

#### Procedure

Change into the newly created release snapshot.

```
smw# snaputil chroot ${SNAPSHOT}
chroot-smw#
```

Note that the prompt has changed from `smw#` to `chroot-smw#`.

## 4.3.2 Update Node Groups with Cray Defaults

### Prerequisites

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02.

This procedure assumes the following:

- This site is updating from SMW 8.0.UP01 / CLE 6.0.UP01 to SMW 8.0.UP03 / CLE 6.0.UP03.
- New software (SMW, CLE, and SLE updates) has been installed into a snapshot.
- The administrator performing the software update has used `snaputil` to `chroot` to that snapshot.

### About this task

This procedure regenerates the node groups service from the configuration templates in the newly installed SMW/CLE software. For sites that saved site-specific node group data from a previous release, this procedure also merges that data with the default node groups data in the regenerated node groups service. Repeat these steps for each CLE config set that will be used to boot the system.

**CAUTION: Boot failure possible if using `cfgset` under certain conditions.**

The `cfgset create` and `cfgset update` commands always call pre- and post-configuration scripts. Some of these scripts require HSS daemons and other CLE services to be running. This can cause problems under these conditions:

- If `xtdiscover` is running, `cfgset` may hang or produce incorrect data that can result in system boot failure.
- If `xtbounce` is in progress or if the SMW is not connected to XC hardware, `cfgset` will fail.

In these circumstances, use the `--no-scripts` option with `cfgset create` or `cfgset update` to avoid running the scripts. Because using that option results in an invalid config set, remember to run `cfgset update` without the `--no-scripts` option afterwards, when circumstances permit, to ensure that all pre- and post-configuration scripts are run.

### Procedure

1. Remove the node groups template from the config set.

```
chroot-smw# rm /var/opt/cray/imps/config/sets/p0/config/
cray_node_groups_config.yaml
```

2. Generate a new node groups template and new worksheets in the config set.

```
chroot-smw# cfgset update --mode prepare --no-scripts p0
```

3. Copy the new node groups worksheet to a location where it can be edited.

```
chroot-smw# cp /var/opt/cray/imps/config/sets/p0/worksheets/\
cray_node_groups_worksheet.yaml /tmp/cray_node_groups_worksheet.yaml
```

If the CLE 6.0.UP01 node groups worksheet was saved in *Prepare to Migrate Node Groups Configuration Data* on page 246, proceed to the next step. Otherwise, skip to step *5* on page 258.

**4.** (If CLE 6.0.UP01 node groups worksheet saved previously) Manually merge selected worksheet entries from the saved CLE 6.0.UP01 node groups worksheet with the new node groups worksheet in `/tmp`.

If the CLE 6.0.UP01 node groups worksheet was saved, copy all CLE 6.0.UP01 worksheet entries beginning with "cray_node_groups.settings.groups.data" into the following location in `/tmp/cray_node_groups_worksheet.yaml`. Do not copy any other data from the CLE 6.0.UP01 worksheet (e.g., cray_node_groups.enabled). This example shows an entry for *`my_node_group`*.

```
<snip>
#------------------------------ groups : members -----------------------------
# Node Group Member List (type=list, level=REQUIRED)
# Guidance:
#    Supply the name of the nodes that are members of this group. For CLE nodes,
#    this is the c-name of the node. If an SMW node is included in the group,
#    enter the output of the 'hostid' command from the SMW node. For eLogin
<snip>
#
#
# ** END 'groups' FIELD SPECIFICATION
#
# ** 'groups' DATA **


#@@@@@@@@@@@@@ >>>>>>>>>>>>>
#@@@@@@@@@@@@@ PLACE NODE GROUPS DATA FROM
#@@@@@@@@@@@@@   /some/safe/location/cray_node_groups_worksheet.yaml HERE
#@@@@@@@@@@@@@ >>>>>>>>>>>>>
cray_node_groups.settings.groups.data.my_node_group.members:
- c0-0c0s0n0
- c1-1c1s1n1


### DEFAULT CRAY DATA BEGINS HERE ###
cray_node_groups.settings.groups.data.group_name.compute_nodes: null
cray_node_groups.settings.groups.data.compute_nodes.description: Default node
group
    which contains all the compute nodes for the current partition.
cray_node_groups.settings.groups.data.compute_nodes.members:
- platform:compute
<snip>
```

The resulting `/tmp/cray_node_groups_worksheet.yaml` file should now include node groups from before the CLE software update as well as the default node groups provided by Cray in the software update.

**5.** Uncomment `cray_node_groups.enabled` and ensure that it is set to `true`.

```
# Enable 'cray_node_groups' Service? (boolean, level=required)
cray_node_groups.enabled: true
```

**6.** Update the config set with the new node groups worksheet.

```
chroot-smw# cfgset update --no-scripts \
-w '/tmp/cray_node_groups_worksheet.yaml' p0
```

**7.** Verify that the default node groups are now present.

To verify that the default node groups are now in the config set, output the membership of the node groups from the config set. For sites that saved site-specific node group data from a previous release, verify that the previously defined node groups were correctly merged into the new worksheet and applied to the config set properly by comparing with the previously defined groups. In this example, the first node group (my_node_group) is an example of a previously defined node group. The remaining node groups are default Cray data and can be left as they are for now. They will be customized later in the software update process.

```
chroot-smw# cfgset search -s cray_node_groups -t members p0

# 9 matches for 'members' from cray_node_groups_config.yaml
#--------------------------------------------------------------------------------
cray_node_groups.settings.groups.data.my_node_group.members: [c0-0c0s0n0,
c1-1c1s1n1]
cray_node_groups.settings.groups.data.compute_nodes.members: platform:compute
cray_node_groups.settings.groups.data.service_nodes.members: platform:service
cray_node_groups.settings.groups.data.smw_nodes.members: [ ] # (empty)
cray_node_groups.settings.groups.data.boot_nodes.members: [ ] # (empty)
cray_node_groups.settings.groups.data.sdb_nodes.members: [ ] # (empty)
cray_node_groups.settings.groups.data.login_nodes.members: [ ] # (empty)
cray_node_groups.settings.groups.data.all_nodes.members: [ ] # (empty)
cray_node_groups.settings.groups.data.tier2_nodes.members: [ ] # (empty)
```

**Trouble?** If the verification shows that the default node groups are not present or the merge (if applicable) was not done correctly, repeat this procedure, beginning at step 1.

When this procedure has been done for each CLE config set that will be used to boot the system, proceed to *Update All Config Sets after a Software Update* on page 259.

## 4.3.3 Update All Config Sets after a Software Update

### Prerequisites

This procedure assumes the following:

- CLE and global config sets from the previous release have been backed up.
- New software (SMW, CLE, and SLE updates) has been installed into a snapshot.
- The administrator performing the update has used `snaputil` to `chroot` to that snapshot.
- (For sites updating from SMW 8.0.UP01 / CLE 6.0.UP01 to the current release only) These procedures for preparing to migrate node groups have been completed:

  *Prepare to Migrate Node Groups Configuration Data* on page 246
  *Update Node Groups with Cray Defaults* on page 256

### About this task

The software update brings in all new configuration templates, so this procedure runs the configurator in auto mode to merge the new content with CLE and global config sets already on the system.

> **NOTICE:** As the command prompts indicate, these steps occur within a chrooted environment—the snapshot that has the newly installed software updates.

### Procedure

**1.** Update the current CLE config set (p0 in the example) to merge the new content.

With this command, the configurator will prompt only for settings that are level required or basic and have not been set before. Repeat this step for each CLE config set that will be used to boot the system.

```
chroot-smw# cfgset update p0
```

For sites updating from UP02 to the current release: skip the rest of this step and proceed to step *3* on page 261.

-----------------------------------------------

**For sites updating from UP01 to the current release only:**

The configurator will prompt for a new node groups setting in each of the configuration services listed below. When prompted for one of these settings, enter `v` to view existing node groups. If none of the existing node groups are appropriate for this setting, enter `!`, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group. The `v` and `!` menu commands are available because this is a "lookup" field. For details about how lookup fields work, see the "Advanced: Lookup" section in "Configurator Data Types and How to Set Them," which is in *XC™ Series Configurator User Guide* (S-2560).

For guidance about how to migrate UP01 node groups data in a particular service, refer to the migration procedure for that service. The order in which services are configured may not be the same as the order of this list.

- *Migrate Node Groups in cray_alps* on page 265
- *Migrate Node Groups in cray_auth* on page 266
- *Migrate Node Groups in cray_boot* on page 267
- *Migrate Node Groups in cray_dvs* on page 267
- *Migrate Node Groups in cray_dws* on page 268
- *Migrate Node Groups in cray_lnet* on page 269
- *Migrate Node Groups in cray_local_users* on page 270
- *Migrate Node Groups in cray_login* on page 271
- *Migrate Node Groups in cray_lustre_client* on page 272
- *Migrate Node Groups in cray_lustre_server* on page 273
- *Migrate Node Groups in cray_net* on page 274
- *Migrate Node Groups in cray_node_groups* on page 276
- *Migrate Node Groups in cray_persistent_data* on page 277
- *Migrate Node Groups in cray_rsip* on page 278
- *Migrate Node Groups in cray_scalable_services* on page 279
- *Migrate Node Groups in cray_sdb* on page 280
- *Migrate Node Groups in cray_simple_shares* on page 281

**2.** (Sites updating from UP01 only) Update selected services in the current config set to complete node groups migration.

Perform this step after the configuration in step *1* on page 259 is complete for all CLE config sets that will be used to boot the system. Repeat this step for the same CLE config sets.

a. Update the level advanced settings in the cray_rsip configuration service.

Some node groups settings in the cray_rsip service are level advanced (as indicated in the table in *Migrate Node Groups in cray_rsip* on page 278), so they were not presented for configuration in step *1* on page 259. To update them, use this command (the example uses config set *p0*).

```
chroot-smw# cfgset update --service cray_rsip --level advanced p0
```

The configurator may prompt for other level advanced settings as well: configure those settings by accepting the defaults, or use the configurator interface to skip them.

b. Update `hostid` setting for eLogin nodes in the cray_net configuration service.

See *Migrate Node Groups in cray_net* on page 274 for instructions.

c. Update `domain_groups` settings in the cray_local_users configuration service.

To simplify migration, Cray has set the default values of all `domain_groups` settings to the 'all_nodes' pre-defined node group. Sites may leave those values as set (no action needed) or change them using the procedure in *Migrate Node Groups in cray_local_users* on page 270.

**3.** (For sites updating from UP01 or UP02 only) Update the current CLE config set (p0 in the example) interactively to change cray_liveupdates, cray_zonesort, cray_persistent_data, and cray_multipath.

a. Update the CLE config set in interactive mode and level advanced.

```
chroot-smw# cfgset update -m interactive -l advanced p0
```

b. Update cray_liveupdates to set its inheritance.

The CLE 6.0.UP03 release introduces a global live updates service, so the CLE live updates service must be set to inherit from the new global one.

In this example, cray_liveupdates is number 34 in the list of config services. Enter **`34`** to select it.

```
Service List Menu [default: save & exit - Q] $ 34
```

At the prompt, enter **`I`** to toggle inheritance and set it to inheriting.

```
Service List Menu [default: configure - C] $ I
```

DO NOT save changes and exit the configurator yet.

c. Update cray_zonesort to enable it.

The CLE 6.0.UP03 release introduces a zone sort service to configure loading of the zonesort_module kernel module, which improves the predictability of MCDRAM (multi-channel dynamic random-access memory) cache performance.

In this example, cray_zonesort is number 49 in the list of config services. Enter **`49`** to select it, then enter **`E`** to toggle the service status from unconfigured to disabled. Repeat both entries to select it and toggle the service status to enabled.

```
Service List Menu [default: save & exit - Q] $ 49
...
Service List Menu [default: save & exit - Q] $ E
```

```
Service List Menu [default: save & exit - Q] $ 49
...
Service List Menu [default: save & exit - Q] $ E
```

DO NOT save changes and exit the configurator yet.

d. Update cray_persistent_data to add an NFS mount.

An NFS mount must be added to ensure that NFS data for the boot and SDB nodes will be persistent when failing over from the primary node to the alternate node.

In this example, cray_persistent_data is number 32 in the list of config services. Enter **`32`** to select it, then enter **`v`** to view its settings.

```
Service List Menu [default: save & exit - Q] $ 32
...
Service List Menu [default: configure - C] $ v
```

At the prompt for this service, enter **`2`** to select the mounts setting, then enter **`C`** to configure it.

```
Cray Persistent Data Configuration Service Menu [default: save & exit - Q] $
2
...
Cray Persistent Data Configuration Service Menu [default: configure - C] $ C
```

At the prompt for the mounts setting, enter **`+`** to add a new mount entry, then enter **`C`** to configure it.

```
cray_persistent_data.settings.mounts
[<cr>=set 4 entries, +=add an entry, ?=help, @=less] $ +
```

```
cray_persistent_data.settings.mounts.data.mount_point
[<cr>=set '', <new value>, ?=help, @=less] $ /var/lib/nfs
```

```
cray_persistent_data.settings.mounts.data./var/lib/nfs.alt_storage_path
[<cr>=set '', <new value>, ?=help, @=less] $ <cr>
```

```
cray_persistent_data.settings.mounts.data./var/lib/nfs.options
[<cr>=set '', <new value>, ?=help, @=less] $ rw
```

```
cray_persistent_data.settings.mounts.data./var/lib/nfs.ancestor_def_perms
[<cr>=set '0771', <new value>, ?=help, @=less] $ 0755
```

```
cray_persistent_data.settings.mounts.data./var/lib/nfs.client_groups
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add client_groups (Ctrl-d to exit) $ boot_nodes
Add client_groups (Ctrl-d to exit) $ sdb_nodes
Add client_groups (Ctrl-d to exit) $ <Ctrl-d>
```

```
cray_persistent_data.settings.mounts.data./var/lib/nfs.client_groups
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
```

```
cray_persistent_data.settings.mounts
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ <cr>
```

DO NOT save changes and exit the configurator yet. Instead, enter **`^^`** to return to the service list menu within the configurator.

```
Cray Persistent Data Configuration Service Menu [default: save & exit - Q] $
^^
```

e. Update cray_multipath to correct several device settings.

If the CLE cray_multipath service is disabled or is set to inherit from the global multipath service, skip the rest of this step and proceed to the next step. Otherwise, continue with these instructions to correct the values of three pre-populated device settings.

In this example, cray_multipath is number 26 in the list. Enter **`26`** to select it, then enter **`v`** to view its settings.

```
Service List Menu [default: save & exit - Q] $ 26
...
Service List Menu [default: configure - C] $ v
```

At the prompt for this service, enter **`33`** to select the enable_devices setting, then enter **`C`** to configure it.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ 33

Cray Multipath Configuration Service Menu [default: configure - C] $ C
```

At the prompt for this setting, enter **`*`** to view all of the pre-populated device settings.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ *
```

Find the DDN_EF3015 device in the list of enabled devices, and enter its number (5 in this example) followed by 'd' and '*' to select and edit the path_grouping_policy field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 5d*
```

Change the value to **`group_by_prio`**.

```
cray_multipath.settings.enabled_devices.data.DDN_EF3015.path_grouping_policy
[<cr>=keep 'multibus', <new value>, ?=help, @=less] $ group_by_prio
```

Find the DDN_SFA12K_20 device in the list of enabled devices, and enter its number (10 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 10b*
```

Change the value to **`SFA12K-20`**.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_20.product
[<cr>=keep 'SFA12K20', <new value>, ?=help, @=less] $ SFA12K-20
```

Find the DDN_SFA12K_40 device in the list of enabled devices, and enter its number (11 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 11b*
```

Change the value to **`SFA12K-40|SFA12KX*`**.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_40.product
[<cr>=keep 'SFA12K40', <new value>, ?=help, @=less] $ SFA12K-40|SFA12KX*
```

Set the enabled_devices entries, then save changes and exit the configurator.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ <cr>
```

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ Q
```

4. (For sites updating from UP01 or UP02 only) Update cray_liveupdates and cray_multipath in the global config set.

   a. Update the global config set in interactive mode and level advanced.

   ```
   chroot-smw# cfgset update -m interactive -l advanced global
   ```

   b. Update cray_liveupdates to enable it.

   The CLE 6.0.UP03 release introduced a global live updates service, so the new global live updates service must be enabled.

   In this example, cray_liveupdates is number 5 in the list. Enter **5** to select it, then enter **E** to toggle the service status from unconfigured to disabled. Repeat both entries to select it and toggle the service status to enabled.

   ```
   Service List Menu [default: save & exit - Q] $ 5
   ...
   Service List Menu [default: save & exit - Q] $ E
   ```

   ```
   Service List Menu [default: save & exit - Q] $ 5
   ...
   Service List Menu [default: save & exit - Q] $ E
   ```

   c. Update cray_multipath to correct the values of three pre-populated device settings.

   In this example, cray_multipath is number 7 in the list of services. Enter **7** to select it, then enter **v** to view its settings.

   ```
   Service List Menu [default: save & exit - Q] $ 7
   ...
   Service List Menu [default: configure - C] $ v
   ```

   At the prompt for this service, enter **33** to select the enable_devices setting, then enter **C** to configure it.

   ```
   Cray Multipath Configuration Service Menu [default: save & exit - Q] $ 33

   Cray Multipath Configuration Service Menu [default: configure - C] $ C
   ```

   At the prompt for this setting, enter ***** to view all of the pre-populated device settings.

   ```
   cray_multipath.settings.enabled_devices
   [<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ *
   ```

   Find the DDN_EF3015 device in the list of enabled devices, and enter its number (5 in this example) followed by 'd' and '*' to select and edit the path_grouping_policy field.

   ```
   cray_multipath.settings.enabled_devices
   [<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 5d*
   ```

   Change the value to **group_by_prio**.

   ```
   cray_multipath.settings.enabled_devices.data.DDN_EF3015.path_grouping_policy
   [<cr>=keep 'multibus', <new value>, ?=help, @=less] $ group_by_prio
   ```

   Find the DDN_SFA12K_20 device in the list of enabled devices, and enter its number (10 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 10b*
```

Change the value to **SFA12K-20**.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_20.product
[<cr>=keep 'SFA12K20', <new value>, ?=help, @=less] $ SFA12K-20
```

Find the DDN_SFA12K_40 device in the list of enabled devices, and enter its number (11 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 11b*
```

Change the value to **SFA12K-40|SFA12KX***.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_40.product
[<cr>=keep 'SFA12K40', <new value>, ?=help, @=less] $ SFA12K-40|SFA12KX*
```

Set the enabled_devices entries, then save changes and exit the configurator.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ <cr>
```

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ Q
```

5. Update the global config set.

   ```
   chroot-smw# cfgset update global
   ```

- For sites updating from SMW 8.0.UP02 / CLE 6.0.UP02 or higher to the current release: skip all node-group migration procedures and proceed to *Display All Config Set Information* on page 283.
- For sites updating from UP01 to the current release: perform all node-group migration procedures, beginning with *Migrate Node Groups in cray_alps* on page 265.

### 4.3.3.1 Migrate Node Groups in cray_alps

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_alps replaces `alps_nodes`, a list of nodes, with `alps_node_groups`, a list of node groups.

*Table 16. cray_alps*

All settings begin with the `cray_alps.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| .common.data.alps_nodes | .common.data.alps_node_groups |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_alps p0-preupdate
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.2 Migrate Node Groups in cray_auth

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_auth replaces `config_id_service`, a list of nodes, with `config_id_service_groups`, a list of node groups.

*Table 17. cray_auth*

All settings begin with the `cray_auth.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.access.data.config_id_service` | `.access.data.config_id_service_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_auth p0-preupdate
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

**3.** Set the value of the replacement setting to the node group(s) identified or created in the previous step.

Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.3 Migrate Node Groups in cray_boot

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The SMW 8.0.UP02 / CLE 6.0.UP02 release introduced cray_boot, a new configuration service/template. The cray_boot service has one node groups field to set, which is a set of node groups that define the CLE boot nodes. This service must be enabled in order for the system to boot properly.

Do this procedure only while updating the new config set.

*Table 18. cray_boot*

All settings begin with the `cray_boot.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| N/A | `.node_groups.data.boot_groups` |

Update this field with the name(s) of the node group(s) that define the boot node and the boot failover node, if applicable. Cray recommends that sites use the pre-defined 'boot_nodes' node group rather than create a custom node group.

## Procedure

**1.** Ensure that the cray_boot service is enabled.

This service must be enabled in order for the system to boot properly.

```
cray_boot.enabled
[<cr>=set 'true', <new value>, ?=help, @=less] $ <cr>
```

**2.** In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

**3.** Set the value of `cray_boot.settings.node_groups.data.boot_groups` to the pre-defined node group `boot_nodes`.

### 4.3.3.4 Migrate Node Groups in cray_dvs

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_dvs replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 19. cray_dvs*

All settings begin with the `cray_dvs.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.client_mount.data.`*`reference`*`.servers` | `.client_mount.data.`*`reference`*`.server_groups` |
| `.client_mount.data.`*`reference`*`.clients` | `.client_mount.data.`*`reference`*`.client_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_dvs p0-pre-update
   ```

   If the previously defined values for the `servers` and `clients` settings include the deprecated values in the left column of the following table, use the values in the right column for the replacement settings (`server_groups` and `client_groups`). If the `clients` setting was an empty list previously, the `client_groups` setting can be left empty.

   *Table 20. Map DVS servers and clients to DVS server_groups and client_groups*

   | Value in deprecated setting | Node group(s) to use in replacement setting |
   |---|---|
   | tier2 | tier2_nodes |
   | compute | compute_nodes |

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.5 Migrate Node Groups in cray_dws

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_dws replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 21. cray_dws*

All settings begin with the `cray_dws.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.service.data.managed_nodes` | `.service.data.managed_nodes_groups` |
| `.service.data.api_gateway_nodes` | `.service.data.api_gateway_nodes_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_dws p0-pre-update
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.6 Migrate Node Groups in cray_lnet

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_lnet replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 22. cray_lnet*

All settings begin with the `cray_lnet.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| .flat_routes.data.*dest_lnet*.routers | .flat_routes.data.*dest_lnet*.router_groups |
| .fgr_routes.data.*dest_name*.routers | .fgr_routes.data.*dest_name*.router_groups |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_lnet p0-preupdate
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.7 Migrate Node Groups in cray_local_users

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_local_users replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 23. cray_local_users*

All settings begin with the `cray_local_users.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| .users.data.*userid*.domains | .users.data.*userid*.domain_groups |
| .groups.data.*groupid*.domains | .groups.data.*groupid*.domain_groups |

**Action required. Maybe.** To simplify migration, Cray has set the default values of all `domain_groups` settings to the 'all_nodes' pre-defined node group. Sites may leave those values as set (skip this procedure) or change them (use this procedure).

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_local_users --level advanced p0-pre-update
   ```

   If the previously defined values for the `domains` settings include the deprecated values in the left column of the following table, use the values in the right column for the `domain_groups` settings.

   *Table 24. Map domains to domain_groups*

   | Value in deprecated setting | Node group(s) to use in replacement setting |
   |---|---|
   | admin | all_nodes |
   | compute | compute_nodes |
   | login | login_nodes |

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. Update only the `domain_groups` settings in the cray_local_users configuration service.

   This step is done after the new config set has been updated using the configurator mode that prompts only for settings that are level required or basic and have not been set before. With this command, the configurator will prompt for all settings of level required or basic, regardless of whether the setting has been configured. Use the configurator interface to skip all settings that are not the `domain_groups` settings shown in the first table.

   ```
   chroot-smw# cfgset update --service cray_local_users --state all p0
   ```

3. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

4. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.8 Migrate Node Groups in cray_login

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_login replaces the deprecated setting listed in the table, which is a list of nodes, with the replacement setting, which is a list of node groups.

*Table 25. cray_login*

All settings begin with the `cray_login.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.login_nodes.data.members` | `.login_nodes.data.member_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_login p0-pre-update
   ```

   Cray recommends that sites add the nodes shown in the `members` list (and any other login nodes on the system) to the pre-defined 'login_nodes' node group rather than create a custom node group.

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of `cray_login.settings.login_nodes.data.member_groups` to the pre-defined node group `login_nodes`.

### 4.3.3.9 Migrate Node Groups in cray_lustre_client

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_lustre_client replaces the deprecated setting listed in the table, which is a list of nodes, with the replacement setting, which is a list of node groups.

*Table 26. cray_lustre_client*

All settings begin with the `cray_lustre_client.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.client_mounts.data.`*`fs_name`*`.mount_locations` | `.client_mounts.data.`*`fs_name`*`.client_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_lustre_client p0-pre-update
   ```

   If the previously defined value for the `mount_locations` setting includes the deprecated values in the left column of the following table, use the values in the right column for the replacement setting (`client_groups`).

   *Table 27. Map mount_locations to client_groups*

   | Value in deprecated setting | Node group(s) to use in replacement setting |
   |---|---|
   | service | service_nodes |
   | compute | compute_nodes |
   | login | login_nodes |
   | elogin | elogin_nodes |

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.10 Migrate Node Groups in cray_lustre_server

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_lustre_server replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 28. cray_lustre_server*

All settings begin with the `cray_lustre_server.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.lustre_servers.mgs` | `.lustre_servers.mgs_group` |
| `.lustre_servers.mds` | `.lustre_servers.mds_groups` |

| Deprecated Setting | Replacement Setting |
|---|---|
| `.lustre_servers.oss` | `.lustre_servers.oss_groups` |

> **NOTICE: Why `mgs_group` instead of `mgs_groups`?** Because the replacement settings are lists of node groups, it is possible to enter multiple node groups for each setting, and each node group could contain multiple nodes. However, `mgs_group` should contain only one node: the MGS node for the Lustre file system. Calling that setting `mgs_group` (singular) instead of `mgs_groups` (plural) is intended to convey this restriction.

## Procedure

**1.** Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

```
smw# cfgset search -s cray_lustre_server p0-preupdate
```

DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

**2.** In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

**3.** Set the value of the replacement setting to the node group(s) identified or created in the previous step.

Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.11 Migrate Node Groups in cray_net

## About this task

> **NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_net does not add any new settings/fields. However, one setting has been deprecated, and one setting must be updated for eLogin nodes so that those nodes can be added to node groups.

**For informational purposes only.** The roles field in all cray_net host entries (`cray_net.settings.hosts.data.`*`host`*`.roles`) has been deprecated; the node groups settings in the cray_boot and cray_sdb services will be used instead. No change to cray_net with regard to the roles field is necessary.

> **NOTICE:** If a site-local Ansible play uses any cray_net host roles field, revise the play to use node groups instead.

*Table 29. cray_net*

All settings begin with the `cray_net.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| .hosts.data.*host*.roles | N/A |

**ACTION REQUIRED.** Prior to UP02, the hostid field in all eLogin host entries (cray_net.settings.hosts.data.*host*.hostid) was left blank (unlike CLE nodes, eLogin nodes do not have cname identifiers). For UP02 and later releases, that field in all eLogin host entries must be set to the same value as the hostname field for eLogin host entries (cray_net.settings.hosts.data.*host*.hostname), so that eLogin nodes can be added to node groups. Follow these steps to update the hostid setting.

## Procedure

1. Update the cray_net service using the configurator in interactive mode.

```
chroot-smw# cfgset update --mode interactive --service cray_net p0
```

2. Select and configure the hosts setting.

   In the configurator Service Configuration Menu, enter **2** to select the hosts settings, then at the next prompt enter **C** to configure them.

```
Service Configuration Menu (Config Set: p0, type=cle)

  cray_net         [ status: enabled ]  [ validation: valid ]
...
          1)   networks
                  name: hsn
                  name: login
...
          2)   hosts
                  common_name: bootnode
                  common_name: sdb_node
...
Cray Networking Configuration Service Menu [default: save & exit - Q] $ 2
Cray Networking Configuration Service Menu [default: configure - C] $ C
```

3. Select and modify the hostid field of the eLogin host entry.

   In the configurator setting screen, enter ***** to see all configured host entries. Find the entry for the eLogin node (for this example, assume the entry has common_name 'elogin_node' and is #3 in a list of five host entries), then enter **3c*** to modify the hostid setting, which is item 'c' in every host entry. Set the hostid value to be the same as the value for hostname, which is item 'e' in every host entry.

```
...
cray_net.settings.hosts
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ *
...
cray_net.settings.hosts
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ 3c*
...
cray_net.settings.hosts.data.elogin_node.hostid
[<cr>=set '', <new value>, ?=help, @=less] $ elogin_hostname
```

4. Save changes and exit the configurator.

After entering the value, enter **<cr>** to set the host entries and then enter **Q** to save changes and exit the configurator.

```
cray_net.settings.hosts
[<cr>=set 5 entries, +=add an entry, ?=help, @=less] $ <cr>
...
Cray Networking Configuration Service Menu [default: save & exit - Q] $ Q
```

### 4.3.3.12 Migrate Node Groups in cray_node_groups

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

Default node groups are groups of nodes that

- are likely to be customized and used by many sites
- support useful default values for many of the migrated services

Several of the default node groups require customization by a site to provide the appropriate node membership information. This table lists the Cray default groups and indicates which ones require site customization.

*Table 30. cray_node_groups*

| Default Node Group | Requires Customization? | Notes |
|---|---|---|
| compute_nodes | No | Defines all compute nodes for the given partition. The list of nodes is determined at runtime. |
| service_nodes | No | Defines all service nodes for the given partition. The list of nodes is determined at runtime. |
| smw_nodes | Yes | Add the output of the `hostid` command for the SMW. For an SMW HA system, add the host ID of the second SMW also. |
| boot_nodes | Yes | Add the cname of the boot node. If there is a failover boot node, add its cname also. |
| sdb_nodes | Yes | Add the cname of the SDB node. If there is a failover SDB node, add its cname also. |
| login_nodes | Yes | Add the names of internal login nodes on the system. |
| all_nodes | Maybe | Defines all compute nodes and service nodes on the system. Add external nodes (e.g., eLogin nodes), as needed. |
| tier2_nodes | Yes | Add the cnames of nodes that will be used as tier2 servers in the cray_scalable_services configuration. |

**Why is there no "tier1_nodes" default node group?** Cray provides a default tier2_nodes node group to support defaults in the cray_simple_shares service. Cray does not provide a tier1_nodes node group because no default data in any service requires it. Because it is likely that tier1 nodes will consist of only the boot node and the SDB node, for which node groups already exist, Cray recommends using those groups to populate the cray_scalable_services tier1_groups setting rather than defining a tier1_nodes group.

**About eLogin nodes.** To add eLogin nodes to node groups, use their 'hostname' values instead of cnames, because unlike CLE nodes, eLogin nodes do not have cname identifiers. If eLogin nodes are intended to receive configuration settings associated with the all_nodes group, add them to that group, or create a new group for eLogin nodes only (elogin_nodes), and then change the appropriate settings in other configuration services to include both all_nodes and elogin_nodes.

## Procedure

**1.** Ensure that the cray_node_groups service is enabled.

If cray_node_groups was not used in the UP01 release, it may have been disabled. This service must be enabled in order for the system to boot.

```
cray_node_groups.enabled
[<cr>=set 'true', <new value>, ?=help, @=less] $ <cr>
```

**2.** In the configurator, add nodes to the pre-defined node groups, as needed, when prompted for the members of each unset, pre-defined node group.

In this example, the configurator is prompting for the members of the smw_nodes node group. Enter **+** to add a node to this group.

```
cray_node_groups.settings.groups.smw_nodes
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
```

**3.** View all defined node groups.

When the configurator has finished prompting for unset node groups, enter `*` at this prompt to view a list of all configured node groups. The output will show at least eight entries, corresponding to the Cray default node groups; there may be more if any new node groups have been defined during the update of other services in this config set.

```
cray_node_groups.settings.groups
[<cr>=set 8 entries, +=add an entry, ?=help, @=less] $ *
```

Additional node groups can be added at this time or when configuring the new node groups settings in other configuration services as the configurator prompts for them during the update of this config set.

### 4.3.3.13 Migrate Node Groups in cray_persistent_data

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_persistent_data replaces the deprecated setting listed in the table, which is a list of nodes, with the replacement setting, which is a list of node groups.

*Table 31. cray_persistent_data*

All settings begin with the `cray_persistent_data.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.mounts.data.`*`path`*`.clients` | `.mounts.data.`*`path`*`.client_groups` |

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_persistent_data p0-pre-update
   ```

   If the previously defined values for the `clients` settings include the deprecated values in the left column of the following table, use the values in the right column for the replacement settings (`client_groups`).

   *Table 32. Map clients to client_groups*

   | Value in deprecated setting | Node group(s) to use in replacement setting |
   |---|---|
   | service | service_nodes |

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.14 Migrate Node Groups in cray_rsip

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_rsip replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 33. cray_rsip*

All settings begin with the `cray_rsip.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.service.data.servers` | `.service.data.server_groups` |

| Deprecated Setting | Replacement Setting |
|---|---|
| `.service.data.nodes_as_client` (level advanced) | `.service.data.node_groups_as_client` (level advanced) |
| `.service.data.method_exceptions` (level advanced) | `.service.data.method_exception_groups` (level advanced) |

**NOTE:** The configurator will not prompt for the settings that are marked 'level advanced' in the table. These will be set later.

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_rsip --level advanced p0-pre-update
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **v** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **!**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.15 Migrate Node Groups in cray_scalable_services

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_scalable_services replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 34. cray_scalable_services*

All settings begin with the `cray_scalable_services.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.scalable_service.tier1` | `.scalable_service.tier1_groups` |
| `.scalable_service.tier2` | `.scalable_service.tier2_groups` |

Note that there is a default node group named tier2_nodes that should be populated with the tier2 nodes and used in the `tier2_groups` setting. There is no default node group for tier1 nodes. For the `tier1_groups` setting, use the appropriate existing node groups, such as the boot_nodes and sdb_nodes groups, or create a new node group that contains all tier1 nodes.

> **IMPORTANT:** If the cray_scalable_services settings are correctly migrated to node groups for the current release, services on the node will be unchanged. However, if these settings are migrated incorrectly, problems may occur with a wide range of system functionality.

For more information, see *About Cray Scalable Services* on page 13.

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

   ```
   smw# cfgset search -s cray_scalable_services p0-preupdate
   ```

   DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

2. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

3. Set the value of the replacement setting to the node group(s) identified or created in the previous step.

   Repeat the previous step and this step for each replacement setting in the table.

### 4.3.3.16 Migrate Node Groups in cray_sdb

## About this task

> **NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_sdb introduces one new node groups setting, which must be set to the group(s) that contain the CLE SDB nodes (including the SDB failover node, if applicable).

*Table 35. cray_sdb*

All settings begin with the `cray_sdb.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| N/A | `.node_groups.data.sdb_groups` |

Update this field with the name(s) of the node group(s) that define the SDB node and the SDB failover node, if applicable. Cray recommends that sites use the pre-defined 'sdb_nodes' node group rather than create a custom node group.

## Procedure

1. In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

   When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

2. Set the value of `cray_sdb.settings.node_groups.data.sdb_groups` to the pre-defined node group `sdb_nodes`.

### 4.3.3.17 Migrate Node Groups in cray_simple_shares

## About this task

**NOTICE:** Skip this procedure if updating from SMW 8.0.UP02 / CLE 6.0.UP02 or a higher release.

The current release of cray_simple_shares replaces the deprecated settings listed in the table, which are all lists of nodes, with the replacement settings, which are all lists of node groups.

*Table 36. cray_simple_shares*

All settings begin with the `cray_simple_shares.settings` string.

| Deprecated Setting | Replacement Setting |
|---|---|
| `.NFS.data.`*`path`*`.servers` | `.NFS.data.`*`path`*`.server_groups` |
| `.NFS.data.`*`path`*`.clients` | `.NFS.data.`*`path`*`.client_groups` |
| | `.NFS.data.`*`path`*`.client_exclude_groups` |
| `.DVS.data.`*`path`*`.clients` | `.DVS.data.`*`path`*`.client_groups` |

Note that the `clients` setting of each NFS mount is replaced by two node groups settings: `client_groups` and `client_exclude_groups`. The `client_groups` setting contains the list of node groups that should mount the file system, and the `client_exclude_groups` setting contains the list of node groups that should be excluded from mounting the file system.

**Why is `client_exclude_groups` needed?** It is possible to exclude specific nodes or platforms by using the tilde character (~) when defining a node group. The `client_exclude_groups` setting was created to maintain the pre-populated data in the cray_simple_shares configurator template without requiring extra user input. There is no `client_exclude_groups` field for a DVS mount because it is not needed to support the pre-populated cray_simple_shares data, and simpler exclusion can be used when defining node groups.

## Procedure

1. Search for the previously defined value(s) of the deprecated setting(s) in the pre-update CLE configuration set(s).

   This step can be done earlier in the update process, as part of *Prepare to Migrate Node Groups Configuration Data* on page 246.

```
smw# cfgset search -s cray_simple_shares p0-preupdate
```

Use these tables to map previously defined values for the `servers` and `clients` settings to values for the replacement settings.

Map NFS servers to NFS server_groups:

| **Value in deprecated setting** | **Node group(s) to use in replacement setting** |
|---|---|
| tier1 | tier1_nodes (if this node group has been created)<br>OR<br>boot_nodes, sdb_nodes (assumes these are the tier1 servers for this system) |
| boot | boot_nodes |
| sdb | sdb_nodes |

Map NFS clients to NFS client_groups and client_exclude_groups:

| **Value in deprecated setting** | **Node group(s) to use in `client_groups`** | **Node group(s) to use in `client_exclude_groups`** |
|---|---|---|
| compute | compute_nodes | N/A |
| service | service_nodes | N/A |
| tier2 | tier2_nodes | N/A |
| boot | boot_nodes | N/A |
| sdb | sdb_nodes | N/A |
| !boot | N/A | boot_nodes |
| !sdb | N/A | sdb_nodes |

Map DVS clients to DVS client_groups:

| **Value in deprecated setting** | **Node group(s) to use in replacement setting** |
|---|---|
| all | all_nodes |
| compute | compute_nodes |
| service | service_nodes |

DO THE REMAINING STEPS ONLY WHILE UPDATING THE NEW CONFIG SET

**2.** In the configurator, view existing node groups and use, modify, or create node groups, as needed, that contain the correct nodes.

When prompted for a replacement setting, enter **`v`** to view existing node groups. If none of the existing node groups are appropriate for this setting, enter **`!`**, which temporarily switches context into the `cray_node_groups.settings.groups` setting to add or modify a node group.

**3.** Set the value of the replacement setting to the node group(s) identified or created in the previous step.

Repeat the previous step and this step for each replacement setting in the table.

## 4.3.4 Display All Config Set Information

### Prerequisites

This procedure assumes that the administrator is still working within the chrooted release snapshot.

### About this task

This procedure is not required, but it may aid in troubleshooting. It displays all of the configuration settings and writes them to a file for the CLE settings, a file for the global settings, and the typescript file started at the beginning of this session.

### Procedure

1. Display the CLE config set (p0 in this example).

   Repeat this step for each CLE config set that will be used to boot the system.

   ```
   chroot-smw# cfgset search -l advanced p0 | tee \
   /var/adm/cray/release/p0.${TODAY}.update.advanced.conf
   ```

2. Display the global config set.

   ```
   chroot-smw# cfgset search -l advanced global | tee \
   /var/adm/cray/release/global.${TODAY}.update.advanced.conf
   ```

## 4.3.5 End Configuration in Release Snapshot

### Prerequisites

This procedure assumes that configuration within the chrooted release snaphot environment is complete.

### Procedure

Exit the chroot snapshot.

```
chroot-smw# exit
smw#
```

Note that the prompt has changed from `chroot-smw#` to `smw#`.

## 4.4 Update Programming Environment (PE) Software

### Prerequisites

Cray Programming Environment (PE) software should be updated with the PE Installer.

### About this task

The same PE image can be used for several of the monthly releases of PE software, but a fresh image must be created and used with each new CLE release. This procedure creates a fresh PE image for this CLE software release, and then updates PE software content and makes it available on compute and login nodes.

Note that although the PE image name has 'compute' in it, the same image is also used for login nodes.

### Procedure

**1.** Change into the newly created release snapshot.

```
smw# snaputil chroot $SNAPSHOT
chroot-smw#
```

**2.** Set an environment variable for the PE image name.

The old name for the PE image set in the Cray Image Binding Service of the CLE config set will need to be changed to this new name.

```
chroot-smw# export PEIMAGE=pe_compute_cle_6.0up03_sles_12
chroot-smw# echo $PEIMAGE
```

**3.** Create fresh PE image root on the SMW for this software release.

a. Get the name of the PE image recipe on the system.

```
chroot-smw# recipe list | grep ^pe
pe_image_cle_6.0up03_sles_12
```

b. Create `$PEIMAGE` image.

Substitute for *<pe recipe name>* the name of the PE image recipe found in step b.

```
chroot-smw# image create -r <pe recipe name> $PEIMAGE
```

c. Update the image name in the config set for `cray_image_binding.settings.profiles.data.PE.image`.

The name of the PE image (`$PEIMAGE`) should be updated in the PE profile of the cray_image_binding service for the CLE config set.

**full system** For a full, unpartitioned system:

```
chroot-smw# cfgset update -s cray_image_binding -m interactive p0
```

**partitioned** For a partitioned system:

```
chroot-smw# cfgset update -s cray_image_binding -m interactive pN
```

d. Validate the config set.

**full system** For a full, unpartitioned system:

```
chroot-smw# cfgset validate p0
```

**partitioned** For a partitioned system:

```
chroot-smw# cfgset validate pN
```

e. Update the `IMAGE_DIRECTORIES` field in the installer configuration file (`/var/adm/cray/release/pe/install-cdt.yaml`).

If PE is to be installed in the new PE image only, update `IMAGE DIRECTORIES` as follows:

```
 IMAGE_DIRECTORIES :
   - /var/opt/cray/imps/image_roots/pe_compute_cle_6.0up03_sles_12
```

If PE is to be installed in both the old and new images, update `IMAGE DIRECTORIES` as follows:

```
 IMAGE_DIRECTORIES :
   - /var/opt/cray/imps/image_roots/pe_compute_cle_6.0up02_sles_12
   - /var/opt/cray/imps/image_roots/pe_compute_cle_6.0up03_sles_12
```

**4.** Exit from the release snapshot.

```
chroot-smw# exit
smw#
```

**5.** Copy the most recent PE ISO to the SMW and mount the ISO.

```
smw# mkdir -p /var/adm/cray/release/pe
smw# cd /var/adm/cray/release/pe
smw# mkdir -p /var/adm/cray/release/pe/mount_iso
smw# mount -o loop,ro <downloaded PE ISO> /var/adm/cray/release/pe/mount_iso
```

**6.** Update the `craype-installer` RPM on the SMW, from the PE ISO.

```
smw# rpm -Uvh /var/adm/cray/release/pe/mount_iso/\
installer/craype-installer-*.x86_64.rpm
```

**7.** Install PE software from the most recent PE installation media and installer.

a. Run the PE installer.

```
smw# module load craype-installer
smw# craype-installer.pl --install --install-yaml-path ./install-cdt.yaml
```

When the installation completes, the following output will be shown, summarizing the installed packages.

```
1) atp-1.7.5-0_3605.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
2) cray-ccdb-1.0.3-0_3575.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
3) cray-dwarf-14.2.0-0.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
 <snip>
71) perftools-clients-6.2.2-1.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
```

b. Set the default versions for PE with `set_default` scripts, if the above install succeeds.

```
smw# craype-installer.pl --set-default --install-yaml-path ./install-cdt.yaml
```

c. Unmount the ISO.

```
smw# umount ./mount_iso
```

d. Clean up the PE ISO and PE RPMs.

These RPMs are large and use up disk space, so they can be removed.

```
smw# rm *.iso *.rpm *.tar.gz
```

**8.** Install as many older monthly PE releases to this UP03 PE image root as desired.

For each of the older monthly PE release ISOs, do the following steps to install them to the new `$PEIMAGE` image root.

a. Mount the PE ISO.

```
smw# mount -o loop,ro <downloaded PE ISO> /var/adm/cray/release/pe/mount_iso
```

b. Install PE software from the most recent PE installation media and installer.

Run the PE installer. This will install the older PE software release to the new $PEIMAGE image root.

```
smw# craype-installer.pl --install --install-yaml-path ./install-cdt.yaml
```

When the installation completes, the following output will be shown, summarizing the installed packages.

```
1) atp-1.7.5-0_3605.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
2) cray-ccdb-1.0.3-0_3575.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
3) cray-dwarf-14.2.0-0.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
<snip>
71) perftools-clients-6.2.2-1.x86_64 (/var/opt/cray/imps/image_roots/
pe_compute_cle_6.0up03_sles_12_x86-64_ari)
```

c. If desired, set this older PE release as the default version.

   If this version of PE should be the default and not the most recent version of PE software installed earlier, then set it to default with this command.

   ```
   smw# craype-installer.pl --set-default --install-yaml-path ./install-cdt.yaml
   ```

d. Unmount the ISO.

   ```
   smw# umount ./mount_iso
   ```

e. Clean up the PE ISO and PE RPMs.

   These RPMs are large and use up disk space, so they can be removed.

   ```
   smw# rm *.iso *.rpm *.tar.gz
   ```

**9.** Push the PE image root to the boot node.

For p0:

```
smw# image push -d boot $PEIMAGE
INFO - Remotely cloning Image '<name of image>' to 'boot'...
INFO - Checking remote destination...
INFO - Passwordless SSH not established; prompting for password for root@boot:
Password:
INFO - Transferring Image '<name of image>' to 'root@boot:/var/opt/cray/imps/
image_roots/<name of image>'...
Password:
INFO - Cloned Image '<name of image>' to remote host 'root@boot:/var/opt/cray/
imps/image_roots/<name of image>'.
```

For partitioned systems, push to the boot node for that partition, `boot-p1`:

```
smw# image push -d boot-p1 $PEIMAGE
```

**Trouble?** Once the new PE image starts being pushed, PE may be in an inconsistent state and users could see errors until the `cray_image_binding` service is restarted and users are able to log in again. Do the following steps if errors are reported during the image push.

a. Remove all files from the PE image directory `/var/opt/cray/imps/image_roots/$PEIMAGE` on the boot node.

   ```
   boot# export PEIMAGE=pe_compute_cle_6.0up03_sles_12
   boot# rm -rf /var/opt/cray/imps/image_roots/$PEIMAGE/*
   ```

b. Run the image push on the SMW again.

**10.** Back up the CLE and global config sets post PE installation.

```
smw# cfgset create --clone global global-postpe-${TODAY}
```

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postpe-${TODAY}
```

# 4.5 Boot SMW to Newly Installed Release

These procedures push image roots to the boot node, if needed, update the diags bind mount profile, shut down the CLE system, then boot the SMW to the newly installed and configured release snapshot.

1. *Push New Netroot and Diag Image Roots to Boot Node*
2. *Update the Diags Bind Mount Profile* on page 289
3. *Shut Down the CLE System*
4. *Set Default Snapshot and Boot the SMW*

## 4.5.1 Push New Netroot and Diag Image Roots to Boot Node

### Prerequisites

This procedure assumes the following:

- The system is set up to create Netroot compute or Netroot login image roots and/or the diagnostics (diag) image root when `SMWinstall` is run.
- The CLE system is currently booted (the boot node must be up).

### About this task

This procedure pushes the new Netroot compute or Netroot login image roots and the diag image root to the boot node. Pushing image roots now saves time later in the process by avoiding a staged boot for image pushes.

### Procedure

1. Push the Netroot image root for compute nodes.

   Find the Netroot compute image root created during this session, set the NETROOT_COMPUTE environment variable, and use it to push the Netroot compute image root to the boot node.

   Note that the name of the Netroot compute image root may be different for this site. Use the correct image root name for this site.

   ```
   smw# export NETROOT_COMPUTE=$(basename `ls -d \
   /var/opt/cray/imps/image_roots/compute-large*${TODAY}*`)

   smw# echo $NETROOT_COMPUTE

   smw# image push -d boot $NETROOT_COMPUTE
   ```

2. Push the Netroot image root for login nodes.

   Find the Netroot login image root created during this session, set the NETROOT_LOGIN environment variable, and use it to push the Netroot login image root to the boot node.

Note that the name of the Netroot login image root may be different for this site. Use the correct image root name for this site.

```
smw# export NETROOT_LOGIN=$(basename `ls -d \
/var/opt/cray/imps/image_roots/login-large*${TODAY}*`)

smw# echo $NETROOT_LOGIN

smw# image push -d boot $NETROOT_LOGIN
```

3. Push the diag image root for login nodes.

   In this example, the diag image root is diag-all_cle_60up03_sles_12_x86-64.

   ```
   smw# image push -d boot diag-all_cle_60up03_sles_12_x86-64
   ```

## 4.5.2 Update the Diags Bind Mount Profile

### Prerequisites

This procedure assumes that diag image root has been pushed to the boot node.

### About this task

The online diagnostics image provides some useful tools that are made available on CLE nodes through the Cray Image Binding service using the profile for the diag image root. This procedure describes how to set the correct diags image root and enable the diags profile.

### Procedure

1. Update cray_image_binding, which is in the CLE config set (p0 in this example).

   ```
   smw# cfgset update -s cray_image_binding -m interactive p0
   ```

   The configurator displays the **Service Configuration Menu**. The service name and status appear at the top of the menu. That menu also includes a list of settings. The Cray Image Binding service has a single setting: `profiles`. Under it is a list of bind mount profile entries.

2. If this service is not yet enabled, enable it now.

   This example shows the service as disabled. Enter **E** to enable it.

   ```
   Service Configuration Menu (Config Set: p0, type: cle)
     cray_image_binding       [status: disabled] [validation: skipped]
   ...
   IMPS Image Binding Service Menu [default: save & exit - Q] $ E
   ```

3. Select the `profiles` setting to configure it.

   Enter **1** to select the `profiles` setting, and then enter **C** to configure that setting.

```
IMPS Image Binding Service Menu [default: save & exit - Q] $ 1
...
IMPS Image Binding Service Menu [default: save & exit - Q] $ C
```

The configurator displays guidance about the `profiles` setting and a numbered list of profile entries that have already been added. A 'PE' profile and a 'diags' profile should be in that list.

**4.** Change the value of the 'diag' profile image field.

a. Enter the number for the 'diag' profile followed by 'a' and '*' to select and edit the field for the diag profile image name.

In this example, the number of the 'diag' profile is 2.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 2a*
```

b. Enter the name of the diag image.

In this example, the diag image root is diags_cle_6.0up03_sles_12_x86-64. This must be the same as the image root that was pushed to the boot node.

```
cray_image_binding.settings.profiles.data.diags.image
[<cr>=keep 'diags_cle_6.0up03_sles_12_x86-64', <new value>, ?=help, @=less]
$ <cr>
```

**5.** Enable the diags profile.

Enable the profile only after the diag image root has been pushed to the boot node.

a. Enter the number for the diags profile followed by 'd' and '*' to select the field for enabling the diags profile.

In this example, the number of the diags profile is 2.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 2d*
```

b. Enter `true`, then press **Enter**.

```
cray_image_binding.settings.profiles.data.diags.enabled
[<cr>=keep 'false', <new value>, ?=help, @=less] $ true
```

**6.** Set the profile entries, and then save changes and exit the configurator.

```
cray_image_binding.settings.profiles
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
...

IMPS Image Binding Service Menu [default: save & exit - Q] $ Q
```

## 4.5.3 Shut Down the CLE System

### About this task

To shut down the CLE system, if it is booted, use the shutdown automation file to shut it down gracefully.

### Procedure

1. Check whether the boot node is up.

   **full system** For a full, unpartitioned system:

   ```
   smw# ping -c3 boot
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# ping -c3 boot-p1
   smw# ping -c3 boot-p2
   ```

2. If the boot node is up, then shut down the CLE system.

   **full system** For a full, unpartitioned system:

   ```
   smw# su - crayadm
   crayadm@smw> xtbootsys -s last -a auto.xtshutdown
   crayadm@smw> exit
   smw#
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# su - crayadm
   crayadm@smw> xtbootsys --partition p1 -s last -a auto.xtshutdown
   crayadm@smw> xtbootsys --partition p2 -s last -a auto.xtshutdown
   crayadm@smw> exit
   smw#
   ```

## 4.5.4 Set Default Snapshot and Boot the SMW during a Software Update

### Prerequisites

This procedure assumes that the snapshot variable has been set and the SMW and CLE software has been installed.

### About this task

This procedure uses `snaputil` to ensure that the correct snapshot is used to boot the SMW.

### Procedure

1. Set the release snapshot as the default.

**IMPORTANT:** Do not skip this step. If the SMW is rebooted without first setting the default snapshot, the SMW becomes unbootable.

```
smw# snaputil default ${SNAPSHOT}
```

2. Verify that the correct snapshot is the default.

```
smw# snaputil list
```

3. Reboot the SMW to switch to the new release.

```
smw# reboot
```

**ATTENTION:** After the SMW has completed booting, wait 10 to 15 minutes before logging in, which allows time for all startup tasks to complete. Logging in too soon can result in SSH connections being dropped.

## 4.6 Configure SMW for CLE System Hardware during a Software Update

In this part of the software update process, use these procedures to discover hardware, update firmware, update and validate config sets, and check the status of all SMW components.

1. *Start a Typescript File* on page 78
2. *Make a Post-install Snapshot using snaputil* on page 98
3. *Make a Post-install Backup of Current Global and CLE Config Sets* on page 294
4. *Compare Previous Snapshot to Current Snapshot* on page 294
5. *Discover Cray Hardware* on page 295
6. *Update Firmware* on page 111
7. *Update Config Sets* on page 299
8. *Validate Config Sets* on page 184
9. *Finish Configuring the SMW for the CLE System Hardware* on page 114

### 4.6.1 Start a Typescript File

### About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file at these milestones:

- just before installing a new software release
- just before configuring the newly installed software

## Procedure

1. Log in as root to the SMW.

2. (First time only) Create a release directory for the typescript file.

```
smw# mkdir -p /var/adm/cray/release
```

3. Change to the release directory.

```
smw# cd /var/adm/cray/release
```

4. Set a variable equal to today's date.

```
smw# export TODAY=`date +%Y%m%d`
smw# echo $TODAY
```

5. Start a typescript file.

```
smw# script -af ${TODAY}.suffix
```

   For *suffix*, substitute a unique string to distinguish among typescript files, such as `install.1` or `update.2`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

## 4.6.2 Make a Post-install Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

### Procedure

1. List the available snapshots on the system.

```
smw# snaputil list
```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

```
smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
smw# echo $SNAPSHOT
```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

**3.** Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postinstall
```

## 4.6.3 Make a Post-install Backup of Current Global and CLE Config Sets

### About this task

This procedure uses the `cfgset` command to create a post-install backup of the global and CLE config sets after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware.

### Procedure

**1.** Back up the current global config set.

```
smw# cfgset create --clone global global-postinstall-${TODAY}
```

**2.** Back up the current CLE config set.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postinstall-${TODAY}
```

## 4.6.4 Compare Previous Snapshot to Current Snapshot

### About this task

This optional procedure compares a previous snapshot to the current one to see whether any files were changed before the system was rebooted to the current snapshot, as when performing a software update.

### Procedure

**1.** List the available snapshots on the system.

```
smw# snaputil list
```

**2.** Compare a previous snapshot to the current one.

This example uses the pre-update archival snapshot as the previous one to be compared to the current snapshot.

```
smw# snaputil diff ${SNAPSHOT}.preupdate ${SNAPSHOT}
```

3. If a file has been changed, check for differences in the contents of the text file.

```
smw# snaputil diff ${SNAPSHOT}.preupdate ${SNAPSHOT} filename
```

## 4.6.5 Discover Cray Hardware

### About this task

**About Hardware Discovery**. This procedure uses `xtdiscover` to ensure that any changes made to the HSS database schema for new features are captured. To display the configuration, use the `xtcli` command after running `xtdiscover`. For more detailed information, see the `xtdiscover(8)` man page.

**About STONITH.** This procedure prepares STONITH (shoot the other node in the head), a Linux service that automatically powers down a node that is not working correctly. If either boot node failover or SDB node failover will be used, then STONITH needs to be set on the primary blade.

> **IMPORTANT:** The primary boot node and primary SDB node should not be on the same blade. Likewise the secondary boot node and secondary SDB node should not be on the same blade. Four different blades should be used if there are two boot nodes and two SDB nodes.

**Trouble?** If a step in this procedure fails because of a hardware issue, such as a cabinet failing to power up, resolve that issue and then go back to the last successful step in the procedure and continue from there. Do not skip steps or continue out of order.

### Procedure

1. Power down the system.

```
smw# xtcli power down s0
Turning off power to cabinet and waiting for confirmation...

           Component|       Flags:    Result
---------------------------------------------------------------------
                c0-0|    noflags|:    Success
            c0-0c0s0|    noflags|:    Success
            c0-0c0s1|    noflags|:    Success
            c0-0c0s2|    noflags|:    Success
            c0-0c0s3|    noflags|:    Success
```

2. Reboot the cabinet controllers (CC), then verify that all CCs are up.

   a. Reboot the cabinet controllers.

```
smw# xtccreboot -c all
xtccreboot: reboot sent to specified CCs
smw# sleep 180
```

   b. Are all cabinet controllers up now? Repeat this command until all of the cabinet controllers report in.

```
smw# xtalive -a l1sysd -l l1 s0
The expected responses were received.
```

3. Power up the system, then verify the blades are powered on.

a. Power up the system.

```
smw# xtcli power up s0
Turning on power to cabinet and waiting for confirmation...

           Component|        Flags:    Result
-------------------------------------------------------------------
                c0-0|    noflags|:    Success
            c0-0c0s0|    noflags|:    Success
            c0-0c0s1|    noflags|:    Success
            c0-0c0s2|    noflags|:    Success
            c0-0c0s3|    noflags|:    Success
```

b. Verify the blades are powered on and the necessary daemons are responding.

**full system**

For a full system:

```
smw# sleep 60
smw# xtalive
```

**partitioned**

For a partitioned system:

```
smw# sleep 60
smw# xtalive --partition p1
smw# xtalive --partition p2
```

Note that at this point the `xtcli status` output shows that all nodes are "off" because they have not yet been bounced.

DISCOVER CRAY SYSTEM HARDWARE

**4.** Run the `xtdiscover` command.

`xtdiscover` may pause with instructions to bounce the system.

```
smw# xtdiscover
*********** xtdiscover started ***********
...

In a separate window, please bounce the system now to continue discovery.
```

**5.** If prompted, bounce the system (as `crayadm`) in a separate window.

```
crayadm@smw> /opt/cray/hss/default/etc/xtdiscover-bounce-cmd
```

**6.** After the `xtbounce` command from the previous step has finished, return to the `xtdiscover` window and enter "c" to continue the hardware discovery.

```
After bounce completes, enter 'c' to complete discovery
or 'q' or 'a' to abort [c]: c
```

**7.** Commit the results of `xtbounce` to the database.

When asked whether to commit the `xtdiscover` results to the database, enter **`y`**.

If `xtdiscover` reports that it saved configuration changes in this file, run this command to show what differences were detected:

```
smw# cat /opt/cray/hss/default/etc/xtdiscover-config-changes.diff
```

(optional) PREPARE STONITH FOR BOOT NODE AND SDB NODE FAILOVER

**8.** For sites using boot node failover, set STONITH for the primary boot node's blade.

Skip this step if there will be no boot node failover at this site.

In the example, the primary boot node is c0-0c0s0n1, so its blade is c0-0c0s0.

```
smw# xtdaemonconfig c0-0c0s0 stonith=true
```

**9.** For sites using SDB failover, set STONITH for primary SDB node's blade.

Skip this step if there will be no SDB node failover at this site.

In the example, the primary SDB node is c0-0c2s0n1, so its blade is c0-0c2s0.

```
smw# xtdaemonconfig c0-0c2s0 stonith=true
```

DISCOVER HSN ROUTING CONFIGURATION

**10.** Discover the routing configuration of the high-speed network (HSN).

After `xtdiscover` finishes, run the `rtr` command as crayadm to determine the exact configuration of the HSN.

a. Switch to crayadm.

```
smw# su crayadm

crayadm@smw> PS1="\u@\h:\w \t> "
```

b. Run the `rtr` command.

**full system** For a full system:

```
crayadm@smw> rtr --discover
```

If the system was not bounced previously, this message may be displayed:

```
System was not bounced in diagnostic mode, should I re-bounce?
Continue (y/n)?
```

If so, enter **y**.

**partitioned** For a partitioned system, first disable the partitions, run `rtr` for the full system, and then enable the partitions again. This is most important when `xtdiscover` has identified a hardware change.

```
crayadm@smw> xtcli part_cfg deactivate p1
crayadm@smw> xtcli part_cfg deactivate p2
crayadm@smw> xtcli part_cfg activate p0

crayadm@smw> rtr --discover

crayadm@smw> xtcli part_cfg deactivate p0
crayadm@smw> xtcli part_cfg activate p1
crayadm@smw> xtcli part_cfg activate p2
```

### 4.6.6 Update Firmware

## Prerequisites

This procedure assumes that Cray hardware discovery has been completed successfully.

## About this task

This procedure first checks whether the firmware of these components (controllers) needs to be updated, then updates the firmware only if there are Revision Mismatches.

**all cabinet-level components**

cc_mc (CC Microcontroller)
cc_bios (CC Tolapai BIOS)
cc_fpga (CC FPGA)
chia_fpga (CHIA FPGA)

**all blade-level components**

cbb_mc (CBB BC Microcontroller)
ibb_mc (IBB BC Microcontroller)
anc_mc (ANC BC Microcontroller)
bc_bios (BC Tolapai BIOS)
lod_fpga (LOD FPGA)
node_bios (Node BIOS)
loc_fpga (LOC FPGA)
qloc_fpga (QLOC FPGA)

## Procedure

1. Check firmware.

   Check whether any firmware needs to be updated on the various controllers.

   ```
   crayadm@smw> xtzap -r -v s0
   ```

   If the firmware on any controllers is out of date, the output looks like this, and the firmware needs to be updated (reflashed).

   ```
   Individual Revision Mismatches:

   Type       ID                Expected   Installed
   ---------- ----------------- ---------- ----------------------------------------
   cc_bios    c0-0              0013       0012
   bc_bios    c0-0c0s0          0013       0012
   bc_bios    c0-0c0s1          0013       0012
   bc_bios    c0-0c0s2          0013       0012
   bc_bios    c0-0c0s3          0013       0012
   ```

2. Update firmware, if any components are not current.

   ⚠ **CAUTION:** The `xtzap` command is normally intended for use by Cray Service personnel only. Improper use of this restricted command can cause serious damage to the computer system.

   Run `xtzap -a` to update all components.

```
crayadm@smw> xtzap -a s0
```

Note that it is possible to update firmware in cabinets or blades only rather than the entire system. For more information, see *XC™ Series System Administration Guide* (S-2393).

3. Run `xtbounce --linktune` if any components were not current.

   Force `xtbounce` to do a `linktune` on the full system before checking firmware again.

   **full system** For a full, unpartitioned system or a fresh install:

   ```
   crayadm@smw> xtbounce --linktune=all s0
   ```

   **partitioned** For a partitioned system (software update process only, because partition instructions come later in the fresh install process):

   ```
   crayadm@smw> xtbounce --linktune=all p1
   crayadm@smw> xtbounce --linktune=all p2
   crayadm@smw> xtbounce --linktune=all p3
   ```

4. Confirm that all components with out-of-date firmware have been updated.

   Check firmware again after updating and linktuning those components.

   ```
   crayadm@smw> xtzap -r -v s0
   ```

## 4.6.7 Update Config Sets

### About this task

It is necessary to update all config sets at serveral points in the fresh install or software update process, such as after hardware discovery. If any nodes or blades were enabled or added prior to running `xtdiscover` and the config sets are not updated afterward, then the system `/etc/hosts` files will not have entries generated for the respective nodes and the nodes will not boot (the boot error will indicate "not in any tier" in an ansible failure). The update ensures that pre- and post-configuration scripts have been properly executed for the global and CLE config sets.

### Procedure

Update the CLE and global config sets.

**full system** Update the CLE and global config sets for a full, unpartitioned system:

```
smw# cfgset update p0
smw# cfgset update global
```

**partitioned** Update the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfgset update p1
smw# cfgset update p2
```

```
smw# cfgset update global
```

## 4.6.8 Validate Config Sets

### About this task

It is important to validate any config set that has been modified, because there is currently no mechanism to prevent the system from trying to use an invalid config set. Validation is useful for determining if the config set is minimally viable for use with the system it is intended to configure.

> **IMPORTANT:** Validation ensures that a config set passes all rules stored on the system. A validated config set does not necessarily equate to a config set with configuration data that will result in a properly configured system.

When validating a config set, the configurator checks the following:

- Config set has the proper directory structure and permissions.
- All configuration templates have correct YAML syntax.
- All configuration templates adhere to the configurator schema.
- All fields of type `lookup` reference values and settings that exist in the available configuration services.
- All level `required` fields in enabled services are configured (i.e., their state is `set`).
- Pre-configuration and post-configuration callback scripts ran successfully during the latest config set update.
- `cfgset validate` has run all validation rules installed on the system.

For more information on how `lookup` fields work, see the "Advanced: Lookup" section in "Configurator Data Types and How to Set Them," which is in *XC™ Series Configurator User Guide* (S-2560). For more information about validation rules, see "Validate a Config Set and List Validation Rules," also in that publication.

### Procedure

Validate the CLE and global config sets.

These example commands use CLE config sets named p0, p1, and p2. Substitute the correct config set names for this site.

**full system** Validate the CLE and global config sets for a full, unpartitioned system:

```
smw# cfgset validate global

smw# cfgset validate p0
```

**partitioned** Validate the CLE and global config sets for a partitioned system, with partitions p1 and p2:

```
smw# cfgset validate global

smw# cfgset validate p1
smw# cfgset validate p2
```

### 4.6.9 Finish Configuring the SMW for the CLE System Hardware

#### Prerequisites

This procedure assumes that Cray hardware has been discovered and component firmware has been updated (if needed).

#### About this task

This procedure contains the final steps of configuring the SMW for the CLE system hardware. Note that a full system is referred to as "s0" here. The term "p0" could have been used, because in this context, the two terms are interchangeable. In contrast, commands that operate on config sets use only the term "p0" when referring to a full system. In the config set context, the terms are not interchangeable.

#### Procedure

**1.** Check status on all components.

**full system** For a full, unpartitioned system:

```
crayadm@smw> xtcli status s0
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xtcli status p1
```

```
crayadm@smw> xtcli status p2
```

**2.** Check routing configuration of the system.

**full system** For a full, unpartitioned system:

```
crayadm@smw> rtr -R s0
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> rtr -R p1
```

```
crayadm@smw> rtr -R p2
```

Note that the `rtr -R` command produces no output unless there is a routing problem.

**3.** Examine the hardware inventory and verify that all nodes are visible to the SMW.

**full system** For a full, unpartitioned system:

```
crayadm@smw> xthwinv s0 > xthwinv.out
```

```
crayadm@smw> xthwinv -x s0 > xthwinv.xml
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xthwinv p1 > xthwinv.p1.out
crayadm@smw> xthwinv -x p1 > xthwinv.p1.xml
```

```
crayadm@smw> xthwinv p2 > xthwinv.p2.out
crayadm@smw> xthwinv -x p2 > xthwinv.p2.xml
```

4. Check microcontroller information.

   Execute the `xtmcinfo -u` command to retrieve microcontroller information from cabinet control processors and blade control processors. Ensure that all blade controllers have output and show similar uptime values.

   ```
   crayadm@smw> xtmcinfo -u s0
   ```

5. Exit from crayadm back to root account.

   ```
   crayadm@smw> exit
   smw#
   ```

# 4.7 Install Patches

## About this task

This procedure finds, downloads, and installs patches for a Cray XC Series system. This is done just prior to booting the CLE system.

System administrators that prefer to boot the CLE system first and perform post-boot tests before installing patches may defer this procedure until after the first system boot, unless the release notes indicate that one or more patches are required for a successful boot.

## Procedure

1. Check CrayPort for patches released by Cray.

2. Make a directory on the SMW (if it does not already exist) to hold any patches that may be available on CrayPort.

   ```
   smw# mkdir -p /var/adm/cray/release/patchsets
   ```

3. Download SMW and CLE patches to the patchset directory on the SMW, as described in the release notes.

4. Install SMW and CLE patches.

   - To install a single patch, follow the instructions provided in the patch README file.
   - To load and install multiple patches, complete the following substeps. When installing more than one patch, Cray recommends postponing the building and mapping of images until the last patch is installed.

     **NOTE:** (SMW HA only) Make a note of all patch sets that will be applied on the first SMW. The second SMW must have exactly the same patch sets.

   a. Temporarily suppress building and mapping images.

```
smw# export PATCHSET_BUILD_IMAGES=false
smw# echo $PATCHSET_BUILD_IMAGES

smw# export PATCHSET_NIMS_TIMING=deferred
smw# echo $PATCHSET_NIMS_TIMING
```

b. Follow all of the instructions in the patch README files.

These instructions will include running the LOAD script and the INSTALL script for each patch, and there may be additional steps for some patches, such as running `xtzap` again to update firmware from an SMW patch.

**5.** Build and map new images.

```
smw# imgbuilder --map
```

## 4.8 Boot the CLE System during a Software Update

The SMW/CLE software update process is nearly complete. These procedures boot the CLE system and perform post-boot activities such as running tests and making a snapshot.

**1.** *Check NIMS Information during a Software Update* on page 303

**2.** *Boot the System during a Software Update* on page 305

**3.** Perform post-boot activities.

**a.** *Run Tests after Boot is Complete*

**b.** *Test xtdumpsys and cdump*

**c.** *Make a Post-boot Snapshot using snaputil* on page 204

**d.** *Make a Post-boot Backup of Current Global and CLE Config Sets* on page 205

**4.** If the installation of patches was deferred until after the first system boot, install patches now, and then repeat steps 2 and 3.

*Install Patches* on page 302

### 4.8.1 Check NIMS Information during a Software Update

### About this task

This procedure lists NIMS (Node Image Mapping Service) information: which maps are active on the SMW and what NIMS information is stored for each node.

### Procedure

**1.** Check active NIMS maps.

**full system** For a full, unpartitioned system:

```
smw# cmap list
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
smw# cmap list --partition p1
```

```
smw# cmap list -p p2
```

2. Set a map to be active.

   If a new NIMS map has been created as part of the software update process, ensure that the new map is active.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap setactive map_name
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cmap setactive p1_map_name -p p1
   ```

   ```
   smw# cmap setactive p2_map_name -p p2
   ```

3. Check the default config set of the active NIMS map.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap list --fields default_config_set map_name
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cmap list --fields default_config_set --partition p1 p1_map_name
   ```

   ```
   smw# cmap list --fields default_config_set --partition p2 p2_map_name
   ```

   If this is not the desired default config set, use *Set Default Config Set for a NIMS Map* on page 338 to change it. If selected nodes need to use a different config set, see *Set Config Set for a Node* on page 338.

4. Check NIMS information for each node.

   **full system** For a full, unpartitioned system:

   ```
   smw# cnode list
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cnode list -p p1
   ```

   ```
   smw# cnode list -p p2
   ```

5. Check NIMS information for each NIMS group.

   ```
   smw# cnode list --filter group=admin
   smw# cnode list --filter group=service
   smw# cnode list --filter group=login
   smw# cnode list --filter group=compute
   ```

Check any additional NIMS groups that may have been created for Netroot compute and login nodes (typically only when Netroot is used on only a subset of compute and login nodes instead of all of them, so the NIMS compute and login groups cannot be used for that subset).

```
smw# cnode list --filter group=compute_netroot
smw# cnode list --filter group=login_netroot
```

Check any additional NIMS groups that may have been created for DataWarp with Fusion IO SSDs.

```
smw# cnode list --filter group=fio-service
```

Check any additional NIMS groups that may have been created with WLM (workload manager) or other site names.

```
smw# cnode list --filter group=wlm-admin
smw# cnode list --filter group=wlm-service
smw# cnode list --filter group=wlm-login
```

## 4.8.2 Boot the System during a Software Update

### Prerequisites

This procedure assumes that configuration and image preparation are complete and the system is now ready to boot.

### About this task

This procedure describes how to update site-specific boot and shutdown automation files and then use the boot automation file to boot the CLE system with `xtbootsys`.

### Procedure

**1.** Update any site boot and shutdown automation files.

If this site has site-specific boot and shutdown automation files, compare their contents to the newly distributed `auto.generic` and `auto.xtshutdown` files, and then edit the site files to merge in any new content.

```
smw# diff /opt/cray/hss/default/etc/auto.generic \
/opt/cray/hss/default/etc/auto.hostname.start
```

```
smw# diff /opt/cray/hss/default/etc/auto.xtshutdown \
/opt/cray/hss/default/etc/auto.hostname.stop
```

**2.** (New with CLE 6.0.UP03) If boot or SDB node failover is used, ensure that the `xtfailover_halt` command is included and enabled in the shutdown automation file (`auto.`*`hostname`*`.stop`).

If not added already as a result of the comparison in step 1, add the following lines to `auto.`*`hostname`*`.stop`. Ensure that the second line is uncommented. The `xtfailover_halt` command must be enabled so that the `xtbootsys` shutdown process sends a STOP NMI to the failover nodes.

```
# Enable the following line if boot or sdb failover is enabled:
lappend actions { crms_exec \
"/opt/cray/hss/default/bin/xtfailover_halt --partition $data(partition,given) --shutdown" }
```

3. Run `xtbootsys` with `auto.`*`hostname`*`.start`.

   **full system** For a full, unpartitioned system:

   ```
   smw# su - crayadm
   crayadm@smw> xtbootsys -a auto.hostname.start
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# su - crayadm
   ```

   ```
   crayadm@smw> xtbootsys -p p1 -a auto.hostname.start.p1
   ```

   ```
   crayadm@smw> xtbootsys -p p2 -a auto.hostname.start.p2
   ```

   The boot automation files have different names in this example to indicate that partitions may use different automation files, if needed.

   **Trouble?** If there are any problems booting CLE, see the *XC™ Series Boot Troubleshooting Guide* (S-2565) for techniques to determine what might be causing the problem.

## 4.8.3 Run Tests after Boot is Complete

### Prerequisites

This procedure assumes the following:

- The system has completed booting.
- The compute nodes are “interactive," not under workload manager (WLM) control.
- ALPS is available.

If ALPS is not available and Slurm is used as the WLM, then the compute nodes can be either "interactive" or "batch," and `srun` (the equivalent Slurm command) should be used instead of the `aprun` commands in the steps that follow.

### About this task

Log in to the login node as crayadm. This can be done from the SMW to the boot node to the login node or directly from another computer to the login node without passing through the SMW and boot node. Then perform these rudimentary functionality checks.

### Procedure

1. Run `apstat` to get the number of nodes to use for the following commands.

   ```
   crayadm@login> NUMNODES=$(($(apstat -v | grep XT | awk "{print \$3}")))
   crayadm@login> echo NUMNODES is $NUMNODES
   ```

2. Verify that all nodes run (from `/tmp`).

```
crayadm@login> cd /tmp
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

3. Verify that the home directory is working by running a job.

```
crayadm@login> cd ~
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

4. Verify that the Lustre directory is working by running a job.

```
crayadm@login> cd /lustre_file_system
crayadm@login> aprun -b -n $NUMNODES -N 1 /bin/cat /proc/sys/kernel/hostname
```

CHECK CURRENT STATE OF COMPUTE NODE SSDs

The next step is intended only for XC systems that have compute nodes with SSDs, for example, systems with DataWarp SSDs or Intel® Xeon Phi™ "Knights Landing" processors.

5. Run xtcheckssd to ensure that SMW databases have the current state of compute node SSDs.

```
root@login# pcmd -r -n ALL_COMPUTE "/opt/cray/ssd/bin/xtcheckssd"
```

## 4.8.4 Test xtdumpsys and cdump

### Prerequisites

This procedure assumes that the system has been booted.

### About this task

This procedure tests the xtdumpsys and cdump tools. The example output is for illustrative purposes only. Actual output may differ for the current release.

### Procedure

1. Start an xtdumpsys typescript.

   Start a new window. Start a typescript session for xtdumpsys in that new window.

```
smw# su - crayadm
crayadm@smw> export TODAY=`date +%Y%m%d`
crayadm@smw> . /etc/opt/cray/release/cle-release
crayadm@smw> mkdir -p /home/crayadm/dump/${TODAY}_${BUILD}
crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
crayadm@smw> script -af hss.xtdumpsys
```

2. Start xtdumpsys.

   Start the dump, but do not press **Ctrl-d** until step 5 on page 309. When xtdumpsys asks for a dump reason, it will have created the dump directory.

For a full system:

```
crayadm@smw> xtdumpsys
  INFO: Beginning dump
 INFO: Gathering system partition information
 INFO: Gathering system hardware information
 INFO: No session specified, defaulting to current.
 INFO: Moving temporary log files to the dump directory.
 INFO:
#############################################################################
 INFO: # Your dump is available in /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-
NNNNNNNNNN #
 INFO:
#############################################################################
 Enter reason for dump:
 (an EOF terminates input, usually CTRL-D)
```

For a partitioned system:

```
crayadm@smw> xtdumpsys  -p p1
crayadm@smw> xtdumpsys  -p p2
```

3. Start a `cdump` typescript in a different window.

   Start another window. Start a typescript session for `cdump` in that window.

```
smw# su - crayadm
cdump crayadm@smw> export TODAY=`date +%Y%m%d`
cdump crayadm@smw> . /etc/opt/cray/release/cle-release
cdump crayadm@smw> cd /home/crayadm/dump/${TODAY}_${BUILD}/
cdump crayadm@smw> script -af hss.cdump
```

4. Dump a node with `cdump`.

   Change to the directory created in the `xtdumpsys` window (after `INFO: # Your dump is available in`), then use `cdump` to dump a compute node that successfully booted.

```
cdump crayadm@smw> cd /var/opt/cray/dump/p0-YYYYMMDDtHHMMSS-NNNNNNNNNN
cdump crayadm@smw> mkdir cdumps; cd cdumps
```

   For a full system (example uses the c0-0c0s3n0 node):

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot c0-0c0s3n0
 Wed Mar 1 09:08:08 CDT 2017 start cdump
 ...
makedumpfile Completed.
- done
Wed Mar 1 09:08:08 CDT 2017 cdump: # of nodes 1
 success 1
 failed  0
 skipped 0
cdump crayadm@smw> exit
```

   For a partitioned system, use the host name to specify which boot node. This example uses boot-p1 to `cdump` the c0-0c0s4n0 node in the p1 partition.

```
cdump crayadm@smw> cdump -AmD -r xt-hsn@boot-p1 c0-0c0s4n0
cdump crayadm@smw> exit
```

5. **Continue** `xtdumpsys`: enter a reason.

   After `cdump` completes, return to the `xtdumpsys` window and enter a reason.

   ```
   xtdumpsys window> testdump
   ```

   Then enter an end-of-file (**Ctrl-d**) to end the dump reason.

   ```
   xtdumpsys window> <Ctrl-d>
    testdump
    INFO: Dump reason:
    ...
    INFO:
   ##############################################################################
    INFO: # Your dump is available in /var/opt/cray/dump/
   p0-20170301t081927-1304240904 #
    INFO:
   ##############################################################################
    INFO: No post-processing plugin found at '/etc/opt/cray/dumpsys/
   postprocessing.py'
    INFO: Example plugins can be found at '/opt/cray/dumpsys/
   1.2.5-1.0000.35873.20.1/bin/plugins/examples/postprocessing.py.*'
    INFO: Cleaning up
   ```

   ```
   xtdumpsys crayadm@smw> exit
   ```

6. **Remove dump directory, if desired.**

   If there are no errors, it is probably safe to delete the dump directory.

   ```
   xtdumpsys crayadm@smw> rm -rf /var/opt/cray/dump/pX-YYYYMMDDtHHMMSS-NNNNNNNNNN
   crayadm@smw> exit
   ```

## 4.8.5 Make a Post-boot Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after booting the CLE system.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

### Procedure

1. List the available snapshots on the system.

   ```
   smw# snaputil list
   ```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

   ```
   smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
   smw# echo $SNAPSHOT
   ```

Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

(SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

**3.** Create a new snapshot.

```
smw# snaputil create ${SNAPSHOT}.postboot
```

## 4.8.6 Make a Post-boot Backup of Current Global and CLE Config Sets

### About this task

This procedure uses the cfgset command to create a post-boot backup of the global and CLE config sets.

### Procedure

**1.** Back up the current global config set.

```
smw# cfgset create --clone global global-postboot-${TODAY}
```

**2.** Back up the current CLE config set.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset create --clone p0 p0-postboot-${TODAY}
```

# 5 Customize Preinstalled SMW/CLE Software

Cray ships System Management Workstation (SMW) systems that are completely installed and configured with Cray-specific hostnames and IP addresses, among other things. To complete the configuration on site, reconfigure the system using these procedures.

Note that many commands require root privilege.

⚠ **CAUTION: Boot failure possible if using `cfgset` under certain conditions.**

The `cfgset create` and `cfgset update` commands always call pre- and post-configuration scripts. Some of these scripts require HSS daemons and other CLE services to be running. This can cause problems under these conditions:

- If `xtdiscover` is running, `cfgset` may hang or produce incorrect data that can result in system boot failure.
- If `xtbounce` is in progress or if the SMW is not connected to XC hardware, `cfgset` will fail.

In these circumstances, use the `--no-scripts` option with `cfgset create` or `cfgset update` to avoid running the scripts. Because using that option results in an invalid config set, remember to run `cfgset update` without the `--no-scripts` option afterwards, when circumstances permit, to ensure that all pre- and post-configuration scripts are run.

1. *Update Site Information and Install Needed Patches* on page 312.
2. *Change the Default SMW Passwords* (includes instructions for logging in as root).
3. *Change the Time Zone* on page 314.
4. (Optional) *Configure the SMW Firewall* on page 317.
5. *Configure LAN on the SMW* on page 318.
6. *Change Networks and IP Addresses in Global Config Set* on page 319.
7. *Change Networks and IP Addresses in CLE Config Set* on page 320.
8. Configure iDRAC network information.
   - For a Dell R630 SMW: *Set Up iDRAC for a Dell R630 SMW* on page 323.
   - For a Dell R815 SMW: *Set Up iDRAC for a Dell R815 SMW* on page 326.
9. *Change the Default iDRAC Password* on page 329.
10. (Optional) *Configure the Simple Event Correlator (SEC)* on page 225.
11. (Optional) *Configure Site Lightweight Log Manager (LLM)* on page 330.
12. (Optional) *Prevent Unintentional Re-creation of Mail Configuration Files* on page 232.
13. *Make a Post-customize Snapshot using snaputil* on page 331.
14. *Make a Post-customize Backup of Current Global and CLE Config Sets* on page 332.

## 5.1 Update Site Information and Install Needed Patches

### Prerequisites

This procedure uses the `xtshowrev` tool. If that module is not yet loaded, see *Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev* on page 201.

### About this task

The first task in customizing a preinstalled system is to ensure that the site name and serial number are set correctly and all needed patches are installed.

### Procedure

1. Determine which patches were installed in the factory.

   Use the `xtshowrev` command. The example shows output for an older release, but its purpose is to indicate where to look for CLE, SLE, and SMW patch information.

```
smw# xtshowrev
Site:                CRAY/INTERNAL
S/N:                 9999
System Type:         XC40
Install Date:        2016-06-01
System Name:         panda1
CNL/CLE Release:     6.0.UP01
XT Release:          6.0.96
CLE Kernel:          3.12.51-52.31.1_1.0600.9146
CLE OS:              SLES12
CLE Patch Sets:      01 02 03      <------------------- CLE patches applied
CLE FNs:
Lustre Version:      2.5.4
OS Type:             CLE
SMW Release:         8.0.UP01
SMW Build:           8.0.96
HSS Release:         8.0__446__ge75851a-49.1
SMW Kernel:          3.12.51-52.39
SMW OS:              SLES12
SLE Patch Sets:                    <------------------- SLE patches applied
SMW Patch Sets:                    <------------------- SMW patches applied
SMW FNs:             5844c
SEC Release:         Cray_SEC 8.0__6__g689802a (sec 2.7.6)
Current Date:        2016-06-01 12:59:21
crayadm@smw>
```

2. Update site information in the `site_config` file.

   For an initial install, the `xtgetrev` command is used to enter site information. For a preinstalled system, enter this information by manually editing the `site_config` file instead.

```
smw# vi /etc/opt/cray/release/pkginfo/site_config
---
site name: CRAY/INTERNAL      <------------------- change this
```

```
serial number: 9999           <------------------ change this
system name: panda1           <------------------ change this, if needed
system type: XC40             <------------------ change this, if needed
install date: 2016-06-01
os type: CLE
```

It is especially important to change/enter the serial number because that is the key into the Site Configurations Database, and it is used to determine whether a site has access to future patches.

**3.** Check for patches released by Cray.

Day-one patches are noted in the Errata docs that are included with the release. For other patches, check CrayPort, which is updated with available patches for the entitled site serial numbers when a patch is released. If patches need to be applied, continue with the remaining steps.

CONTINUE ONLY IF PATCHES NEED TO BE APPLIED

**4.** Make a directory on the SMW (if it does not already exist) to hold any patches that may be available on CrayPort.

```
smw# mkdir -p /var/adm/cray/release/patchsets
```

**5.** Download SMW and CLE patches to the patchset directory on the SMW, as described in the release notes.

**6.** Install SMW and CLE patches.

- To install a single patch, follow the instructions provided in the patch README file.
- To load and install multiple patches, complete the following substeps. When installing more than one patch, Cray recommends postponing the building and mapping of images until the last patch is installed.

  **NOTE:** (SMW HA only) Make a note of all patch sets that will be applied on the first SMW. The second SMW must have exactly the same patch sets.

a. Temporarily suppress building and mapping images.

```
smw# export PATCHSET_BUILD_IMAGES=false
smw# echo $PATCHSET_BUILD_IMAGES

smw# export PATCHSET_NIMS_TIMING=deferred
smw# echo $PATCHSET_NIMS_TIMING
```

b. Follow all of the instructions in the patch README files.

These instructions will include running the LOAD script and the INSTALL script for each patch, and there may be additional steps for some patches, such as running `xtzap` again to update firmware from an SMW patch.

**7.** Build and map new images.

```
smw# imgbuilder --map
```

## 5.2 Change the Default System Management Workstation (SMW) Passwords

### About this task

The SMW contains its own `/etc/passwd` file that is separate from the password file for the rest of the CLE system. After logging on to the SMW for the first time, Cray recommends changing the default passwords, as described in the following instructions.

### Procedure

1. Log in to SMW as root.

   When the login screen is displayed with the crayadm account as the account which will be logged in:

   a. Select **Not listed?**, then enter `root` for the username.

   b. Either press **Enter** or select **Sign In**.

   c. Enter the password for root.

2. Change default passwords on the SMW by executing the following commands.

```
smw# passwd root
```

```
smw# passwd crayadm
```

```
smw# passwd mysql
```

It is also necessary to change the iDRAC password, which uses a different procedure. See *Change the Default iDRAC Password* on page 329.

## 5.3 Change the Time Zone

### Prerequisites

This procedure assumes that the XC system is booted.

### About this task

This procedure changes the time zone of an XC system by changing some configuration and then rebooting components. Most of these commands must be performed as root.

### Procedure

1. Check the current time zone.

   a. Check time zone on SMW.

```
smw# date
```

b. Check time zone on cabinet and blade controllers.

```
smw# xtrsh -l root -s date
```

c. Check time zone on boot node.

```
smw# ssh boot date
```

d. Check time zone on SDB node.

   This command works from the SMW if the SDB node is a tier1 node with an Ethernet connection to the SMW.

```
smw# ssh sdb date
```

e. Check time zone on all service nodes.

```
smw# ssh sdb pcmd -r -n ALL_SERVICE_NOT_ME "date"
```

f. Check time zone on all compute nodes.

```
smw# ssh sdb pcmd -r -n ALL_COMPUTE "date"
```

Continue to the next step only if the time zone needs to be changed.

**2.** Change the SMW local time zone, if needed.

The default time zone on the SMW is **America/Chicago**. To change it:

a. Execute this command:

```
smw# yast2 timezone
```

   yast2 opens a new window for changing the time zone, then a pop-up window appears with this message: "file /etc/ntp.conf has been changed manually. YaST might lose some of the changes."

b. Select the **Do not show this message anymore** checkbox, then select **Continue**.

c. Choose the time zone either by selecting a region on the map or by using the drop-down menus for **Region** and **Time Zone**.

d. Select **Other Settings** if the time is incorrect, then select the **Manually** radio button and enter **Current Time** and **Current Date**. Select **Accept** when done.

e. Select **OK** when done with time zone settings.

The change on the SMW is immediate, but any users on the system need to log out and then log in again to get the new environment. This does not change the time zone for the CLE nodes or the cabinet and blade controllers. Continue to step 3 to make those changes.

**3.** Change the time zone in the global config set.

a. Set cray_time.settings.service.data.timezone to the desired time zone.

   A list of possible time zones is available on the SMW in /usr/share/zoneinfo/zone1970.tab.

```
smw# cfgset update -s cray_time -m interactive global
```

b. Validate the config set.

```
smw# cfgset validate global
```

**4.** Change the time zone in the CLE config set.

If the CLE config set has `cray_time.inherit` set to true, then the time zone and other time settings from the global config set will be inherited by the CLE config set. If the CLE config set has `cray_time.inherit` set to false, then use these commands to change the setting and validate the config set.

a. Set `cray_time.settings.service.data.timezone` to the desired time zone.

A list of possible time zones is available on the SMW in `/usr/share/zoneinfo/zone1970.tab`.

```
smw# cfgset update -s cray_time -m interactive p0
```

b. Validate the config set.

```
smw# cfgset validate p0
```

**5.** Put the SMW time zone setting where the cabinet and blade controllers can access it.

```
smw# cp /etc/localtime /opt/tftpboot/localtime
```

**6.** Reboot to set the new time zone for all components.

a. Shut down CLE.

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.hostname.stop
```

b. Reboot the SMW and verify that the time zone has been reset..

```
crayadm@adm> su - root
smw# reboot
```

After the SMW reboots, check that the SMW has the desired time zone setting.

```
smw# date
```

c. Reboot the cabinet controllers, then verify that all cabinet controllers are up.

```
smw# xtccreboot -c all

smw# sleep 120

smw# xtalive -a l1sysd -l l1 s0
```

Repeat the `xtalive` command until all cabinet controllers are alive.

d. Reboot the blade controllers, then verify that all blade controllers are up.

```
smw# xtccreboot -b all

smw# sleep 120

smw# xtalive s0
```

Repeat the `xtalive` command until all blade controllers are alive.

e. Boot CLE nodes for the new time zone using the site boot automation file.

```
crayadm@smw> xtbootsys -a auto.hostname.start
```

7. Check the current time zone again.

a. Check time zone on SMW.

```
smw# date
```

b. Check time zone on cabinet and blade controllers.

```
smw# xtrsh -l root -s date
```

c. Check time zone on boot node.

```
smw# ssh boot date
```

d. Check time zone on SDB node.

This command works from the SMW if the SDB node is a tier1 node with an Ethernet connection to the SMW.

```
smw# ssh sdb date
```

e. Check time zone on all service nodes.

```
smw# ssh sdb pcmd -r -n ALL_SERVICE_NOT_ME "date"
```

f. Check time zone on all compute nodes.

```
smw# ssh sdb pcmd -r -n ALL_COMPUTE "date"
```

If these checks show the correct time zone, then the time zone has been successfully changed.

## 5.4 Configure the SMW Firewall

### Prerequisites

This procedure assumes that SLES 12 has been installed as the base operating system on the SMW.

### About this task

The SUSE firewall settings may need to be adjusted to match site firewall policy and to customize for site IP addresses. This procedure enables and configures the firewall.

**TIP:** It is not necessary to shut down the system before performing this task.

### Procedure

1. Save the SUSE firewall configuration.

Before modifying the SUSE firewall settings, make a copy of the configuration file.

```
smw# cp -p /etc/sysconfig/SuSEfirewall2 /etc/sysconfig/SuSEfirewall2.orig
```

2. Check current firewall settings.

   Check current firewall settings and change to support any site requirements. During the process of configuring Cray SMW and CLE software, some of the firewall settings may be adjusted. SSH access is one of the protocols permitted through the firewall from the external network to the SMW.

   ```
   smw# iptables -L
   smw# vi /etc/sysconfig/SuSEfirewall2
   ```

3. Start the firewall immediately.

   Invoke the modified configuration.

   ```
   smw# systemctl start SuSEfirewall2_init.service
   smw# systemctl start SuSEfirewall2.service
   ```

4. Ensure that the firewall will start at next boot.

   Execute the following commands to start the firewall at boot time.

   ```
   smw# systemctl enable SuSEfirewall2_init.service
   smw# systemctl enable SuSEfirewall2.service
   ```

5. Verify firewall changes.

   Verify the changes to the `iptables`.

   ```
   smw# iptables -nvL
   ```

## 5.5 Configure LAN on the SMW

### About this task

This procedure sets the network configuration for eth0 and the hostname for the SMW.

### Procedure

1. Execute this command:

   ```
   smw# yast2 lan
   ```

   The **Network Settings** screen appears with the **Overview** tab highlighted.

2. Select the **eth0** line on the **Overview** tab, then select **Edit**.

   The **Network Card Setup** screen appears with the **Address** tab highlighted.

3. Select **Statically Assigned IP address** on the **Address** tab and enter values for IP address, subnet mask, and hostname (including the domain name). Then select **Next**.

4. Select the **Hostname/DNS** tab on the **Network Settings** screen.

   1. For the **Hostname and Domain Name** area, enter Hostname and Domain Name.

2. For the **Name Servers and Domain Search List**, enter Name Server 1, Name Server 2, Name Server 3, and Domain Search.

5. Select the **Routing** tab on the **Network Settings** screen, then enter the Default IPv4 Gateway (for the network connected to eth0) and set Device to eth0 using the dropdown menu.

6. Click **OK** after all of the **Network Settings** have been prepared.

## 5.6 Change Networks and IP Addresses in Global Config Set

### Prerequisites

This procedure assumes that the SMW and CLE software has been installed so that the global config set is present.

### About this task

This procedure suggests some settings to change in the Cray Global Networking service (in the global config set) to add site-specific data. It also includes steps to validate the config set and run Ansible plays on the SMW to effect the changes.

### Procedure

1. Check the global config set for information that may need to be changed.

```
smw# cfgset search -t "ipv4_address|management" global
```

2. Update the cray_global_net service to change any settings that need site-specific information.

```
smw# cfgset update -s cray_global_net -m interactive -s advanced global
```

At a minimum, these settings will need to be changed:

- Information for the management network, which is the customer network connected to the SMW.

```
cray_global_net.settings.networks.data.management.ipv4_network
cray_global_net.settings.networks.data.management.ipv4_netmask
cray_global_net.settings.networks.data.management.ipv4_gateway
cray_global_net.settings.networks.data.management.dns_servers
cray_global_net.settings.networks.data.management.ntp_servers
```

- IP address of the SMW on the management network.

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_addr
ess
```

3. Update the SMW host ID.

If the customer Ethernet IP address changes, the output from the `hostid` command will be different. After changing this setting

```
cray_global_net.settings.hosts.data.primary_smw.interfaces.customer_ethernet.ipv4_address
```

in the previous step, ensure that this setting (the SMW host ID) is set to the output of the `hostid` command.

```
cray_global_net.settings.hosts.data.primary_smw.hostid
```

4. Validate the global config set.

   ```
   smw# cfgset validate global
   ```

5. Run Ansible plays on the SMW.

   After the global config set has been updated, reapply any Ansible plays that consume global config set data.

   > **NOTE:** (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

   ```
   smw# /etc/init.d/cray-ansible start
   ```

   Logs from running Ansible plays, such as cray-ansible, are stored on the SMW in `/var/opt/cray/log/ansible`.

# 5.7 Change Networks and IP Addresses in CLE Config Set

## Prerequisites

This procedure assumes that the SMW and CLE software has been installed so that a CLE config set is present.

## About this task

The Cray Networking service defines all network information for CLE nodes. This procedure suggests some settings to change in the Cray Networking service configuration worksheet to add site-specific data. It also includes steps to validate the config set and run Ansible plays on the SMW to effect the changes.

> **REMEMBER:** For partitioned systems, each partition generally has its own config set and associated configuration worksheets. Follow this procedure for each partition.

## Procedure

1. Save a copy of original CLE worksheets.

   Copy the original configuration worksheets into a new directory to preserve them in case they are needed later for comparison.

   ```
   smw# ls -l /var/opt/cray/imps/config/sets/p0/worksheets

   smw# cp -a /var/opt/cray/imps/config/sets/p0/worksheets \
   /var/opt/cray/imps/config/sets/p0/worksheets.orig
   ```

2. Make a work area for CLE worksheets.

   Copy the CLE configuration worksheets to a new work area for editing. The worksheets should not be edited in their original location for two reasons: (1) the configurator will not permit updating a config set from worksheets within that config set, and (2) edits would be overwritten when the config set is updated.

   ```
   smw# cp -a /var/opt/cray/imps/config/sets/p0/worksheets \
   /var/adm/cray/release/p0_worksheet_workarea
   ```

Change to the work area directory to simplify the editing commands in the following steps.

```
smw# cd /var/adm/cray/release/p0_worksheet_workarea
```

3. Check the CLE config set for information that may need to be changed.

```
smw# cfgset search -s cray_net -t ipv4 p0
```

4. Edit `cray_net_worksheet.yaml` to change any settings that need site-specific information.

   At a minimum, these settings will need to be changed:

   a. Change these values to site-specific values for the "Customer network" to which the login nodes connect.

   ```
   cray_net.settings.networks.data.login.ipv4_network
   cray_net.settings.networks.data.login.ipv4_netmask
   ```

   b. (Only for systems with an external Lustre server) Change these values to site-specific values for each external Lustre server.

   ```
   cray_net.settings.networks.data.lnet.ipv4_network
   cray_net.settings.networks.data.lnet.ipv4_netmask
   ```

   c. Change this value to the IP address of the login node's eth0 interface on the "login" network.

   ```
   cray_net.settings.hosts.data.login_node.interfaces.login_ethernet.ipv4_addres
   s
   ```

   When making changes, keep this mind:

   - Add values for the `dns_servers` and `dns_search` settings to the login network only, not to any other network.
   - DO NOT add a value for the `ntp_servers` setting for any network used for CLE nodes, because CLE nodes must source their time/NTP settings from the SMW rather than try to contact NTP servers on the login network.

5. Configure additional hosts, as needed.

   If this system has additional service nodes that need to have hostname or hostname alias or network interface settings, then add a section like this for each of the hosts. The first example shows the host configuration of a DVS node with the hostname set to "dvs1," a hostname alias of "dvs," and one Ethernet interface connected to the "login" network.

```
cray_net.settings.hosts.data.common_name.dvs_node: null
cray_net.settings.hosts.data.dvs_node.description: DVS node
cray_net.settings.hosts.data.dvs_node.aliases:
- dvs
cray_net.settings.hosts.data.dvs_node.roles: []
cray_net.settings.hosts.data.dvs_node.hostid: 'c0-0c0s0n2'
cray_net.settings.hosts.data.dvs_node.host_type: ''
cray_net.settings.hosts.data.dvs_node.hostname: 'dvs1'
cray_net.settings.hosts.data.dvs_node.standby_node: false

cray_net.settings.hosts.data.dvs_node.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.description: Ethernet
    connecting the DVS node to the customer network.
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.network: login
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.ipv4_address: '172.30.50.128'
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.startmode: auto
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.bootproto: static
```

```
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.dvs_node.interfaces.eth0.extra_attributes: []
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.module: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.params: ''
#cray_net.settings.hosts.data.dvs_node.interfaces.eth0.unmanaged_interface: false
```

The second example shows the host configuration for an LNet router node that has two different InfiniBand interfaces (ib0 and ib2) to connect to two different networks.

```
cray_net.settings.hosts.data.common_name.clfs_lnet_1: null
cray_net.settings.hosts.data.clfs_lnet_1.description: CLFS router 1 node
cray_net.settings.hosts.data.clfs_lnet_1.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.roles: []
cray_net.settings.hosts.data.clfs_lnet_1.hostid: 'c0-0c1s0n1'
cray_net.settings.hosts.data.clfs_lnet_1.host_type: ''
cray_net.settings.hosts.data.clfs_lnet_1.hostname: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.standby_node: false
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib0: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.description: InfiniBand
    ib0 connecting the CLFS router 1 node to the lnet network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.ipv4_address: '10.150.10.65'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib0.unmanaged_interface: false

cray_net.settings.hosts.data.clfs_lnet_1.interfaces.common_name.ib2: null
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.name: ib2
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.description: InfiniBand
    ib2 connecting the CLFS router 1 node to the lnet1 network.
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.aliases: []
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.network: lnet1
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.ipv4_address: '10.151.10.65'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mac: ''
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.startmode: auto
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.bootproto: static
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.mtu: '65520'
cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.extra_attributes:
- IPOIB_MODE='connected'
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.module: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.params: ''
#cray_net.settings.hosts.data.clfs_lnet_1.interfaces.ib2.unmanaged_interface: false
```

**6.** Upload the modified worksheet into the CLE config set.

Note that the full filepath must be specified in this `cfgset` command.

```
smw# cfgset update -w \
/var/adm/cray/release/p0_worksheet_workarea/cray_net_worksheet.yaml p0
```

**7.** Validate the CLE config set.

```
smw# cfgset validate p0
```

**8.** Run Ansible plays on the SMW.

After the CLE config set has been updated, reapply any Ansible plays that consume CLE config set data.

> **NOTE:** (SMW HA only) Both SMWs require this command. The procedure to install and configure the second SMW includes this command.

```
smw# /etc/init.d/cray-ansible start
```

Logs from running Ansible plays, such as cray-ansible, are stored on the SMW in `/var/opt/cray/log/ansible`.

## 5.8 Set Up iDRAC for a Dell R630 SMW

### Prerequisites

This procedure requires the following:

- Physical access to the SMW console
- iDRAC6 IP address, subnet mask, and default gateway
- SMW `root` account password

### About this task

An integrated Dell Remote Access Controller (iDRAC) enables remote management of the System Management Workstation (SMW). This procedure sets up and enables an iDRAC for an R630 SMW. For an R815 model, see *Set Up iDRAC for a Dell R815 SMW* on page 326.

### Procedure

1. If the SMW is up, `su` to `root` and shut it down.

   ```
   crayadm@smw> su - root
   smw# shutdown -h now;exit
   ```

2. Connect an Ethernet cable to the iDRAC port. The cable is located on back of the R815 SMW in the lower left corner.

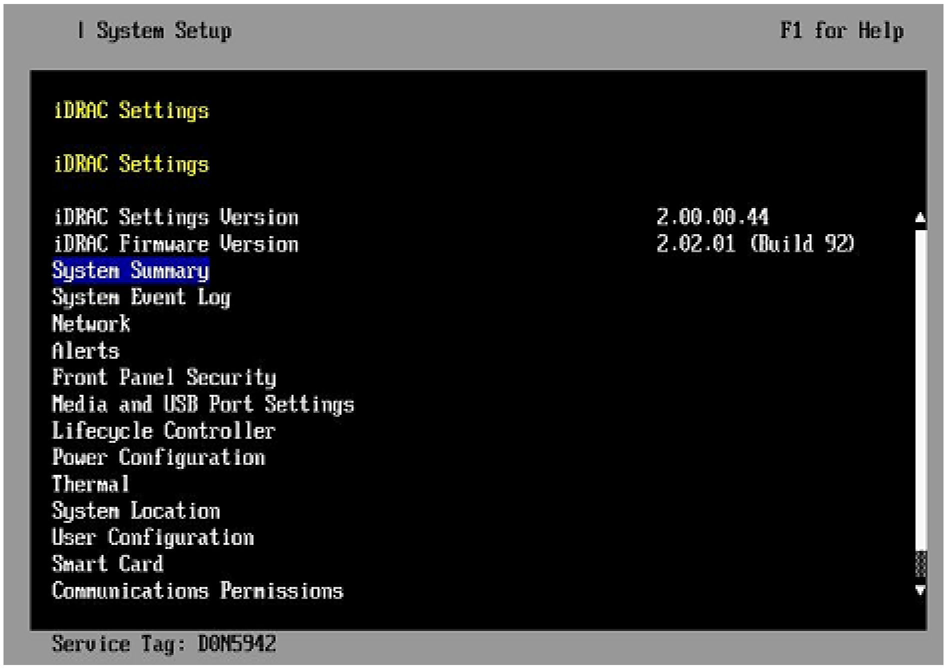
3. Power up the SMW.

4. Change the iDRAC settings.

   Select **iDRAC Settings** on the **System Setup Main Menu**, then press **Enter**.

   The **iDRAC Settings** screen appears.

*Figure 24. Dell R630 iDRAC Settings Screen*



```
 | System Setup                                          F1 for Help

 iDRAC Settings

 iDRAC Settings

 iDRAC Settings Version                      2.00.00.44
 iDRAC Firmware Version                      2.02.01 (Build 92)
 System Summary
 System Event Log
 Network
 Alerts
 Front Panel Security
 Media and USB Port Settings
 Lifecycle Controller
 Power Configuration
 Thermal
 System Location
 User Configuration
 Smart Card
 Communications Permissions

 Service Tag: D0N5942
```

**5.** Change the iDRAC network.

   a. Select **Network** to display a long list of network settings.

   b. Change the DNS DRAC name.

      Use the arrow key to scroll down to **DNS DRAC Name**, then enter an iDRAC hostname that is similar to the SMW node hostname (e.g., cray-drac).

   c. Change the static DNS domain name.

      Use the arrow key to scroll down to **Static DNS Domain Name**, then enter the DNS domain name and press **Enter**.

   d. Change the IPv4 settings.

      Use the arrow key to scroll down to the **IPV4 SETTINGS** list.

      **1.** Ensure that IPv4 is enabled.

         **a.** If necessary, select **Enable IPV4**, then press **Enter**.

         **b.** Select **<Enabled>** in the pop-up window, then press **Enter** to return to the previous screen.

      **2.** Ensure that DHCP is disabled.

         **a.** If necessary, select **Enable DHCP**, then press **Enter**.

         **b.** Select **<Disabled>** in the pop-up window, then press **Enter** to return to the previous screen.

      **3.** Change the IP address.

         **a.** Select **Static IP Address**.

         **b.** Enter the IP address of the iDRAC interface (`ipmi0`) for the SMW, then press **Enter**.

      **4.** Change the gateway.

a. Select **Static Gateway**.

b. Enter the appropriate value for the gateway of the network to which the iDRAC is connected, then press **Enter**.

5. Change the subnet mask.

a. Select **Subnet Mask**.

b. Enter the subnet mask for the network to which the iDRAC is connected (such as `255.255.255.0`), then press **Enter**.

6. Change the DNS server settings.

a. Select **Static Preferred DNS Server**, enter the IP address of the primary DNS server, then press **Enter**.

b. Select **Alternate DNS Server**, enter the IP address of the alternate DNS server, then press **Enter**.

e. Change the IPMI settings.

Change the IPMI settings to enable the Serial Over LAN (SOL) console.

1. Use the arrow key to scroll down to the **IPMI SETTINGS** list.
2. Ensure that **Enable IPMI over LAN** is selected.

   **TIP:** Use the left-arrow or right-arrow to switch between two settings.

3. Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

f. Exit Network screen.

Press the **Esc** key to exit the **Network** screen and return to the **iDRAC Settings** screen.

6. Change hostname in iDRAC LCD display.

Change front panel security to show the hostname in LCD display.

a. Use the arrow key to scroll down and highlight **Front Panel Security** on the **iDRAC Settings** screen, then press **Enter**.

b. Select **Set LCD message**, then press **Enter**.

c. Select **User-Defined String**, then press **Enter**.

d. Select **User-Defined String**, then enter the SMW hostname and press **Enter**.

e. Press the **Esc** key to exit the **Front Panel Security** screen.

7. (Optional) Change the iDRAC **System Location** fields.

Change the **System Location** configuration on the **iDRAC Settings** screen to set any of these fields: **Data Center Name**, **Aisle Name**, **Rack Name**, and **Rack Slot**.

8. Configure iDRAC virtual media.

a. Select **Domain Name**, then press **Enter**.

b. Select **Virtual Media Configuration**, then press **Enter**.

c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.

d. Press **Esc** to exit the **Virtual Media Configuration** menu.

9. Set the password for the iDRAC root account.

   a. Use the arrow key to highlight **User Configuration** on the **iDRAC Settings** screen, then press **Enter**.

   b. Confirm that User Name is root. Select **User Name**, then enter the "root" user name.

   c. Select **Change Password**, then enter a new password.

   d. Reenter the new password in the next pop-up window to confirm it (the default password is "calvin").

   e. Press the **Esc** key to exit the **User Configuration** screen.

10. Exit iDRAC settings.

    a. Press the **Esc** key to exit the **iDRAC Settings** screen.

       A "Settings have changed" message appears.

    b. Select **Yes**, then press **Enter** to save the changes.

       A "Success" message appears.

    c. Select **Ok**, then press **Enter**.

       The main screen (**System Setup Main Menu**) appears.

# 5.9 Set Up iDRAC for a Dell R815 SMW

## Prerequisites

This procedure requires the following:

- Physical access to the SMW console
- iDRAC6 IP address, subnet mask, and default gateway
- SMW `root` account password

## About this task

An integrated Dell Remote Access Controller (iDRAC) enables remote management of the System Management Workstation (SMW). This procedure sets up and enables an iDRAC for an R815 SMW. For an R630 model, see *Set Up iDRAC for a Dell R630 SMW* on page 323.

## Procedure

1. If the SMW is up, `su` to `root` and shut it down.

   ```
   crayadm@smw> su - root
   smw# shutdown -h now;exit
   ```

2. Connect an Ethernet cable to the iDRAC port. The cable is located on back of the R815 SMW in the lower left corner.

3. Power up the SMW.

**4.** Change the iDRAC settings.

Watch the screen carefully as text scrolls until the **iDRAC6 Configuration Utility 1.57** line is visible. When the line **Press <Ctrl-E> for Remote Access Setup within 5 sec...** displays, press **Ctrl-E** within 5 seconds.

```
0 5 0 ATA WDC WD5000BPVT-0 1A01 465 GB
LSI Corporation MPT2 boot ROM successfully installed!
iDRAC6 Configuration Utility 1.57
Copyright 2010 Dell Inc. All Rights Reserved
iDRAC6 Firmware Revision version: 1.54.15
Primary Backplane Firmware Revision 1.07
-------------------------------
IPv6 Settings
-------------------------------
IPv6 Stack : Disabled
Address 1 : ::
Default Gateway : ::
--------------------------------
IPv4 Settings
--------------------------------
IPv4 Stack : Enabled
IP Address : 172. 31. 73.142
Subnet mask : 255.255.255. 0
Default Gateway : 172. 31. 73. 1
Press <Ctrl-E> for Remote Access Setup within 5 sec...
```

The **iDRAC6 Configuration Utility** menu appears.

**5.** Set **iDRAC LAN** to **ON**.

**6.** Configure the iDRAC LAN.

Select **LAN Parameters**, then press **Enter**.

a. Configure iDRAC6 name.

Use the arrow key to scroll down and select **iDRAC6 Name**, then press **Enter**. Enter a value for **Current DNS iDRAC6 Name** (e.g., smw-drac), then press **Enter**.

**Trouble?** If unable to set the iDRAC6 name, try this:

**1.** Temporarily set **Register iDRAC6 Name** to "On."

**2.** Set **iDRAC6 Name**.

**3.** Return to **Register iDRAC6 Name** and set it to "Off."

b. Configure domain name.

Use the arrow key to scroll down and select **Domain Name**, then press **Enter**. Enter a value for **Current Domain Name** (e.g., us.cray.com), then press **Enter**.

c. Configure hostname string.

Use the arrow key to scroll down and select **Host Name String**, then press **Enter**. Enter a value for **Current Host Name String** (e.g., smw-drac), then press **Enter**.

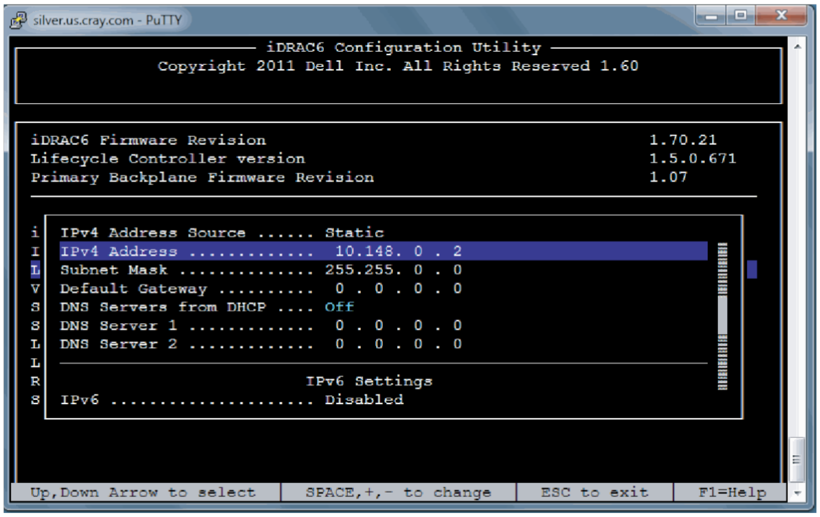
d. Configure IPv4 settings.

Use the arrow key to scroll down into the **IPv4 Settings** group and confirm that the **IPv4 Address Source** is set to **static**. Then enter values for the following:

**IPv4 Address** (the SMW DRAC IP address)

**Subnet Mask** (the SMW iDRAC subnet mask)
**Default Gateway** (the SMW iDRAC default gateway)
**DNS Server 1** (the first site DNS server)
**DNS Server 2** (the second site DNS server)

*Figure 25. Dell R815 SMW DRAC IPv4 Parameter Settings*



```
                    iDRAC6 Configuration Utility
          Copyright 2011 Dell Inc. All Rights Reserved 1.60

iDRAC6 Firmware Revision                                      1.70.21
Lifecycle Controller version                                  1.5.0.671
Primary Backplane Firmware Revision                           1.07

i| IPv4 Address Source ...... Static
I| IPv4 Address ............. 10.148. 0 . 2
L| Subnet Mask .............. 255.255. 0 . 0
V| Default Gateway .........   0 . 0 . 0 . 0
S| DNS Servers from DHCP .... Off
S| DNS Server 1 ............   0 . 0 . 0 . 0
L| DNS Server 2 ............   0 . 0 . 0 . 0
L|
R|                     IPv6 Settings
S| IPv6 ..................... Disabled

Up,Down Arrow to select | SPACE,+,- to change | ESC to exit | F1=Help
```

e. Configure IPv6 settings.

   Use the arrow key to scroll down into the **IPv6 Settings** group and ensure that **IPv6** is disabled.

f. Change the IPMI settings.

   Change the IPMI settings to enable the Serial Over LAN (SOL) console.

   1. Use the arrow key to scroll down to the **IPMI SETTINGS** list.
   2. (Stand-alone SMW only) Ensure that **Enable IPMI over LAN** is NOT selected.
   3. (SMW HA only) Ensure that **Enable IPMI over LAN** is selected. This setting is used for both SMWs in an SMW HA pair.
   4. Ensure that **Channel Privilege Level Limit** is set to **Administrator**.

g. Press **Esc** to exit **LAN Parameters** and return to the **iDRAC6 Configuration Utility** menu.

7. Configure iDRAC virtual media.

   a. Select **Domain Name**, then press **Enter**.

   b. Select **Virtual Media Configuration**, then press **Enter**.

   c. Select the **Virtual Media** line and press the space key until it indicates **Detached**.

   d. Press **Esc** to exit the **Virtual Media Configuration** menu.

8. Set the password for the iDRAC LAN root account.

   Using the arrow keys, select **LAN User Configuration**, then press **Enter**. The following configuration is for both SSH and web browser access to the iDRAC.

   a. Select **Account User Name** and enter the account name "root."

   b. Select **Enter Password** and enter the intended password.

   c. Select **Confirm Password** and enter the intended password again.

   d. Press **Esc** to return to the **iDRAC6 Configuration Utility** menu.

**9.** Exit the iDRAC configuration utility.

   a. Press **Esc** to exit the **iDRAC6 Configuration Utility** menu.

   b. Select **Save Changes and Exit**.

   The **BIOS Boot Manager** menu appears.

## 5.10 Change the Default iDRAC Password

### About this task

This procedure describes how to log in to the iDRAC web interface and change a user password.

### Procedure

**1.** Bring up a web browser.

**2.** Go to: `https://`*`cray-drac`*, where *`cray-drac`* is the name of the iDRAC.

   A login screen appears.

**3.** Log in to the web interface as `root`.

**4.** Select **iDRAC settings** on the left navigation bar.

**5.** Expand **iDRAC settings** on the left navigation bar.

**6.** Select **User Authentication**.

**7.** Select the user whose password is changing. To change the root password, select userid `2`.

**8.** Select **Next**.

**9.** Select the **Change Password** box and enter the new password in the boxes below it.

**10.** Select **Apply** to complete the password change.

The password change is complete.

**Alternative.** Another approach to changing the iDRAC root password is to use `ipmitool` on the SMW command line interface.

```
smw# ipmitool -U root -I lanplus -H <drac-ip-addr> -P <old-drac-password> \
user set password 2 <new-drac-password>
```

## 5.11 Configure the Simple Event Correlator (SEC)

The Simple Event Correlator (SEC) is an SMW utility that parses every line being appended to system log files, watching for specific strings that represent the occurrence of significant system events. When a specified string is detected, SEC sends notification that this has happened, either by email, IRC, writing to a file, or some user-configurable combination of all three.

SEC is enabled by default, and by default is configured to generate email notifications to `crayadm`. The types of notifications generated and the recipients to whom notifications are sent are defined in the SEC configuration file, `/etc/opt/cray/cray_sec_actions_config`.

The System Management Workstation (SMW) release includes `sec-2.7.6` and an SEC support package, `cray-sec-8.0.0`. The SEC support package contains control scripts to manage the starting and stopping of SEC around a Cray mainframe boot session, in addition to other utilities and a rule set designed for Cray systems.

For configuration procedures, see *XC™ Series SEC Configuration Guide* (S-2542) for release CLE 6.0.UP03.

## 5.12 Configure Site Lightweight Log Manager (LLM)

### About this task

If this site uses the Lightweight Log Manager (LLM) to send logs from the SMW to a site loghost, use this procedure to update settings in the cray_logging service, which is in the global config set.

### Procedure

1. Update the cray_logging service.

```
smw# cfgset update -s cray_logging -m interactive -l advanced global
```

2. Update the following settings, as needed.

```
cray_logging.settings.site_loghost.data.name:
cray_logging.settings.site_loghost.data.ip_protocol: tcp
cray_logging.settings.site_loghost.data.ip_port: 514
cray_logging.settings.site_loghost.data.syslog_format: rfc5424
```

3. Validate the global config set

```
smw# cfgset validate global
```

4. Apply configuration changes, if any.

   Run `cray-ansible` so Ansible plays that consume config set data will apply that data to the SMW.

```
smw# /etc/init.d/cray-ansible start
```

## 5.13 Prevent Unintentional Re-creation of Mail Configuration Files

This procedure is optional. It applies to systems where postfix or sendmail are configured on the SMW.

To prevent the `master.cf` and `main.cf` postfix configuration files from being re-created during software updates or fixes, edit the `/etc/sysconfig/mail` file on the SMW and ensure that the `MAIL_CREATE_CONFIG` setting is set to "no."

```
smw# vi /etc/sysconfig/mail

MAIL_CREATE_CONFIG="no"
```

## 5.14 Make a Post-customize Snapshot using snaputil

### About this task

This procedure uses `snaputil` to make an archival snapshot of the system after customizing a preinstalled system.

**Best Practice.** Make a snapshot and back up the config set at the same time to keep them in sync. Name the snapshot and config set backup using the same suffix and date/time stamp to help administrators identify which snapshot and config set backup pairs belong together.

For more information, see *About Snapshots and Config Set Backups* on page 16.

### Procedure

1. List the available snapshots on the system.

   ```
   smw# snaputil list
   ```

2. Set the `SNAPSHOT` environment variable using the currently booted snapshot name.

   ```
   smw# export SNAPSHOT=$(snaputil list |grep ^cur| awk '{print $2}')
   smw# echo $SNAPSHOT
   ```

   Setting a variable for the snapshot name enables better command substitution in later commands dealing with snapshots.

   (SMW HA only) This is especially important for SMW HA systems because it makes it easier to use the exact same snapshot name for both SMWs. Using different snapshots results in HSS database (MySQL) inconsistencies, which causes problems at failover.

3. Create a new snapshot.

   ```
   smw# snaputil create ${SNAPSHOT}.postcustomize
   ```

## 5.15 Make a Post-customize Backup of Current Global and CLE Config Sets

### About this task

This procedure uses the cfgset command to create a backup of the global and CLE config sets after customizing a preinstalled system.

### Procedure

1. Back up the current global config set.

   ```
   smw# cfgset create --clone global global-postcustomize-${TODAY}
   ```

2. Back up the current CLE config set.

   This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

   ```
   smw# cfgset create --clone p0 p0-postcustomize-${TODAY}
   ```

## 5.16 Back Up the Newly Installed and Configured SMW/CLE Software

After installing and configuring the new SMW/CLE software, create a backup of it, if needed.

**Dell R815 SMW with software RAID** If an R815 SMW was configured using the recommended software RAID1 configuration for the boot disk, there is no need to make a backup of the SMW boot disk because the boot disks are mirrored automatically by the RAID software.

**Dell R630 SMW with software RAID** If an R630 SMW was configured using the recommended hardware RAID5 virtual disk, there is no need to make a backup of the SMW boot disk. Any one of the four disks in the RAID5 configuration can fail without losing any data.

# 6 Troubleshoot SMW/CLE Software Installation

The following procedures address issues that may occur while installing, configuring, or booting Cray System Management Workstation (SMW) and Cray Linux Environment (CLE) software.

- For instructions on how to boot the CLE system in debugging mode, see *Boot the System with DEBUG* on page 333.
- For extensive information about troubleshooting a CLE system boot, see *XC™ Series Boot Troubleshooting Guide* (S-2565).

## 6.1 Boot the System with DEBUG

### Prerequisites

This procedure assumes that the SMW and CLE have been configured and images have been built and mapped to nodes.

### About this task

Because booting with DEBUG is not commonly used, these instructions describe it for a full system (p0) and not for a partitioned system.

### Procedure

1. Run rtr to set up the routes.

   ```
   crayadm@smw> rtr -R
   ```

2. Enable DEBUG boot.

   As root, set debug for the boot parameters of all nodes. This ensures they will stop the boot process allowing a console login.

   ```
   smw# cnode update --add-parameter DEBUG=true "*"
   ```

3. Boot the boot node.

   a. Boot the boot node.

      ```
      crayadm@smw> xtcli boot DEFAULT -o bootnode c0-0c0s0n1
      ```

   b. Log in to console.

      If the boot fails, log in to the console to get a shell and debug the problems.

```
smw# xtcon c0-0c0s0n1
```

c. Restart Ansible configuration.

After changing YAML files or other config set files on the SMW, rerun this command on the console.

```
console# /etc/init.d/cray-ansible start
```

4. Boot additional service nodes.

```
crayadm@smw> xtcli boot DEFAULT <node_list>
```

5. Boot compute nodes.

```
crayadm@smw> xtcli boot DEFAULT -o compute p0
```

6. Disable DEBUG boot.

As root, disable debug for the boot parameters of all nodes. This ensures they will not stop the boot process allowing a console login for debugging.

```
smw# cnode update --remove-parameter DEBUG=true "*"
```

# 7 Miscellaneous Installation and Configuration Procedures

Some of these procedures appear in this guide in the context in which they are used and are collected here for easy reference. Others do not fit within the fresh install or software update processes, but are provided here for reference as needed.

- *Back Up Site Data* on page 335
- *Back Up Current Global and CLE Config Sets* on page 337
- *Modify a Config Set for use with Advanced Authentication Configurations* on page 339
- *Install Third-Party Software with a Custom Image Recipe* on page 346
- *Enable Multipath on an Installed XC System* on page 351
- *Change the Time Zone* on page 314
- *Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev* on page 201
- *Shut Down the System* on page 360

## 7.1 Back Up Site Data

This procedure helps sites identify and back up important data from the SMW, boot, and SDB nodes. Back up site data before and after installing new software, depending on circumstances and site policy.

**Before installation** When a fresh install is performed on a system, disks are wiped clean. Before beginning any installation procedures, back up configuration files, log files, or other files that need to be preserved.

**After installation** Sites may also want to archive important SMW and CLE information even if there are no immediate plans to install or reinstall a software release. Saving such information elsewhere will make a later reinstall easier, whether it is planned or part of disaster recovery.

What data should be saved at a particular site depends on several things, such as what is currently installed and where data is stored. A site might have CLE 5.x / SMW 7.x installed, or it might already have CLE 6.0.x / SMW 8.0.x installed and is now planning to do a fresh install and wants to reuse configuration data files. The information to save would be different in each case. And there could be site data in home directories or other parts of the file system unknown to Cray and therefore not listed here. The following suggestions about what data to preserve assume a reinstallation of a CLE 6.x / SMW 8.x release that wipes out an earlier installation of that release.

### SMW Data to Save from a CLE 6.x / SMW 8.x release

**SMW Configuration Data**

**`/var/opt/cray/imps`**

Save the entire directory, which has global config sets (`/var/otp/cray/imps/global`) and CLE config sets (`/var/opt/cray/config/sets/p0`). Saving only the worksheet YAML would miss any site files added for distribution by simple sync or any site Ansible plays. Of particular importance in the global config set is cray_bootraid_config.yaml (or cray_bootraid_worksheel.yaml) which describes how the storage on the Boot RAID is being used.

**`/etc/`**

Save the entire directory. Information related to image recipes is stored in `/etc/opt/cray/imps/image_recipes.d` (especially any site changes to `image_recipes.local.json`) and `/etc/opt/cray/imps/package_collections.d`.

**`/opt/cray/hss/default/etc`**

Save the boot automation files (`/opt/cray/hss/default/etc/auto.*`) and any other files with custom settings.

**`/var/opt/cray/repos`**

Save any site repos which have been created in this directory.

**`/home/crayadm/*fs_defs`**

Save this file if direct-attached Lustre (DAL) was configured.

**`/var/adm/cray/release/pe/install-cdt.yaml`**

Save the PE installer YAML configuration file.

**Command output**

Save output from these commands:

- Are any nodes disabled?

  ```
  smw# xtcli status s0
  ```

- What are the boot and SDB nodes and are any CLE partitions present?

  ```
  smw# xtcli part_cfg show
  ```

**SMW Operational Data**

**`/home`**

Save any user data in this directory, especially in `/home/crayadm`.

**`/var/opt/cray/disk/1`**

Save all files in this directory, which has logs, dumps, and debugging information.

**`/var/opt/cray/imps/image_roots` and `/var/opt/cray/imps/boot_images`**

No need to save data in these two directories as long as the image recipes are saved, because these files can be rebuilt from the image recipes. And when they are rebuilt, they can be pushed to the boot node or CMC (for eLogin).

**`/var/lib/mysql`**

Perform a `mysql dump` of `/var/lib/mysql`. This data will be regenerated by rerunning `xtdiscover`.

### CLE Data to Save from a CLE 6.x / SMW 8.x release

**CLE Boot Node Data**

| | |
|---|---|
| `/var/opt/cray/imps` | No need to save the files in this directory. They are all copies of files on the SMW. |
| `/non-volatile` and `/cray_home` | Save the data in these two directories for possible restoration after the fresh install. |

**CLE SDB Node Data**

| | |
|---|---|
| `/alps_shared` and `/var/lib/mysql` | No need to save the data in these two file systems. It will be regenerated at the first boot with the newly installed software. The only side effect is that all ALPS apids will start over at apid 100. |

# 7.2 Back Up Current Global and CLE Config Sets

## About this task

Sites can back up the current global and CLE config sets as few or as many times as they deem useful. Cray recommends backing up the config sets at these software installation/configuration milestones, which correspond to the suggested milestones for making a snapshot. It is good practice to make a snapshot and back up the config set at the same time to keep them in sync. Cray also recommends naming the snapshot and config set backup using the same suffix and date/time stamp, which helps administrators identify which snapshot and backup pairs belong together.

In the example commands below, replace *`suffix`* with a unique suffix to distinguish among config set backups. Here is a list of suggested suffixes and their associated milestones.

| | |
|---|---|
| **preupdate** | before beginning any software update activities (software update only) |
| **preconfig** | after installing a software update and before updating the global and CLE config sets (software update only) |
| **postinstall** | after installing a new software release (fresh install or software update) and before configuring the SMW for CLE system hardware |
| **postconfig** | after configuring CLE and before booting the CLE system |
| **postboot** | after booting the CLE system |
| **postpe** | after installing Cray PE software |
| **postcustomize** | after customizing a preinstalled system |

## Procedure

**1.** Back up the current global config set.

```
smw# cfgset create --clone global global-suffix-${TODAY}
```

2. Back up the current CLE config set.

   This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

   ```
   smw# cfgset create --clone p0 p0-suffix-${TODAY}
   ```

## 7.3 Set Default Config Set for a NIMS Map

### Prerequisites

This procedure assumes that a NIMS map has been created.

### About this task

Every NIMS map has a default config set, which is the config set that all of the nodes in the map will use if the nodes have not been specifically set to use a different config set. When a NIMS map is created, the default config set can be specified using the `--config-set` option. If the default config set is not specified, then it defaults to the name of the partition associated with that NIMS map. For an unpartitioned system, the default config set would be 'p0,' while for a partitioned system, the default config set for the map associated with partition 1 would be 'p1,' and the default config set for the map associated with partition 2 would be 'p2.'

### Procedure

1. Set the default config set for a NIMS map.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap update -s config_set_name map_name
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cmap update --partition p1 -s p1_config_set_name p1_map_name
   ```

   ```
   smw# cmap update --partition p2 -s p2_config_set_name p2_map_name
   ```

2. Verify that the NIMS map has the specified default config set.

   **full system** For a full, unpartitioned system:

   ```
   smw# cmap list --fields default_config_set map_name
   ```

   **partitioned** For a partitioned system, with partitions p1 and p2:

   ```
   smw# cmap list --fields default_config_set --partition p1 p1_map_name
   ```

   ```
   smw# cmap list --fields default_config_set --partition p2 p2_map_name
   ```

## 7.4 Set Config Set for a Node

### About this task

This procedure sets a config set for an individual node and then verifies the setting. The node will use the specified config set instead of the default config set of the active NIMS map.

### Procedure

1. Set the config set for an individual node.

   ```
   smw# cnode update --set-config-set config_set_name cname
   ```

2. Verify that the node is set to use the specified config set.

   ```
   smw# cnode list --fields config_set cname
   ```

## 7.5 Modify a Config Set for use with Advanced Authentication Configurations

The examples given here show various configurations that enable an XC system to connect to an Active Directory system. Authentication configuration is often implementation-specific, and the examples here should only be used as a reference for modifying a config set for use with an authentication method other than the default LDAP setup.

> **NOTE:** Because of possible boot failure due to an invalid or misconfigured config set, it is recommended to run Ansible and examine the output for improper values once the config set has been modified. Improper values may prevent a system from booting. The following process is recommended when modifying authentication systems on a config set:
>
> 1. Boot system
> 2. Modify config
> 3. Run Ansible to ensure proper values and configuration
> 4. If improper values are returned, revise the config until proper values are attained
> 5. Reboot the system only after ensuring that proper values are returned.

### Using Jinja Substitution and Variables in Config Files

In rare cases it may be necessary to dynamically determine values. For these instances, variable substitution can be used to manipulate config values depending on whether certain conditions are met. Variable statements follow Jinja templating: items in double quotes are treated as literals, while non-quoted items are variables. Begin a variable statement with `{{` and end with `}}`. Ansible variables and facts, as well as config set values, can be used in a variable statement. In this example, variable substitution is used to manipulate *auth_provider.value*:

```
cray_auth.settings.common_ldap_options.data.auth_provider.value: '{{ ''none'' if
ansible_local.cray_system.platform
    == ''compute'' else ''krb5'' }}'
```

## Advanced Authentication Settings: Example 1

This example config shows a system set up for an Active Directory server. In this case, the Active Directory server is a Windows Domain server. Depending on the desired debugger verbosity, a debug level may be set for each config set section. If no *debug_level* is specified, no debugging into will be collected. The `/etc/sssd.conf` file is displayed below:

```
fireball:/etc/sssd # more sssd.conf
[sssd]
config_file_version = 2
services = nss, pam
domains = crayit
debug_level = 7

[nss]
filter_users = root, crayadm
filter_groups = root
debug_level = 7

[pam]
debug_level = 7

[domain/crayit]
override_shell = /bin/bash
override_homedir = /cray_home/%u
ldap_search_base=dc=crayit,dc=com
debug_level = 7
krb5_realm=CRAYIT.COM
krb5_server=crayadserver.us.cray.com
# following allows the system to bind to ldap server via a user/password
ldap_default_authtok=mypassword
ldap_default_authtok_type=password
ldap_default_bind_dn=CN=SLES Test,OU=Users,OU=Cray Objects,DC=crayit,DC=com
#
ldap_group_object_class=group
ldap_user_object_class=user
ldap_uri = ldap://crayadserver.us.cray.com/
id_provider = ldap
auth_provider = krb5
ldap_schema = rfc2307bis

[ssh]
debug_level = 7

[sudo]
```

The corresponding `/etc/nsswitch.conf` file is displayed below. Note that the entries for each key represent the corresponding location a value is first searched for (in this case, `files` is searched first followed by `sss`).

```
fireball:/etc/sssd # grep -v ^# /etc/nsswitch.conf | sort -u
passwd: files sss
group: files sss
hosts: files dns
networks:       files dns
services: files sss
protocols:      files
rpc:            files
ethers:         files
netmasks:       files
```

```
netgroup: files sss
publickey:      files
bootparams:     files
automount: files sss
aliases:        files
shadow: files sss
```

The corresponding `/etc/security/access.conf` file:

```
fireball:/etc/sssd # cat /etc/security/access.conf | grep -v ^#
+:root:LOCAL
+:crayadm:LOCAL
+:@admin_grp:ALL
+:@dev_users:ALL
+:ALL:ALL
```

A sample test run demonstrating Active Directory authentication using the above sample configuration settings:

```
fireball:/etc/sssd # systemctl stop sssd ; rm /var/lib/sss/db/* ; rm /var/log/sssd/
* ; systemctl start sssd

user@user-lnx:~$ ssh -l sles-test
fireball

Password:

*** Welcome to IMPS service node c0-0c0s1n0 (nid 4) ***
    Running 115MB Suse 12 image login-large_cle_rhine_sles_12_x86-64_ari
    CLE release 6.0.UP03, build 6.0.3019(201611291150)
    36 vcores, boot_freemem: 62964mb


 _       _                       _   _
(_)     | |                     | | (_)
 _ _ __ | |_ ___  __ _ _ __ __ _| |_ _  ___  _ __
| | '_ \| __/ _ \/ _` | '__/ _` | __| |/ _ \| '_ \
| | | | | ||  __/ (_| | | | (_| | |_| | (_) | | | |
|_|_| |_|\__\___|\__, |_|  \__,_|\__|_|\___/|_| |_|
                  __/ |
                 |___/

dal_nodes | c0-0c0s1n3
rsip_servers | c0-0c0s1n0
lnet_routers | c0-0c0s1n1
tier2_nodes | c0-0c0s1n2
login_nodes | c0-0c0s1n0
sdb_nodes | c0-0c0s0n2
boot_nodes | c0-0c0s0n1
sles-test@fireball:~> id
uid=10000(sles-test) gid=20000(SLES) groups=20000(SLES)
sles-test@fireball:~> df -k .
Filesystem               1K-blocks   Used Available Use% Mounted on
boot-fireball:/cray_home  52428800 954368  49371136   2% /cray_home
sles-test@fireball:~> pwd
/cray_home/sles-test
sles-test@fireball:~> klist
Ticket cache: FILE:/tmp/krb5cc_10000_lwBoeA
Default principal: sles-test@CRAYIT.COM

Valid starting       Expires              Service principal
11/30/2016 12:47:28  11/30/2016 22:47:28  krbtgt/CRAYIT.COM@CRAYIT.COM
```

```
        renew until 12/01/2016 12:47:27
sles-test@fireball:~> xtprocadmin | grep compute | tail -n 1
    11      0xb  c0-0c0s2n3  compute        up       batch
sles-test@fireball:~> ssh -x -l root c0-0c0s2n3
Password:

*** Welcome to IMPS compute node c0-0c0s2n3 (nid 11) ***
    Running 109MB Suse 12 image compute-large_cle_rhine_sles_12_x86-64_ari
    CLE release 6.0.UP03, build 6.0.3019(201611291150)
    24 vcores, boot_freemem: 62862mb

 _       _                       _   _
(_)     | |                     | | (_)
 _ _ __ | |_ ___  __ _ _ __ __ _| |_ _  ___  _ __
| | '_ \| __/ _ \/ _` | '__/ _` | __| |/ _ \| '_ \
| | | | | ||  __/ (_| | | | (_| | |_| | (_) | | | |
|_|_| |_|\__\___|\__, |_|  \__,_|\__|_|\___/|_| |_|
                  __/ |
                 |___/

dal_nodes | c0-0c0s1n3
rsip_servers | c0-0c0s1n0
lnet_routers | c0-0c0s1n1
tier2_nodes | c0-0c0s1n2
login_nodes | c0-0c0s1n0
sdb_nodes | c0-0c0s0n2
boot_nodes | c0-0c0s0n1
nid00011:~ # id sles-test
uid=10000(sles-test) gid=20000(SLES) groups=20000(SLES)
nid00011:~ # su - sles-test
sles-test@nid00011:~> pwd
/cray_home/sles-test
sles-test@nid00011:~>
```

Sample config set data from this example:

```
fireball-smw:/var/opt/cray/imps/config/sets/p0.integration/worksheets # grep -v ^#
cray_auth_worksheet.yaml
cray_auth.enabled: true
cray_auth.settings.nsswitch_sources.data.database.automount: null
cray_auth.settings.nsswitch_sources.data.automount.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.group: null
cray_auth.settings.nsswitch_sources.data.group.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.hosts: null
cray_auth.settings.nsswitch_sources.data.hosts.sources: files dns
cray_auth.settings.nsswitch_sources.data.database.netgroup: null
cray_auth.settings.nsswitch_sources.data.netgroup.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.passwd: null
cray_auth.settings.nsswitch_sources.data.passwd.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.services: null
cray_auth.settings.nsswitch_sources.data.services.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.shadow: null
cray_auth.settings.nsswitch_sources.data.shadow.sources: files sss
cray_auth.settings.common_ldap_options.data.option.id_provider: null
cray_auth.settings.common_ldap_options.data.id_provider.value: ldap
cray_auth.settings.common_ldap_options.data.option.auth_provider: null
cray_auth.settings.common_ldap_options.data.auth_provider.value: krb5
cray_auth.settings.common_nis_options.data.option.id_provider: null
cray_auth.settings.common_nis_options.data.id_provider.value: proxy
cray_auth.settings.common_nis_options.data.option.auth_provider: null
cray_auth.settings.common_nis_options.data.auth_provider.value: proxy
cray_auth.settings.common_nis_options.data.option.proxy_lib_name: null
cray_auth.settings.common_nis_options.data.proxy_lib_name.value: nis
cray_auth.settings.common_nis_options.data.option.proxy_pam_target: null
cray_auth.settings.common_nis_options.data.proxy_pam_target.value: sssdnisproxy
cray_auth.settings.domain.data.reference.crayit: null
cray_auth.settings.domain.data.crayit.servers:
- ldap://crayadserver.us.cray.com/
cray_auth.settings.domain.data.crayit.schema: rfc2307bis
cray_auth.settings.domain.data.crayit.aux_settings:
- ldap_user_object_class=user
- ldap_group_object_class=group
- ldap_default_bind_dn=CN=SLES Test,OU=Users,OU=Cray Objects,DC=crayit,DC=com
- ldap_default_authtok_type=password
- ldap_default_authtok=mypassword
- krb5_server=crayadserver.us.cray.com
- krb5_realm=CRAYIT.COM
- debug_level = 7
- ldap_search_base=dc=crayit,dc=com
cray_auth.settings.access.data.policy:
- +:root:LOCAL
- +:crayadm:LOCAL
- +:@admin_grp:ALL
- +:@dev_users:ALL
cray_auth.settings.access.data.restrictive: false
cray_auth.settings.access.data.config_computes: true
cray_auth.settings.access.data.config_id_service_groups: []
cray_auth.settings.section.data.section_name.sssd: null
cray_auth.settings.section.data.sssd.options.option_name.debug_level: null
cray_auth.settings.section.data.sssd.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.nss: null
cray_auth.settings.section.data.nss.options.option_name.debug_level: null
cray_auth.settings.section.data.nss.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.ssh: null
cray_auth.settings.section.data.ssh.options.option_name.debug_level: null
cray_auth.settings.section.data.ssh.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.sudo: null
cray_auth.settings.section.data.sudo.options.option_name.debug_level: null
cray_auth.settings.section.data.sudo.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.pam: null
cray_auth.settings.section.data.pam.options.option_name.debug_level: null
cray_auth.settings.section.data.pam.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.autofs: null
cray_auth.settings.section.data.autofs.options.option_name.debug_level: null
cray_auth.settings.section.data.autofs.options.debug_level.value: '7'
cray_auth.config_file: /var/opt/cray/imps/config/sets/p0.integration/config/
cray_auth_config.yaml
cray_auth.configurator.version_template: 1.2.1
cray_auth.configurator.version_product: 0.0.0
cray_auth.configurator.version_configurator: 3.0.0
cray_auth.configurator.template_type:
- cle
fireball-smw:/var/opt/cray/imps/config/sets/p0.integration/worksheets #
```

## Advanced Authentication Settings: Example 2

Example 2 demonstrates an alternative configuration method for using Active Diretory. In this example, a `/etc/krb5.keytab` file has been generated utilizing a temporary Samba configuration and a user account that has permissions to join the host to the domain. The `/etc/krb5.keytab` file was put into place on the node via Simple Sync. Because in this example there is no keytab file each computer node, *`config_computes`* must be set to `false`. This example uses the same `/etc/security/access.conf` and `/etc/nsswitch.conf` files as Example 1. The `sssd.conf` file is displayed below:

```
=========/etc/sssd/sssd.conf======================
[sssd]
debug_level = 7
config_file_version = 2
services = nss, pam
domains = crayit

[nss]
debug_level = 7
filter_users = root, crayadm
filter_groups = root

[domain/crayit]
debug_level = 7
id_provider = ad
auth_provider = ad
access_provider = ad
chpass_provider=ad
ad_server = crayadserver.us.cray.com
ad_domain = CRAYIT.COM
default_shell = /bin/bash
#fallback_homedir = /home/%d/%u
use_fully_qualified_names = False
dyndns_update = false

[ssh]
debug_level = 7

[sudo]
debug_level = 7

[pam]
debug_level = 7

[autofs]
debug_level = 7
```

The corresponding config set:

```
cray_auth.enabled: true
cray_auth.settings.nsswitch_sources.data.database.automount: null
cray_auth.settings.nsswitch_sources.data.automount.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.group: null
cray_auth.settings.nsswitch_sources.data.group.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.hosts: null
cray_auth.settings.nsswitch_sources.data.hosts.sources: files dns
cray_auth.settings.nsswitch_sources.data.database.netgroup: null
cray_auth.settings.nsswitch_sources.data.netgroup.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.passwd: null
cray_auth.settings.nsswitch_sources.data.passwd.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.services: null
cray_auth.settings.nsswitch_sources.data.services.sources: files sss
cray_auth.settings.nsswitch_sources.data.database.shadow: null
cray_auth.settings.nsswitch_sources.data.shadow.sources: files sss
cray_auth.settings.common_ldap_options.data.option.id_provider: null
cray_auth.settings.common_ldap_options.data.id_provider.value: ad
cray_auth.settings.common_ldap_options.data.option.auth_provider: null
cray_auth.settings.common_ldap_options.data.auth_provider.value: ad
cray_auth.settings.common_nis_options.data.option.id_provider: null
cray_auth.settings.common_nis_options.data.id_provider.value: proxy
cray_auth.settings.common_nis_options.data.option.auth_provider: null
cray_auth.settings.common_nis_options.data.auth_provider.value: proxy
cray_auth.settings.common_nis_options.data.option.proxy_lib_name: null
cray_auth.settings.common_nis_options.data.proxy_lib_name.value: nis
cray_auth.settings.common_nis_options.data.option.proxy_pam_target: null
cray_auth.settings.common_nis_options.data.proxy_pam_target.value: sssdnisproxy
cray_auth.settings.domain.data.reference.crayit: null
cray_auth.settings.domain.data.crayit.servers: []
cray_auth.settings.domain.data.crayit.schema: ''
cray_auth.settings.domain.data.crayit.aux_settings:
- access_provider = ad
- chpass_provider = ad
- ad_server = crayadserver.us.cray.com
- ad_domain = CRAYIT.COM
- default_shell = /bin/bash
- use_fully_qualified_names = False
- dyndns_update = false
- override_homedir = /cray_home/%u
cray_auth.settings.access.data.policy:
- +:root:LOCAL
- +:crayadm:LOCAL
- +:@admin_grp:ALL
- +:@dev_users:ALL
cray_auth.settings.access.data.restrictive: false
cray_auth.settings.access.data.config_computes: false
cray_auth.settings.access.data.config_id_service_groups: []
cray_auth.settings.section.data.section_name.sssd: null
cray_auth.settings.section.data.sssd.options.option_name.debug_level: null
cray_auth.settings.section.data.sssd.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.nss: null
cray_auth.settings.section.data.nss.options.option_name.debug_level: null
cray_auth.settings.section.data.nss.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.ssh: null
cray_auth.settings.section.data.ssh.options.option_name.debug_level: null
cray_auth.settings.section.data.ssh.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.sudo: null
cray_auth.settings.section.data.sudo.options.option_name.debug_level: null
cray_auth.settings.section.data.sudo.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.pam: null
cray_auth.settings.section.data.pam.options.option_name.debug_level: null
cray_auth.settings.section.data.pam.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.autofs: null
cray_auth.settings.section.data.autofs.options.option_name.debug_level: null
cray_auth.settings.section.data.autofs.options.debug_level.value: '7'
cray_auth.settings.section.data.section_name.domain/crayit: null
cray_auth.settings.section.data.domain/crayit.options.option_name.debug_level: null
cray_auth.settings.section.data.domain/crayit.options.debug_level.value: '7'
cray_auth.config_file: /var/opt/cray/imps/config/sets/p0.integration/config/
cray_auth_config.yaml
cray_auth.configurator.version_template: 1.2.1
cray_auth.configurator.version_product: 1.0.0
cray_auth.configurator.version_configurator: 3.0.0
cray_auth.configurator.template_type:
- cle
```

Sample output demonstrating authentication functionality:

```
fireball:/etc/sssd # systemctl stop sssd ; rm /var/lib/sss/db/* ; rm /var/log/sssd/
* ; systemctl start sssd
fireball:/etc/sssd # ssh -x -l sles-test localhost
Password:
Last login: Fri Sep 32 17:31:39 2021 from localhost

*** Welcome to IMPS service node c0-0c0s1n0 (nid 4) ***
    Running 116MB Suse 12 image login-large_cle_rhine_sles_12_x86-64_ari
    CLE release 6.1.DV00, build 6.1.262DV(201609300100)
    36 vcores, boot_freemem: 62978mb

 ___ ___   ____  _____ ______    ___  ____
|   |   | /    |/ ___/|      |  /  _]|    \
| _   _ ||  o  (   \_ |      | /  [_ |  D  )
|  \_/  ||     |\__  ||_|  |_||    _]|    /
|   |   ||  _  |/  \ |  |  |  |   [_ |    \
|   |   ||  |  |\    |  |  |  |     ||  .  \
|___|___||__|__| \___|  |__|  |_____||__|\_|

dal_nodes | c0-0c0s1n3
rsip_servers | c0-0c0s0n0
lnet_routers | c0-0c0s1n1
tier2_nodes | c0-0c0s1n2
login_nodes | c0-0c0s1n0
sdb_nodes | c0-0c0s0n2
boot_nodes | c0-0c0s0n1
Could not chdir to home directory /home/sles-test: No such file or directory
sles-test@fireball:/> groups sles-test
sles-test : domain users sles
sles-test@fireball:/>
```

# 7.6 Install Third-Party Software with a Custom Image Recipe

## About this task

Any software that is created independent from Cray *and* that is not delivered with a Cray system is third-party software that administrators install as add-ons to the Cray system. (The information in this section does not pertain to software installed on an external file system that is connected to a Cray system.) There are several ways to install third-party software:

- Add a third-party software package to a custom image recipe (recommended).
- Use the `chroot` command to install the software to an existing image.
- Use the `zypper` command to install software on a node.

> **IMPORTANT:** Do not directly modify a Cray-provided recipe.

Installing software with a custom image recipe is the best method to use because the update to the image is persisted in the recipe. Each time the recipe is built into an image root, packaged into a boot image, and then the node is booted from the boot image, the third-party software is available. This approach follows the ideal model of a *prescriptive recipe*. Using `chroot` or `zypper` to install software is usually less desirable because the installations are not persisted. Software installed using `zypper` is lost the next time the node is booted; software installed with `chroot` is lost when a node image is rebuilt from a recipe. However, using the `chroot` or `zypper` method can be useful when persistence is not important, such as when testing third-party software.

Cray recommends customizing a Cray-provided recipe by creating a new recipe, then adding the Cray-provided recipe as a subrecipe. It is possible to clone a Cray-provided recipe and modify the clone, but cloned recipes do not receive updates from patches. For more information, see *About the Image Management and Provisioning System (IMPS)*.

The following procedure shows how to create a new image recipe, add a Cray-provided image recipe as a sub-recipe, then add the third-party RPM and non-RPM content for the image. This procedure starts with adding the RPMs to a package collection and repository. After the image recipe is defined, the procedure describes how to use the IMPS `image` command build an image root, export the image root into a boot image, and test it on a single node, then assign it to all applicable nodes. (After a recipe has been defined and tested, the `imgbuilder` command can be used to rebuild boot images.)

## Procedure

**1.** Create a new repository and add the third-party packages (RPMs).

a. Use the `repo create` command to create the new repository (for example, `my_sles12_repo`). This command requires the architecture (such as `x86-64`) and operating system type (either `SLES12` or `CentOS`).

```
smw# repo create --arch x86-64 --type SLES12 my_sles12_repo
```

b. Verify that the new repository was created.

```
smw# repo list my*
my_sles12_repo
```

c. Add the third-party RPMs to the repository. This example takes all RPMs starting with `myrpm` in the example repository path */path/to/repos/* and copies them to the example repo `my_sles12_repo`.

```
smw# repo update -a "/path/to/repos/myrpm*.rpm" my_sles12_repo
smw# ls -l /var/opt/cray/repos/my_sles12_repo
-rw-r--r-- 1 crayadm crayadm 485137 Nov 23 08:56 myrpm-11.13.1.1-4.x86_64.rpm
```

d. Validate the repository.

```
smw# repo validate my_sles12_repo
```

**2.** Create a package collection and add the RPM package names.

A package collection represents a logical grouping of RPMs. Cray recommends using a package collection because the RPMs can be used in multiple image types (such as compute and service node images). Package collections are stored on the SMW in `/etc/opt/cray/imps/package_collections.d/`.

a. Create an empty package collection (for example, `my_collection`).

```
smw# pkgcoll create my_collection
```

b. Verify that the package collection was created.

```
smw# pkgcoll list my*
my_collection
```

c. Add the RPM package name or names (for example, `myrpm`) to the package collection.

```
smw# pkgcoll update -p myrpm my_collection
```

d. Display information about the package collection.

```
smw# pkgcoll show my_collection
my_collection:
name: my_collection
package_collections:
myrpm
```

**3.** Create a new recipe and customize it by adding a sub-recipe (the Cray-provided image) and the content for the third-party software.

a. List the existing recipes to determine which image recipe to include.

```
smw# recipe list
compute-large_cle_6.0up01_sles_12_x86-64_ari
compute-large_cle_6.0up02_sles_12_x86-64_ari
compute-large_cle_6.0up03_sles_12_x86-64_ari
compute_cle_6.0up01_sles_12_x86-64_ari
compute_cle_6.0up02_sles_12_x86-64_ari
compute_cle_6.0up03_sles_12_x86-64_ari
dal_cle_6.0up01_centos_6.5_x86-64_ari
dal_cle_6.0up02_centos_6.5_x86-64_ari
dal_cle_6.0up03_centos_6.5_x86-64_ari
elogin_cle_6.0up01_sles_12_x86-64_ari
...
```

b. Create a new image recipe. This example uses the recipe name `site_compute`.

```
smw# recipe create site_compute
```

c. Add the existing image recipe as a sub-recipe. This example uses the Cray-provided recipe `compute_cle_6.0up03_sles_12_x86-64_ari`.

```
smw# recipe update -i compute_cle_6.0up03_sles_12_x86-64_ari site_compute
```

d. Add the package collection that contains the third-party RPMs (in this example, `my_collection`).

```
smw# recipe update -c my_collection site_compute
```

e. Add the repository that contains the third-party RPMs (for example, `my_sles12_repo`).

```
smw# recipe update -r my_sles12_repo site_compute
```

f. Add the objects mentioned in the sub-recipe that are also needed for the parent recipe.

> **IMPORTANT:** The objects mentioned in a sub-recipe are used to build that sub-recipe but are not available to the parent recipe. If a package (RPM) or package collection is specified in the parent recipe, the custom recipe must explicitly contain the set of repositories where the packages can be found.

**1.** Determine which repository contains the necessary RPM or RPMs. This example `find` command identifies the Cray repository that contains the RPM `otherrpm`.

```
smw# find /var/opt/cray/repos -name otherrpm\* -ls
```

**2.** Select the correct repository:

- Choose the repository for the image's operating system type — use a SLES repository for a SLES image recipe; use a CentOS repository for a CentOS recipe.
- Most operating system and Cray repositories come in pairs (base and updates), such as `sles_12_x86-64` and `sles_12_x86-64_updates`. Be sure to select both the *`base`* and *`base`*`_updates` repositories if they exist.

**3.** Add the required repository or repositories (in this example, `otherrepo`).

```
smw# recipe update -r otherrepo site_compute
```

Repeat the `-r` option to add multiple repositories, such as a *`base`* and *`base`*`_updates` repository pair.

```
smw# recipe update -r sles_12_x86-64 -r sles_12_x86-64_updates \
site_compute
```

g. (Optional.) Use post-build actions to add non-RPM content to the recipe. For example, post-build actions could include copying a tar file into the image then using `chroot` to run the commands to untar it and run an install script.

To add post-build actions, manually edit the image recipe: Open the local image recipe `/etc/opt/cray/imps/image_recipes.d/image_recipes.local.json`. Locate the image recipe definition for the custom image (for example, `site_compute`). In the `postbuild_copy` section, add the files to copy into the image. In the `postbuild_chroot` section, add the commands to run in a `chroot` environment for this image root.

```
smw# vi /etc/opt/cray/imps/image_recipes.d/image_recipes.local.json
"site_compute": {
        ...
        "package_collections": { ... },
        "packages": { ... },
        "postbuild_chroot": [
            "chroot_command1",
            ...
            "chroot_commandN"
        ],        "postbuild_copy": [
            "/file/1",
            ...
            "/dir/2/content"
        ],

        ”recipes": { ... },
        "repositories": { ... }
    },
```

**TIP:** Post-build scripts can use the following environmental variables:

- `IMPS_IMAGE_NAME`
- `IMPS_VERSION`
- `IMPS_IMAGE_RECIPE_NAME`
- `IMPS_POSTBUILD_FILES`

h. Validate the image recipe.

```
smw# recipe validate site_compute
INFO - Repository 'my_sles12_repo' validates.
INFO - Recipe 'site_compute' is valid.
```

This command checks that the JSON syntax of the image recipe is correct. It also validates all repositories and package collections referenced by the image recipe and ensures that it can access any files in the `postbuild_copy` section.

**4.** Build the image recipe to create the image root. For the image root name, Cray recommends using the image recipe name plus the current date/time. This example shows the image root name `site_compute_`*`timestamp`*.

The `image create` command builds the image recipe starting with the package manager installation and then proceeds to step through the postbuild copy and `chroot` commands (in that order).

```
smw# image create -r site_compute site_compute_timestamp
INFO - Repository 'my_sles12_repo' validates.
INFO - Recipe 'site_compute' is valid for building.
INFO - Calling Package manager to build new image root; this will take a few
minutes.
INFO - Rebuilding RPM database for Image 'site_compute_timestamp'.
INFO - RPM database does not need to be rebuilt.
INFO - Running post-build scripts for Image 'site_compute_timestamp'.
INFO - Copying postbuild files to /tmp/tmpmAyzGl in Image
'site_compute_timestamp'
INFO - * Executing post-build chroot script: 'chroot_command1'
INFO - post-build chroot script output will be located in /tmp/
site_compute_postbuild_out_20150713-15:55:11g4WA6p
INFO - Build of Recipe 'site_compute' has completed successfully.
```

**5.** (Optional.) Display the build history of the image root.

```
smw# image show site_compute_timestamp
 site_compute_timestamp:
         name: site_compute_timestamp
         created: 2016-07-13T15:54:06
         history:
              2016-07-13T15:55:16:        Successful build of Recipe
                 'site_compute into Image 'site_compute_timestamp'.
              2016-07-13T15:55:17:        Successful rebuild of RPM database.
          path: /var/opt/cray/imps/image_roots/site_compute_timestamp
```

**6.** Package the image root into a boot image.

```
smw# image export site_compute_timestamp

INFO - Copying kernel /var/opt/cray/imps/image_roots/site_compute_timestamp/boot/
bzImage-3.12.28-4.6_1.0000.8685-cray_ari_c into /tmp/temp_tempfs_50LJ93/DEFAULT
INFO - Copying parameters file /var/opt/cray/imps/image_roots/site_compute_timestamp/
boot/parameters-ari_c into /tmp/temp_tempfs_50LJ93/DEFAULT
 .
 .
 .
INFO - Image 'site_compute_timestamp' has been packaged into /var/opt/cray/imps/
boot_images/site_compute_timestamp.cpio.
```

The `image export` command displays the boot image file name at the end of the output. This `cpio` file is used in the next step.

**7.** Test the new boot image on a single node.

a. Assign the boot image to a node with the NIMS `cnode` command. This example assigns the boot image file `site_compute_`*`timestamp`*`.cpio` (in the directory `/var/opt/cray/imps/boot_images/`) to the compute node with the cname `c0-0c0s15n3`.

```
smw# cnode update -i \
/var/opt/cray/imps/boot_images/site_compute_timestamp.cpio c0-0c0s15n3
```

b. Warm-boot the node to test the boot image.

```
smw#  xtcli shutdown c0-0c0s15n3
.
.
.
crayadm@smw> xtbootsys --reboot \
-r "testing new boot image site_compute_timestamp" c0-0c0s15n3
```

**8.** Assign the new boot image to all applicable nodes.

```
smw# cnode update --group compute \
-i /var/opt/cray/imps/boot_images/site_compute_timestamp.cpio
```

**9.** Choose when the compute nodes should switch to the new image.

- To immediately use the new image, warm-boot all applicable nodes with the new image. This example specifies the compute nodes as a comma-separated list of cnames; see the `xtcli(8)` man page for other ways of specifying multiple nodes.

```
smw# xtcli shutdown cname, cname, ... cname
.
.
.

smw# xtbootsys --reboot -r "Booting custom image on all compute nodes" \
cname, cname, ... cname
```

- To have the workload manager (WLM) reboot the node once the current user's job finishes, see *Apply Rolling Patches to Compute Nodes with cnat*.
- Otherwise, wait until the next full system reboot. The nodes will boot with the new image.

## 7.7 Enable Multipath on an Installed XC System

### Prerequisites

This procedure assumes that the Cray XC system has already been installed and configured without multipath having been enabled. If performing a fresh install, this procedure is not necessary if multipath was already set up using *Prepare and Update the Global Config Set* on page 99 or *Update cray_multipath Worksheet* on page 143.

### About this task

This procedure describes how to enable multipath on a Cray XC system that has already been installed and configured. Note that multipath does NOT need to be fully cabled to be used. The multipath driver can handle using one path or many.

**IMPORTANT:** If this system has partitions, repeat any steps that modify 'p0' for each partition. Multipath must be enabled everywhere or nowhere; enabling it on only part of the system causes problems.

## Procedure

1. Start the multipath daemon now.

   ```
   smw# systemctl start multipathd
   ```

   Later in this procedure, the `cray-ansible` command will be used to enable the multipath daemon.

2. Obtain the host ID of the SMW and the cnames of any nodes in the system that are connected to the boot RAID with an HBA (host bus adapter).

   The system should be bounced or booted for `xtcheckhss` to return a proper list.

   ```
   smw# hostid
   {8 digit hostid}
   smw# xtcheckhss --detail=f --pci
   ```

   Look for cnames with HBAs like 'QLogic_ISP2532_8Gb_Fibre_Channel_HBA.'

3. Use the configurator to update cray_multipath in the global config set.

   ```
   smw# cfgset update -s cray_multipath -m interactive global
   ```

   a. Enable multipath.

      Enter **`E`** at the configurator prompt to toggle the enable status of the multipath service, which is disabled by default.

      ```
      Cray Multipath Configuration Service Menu [default: save & exit - Q] $ E
      ```

   b. Add the host ID and cnames obtained in step *2*.

      At the prompt, enter **`1`** to select the node_list setting, then enter **`C`** to configure it. At the prompt for that setting, enter values **+** to add node_list entries: add the host IDs and cnames obtained in step *2*, one per line. When finished, press **Ctrl-d** and then **`<cr>`** to set the entries.

      ```
      Cray Multipath Configuration Service Menu [default: save & exit - Q] $ 1
      ...
      Cray Multipath Configuration Service Menu [default: configure - C] $ C
      ...
      cray_multipath.settings.multipath.data.node_list
      [<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
      Add node_list (Ctrl-d to exit) $
      ```

   c. (If system running CLE 6.0.UP03 or earlier release) Correct the values of three pre-populated device settings.

      At the prompt, enter **`33`** to select the enable_devices setting, then enter **`C`** to configure it.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ 33

Cray Multipath Configuration Service Menu [default: configure - C] $ C
```

At the prompt for this setting, enter `*` to view all of the pre-populated device settings.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ *
```

Find the DDN_EF3015 device in the list of enabled devices, and enter its number (5 in this example) followed by 'd' and '*' to select and edit the path_grouping_policy field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 5d*
```

Change the value to **`group_by_prio`**.

```
cray_multipath.settings.enabled_devices.data.DDN_EF3015.path_grouping_policy
[<cr>=keep 'multibus', <new value>, ?=help, @=less] $ group_by_prio
```

Find the DDN_SFA12K_20 device in the list of enabled devices, and enter its number (10 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 10b*
```

Change the value to **`SFA12K-20`**.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_20.product
[<cr>=keep 'SFA12K20', <new value>, ?=help, @=less] $ SFA12K-20
```

Find the DDN_SFA12K_40 device in the list of enabled devices, and enter its number (11 in this example) followed by 'b' and '*' to select and edit the product field.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ 11b*
```

Change the value to **`SFA12K-40|SFA12KX*`**.

```
cray_multipath.settings.enabled_devices.data.DDN_SFA12K_40.product
[<cr>=keep 'SFA12K40', <new value>, ?=help, @=less] $ SFA12K-40|SFA12KX*
```

Set the enabled_devices entries, then save changes and exit the configurator.

```
cray_multipath.settings.enabled_devices
[<cr>=set 11 entries, +=add an entry, ?=help, @=less] $ <cr>
```

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ Q
```

**4.** Use the configurator to update cray_bootraid in the global config set.

```
smw# cfgset update -s cray_bootraid -m interactive global
```

a. Select the storage sets setting to configure it.

```
Boot RAID Configuration Service Menu [default: save & exit - Q] $ 1
...
```

```
Boot RAID Configuration Service Menu [default: configure - C] $ C
```

b. For each device in the cledefault and smwdefault storage sets, modify the path name from scsi to dm-uuid-mpath.

This example shows selecting the cledefault (1) volume group (a) boot_node_vg (1) devices (b) field. The * indicates that the selection is to be edited.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ 1a1b*
```

Remove the "scsi path name, and replace it with the dm-uuid-mpath name.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_v
g.devices
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ 1-

cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_v
g.devices
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add devices (Ctrl-d to exit) $ /dev/disk/by-id/dm-uuid-
mpath-3600a0980009ec0750000010a5762af70
Add devices (Ctrl-d to exit) $ <Ctrl-d>
```

Set the entries for the boot_node_vg volume group.

```
cray_bootraid.settings.storage_sets.data.cledefault.volume_groups.boot_node_v
g.devices
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <cr>
```

Repeat substep b for each device in the cledefault and smwdefault storage sets. Enter * at the prompt to see all storage set entries.

- To select the next cledefault volume group device (sdb_node_vg), enter **`1a2b*`** at the prompt. If there are more cledefault volume groups, increment the third character to select each one (**`1a3b*`**, **`1a4b*`**, and so forth).
- To select the first smwdefault volume group device (smw_node_vg), enter **`2a1b*`** at the prompt. If there are more smwdefault volume groups, increment the third character to select each one (**`2a2b*`**, **`2a3b*`**, and so forth).

c. Set the storage set entries, then save changes and exit the configurator.

```
cray_bootraid.settings.storage_sets
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
...
Boot RAID Configuration Service Menu [default: save & exit - Q] $ Q
```

**5.** Use the configurator to set up inheritance for multipath in the CLE config set of the active SMW.

This example uses 'p0' as the name of the CLE config set. Substitute the actual name used at this site.

```
smw# cfgset update -s cray_multipath -m interactive p0
```

Enter **I** at the configurator prompt to toggle the inherit status of the multipath service, which is disabled by default. This means that multipath settings in the global config set will be used instead of multipath settings in the CLE config set.

```
Cray Multipath Configuration Service Menu [default: save & exit - Q] $ I
```

Repeat this step for each CLE config set.

6. Validate the config sets and run cray-ansible to apply the config set changes.

   a. Validate the config sets.

   ```
   smw# cfgset validate global

   smw# cfgset validate p0
   ```

   b. Run cray-ansible.

   ```
   smw# /etc/init.d/cray-ansible start
   ```

7. For systems using direct-attached Lustre (DAL), update the `dal.fs_defs` file.

   Repeat these steps for each partition.

   a. Locate the current `fs_defs` files (typically stored in `/home/crayadm`).

   ```
   smw# find /home/crayadm -name "*fs_defs*"
   ```

   b. Find the `fs_defs` files that are currently installed and compare with the one found in `/home/crayadm`.

   ```
   smw# find /var/opt/cray/imps/config/sets/p0 -name "*fs_defs*"
   ```

   ```
   smw# diff /home/crayadm/dal.fs_defs /var/opt/cray/\
   imps/config/sets/p0/lustre/.lctrl/dal.fs_defs.20160205.1454685527
   ```

   c. Edit the `dal.fs_defs` file to ensure that it has the proper mpath paths in it.

   ```
   smw# cd /home/crayadm

   smw# sed -i.nompath 's/\/dev\/disk\/by-id\/scsi/\/dev\/disk\/by-id\/dm-uuid-
   mpath/g' \
   dal.fs_defs

   smw# cp -p dal.fs_defs dal.fs_defs.mpath
   ```

   d. Install the new `dal.fs_defs` file using `lustre_control`.

   ```
   smw# lustre_control install -c p0 /home/crayadm/dal.fs_defs
   ```

8. Shut down all partitions of the Cray system.

9. Reboot the SMW.

10. Boot the Cray system.

# 7.8 Change the Time Zone

## Prerequisites

This procedure assumes that the XC system is booted.

## About this task

This procedure changes the time zone of an XC system by changing some configuration and then rebooting components. Most of these commands must be performed as root.

## Procedure

1. Check the current time zone.

   a. Check time zone on SMW.

   ```
   smw# date
   ```

   b. Check time zone on cabinet and blade controllers.

   ```
   smw# xtrsh -l root -s date
   ```

   c. Check time zone on boot node.

   ```
   smw# ssh boot date
   ```

   d. Check time zone on SDB node.

   This command works from the SMW if the SDB node is a tier1 node with an Ethernet connection to the SMW.

   ```
   smw# ssh sdb date
   ```

   e. Check time zone on all service nodes.

   ```
   smw# ssh sdb pcmd -r -n ALL_SERVICE_NOT_ME "date"
   ```

   f. Check time zone on all compute nodes.

   ```
   smw# ssh sdb pcmd -r -n ALL_COMPUTE "date"
   ```

   Continue to the next step only if the time zone needs to be changed.

2. Change the SMW local time zone, if needed.

   The default time zone on the SMW is **America/Chicago**. To change it:

   a. Execute this command:

```
smw# yast2 timezone
```

`yast2` opens a new window for changing the time zone, then a pop-up window appears with this message: "file `/etc/ntp.conf` has been changed manually. YaST might lose some of the changes."

b. Select the **Do not show this message anymore** checkbox, then select **Continue**.

c. Choose the time zone either by selecting a region on the map or by using the drop-down menus for **Region** and **Time Zone**.

d. Select **Other Settings** if the time is incorrect, then select the **Manually** radio button and enter **Current Time** and **Current Date**. Select **Accept** when done.

e. Select **OK** when done with time zone settings.

The change on the SMW is immediate, but any users on the system need to log out and then log in again to get the new environment. This does not change the time zone for the CLE nodes or the cabinet and blade controllers. Continue to step 3 to make those changes.

**3.** Change the time zone in the global config set.

a. Set `cray_time.settings.service.data.timezone` to the desired time zone.

A list of possible time zones is available on the SMW in `/usr/share/zoneinfo/zone1970.tab`.

```
smw# cfgset update -s cray_time -m interactive global
```

b. Validate the config set.

```
smw# cfgset validate global
```

**4.** Change the time zone in the CLE config set.

If the CLE config set has `cray_time.inherit` set to true, then the time zone and other time settings from the global config set will be inherited by the CLE config set. If the CLE config set has `cray_time.inherit` set to false, then use these commands to change the setting and validate the config set.

a. Set `cray_time.settings.service.data.timezone` to the desired time zone.

A list of possible time zones is available on the SMW in `/usr/share/zoneinfo/zone1970.tab`.

```
smw# cfgset update -s cray_time -m interactive p0
```

b. Validate the config set.

```
smw# cfgset validate p0
```

**5.** Put the SMW time zone setting where the cabinet and blade controllers can access it.

```
smw# cp /etc/localtime /opt/tftpboot/localtime
```

**6.** Reboot to set the new time zone for all components.

a. Shut down CLE.

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.hostname.stop
```

b. Reboot the SMW and verify that the time zone has been reset..

```
crayadm@adm> su - root
smw# reboot
```

After the SMW reboots, check that the SMW has the desired time zone setting.

```
smw# date
```

c. Reboot the cabinet controllers, then verify that all cabinet controllers are up.

```
smw# xtccreboot -c all

smw# sleep 120

smw# xtalive -a l1sysd -l l1 s0
```

Repeat the `xtalive` command until all cabinet controllers are alive.

d. Reboot the blade controllers, then verify that all blade controllers are up.

```
smw# xtccreboot -b all

smw# sleep 120

smw# xtalive s0
```

Repeat the `xtalive` command until all blade controllers are alive.

e. Boot CLE nodes for the new time zone using the site boot automation file.

```
crayadm@smw> xtbootsys -a auto.hostname.start
```

**7.** Check the current time zone again.

a. Check time zone on SMW.

```
smw# date
```

b. Check time zone on cabinet and blade controllers.

```
smw# xtrsh -l root -s date
```

c. Check time zone on boot node.

```
smw# ssh boot date
```

d. Check time zone on SDB node.

This command works from the SMW if the SDB node is a tier1 node with an Ethernet connection to the SMW.

```
smw# ssh sdb date
```

e. Check time zone on all service nodes.

```
smw# ssh sdb pcmd -r -n ALL_SERVICE_NOT_ME "date"
```

f. Check time zone on all compute nodes.

```
smw# ssh sdb pcmd -r -n ALL_COMPUTE "date"
```

If these checks show the correct time zone, then the time zone has been successfully changed.

## 7.9 Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev

### Prerequisites

To run `xtgetrev`, the boot node must be booted and accessible.

### About this task

System administrators use the `xtgetrev` and `xtshowrev` commands to gather and display machine, software revision, Field Notice (FN), and patch set information. The `xtgetrev` command collects information from the administrator and from the SMW and boot node. The `xtshowrev` command displays that information, even when CLE is not running. These tools are useful for gathering information to send to Cray after installing a software upgrade, FN, or patch set and for help with troubleshooting.

This procedure describes how to use these two tools on a Cray XC Series system. These steps (except for running `xtshowrev`) must be executed as root.

> **ATTENTION:** Any information that is submitted to site_install_data@cray.com will be used only within Cray, Inc. and will not be made public. The `xtshowrev` command does not submit any information to Cray automatically.

### Procedure

1. Load the module to enable use of the tools.

   ```
   smw# module load xtshowrev
   ```

2. Run `xtgetrev` to create and populate the initial files.

   Only root can run this command. The first time `xtgetrev` is executed, when there are no files populated, the tool will prompt for site information. If the boot node does not have passwordless ssh, then the tool will prompt for the password.

   This example uses *`CRAY/INTERNAL`* as the site name and *`9999`* as the serial number of the machine. Substitute the actual values for this site.

   ```
   smw# xtgetrev
   xtgetrev: No site information has been defined.

   Site name: CRAY/INTERNAL
   Serial Number: 9999
   System Name [panda1]:
   System Type [XC40]:

   <snip>
   ```

   **Trouble?** If `xtgetrev` does not allow entry of those values, it may be because the initial configuration files have been created already. In that case, manually edit `/etc/opt/cray/release/pkginfo/site_config` and modify 'site name:' and 'serial number:' values.

```
smw# vi /etc/opt/cray/release/pkginfo/site_config
```

3. Run `xtshowrev` to see the formatted information.

   Note the prompt, which indicates that any user can run this command.

```
user@smw> xtshowrev
Site:                    CRAY/INTERNAL
S/N:                     9999
System Type:             XC40
Install Date:            2016-06-01

<snip>
user@smw>
```

## 7.10 Shut Down the System

If it is necessary to shut down the system, use one of these commands.

**full system** For a full, unpartitioned system:

```
crayadm@smw> xtbootsys -s last -a auto.xtshutdown
```

**partitioned** For a partitioned system, with partitions p1 and p2:

```
crayadm@smw> xtbootsys -s last -a auto.xtshutdown -p p1
```

```
crayadm@smw> xtbootsys -s last -a auto.xtshutdown -p p2
```

# 8 Checklists for XC™ Series Software Installation

The process of installing and configuration software for a Cray XC Series system is not necessarily sequential: some steps are optional, and some can be performed in parallel by different people. Cray recommends using the provided checklists to track progress through the installation/configuration process.

*Master Checklist for XC™ Series Software Installation and Configuration* on page 361 lists all of the high-level tasks needed for initial installation and configuration of software on an XC system. It includes links to more detailed checklists for some tasks.

## 8.1 Master Checklist for XC™ Series Software Installation and Configuration

*Table 37. Master Checklist for XC™ Series Software Installation and Configuration*

| ✔ | Task | Notes |
|---|---|---|
| | *Prepare for an SMW/CLE Fresh Install* on page 27 | |
| | *Installation Checklist 1: Install the Base Operating System on the SMW* on page 362 | |
| | *Installation Checklist 2: Install the SMW and CLE Software* on page 362 | |
| | *Installation Checklist 3: Configure SMW for CLE Hardware during a Fresh Install* on page 363 | |
| | *Installation Checklist 4: Configure CLE* on page 364 | |
| | *Installation Checklist 5: Update CLE Configuration Worksheets* on page 365 | |
| | *Installation Checklist 6: Prepare Boot Images and Boot the CLE System during a Fresh Install* on page 366 | |
| | *Installation Checklist 7: Configure Other Features and Services* on page 367 | |
| | *Installation Checklist 8: Install Additional Software* on page 368 | |
| | *Back Up the Newly Installed and Configured SMW/CLE Software* on page 241 | |

## 8.2 Installation Checklist 1: Install the Base Operating System on the SMW

*Table 38. Installation Checklist 1: Install the Base Operating System on the SMW*

| ✔ | Task | Notes |
|---|---|---|
| *Prepare to Install the Base Linux Distribution* on page 33 | | |
| | (Dell R815 SMW only) *R815 SMW: Change the BIOS and iDRAC Settings* on page 34 | |
| | (Dell R630 SMW only) *Configure the Dell R630 SMW RAID Virtual Disks* on page 39 | |
| | (Dell R630 SMW only) *R630 SMW: Change the BIOS and iDRAC Settings* on page 43 | |
| Install the base OS | | |
| | *Install the SLES 12 Base Linux Distribution on the SMW* on page 53 | |
| *Configure Boot RAID Devices* on page 58 | | |
| | *Install SANtricity Storage Manager for NetApp, Inc. Devices* on page 60 | |
| | *Set Up Boot RAID Space for Direct-attached Lustre* on page 61 | |
| | *Create Boot RAID Volume Group and Volumes for NetApp, Inc. devices* on page 62 | |
| | Zone the SAS (Serial Attached SCSI) or FC (Fibre Channel) switch using one of these procedures:<br>● *Zone the QLogic FC Switch* on page 64<br>● *Zone the Brocade FC Switch* on page 67<br>● *Zone the LSI SAS Switch* on page 73 | |
| | *Reboot the SMW and Verify LUNs are Recognized* on page 76 | |
| | *Make a Snapshot Manually* on page 77 | |

## 8.3 Installation Checklist 2: Install the SMW and CLE Software

*Table 39. Installation Checklist 2: Install the SMW and CLE Software*

| ✔ | Task | Notes |
|---|---|---|
| | *Start a Typescript File* on page 78 | |
| Prepare to bootstrap the installation | | |
| | *Prepare to Bootstrap the SMW Installation* on page 79 | |
| | *Determine the Persistent Device Name for a LUN* on page 81 | |
| | *RAID Disk Space Requirements* on page 84 | |
| Bootstrap and install the SMW and CLE software | | |
| | *Bootstrap the SMW Installation* on page 85 | |
| | *Provision SMW Storage* on page 92 | |
| | *Run the Installer for an Initial Installation* on page 93 | |
| | *Set Default Snapshot and Boot the SMW* on page 95 | |

## 8.4 Installation Checklist 3: Configure SMW for CLE Hardware during a Fresh Install

*Table 40. Installation Checklist 3: Configure SMW for CLE Hardware during a Fresh Install*

| ✔ | Task | Notes |
|---|---|---|
| Prepare to configure the SMW for CLE hardware | | |
| | *Set or Change the HSS Data Store (MariaDB) Root Password* on page 96 | |
| | *Start a Typescript File* on page 78 | |
| | *Make a Post-install Snapshot using snaputil* on page 98 | |
| | *Update install.cle.conf for Software Updates* on page 99 | |
| Prepare the global config set and the CLE configuration worksheets | | |
| | *Prepare and Update the Global Config Set* on page 99 | |
| | *Prepare the CLE Configuration Worksheets* on page 106 | |

| ✔ | Task | Notes |
|---|---|---|
| Discover hardware | | |
| | *Bootstrap Hardware Discovery* on page 107 | |
| | *Update Firmware* on page 111 | |
| | *Discover Hardware and HSN Routing, Prepare STONITH* on page 109 | |
| | *(Optional) Configure Partitions* on page 112 | |
| | *Repurpose Compute Nodes* on page 113 | |
| | *Finish Configuring the SMW for the CLE System Hardware* on page 114 | |

## 8.5 Installation Checklist 4: Configure CLE

*Table 41. Installation Checklist 4: Configure CLE*

| ✔ | Task | Notes |
|---|---|---|
| | *Installation Checklist 5: Update CLE Configuration Worksheets* on page 365 | |
| | *Create New CLE Config Set from Worksheets* on page 177 | |
| | *Update CLE Config Set after a Fresh Install* on page 178 | |
| | *Check CLE Hostnames in /etc/hosts File* on page 180 | |
| | *Update /etc/motd for Nodes* on page 181 | |
| | *Copy Files for External Lustre Fine-grained Routing* on page 181 | |
| | *Configure Files for Cray Simple Sync Service* on page 182 | |
| | *Display and Capture all Config Set Information* on page 183 | |
| | *Validate Config Sets* on page 184 | |
| | *Make a Post-config Snapshot using snaputil* on page 185 | |
| | *Make a Post-config Backup of Current Global and CLE Config Sets* on page 185 | |

## 8.6 Installation Checklist 5: Update CLE Configuration Worksheets

*Table 42. Installation Checklist 4.1: Update CLE Configuration Worksheets*

| ✔ | Task | Notes |
|---|---|---|
| | *Update cray_alps Worksheet* on page 117 | |
| | *Update cray_auth Worksheet* on page 118 | |
| | *Update cray_batchlimit Worksheet* on page 122 | |
| | *Update cray_boot Worksheet* on page 123 | |
| | *Update cray_ccm Worksheet* on page 124 | |
| | *Update cray_cnat Worksheet* on page 124 | |
| | *Update cray_drc Worksheet* on page 127 | |
| | *Update cray_dvs Worksheet* on page 129 | |
| | *Update cray_dw_wlm Worksheet* on page 130 | |
| | *Update cray_dws Worksheet* on page 129 | |
| | Update cray_elogin_lnet Worksheet (*Update Cray eLogin Service Worksheets* on page 130) | |
| | Update cray_elogin_motd Worksheet (*Update Cray eLogin Service Worksheets* on page 130) | |
| | Update cray_elogin_networking Worksheet (*Update Cray eLogin Service Worksheets* on page 130) | |
| | Update cray_eswrap Worksheet (*Update Cray eLogin Service Worksheets* on page 130) | |
| | *Update cray_firewall Worksheet* on page 131 | |
| | *Update cray_image_binding Worksheet* on page 132 | |
| | *Update cray_ipforward Worksheet* on page 133 | |
| | *Update cray_liveupdates Worksheet* on page 133 | |
| | *Update cray_lmt Worksheet* on page 134 | |
| | *Update cray_lnet Worksheet* on page 134 | |
| | *Update cray_local_users Worksheet* on page 137 | |
| | *Update cray_logging Worksheet* on page 139 | |
| | *Update cray_login Worksheet* on page 139 | |
| | *Update cray_lustre_client Worksheet* on page 140 | |

| ✔ | Task | Notes |
|---|---|---|
| | *Update cray_lustre_server Worksheet* on page 142 | |
| | *Update cray_multipath Worksheet* on page 143 | |
| | *Update cray_munge Worksheet* on page 145 | |
| | *Update cray_net Worksheet* on page 146 | |
| | *Update cray_netroot_preload Worksheet* on page 155 | |
| | *Update cray_node_groups Worksheet* on page 156 | |
| | *Update cray_node_health Worksheet* on page 160 | |
| | *Update cray_persistent_data Worksheet* on page 160 | |
| | *Update cray_rsip Worksheet* on page 163 | |
| | *Update cray_rur Worksheet* on page 164 | |
| | *Update cray_scalable_services Worksheet* on page 165 | |
| | *Update cray_sdb Worksheet* on page 167 | |
| | *Update cray_service_node Worksheet* on page 168 | |
| | *Update cray_shifter Worksheet* on page 168 | |
| | *Update cray_simple_shares Worksheet* on page 169 | |
| | *Update cray_simple_sync Worksheet* on page 171 | |
| | *Update cray_ssh Worksheet* on page 171 | |
| | *Update cray_storage Worksheet* on page 172 | |
| | *Update cray_sysconfig Worksheet* on page 172 | |
| | *Update cray_sysenv Worksheet* on page 173 | |
| | *Update cray_time Worksheet* on page 174 | |
| | *Update cray_user_settings Worksheet* on page 174 | |
| | *Update cray_wlm_detect Worksheet* on page 175 | |
| | *Update cray_wlm_trans Worksheet* on page 176 | |
| | *Update cray_zonesort Worksheet* on page 176 | |

## 8.7 Installation Checklist 6: Prepare Boot Images and Boot the CLE System during a Fresh Install

*Table 43. Installation Checklist 6: Prepare Boot Images and Boot the CLE System during a Fresh Install*

| ✔ | Task | Notes |
|---|---|---|
| | Decide whether to use Netroot (see *Where to Place the Root File System—tmpfs versus Netroot* on page 186) | |
| | *Create a NIMS Map* on page 188 | |
| | *Build Boot Images for a Fresh Install* on page 190 | |
| | *Set the Turbo Boost Limit* on page 195 | |
| | *Check NIMS Information during a Fresh Install* on page 196 | |
| | *Boot the System using a Boot Automation File* on page 197 | |
| | *Run Tests after Boot is Complete* on page 200 | |
| | *Prepare Site and Software Revision Information Reporting using xtgetrev and xtshowrev* on page 201 | |
| | *Test xtdumpsys and cdump* on page 202 | |
| | *Make a Post-boot Snapshot using snaputil* on page 204 | |
| | *Make a Post-boot Backup of Current Global and CLE Config Sets* on page 205 | |

## 8.8 Installation Checklist 7: Configure Other Features and Services

*Table 44. Installation Checklist 7: Configure Other Features and Services*

| ✔ | Task | Notes |
|---|---|---|
| | *Configure Power Management* on page 206 | |
| | *Push Diag Image to Boot Node and Update the Diags Bind Mount Profile* on page 210 | |
| | *Reduce Impact of Btrfs Periodic Maintenance on SMW Performance* on page 212 | |
| | (if using DAL) *Configure Direct-attached Lustre (DAL)* on page 213 | |

| ✔ | Task | Notes |
|---|---|---|
| | (if using DAL) *LMT Configuration for DAL* on page 220 (Lustre Monitoring Tool for direct-attached Lustre) | |
| | (if using SEC) *Configure the Simple Event Correlator (SEC)* on page 225 | |
| | (if using SEDC) *Enable System Environmental Data Collections (SEDC)* on page 225 | |
| | *Configure Netroot* on page 225 | |
| | *Prevent Unintentional Re-creation of Mail Configuration Files* on page 232 | |

## 8.9 Installation Checklist 8: Install Additional Software

*Table 45. Installation Checklist 8: Install Additional Software*

| ✔ | Task | Notes |
|---|---|---|
| | *Install the Dell Systems Management Tools and Documentation DVD* on page 233 | |
| | *Install and Configure DataWarp* on page 234 | |
| | *Install Cray Programming Environment (PE) Software* on page 234 | |
| | *Install and Configure a Workload Manager (WLM)* on page 240 | |
| | *Install and Configure CMC/eLogin* on page 241 | |