



XC™ Series SMW-managed eLogin Installation Guide

(CLE 6.0.UP06)

S-3020 Rev C

Contents

1 About XC™ Series SMW-managed eLogin Installation Guide (S-3020).....	5
1.1 Related Publications.....	8
2 Introduction to the Installation and Configuration of SMW-managed eLogin Nodes.....	10
2.1 About eLogin Network Architecture.....	13
2.2 About eLogin Security.....	17
2.3 About the Firewall for SMW and eLogin Nodes.....	19
2.4 About the eLogin Node Registry and Node Enrollment.....	21
2.5 About eLogin and Simple Sync.....	22
2.6 About SSH Host Keys for eLogin Nodes.....	24
2.7 About eLogin and Cray Scalable Services.....	26
2.8 About eLogin Image and Configuration Management.....	26
2.9 About the eLogin Boot and Provisioning Process.....	28
2.10 About Storage Profiles for eLogin Nodes.....	29
2.11 About the External State Daemon and eLogin Node States.....	32
3 Migrate from CIMS/CDL or from CMC/eLogin.....	37
4 Install and Configure eLogin.....	39
4.1 Collect Required Configuration Information.....	40
4.1.1 Determine Boot Interface and MAC Address.....	41
4.2 Prepare SMW Network Hardware and Cabling.....	43
4.2.1 Install the SMW Hardware to Manage eLogin Nodes.....	44
4.2.2 Power Off eLogin Nodes and Migrate eLogin Hardware Cabling.....	45
4.2.3 Cable the SMW and eLogin Hardware.....	46
4.2.4 Configure the SMW Networks to Manage eLogin Nodes.....	47
4.3 Configure and Deploy eLogin.....	49
4.3.1 Start a Typescript File Before Configuring eLogin Software.....	49
4.3.2 Change the esd Port.....	50
4.3.3 Start the External State Daemon.....	50
4.3.4 Update the Config Set for eLogin.....	51
4.3.5 Change the eLogin BIOS and iDRAC Settings.....	73
4.3.6 Add eLogin Nodes to the Node Registry.....	85
4.3.7 Configure the eLogin RAID Virtual Disks.....	91
4.3.8 Create an eLogin Image.....	99
4.3.9 Export an eLogin Image.....	100
4.3.10 Assign Image to eLogin Nodes.....	100
4.3.11 Export a PE Image.....	101

4.3.12 Add eLogin Resources to the SMW HA Cluster.....	103
4.3.13 Provision eLogin Nodes.....	105
4.3.14 Modify Config Set or Node Registry After Initial PXE Boot.....	107
5 Manage eLogin Nodes.....	110
5.1 Manipulate eLogin Node Registry.....	110
5.1.1 Register eLogin Nodes.....	110
5.1.2 Enroll eLogin Nodes.....	114
5.1.3 List eLogin Nodes.....	115
5.1.4 Delete eLogin Nodes.....	116
5.1.5 Update eLogin Nodes.....	117
5.2 Manage eLogin Node Life Cycle.....	122
5.2.1 Check eLogin Node Status.....	122
5.2.2 Boot eLogin Nodes.....	123
5.2.3 Shutdown eLogin Nodes.....	126
5.2.4 Reboot eLogin Nodes.....	127
5.3 Create eLogin Images.....	131
5.3.1 Create an eLogin Image.....	131
5.3.2 Export an eLogin Image.....	132
5.3.3 Create eLogin Images with imgbuilder.....	133
5.4 Stage an eLogin Node.....	134
5.5 Deploy Images to eLogin Nodes.....	135
5.5.1 Push eLogin Image Root to eLogin Node.....	135
5.5.2 Push PE Image Root to eLogin Node.....	136
5.6 Transfer Config Set and Apply Changes to eLogin Nodes.....	137
5.6.1 Push Config Set to eLogin Node.....	137
5.6.2 Run cray-ansible on eLogin Node.....	138
6 eLogin Configuration Options.....	140
6.1 Update eLogin Firmware.....	140
6.2 Configure Log Rotation for eLogin Logs on the SMW.....	149
6.3 Change the Firewall Configuration.....	151
6.4 Manage Partitions and Persistent Data on an eLogin Node.....	155
6.4.1 Reprovision a Persistent Disk on an eLogin Node.....	156
6.4.2 Reprovision a Nonpersistent Disk on an eLogin Node.....	160
6.5 Enable LiveUpdates Support for eLogin Nodes.....	162
6.6 Optional Network Configurations.....	166
6.6.1 Configure an IPv4 Interface to Include IPv6 Address.....	166
6.6.2 Configure Tagged VLANs for eLogin.....	168
6.6.3 Configure Bonded Interfaces for eLogin.....	170

6.6.4 Configure Bonded VLAN Interfaces for eLogin.....	173
6.7 Configure SSDs on eLogin Nodes.....	177
7 Gather Diagnostics for eLogin.....	186
7.1 Access the eLogin Console.....	186
7.2 The journalctl Command.....	187
7.3 Log File Locations.....	188
7.3.1 Ansible Logs.....	189
7.4 Collect Dump and Log Information for eLogin.....	190
7.4.1 Enable and Start kdump.....	190
7.4.2 Configure and Run edumpsys.....	194
8 Troubleshoot eLogin Issues.....	200
8.1 Boot the eLogin Node with the DEBUG Shell.....	200
8.2 Troubleshoot Disk Space Issues.....	201
8.3 Use the iDRAC.....	202
9 Supplemental Information.....	204
9.1 Glossary.....	204
9.2 Prefixes for Binary and Decimal Multiples.....	206

1 About XC™ Series SMW-managed eLogin Installation Guide (S-3020)

XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020 Rev C, published 04 April 2018, supersedes *XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020 Rev B*, which was published 27 March 2018.

Scope and Audience

The *XC™ Series SMW-managed eLogin Installation Guide (S-3020)* provides overview information and detailed procedures to install, configure, and deploy eLogin nodes on a Cray XC™ Series system where a System Management Workstation (SMW) is used to manage both internal and external nodes.

This publication applies to an initial deployment of eLogin and to a migration of eLogin nodes from CIMS/CDL or CMC/eLogin to SMW-managed eLogin. To administer a booted system of SMW-managed eLogin nodes, see *XC™ Series SMW-managed eLogin Administration Guide (S-3021)*.

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.UP06 / SMW 8.0.UP06 Release

XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020 supports Cray software release CLE 6.0.UP06 / SMW 8.0.UP06 for Cray XC™ Series systems, released on 01 March 2018.

Changes Published in Revision C

- Information has been added regarding how to obtain the Cray migration tool (`smw_enode_migration`) and instructions for its use. See [Migrate from CIMS/CDL or from CMC/eLogin](#) on page 37.

Changes Published in Revision B

- The procedure to place the external state daemon (`esd`) under cluster management in an SMW HA system was moved to just before eLogin nodes are provisioned and booted. See [Add eLogin Resources to the SMW HA Cluster](#) on page 103.
- Errors have been corrected in [Configure SSDs on eLogin Nodes](#) on page 177.
- Various minor corrections and editorial changes were made.

Changes Published in Revision A

- A new procedure prompts sites to add or migrate site-specific host entries and comments to the `/etc/hosts` file that is installed on eLogin nodes. This is especially important for sites migrating from an earlier implementation of eLogin, because old host entries for management nodes other than the SMW can cause an eLogin node boot to fail. See [Add or Migrate Site Data to /etc/hosts File](#) on page 55.

- A new procedure checks the firewall to make sure the correct ports are open. If the NFS port is not open, eLogin nodes will be unable to boot because they will be unable to NFS-mount the necessary file systems. See [Ensure that NFS Port is Open in Firewall](#) on page 70.
- Several issues were discovered late in the release cycle: they will be corrected in a future release, but for now, documented workarounds are provided. See "Known Issues that Occur at SMW Reboot or Failover" in [Install and Configure eLogin](#) on page 39.
- In the original publication of this guide, the steps for configuring all eLogin-related configuration services were in a single, very long procedure. In this revision, that procedure has been split into separate procedures, one for each configuration service.
- In the original publication of this guide, the procedure for staging an eLogin node was omitted from the "Manage eLogin Nodes" section. In this revision, that procedure has been included. See [Stage an eLogin Node](#) on page 134.
- Various minor corrections were made to this revision.

New in the CLE 6.0.UP06 / SMW 8.0.UP06 Release

This was the first publication of this guide. For an overview of the changes required for managing eLogin nodes from the SMW, see "Overview of What's New" in [Introduction to the Installation and Configuration of SMW-managed eLogin Nodes](#) on page 10.

Table 1. Record of Revision

Publication Title	Date	Release
<i>XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020 Rev B</i>	27 Mar 2018	CLE 6.0.UP06 / SMW 8.0.UP06
<i>XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020 Rev A</i>	12 Mar 2018	CLE 6.0.UP06 / SMW 8.0.UP06
<i>XC™ Series SMW-managed eLogin Installation Guide (CLE 6.0.UP06) S-3020</i> Note that S-3020 supersedes S-2566. There will be no revisions of S-2566 for this release and all future releases.	01 Mar 2018	CLE 6.0.UP06 / SMW 8.0.UP06
<i>XC™ Series eLogin Installation Guide (CLE 6.0.UP05) S-2566</i>	05 Oct 2017	CLE 6.0.UP05 / CSMS 1.1.4

Command Prompt Conventions

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`.

<code>smw#</code>	Run the command on the SMW as <code>root</code> .
<code>sdb#</code>	Run the command on the SDB node as <code>root</code> .
<code>elogin#</code>	Run the command on the eLogin node as <code>root</code> .
<code>crayadm@boot></code>	Run the command on the boot node as the <code>crayadm</code> user.
<code>user@login></code>	Run the command on any login node as any <code>non-root</code> user.
<code>hostname#</code>	Run the command on the specified system as <code>root</code> .
<code>user@hostname></code>	Run the command on the specified system as any <code>non-root</code> user.

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.
<i>Oblique or Italics</i>	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.

Trademarks

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, Urika-GX, 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

XC™ Series SMW-managed eLogin Installation Guide (S-3020) supersedes *XC™ Series eLogin Installation Guide (S-2566)*.

This publication contains all that is necessary to install eLogin-related hardware on the SMW, configure the SMW for eLogin, and configure eLogin hardware and software.

Because eLogin software is installed during the installation of SMW and CLE software, familiarity with the following documentation is necessary. The release errata and readme files are available on CrayPort, and all other publications listed here are available 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 Software Installation and Configuration Guide (S-2560)*

The following publications contain additional information that may be helpful while setting up an XC system for eLogin.

- *XC™ Series SMW-managed eLogin Administration Guide (S-3021)*
- *XC™ Series Configurator User Guide (S-2560)*
- *XC™ Series Ansible Play Writing Guide (S-2582)*
- *XC™ Series System Administration Guide (S-2393)*
- *XC™ Series Boot Troubleshooting Guide (S-2565)*
- *XC™ Series Lustre® Administration Guide (S-2648)*
- *Cray® Programming Environments Installation Guide (S-2372)*

- *XC™ Series Programming Environment User Guide (17.05) (S-2529)*
- *Cray Compiling Environment Release Overview and Installation Guide*
- 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)*

2 Introduction to the Installation and Configuration of SMW-managed eLogin Nodes

External login (eLogin) nodes enable user access to a Cray XC system, providing a place for users to:

- log in
- access their home directories that are external to the XC system
- do application development using the Cray Programming Environment (PE)
- access the parallel file system, such as Lustre or GPFS
- submit jobs to the workload manager (WLM) used by the site, such as Slurm, Moab/TORQUE, PBS, and ALPS

An eLogin node must be usable for its primary duties even when the XC system or the System Management Workstation (SMW) is unavailable. Therefore, an eLogin node uses local storage for its operating system and local persistent storage for anything that must remain persistent (such as SSH host keys) across reprovisioning of the eLogin node with new images. Furthermore, because an eLogin node is the access point for the user community, it must have a strong security profile that can be adjusted for the needs of each site. Risk of attack from a compromised eLogin node to other components in the XC system must be as low as possible.

Beginning with the CLE 6.0.UP06 release, management support for eLogin nodes is provided entirely by the SMW; no other management node is needed.

In CLE 5.2 releases, external login nodes, then called CDL (Cray Development and Login) nodes, were managed on a CIMS (Cray Integrated Management Server) node running Bright Computing software with Cray additions to configure CDL nodes and build the software image provisioned to the nodes by Bright Computing software. In CLE 6.0 releases prior to UP06, the SMW was used to prepare the image root and data in the config set for eLogin nodes, but the CMC (Cray Management Controller) node, running CSMS (Cray System Management Software) and OpenStack software, was used to provision and manage eLogin nodes.

Now, the SMW (running SLES 12 SP3) is used not only for the config set and eLogin image root but also to provision the image root to the node via PXE boot and perform other management functions.

Rationale

The switch from using a CMC and OpenStack to using the SMW and SLES 12 SP3 for the management of eLogin nodes provides the following benefits:

- **Improved Usability**
 - **Reduced complexity.** Only one management node is required instead of two that run different software.
 - **Consistency.** eLogin nodes can be managed within the same SMW/CLE image and configuration management framework as internal nodes, using extended or similar commands and the same SMW installation, update, security update processes.
 - **Scalability.** This new approach to managing eLogin nodes is general enough to apply to other external nodes as well, where an external node is defined as any node that is not connected to the Aries high-

speed network (HSN) of a Cray XC system. Examples include purge servers, data movers, and visualization servers.

- **Increased Security**

- **Network security.** Switch-based VLANs and port access control lists (ACL), where applicable, physically restrict eLogin-to-SMW network port access. The software firewall (iptables) on the SMW provides a second layer of access protection. In addition, this approach enables the use of separate management networks for each external node type.
- **Data security.** The SMW config set is sanitized before being pushed to the eLogin node, which means that config set data not needed on the eLogin node is excluded, such as SDB accounts and passwords.
- **Console security.** Plain text passwords in files or environment variables are not exposed, as they are with OpenStack-based operations, to protect the BMC account name and password used for administrative access to the BMC device (iDRAC).

- **Reduced Deployment Time**

SMW-managed eLogin nodes demonstrate significantly faster deployment times compared with CMC-managed eLogin nodes.

- CMC-managed eLogin nodes require two passes through the POST/BIOS process, while SMW-managed eLogin nodes require only one, saving 7–10 minutes per node.
- A small initramfs is used for PXE boot.
- A smaller SquashFS format of image root is used for the operating system and PE.

- **Improved Reliability/Availability**

- Diskfull nodes are not dependent on the SMW for operation. They continue to operate normally and can be manually rebooted in the absence of the SMW.
- SMW-managed eLogin nodes can be managed on the active SMW in SMW HA pair, thereby enabling high availability for eLogin systems.
- Ongoing image and configuration management can be performed without requiring a node reboot: stage a new eLogin image root and/or PE image root to the eLogin node while the node is booted, deliver new config set data to the eLogin node while the node is booted, and then run `cray-ansible` on the eLogin node.

Overview of the Initial Deployment Process

For the initial configuration and deployment of one or more eLogin nodes, a system administrator performs these tasks on the SMW:

1. Register information about each eLogin node needed to control the PXE boot process.
2. Create eLogin images using Image Management and Provisioning System (IMPS) recipes.
3. Modify data in the global and CLE config set for each eLogin node to consume.
4. Define the disk layout for internal storage on each eLogin node.
5. Associate information about images, config sets, and kernel parameters to each eLogin node.
6. Configure each eLogin node to boot from the network (can configure to boot from local storage for subsequent boots).
7. Power on an eLogin node.

8. Monitor the state of an eLogin node as it boots.
9. Connect to the console of an eLogin node via an interactive terminal session.
10. View eLogin logs and initiate a dump of an eLogin node, as needed.

Overview of What's New

New SMW Daemon. An SMW-managed eLogin node is connected to the SMW through the IPMI network and the management network. Because the SMW resides in a higher trust domain than the eLogin, most actions must be initiated on the SMW (SMW pushes to eLogin node) rather than being initiated on the eLogin node (eLogin node pulls from SMW). This is unlike the internal CLE nodes, which generally pull to the node via IDS and Cray Scalable Services. Therefore, a new daemon, `esd` (external state daemon) has been added to the SMW so that the SMW can determine the state of the eLogin node and the eLogin node can request the SMW to push content to the node. The `esd` daemon controls what content is available to an eLogin node based on the state of the node (similar to the state manager and boot manager daemons for internal nodes). More generally, `esd` maintains eLogin node registry and state, and it performs all eLogin node life-cycle tasks.

New SMW Command. A new command, `enode`, has been added to the SMW as the command line interface to `esd`, the new SMW daemon. The `enode` command has subcommands for node enrollment, for managing eLogin nodes via their BMC devices, and for associating boot information (image, config set, and kernel parameters) with the node.

Table 2. Comparison of Node Management Commands for Internal CLE Nodes and eLogin Nodes

Task	Internal CLE Node	eLogin Node
Discover nodes	automatic with <code>xtdiscover</code>	manual
Enroll/register nodes	<code>xtdiscover</code> (registers nodes with HSS)	<code>enode create</code> , <code>enode enroll</code>
Map image, config set, and kernel parameters to nodes	<code>cnode update</code>	<code>enode create</code> , <code>enode update</code>
List node attributes	<code>cnode list</code>	<code>enode list</code>
Delete nodes	N/A	<code>enode delete</code>
Boot nodes	<code>xtbootsys</code> or <code>xtcli boot</code>	<code>enode boot</code> , <code>enode reboot</code>
Shut down nodes	<code>xtbootsys</code> or <code>xtcli shutdown</code>	<code>enode shutdown</code>
Check node status	<code>xtcli status</code>	<code>enode status</code>
Dump nodes	<code>cdump</code> and <code>xtdumpsys</code>	<code>kdump</code> and <code>edumpsys</code> (<code>xtdumpsys</code> with a new external node plugin)
Initiate interactive console session with nodes	<code>xtcon</code>	<code>conman</code>

New/Extended CLE Config Services and Ansible Plays. One existing CLE configuration service has been extended and two new CLE config services have added to support SMW-managed eLogin nodes. New Ansible plays have been added to the software installed on eLogin nodes to consume these new configuration settings.

- The `cray_storage` config service has been extended to specify the storage layout for local storage on eLogin nodes.
- The `cray_cfgset_exclude` config service has been added to specify which parts of the config set should be excluded when the config set is pushed from the SMW to the eLogin node.
- The `cray_kdump` config service has been added to configure the kernel dump tool on eLogin nodes.

Networks New to the SMW. Two networks are required to manage an eLogin node from the SMW: `external-ipmi-net` and `external-management-net`. The `external-ipmi-net` network enables the SMW to remotely control the BMC device with IPMI commands. The `external-management-net` network is used for transfer of data, such as config sets and images, to the eLogin nodes.

- Two Ethernet ports must be available on the SMW to connect to these networks. A quad port Ethernet card must be added to the SMW if it is a single, stand-alone SMW. If the SMW is already part of an SMW HA pair, then no additional quad port Ethernet card is required because one was added earlier to support SMW HA. To keep the two types of SMW (stand-alone and HA) consistent, two previously unused ports have been assigned for these two networks:
 - `eth6` for `external-ipmi-net`
 - `eth7` for `external-management-net`
- The Ethernet switch between the SMW and eLogin nodes must be configured with two VLANs for isolation of network traffic between the `external-ipmi-net` network and the `external-management-net` network.

2.1 About eLogin Network Architecture

eLogin Networks

The following networks are used to connect eLogin nodes to the SMW, users, and the Cray XC system.

site-admin-net	External administration network that enables site administrators to log into the SMW. The IP address of this network can be customized during SMW software installation.
site-ipmi-net	The SMW's iDRAC (IPMI device) can be connected to a network for remote console and power management of the SMW. Cray recommends that the IPMI interface of the SMW not be connected to <code>site-admin-net</code> , but instead be connected to a separate network with tighter access control.
external-ipmi-net	External management network that connects the SMW to the eLogin IPMI devices. This network enables remote console and power management of eLogin nodes. The dedicated IPMI device port of each eLogin node must be connected to the IPMI network.
external-management-net	External management network that connects the SMW to the eLogin nodes for PXE booting and other data transfer between the SMW and the eLogin nodes. The first 1GbE device of each eLogin node must be connected to <code>external-management-net</code> . Depending on the eLogin hardware configuration, this may be the first Ethernet device in the case of a 4x1GbE LOM network adapter, or the third Ethernet device in the case of a 2x10GbE+2x1GbE LOM network adapter.

site-user-net External user (site) network used by eLogin nodes. This network provides user access and may be used to access authentication services like LDAP. The name and IP addresses on this network are added in the config set. Connections to additional site-specific networks are optional.

IMPORTANT: The site-user-net network must be configured as `site` in the network section of the `cray_net` configuration service.

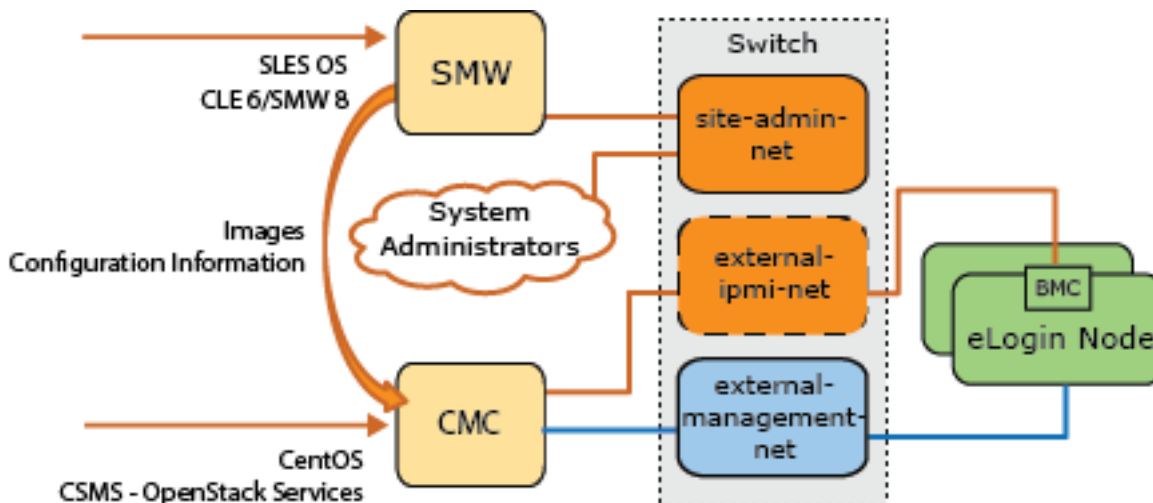
One Ethernet interface of each eLogin node must be connected to site-user-net. This Ethernet interface may be 1GbE, 10GbE, or 40GbE, depending on site infrastructure.

IB Net Internal Infiniband® network used for high-speed Lustre LNet traffic.

Connection of eLogin Nodes to the SMW

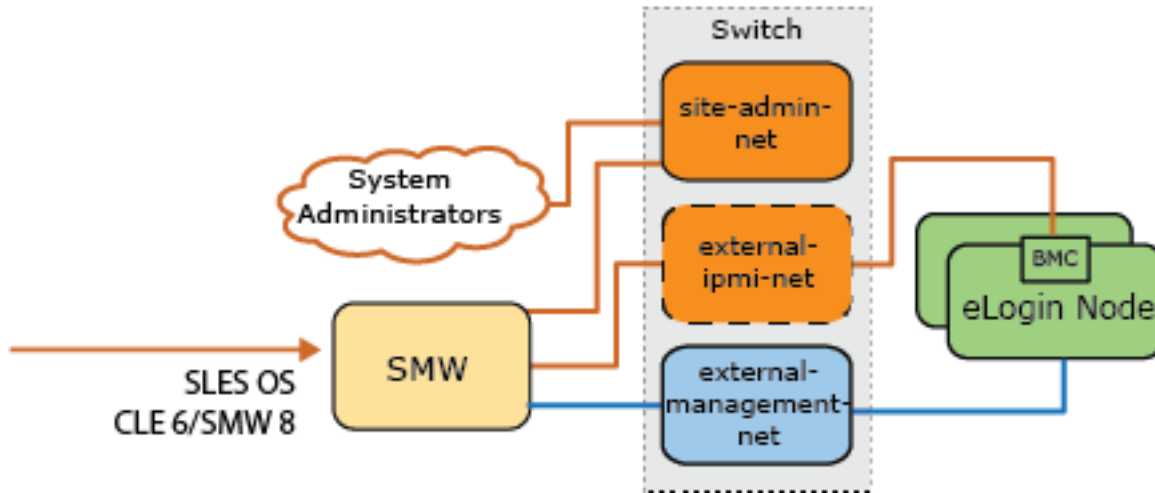
In CLE 6.0 releases prior to UP06, the SMW was used to prepare image and configuration data for eLogin nodes and push that data to the CMC (Cray Management Controller), which was used to provision and manage eLogin nodes. The following figure shows the network topology connecting SMW, CMC, and eLogin nodes.

Figure 1. eLogin Management Topology for Releases Prior to CLE 6.0.UP06



Beginning with the CLE 6.0.UP06 release, the SMW is used to provide all management support of eLogin nodes. The CMC is no longer needed. The following figure shows the network topology connecting the SMW and eLogin nodes.

Figure 2. eLogin Management Topology for Releases Beginning with CLE 6.0.UP06

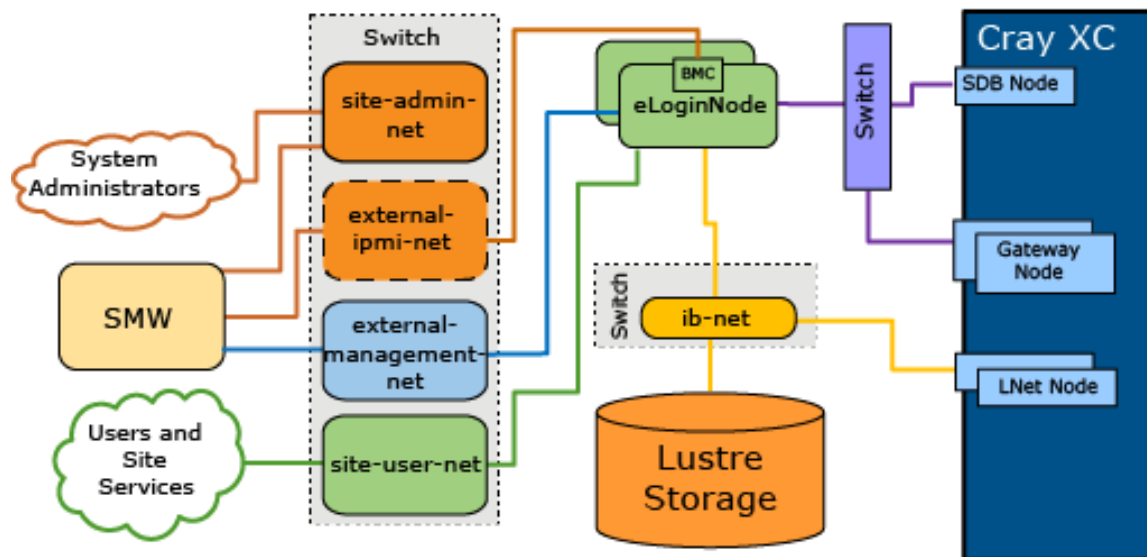


Connection of eLogin Nodes to the XC System

There are four distinct topologies for connecting eLogin nodes to a Cray XC system. All four topologies are the same with regard to how the eLogin nodes connect to the SMW, to users, and to LNet nodes and Lustre storage. However, they differ in how they are connected to the service database (SDB) node and the gateway node. eLogin nodes can connect to the SDB directly through a switch or indirectly through a routed connection from the gateway node. eLogin nodes can connect to the gateway node directly through a switch or through the site-user-net network. The four possible topologies are shown in the following figures.

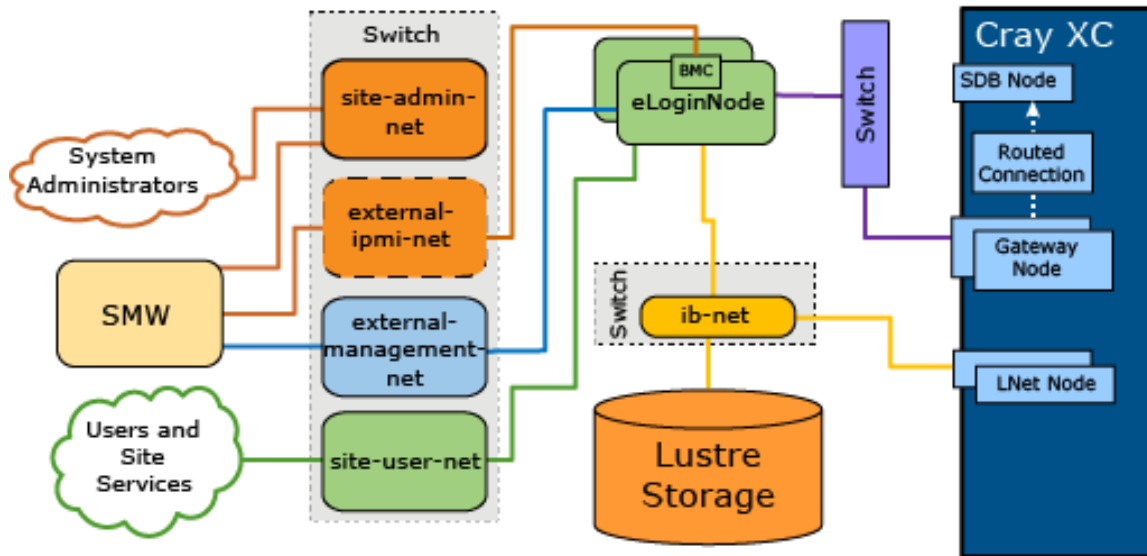
- **eLogin Nodes Connected to SDB Node and to Gateway Node via Switch.** Jobs are submitted from an eLogin node directly to the SDB node.

Figure 3. eLogin Connected to SDB and Gateway via Switch



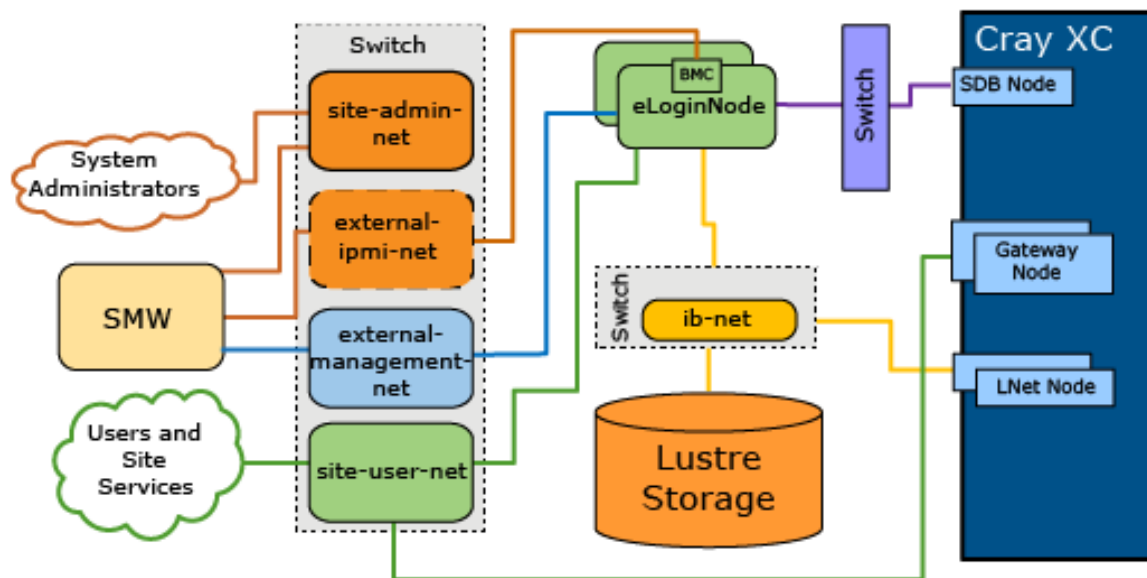
- **eLogin Nodes Connected to SDB Node via Routed Connection from Gateway Node and to Gateway Node via Switch.** Jobs are submitted from an eLogin node through the gateway node to the SDB node.

Figure 4. eLogin Connected to SDB Routed from Gateway and to Gateway via Switch



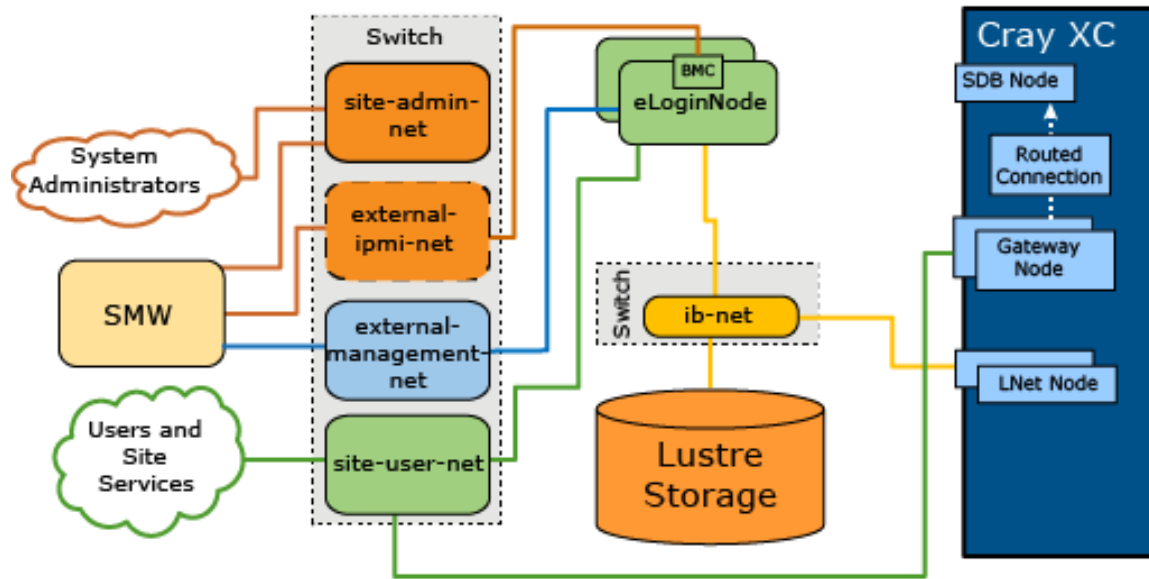
- **eLogin Nodes Connected to SDB Node via Switch and to Gateway Node via Site User Net.** Jobs are submitted from an eLogin node directly to the SDB node. Users access the gateway node directly from the site user net.

Figure 5. eLogin Connected to SDB via Switch and to Gateway via Site User Net



- **eLogin Nodes Connected to SDB Node via Routed Connection from Gateway Node and to Gateway Node via Site User Net.** Users access the gateway node directly from the site user net. Jobs are submitted from an eLogin node through the gateway node to the SDB node.

Figure 6. eLogin Connected to SDB Routed from Gateway and to Gateway via Site User Net



2.2 About eLogin Security

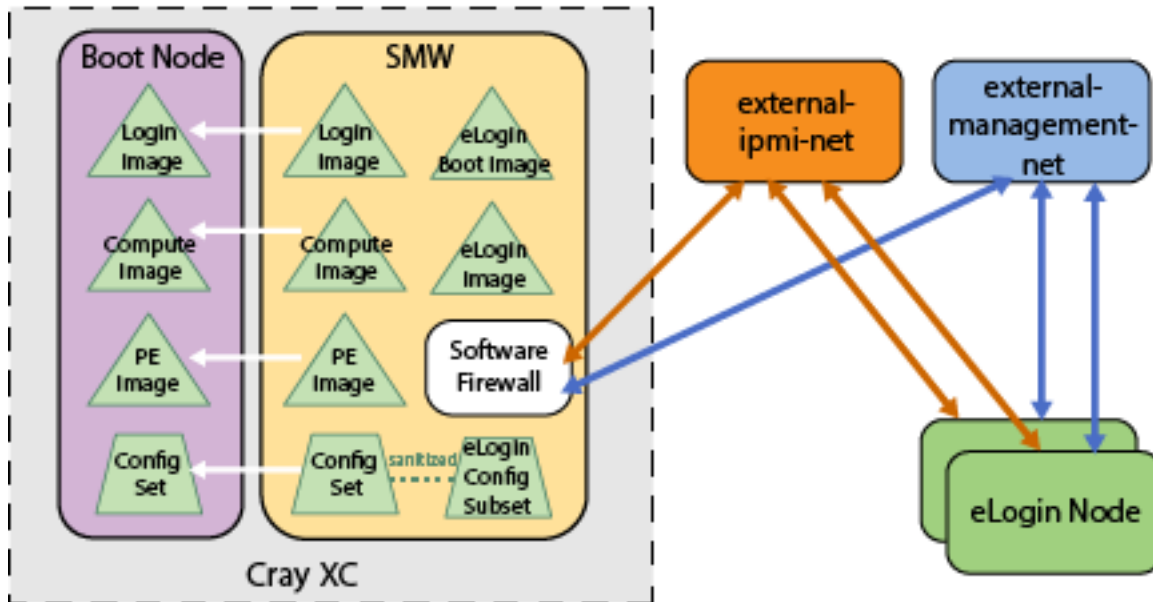
Security for a Cray XC system with SMW-managed eLogin nodes is implemented through network security, data security, and console security.

Network Security

Network security is achieved by the following:

- The external-ipmi-net network and the external-management-net network are defined to be different networks (whether done via VLAN or physically separate networks) because the traffic on the external-ipmi-net network between the SMW and the BMC devices of the nodes includes clear-text passwords for the BMC devices. A Linux user on the node is unable to use the Ethernet interface on the BMC device to capture network packets and see this clear-text password traffic. If the external-ipmi-net and the external-management-net are the same network, then there would be the potential security exposure of a Linux user being able to capture those passwords and hence control other BMC devices on external-ipmi-net.
- An Ethernet switch with VLANs and optional access control lists (ACL) physically restrict eLogin-to-SMW network port access to reduce the network exposure of the SMW to devices on the networks.
- Software firewall rules (iptables) on the SMW provide a layer of access protection for the SMW. The SMW iptables have entries that restrict access on the two network interfaces to external-ipmi-net (SMW eth6) and external-management-net (SMW eth7). See [About the Firewall for SMW and eLogin Nodes](#) on page 19.

Figure 7. SMW-managed eLogin Network Security



Communications between the SMW and the BMC device of the eLogin node are not encrypted, which is why the external-ipmi-net network connects only the SMW and the BMC devices of eLogin nodes. The external-ipmi-net should be completely isolated so that no other device is connected to it other than the SMW and the BMC of any eLogin node being managed by that SMW.

Data Security

The SMW resides in a higher trust domain than an eLogin or other external node because “untrusted” users are allowed direct access to such nodes. Most actions must be initiated on the SMW (push from SMW to an eLogin node) rather than being initiated on an eLogin node (pull to an eLogin node from the SMW). In contrast, internal CLE nodes can initiate data transfer (pull) from the SMW via the IMPS Distribution System (IDS) and Cray Scalable Services.

Authenticated requests for data. Because an eLogin node resides in a lower trust domain than the SMW and cannot pull data from the SMW, it must announce that it is in a state where the SMW can push data to it. The `esd` daemon, which resides on the SMW, keeps track of the state of eLogin nodes and triggers push operation whenever an eLogin node requests data. For example, during the course of booting an eLogin node, the node must announce that it is in a state ready to receive a push from the SMW of configuration data, image root, or PE image root at the appropriate time in the boot process. Except for the initial transmission of certificates, all communication to the `esd` daemon on the SMW is done using X.509 authenticated messages to verify the eLogin node.

Sanitized Config Set. Configuration data is stored on the SMW in config sets. An eLogin node needs most of that configuration data, but there is some data that it must not have access to, such as SDB accounts and passwords. Therefore, that data is excluded from the config set when it is pushed from the SMW to an eLogin node. The resulting sanitized, eLogin-specific subset of the SMW config set is stored on the eLogin node and used to boot the node. The configuration data to be excluded is configurable through the `cray_cfgset_exclude` configuration service on the SMW. Typically, the following directories and files are excluded:

- worksheets (all config worksheet YAML files)
- `config/cray_sdb_config.yaml` (SDB configuration)

- `files/roles/common/etc/ssh` (SSH keys)
- `files/roles/common/root` (SSH and node health)
- `files/roles/munge` (munge)
- `files/roles/common/etc/opt/cray/xtremoted-agent`
- `files/roles/merge_account_files` (site-provided user account info)
- `files/simple_sync/common/files/etc/ssh` (SSH host keys)
- `files/simple_sync/common/files/root/.ssh` (root user SSH public/private key pairs)

Console Security

Console security refers to protecting the BMC account name and password used for administrative access to the BMC device (iDRAC). The BMC account name and password must be securely stored on the SMW to enable the actions that control the eLogin nodes, including powering on or off the node, changing the BIOS boot order, and accessing the console via IPMI SOL to the eLogin node using ConMan. The `enode` command does most of these actions, and it can place the BMC account name and password into the correct location for ConMan to use them. ConMan stores this information in clear text in its configuration file, which is set to permissions preventing non-root users from reading or writing it. `enode` and `esd` use keychains for BMC credentials. The `conman.conf` file contains clear-text passwords, but that file can be viewed only by root on the SMW.

2.3 About the Firewall for SMW and eLogin Nodes

Software firewall rules on the SMW and the eLogin nodes provide a layer of access protection for the external-management-net network. Cray provides firewall configuration templates to enable/disable the firewalls and Ansible plays to manage firewall configuration on both the SMW and all eLogin nodes.

The firewall on both the SMW and eLogin nodes is implemented using `SuSEfirewall12`, a script that uses the configuration settings stored in `/etc/sysconfig/SuSEfirewall12` to create iptables rules. Sites should not make changes directly to that configuration file because it is changed whenever Cray Ansible plays are run, and site changes (e.g., custom zoning) could be overwritten.

SMW firewall configuration. The firewall on the SMW is configured through the `cray_firewall` configuration service in the global config set. The global firewall config template controls enabling and disabling the service on the SMW, and Ansible plays configure the firewall. Plays run automatically when the SMW is rebooted, and administrators can run them manually at any time to apply changes.

eLogin firewall configuration. The firewall on an eLogin node is configured through the `cray_firewall` configuration service in the CLE config set. The CLE firewall config template controls enabling and disabling the firewall for all internal CLE nodes and eLogin nodes in the system, or it can be configured to inherit the settings from the global firewall config template, which would then control firewall enabling/disabling for the SMW and the entire system, including eLogin nodes. Ansible plays configure the firewall. Plays run automatically when nodes are booted, and administrators can run them manually on nodes at any time to apply changes.

Firewall Zones

Firewall zones have been created within the SMW and eLogin firewalls to enable different sets of rules to be applied to each zone. Each network interface is assigned to one of the firewall zones (note that an interface can belong to only one zone). The firewall rules for a particular zone are applied to all interfaces assigned to that zone.

Cray has created a custom firewall zone named MGMT (management) to handle traffic between the SMW and other nodes on the external-management-net network. The SMW and eLogin nodes communicate on this network.

Other zones used by Cray include the INT (internal) zone, which is used for internal CLE nodes, and the EXT (external) firewall zone, which is the least trusted zone.

Interfaces and Ports eLogin Firewall

Cray places an eLogin node's network interfaces into firewall zones as follows:

- An eLogin node's connection to the site network (site-user-net) is placed into the EXT firewall zone.
- An eLogin node's connection to the management network (external-management-net), whose interface is passed along as a kernel parameter, is placed into the MGMT firewall zone.

The SSH port (port 22) is open in the EXT zone so that users can log in. In the MGMT zone of the eLogin firewall, the following ports are open. Note that a port can be opened on multiple zones.

Table 3. ELogin Ports Open in the Firewall MGMT Zone

Service	Port	Protocol
SSH	22	TCP
NTP	123	UDP

SMW Firewall Interfaces and Ports

Cray places the SMW network interfaces into firewall zones as follows:

- The eth6 interface on the SMW is connected to the IPMI network (external-ipmi-net), which is used for IPMI communication with the IDRAC controlling the eLogin nodes. It is placed into the EXT zone.
- The eth7 interface on the SMW is connected to the management network (external-management-net). It is placed into the MGMT zone of the firewall.

In the MGMT zone of the SMW firewall, the following ports are open.

Table 4. SMW Ports Open in the Firewall MGMT Zone

Service	Port	Protocol
esd	8449 (default) or as configured in <code>/etc/opt/cray/esd/esd.ini</code>	TCP
TFTP	69	UDP
DHCP	67	UDP
NFS	48451	UDP
NFS	49478	TCP
NFS	41276	UDP
NFS	35938	TCP
NFS	Mountd (20048)	UDP

Service	Port	Protocol
NFS	Mountd (20048)	TCP
NFS	NFS (2049)	UDP
NFS	NFS (2049)	TCP
NFS	NFS (2049)	UDP
NFS	NFS (2049)	TCP
NFS	SUNRPC (111)	UDP
NFS	SUNRPC (111)	TCP
LiveUpdates	2526	TCP

Make and Apply Firewall Configuration Changes

The Cray firewall configuration services and Ansible plays are designed to make it unnecessary for site system administrators to change the SMW and eLogin firewall configuration. However, there are several basic changes a site may wish to make, and there are certain steps that must be taken to apply those changes.

- For an initial deployment or migration, firewall configuration steps are included where appropriate in the procedures for configuring eLogin software.
- For reconfiguration of the firewall of a system with SMW-managed eLogin already deployed, see [Change the Firewall Configuration](#) on page 151.

2.4 About the eLogin Node Registry and Node Enrollment

For internal CLE nodes, node enrollment (or registration) occurs during hardware discovery, and node information is stored in the Hardware Supervisory System (HSS) database. For eLogin nodes, node enrollment is a manual process that uses the `enode` command, and the information is stored in the node registry.

Enrollment of an eLogin node requires the following information, though additional information must be added before the node can be booted.

- IP address of the BMC device (iDRAC) on the IPMI network (external-ipmi-net)
- BMC account name and BMC password for administrative access to BMC device (iDRAC)
- MAC address of the eLogin node interface on the management network (external-management-net), used for PXE boot
- IP address of the eLogin node interface on the management network (external-management-net)

All eLogin nodes to be managed by the SMW must be added to the node registry using either `enode create` or `enode enroll`.

- The `enode create` command adds a single eLogin node to the registry by specifying all of the required data on the command line. The information in the node registry can be changed later using the `enode update` command. This is typically used for an initial deployment.
- The `enode enroll` command adds multiple eLogin nodes by importing the `inventory.csv` file, which is generated by the `smw_enode_migration` tool. This is typically used for a migration.

Because `inventory.csv` does not contain all of the data needed to manage eLogin nodes from the SMW, such as the IP address of the eLogin interface on `external-management-net`, the missing data must be added to the node registry later using the `enode update` command.

Whenever eLogin nodes are added to the node registry using either `enode create` or `enode enroll`, `esd` also does the following:

- Updates the `/etc/conman.conf` file for remote console.
- Updates the DHCP configuration and TFTP configuration files and restarts `dhcpd` (if `enode enroll` was used, this occurs only after the necessary data is added with `enode update`).
- Adds the eLogin node host names on the IPMI network and management network to the SMW `/etc/hosts` file. All old entries for this node in `/etc/hosts` are cleaned up.
- Adds a “console” entry for each eLogin node to `/etc/conman.conf` so that its console output can be redirected to the logging directory for that node’s host name, `/var/opt/cray/log/external/conman/console.<hostname>`. This directory is on the boot RAID so that it can be mounted by the active SMW in an SMW HA pair. Any previous definition for the eLogin node’s `external_ipmi_net` IP address in `/etc/conman.conf` is removed.

ATTENTION: After a change to `/etc/conman.conf`, the `conmand` daemon is restarted. This will disconnect all active console sessions for any eLogin nodes. Logging of console messages will be restored as soon as the service restarts.

2.5 About eLogin and Simple Sync

The Cray Simple Sync service (`cray_simple_sync`) provides a simple, generic mechanism for copying user-defined content to internal and external nodes in a Cray XC system. When executed, the service automatically copies files found in source directories in the config set to one or more target nodes. 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. For more information, see `man cfgset(8)`.

With regard to external nodes like eLogin nodes, the exclusions specified in the `cray_cfgset_exclude` configuration service are applied when the CLE config set is transferred to the node, and some portions of the Simple Sync directory in the config set are excluded. The “Files Excluded from eLogin Nodes” section contains more details.

How Simple Sync Works

When enabled, the Simple Sync service is executed on all internal CLE nodes and eLogin nodes at boot time and whenever the administrator executes `/etc/init.d/cray-ansible start` on a CLE node or eLogin node. When Simple Sync is executed, files placed in the following directory structure are copied to the root file system (`/`) on the target nodes.

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 in those directories are copied to their associated target nodes.

`./common/files/`

Targets all nodes, both internal CLE nodes and eLogin nodes.

<code>./hardwareid/<hardwareid>/files/</code>	Not applicable to eLogin nodes.
<code>./hostname/<hostname>/files/</code>	Used ONLY for eLogin nodes. Targets a node with the specified host name. An admin must create both the <code><hostname></code> directory and the <code>files</code> directory.
<code>./nodegroups/<node_group_name>/files/</code>	Targets all nodes in the specified node group. The directories for this <code>nodegroups</code> directory are automatically stubbed out when the config set is updated after node groups are defined and configured in the <code>cray_node_groups</code> service.
<code>./platform/[compute service]/files/</code>	Not applicable to eLogin nodes.
<code>./README</code>	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.

The ownership and permissions of copied directories and files are preserved when they are copied to root on the 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 target nodes.

Files Excluded from eLogin Nodes

Because eLogin nodes use the `cray_cfgset_exclude` configuration service, some directories within the Simple Sync directory structure on the SMW can be excluded from transfer to eLogin nodes. The default “`elogin_security`” profile will exclude the following config set directories from being transferred to an eLogin node when the CLE config set is pushed to the node from the SMW.

- `files/simple_sync/common/files/etc/ssh`
- `files/simple_sync/common/files/root/.ssh`

To specify other areas within the Simple Sync directory structure that should not be transferred to eLogin nodes, create a customized site profile in `cray_cfgset_exclude`.

Examples Using Simple Sync for eLogin Nodes

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 particular eLogin node

1. Create the `<hostname>/` and `files/` directories under `./hostname/`
2. Place `etc/myfile` under `./hostname/<hostname>/files/` in the Simple Sync directory structure.
3. Simple Sync copies it to `/etc/myfile` on the eLogin node.

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/myfile` under `./nodegroups/my_nodes/files/` in the Simple Sync directory structure.
4. Simple Sync copies it to `/etc/myfile` on all nodes listed in node group `my_nodes`.

For cautions about the use of Simple Sync and more information and examples, see "About Simple Sync" in *XC™ Series Software Installation and Configuration Guide (S-2559)*.

2.6 About SSH Host Keys for eLogin Nodes

Cray offers several options for how SSH host keys are handled for an eLogin node:

- SSH host keys can be shared or unique for each node.
- SSH host keys can be generated automatically or supplied by the site.

The external state daemon (`esd`) handles SSH host keys for an eLogin node based on the value of the `ssh_host_keys` field in the node registry and whether host keys are present in the Simple Sync directory structure for that node. The value of `ssh_host_keys` can be specified when the node is created or updated using the `enode` command.

The `ssh_host_keys` field must have one of the following values: `simple_sync`, `generate`, or `/some/directory/path`. If `ssh_host_keys` is not specified for an eLogin node when it is created, the default value, `simple_sync`, is used when the node is PXE booted.

If `ssh_host_keys` is set to a value other than `simple_sync`, then SSH host keys must not be present in the Simple Sync directory structure. If they are present, `esd` will not allow the node to be PXE booted and will report an error.

simple_sync

The `ssh_host_keys` field is set to `simple_sync` by default. When this value is specified, `esd` looks for SSH host key files in the config set under the Simple Sync directory structure:

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

The following subdirectories (under `simple_sync/`) are searched in the order shown, moving from more specific to more general:

- `hostname/<name>/files/etc/ssh`

For `<name>` substitute the name of the eLogin node.

- `nodegroups/<node_group_name>/files/etc/ssh`

This subdirectory is the location where SSH host keys are placed that are common to eLogin nodes in the specified node group. If the eLogin node is in multiple node groups, then the `nodegroups` subdirectories are searched in reverse alphabetical order.

- `common/files/etc/ssh`

This subdirectory is the location where SSH keys common to all CLE nodes are placed.

If `cray_ssh` is configured to generate SSH host keys automatically, then when the config set is updated, new SSH host keys will be placed both in `nodegroups/<node_group_name>/files/etc/ssh` and in `common/files/etc/ssh`, if no SSH host keys are already present in those locations. If SSH host keys are already present there, the newly generated keys will not overwrite them.

If `esd` finds SSH host key files in one of these directories, then `esd` copies them to the eLogin node's directory when PXE booting the node. Note that several of these directories may contain the SSH host keys for a node, but `esd` will stop after finding the first instance of SSH host keys applicable to the node, and will copy them to the node when booting the node.

If none of these Simple Sync directories contains SSH host key files, then `esd` will generate the keys as if `ssh_host_keys` were set to `generate`.

If one of these Simple Sync directories does contain SSH host key files, but `ssh_host_keys` is set to a value other than `simple_sync`, then the PXE boot of the eLogin node will fail.

generate

When `ssh_host_keys` is set to `generate`, `esd` generates SSH host key files when PXE booting the eLogin node.

For this way of handling SSH host keys to work properly, SSH host keys must NOT be present in the Simple Sync directory structure in the CLE config set for the node. If they are present, the PXE boot of that node will fail.

/some/directory/path

When `ssh_host_keys` is set to an absolute path (a path starting with `/`), `esd` copies the site-supplied SSH host key files from the specified path location when PXE booting the eLogin node.

For this way of handling SSH host keys to work properly, SSH host keys must NOT be present in the Simple Sync directory structure in the CLE config set for the node. If they are present, the PXE boot of that node will fail.

The instructions for assigning a value to `ssh_host_keys` and disabling/enabling automatic SSH key generation in the CLE config set are provided in several procedures for configuring eLogin software.

For more information about how to configure SSH on a Cray XC, see "About Secure Shell Configuration" in *XC™ Series Software Installation and Configuration Guide (S-2559)*.

2.7 About eLogin and Cray Scalable Services

Cray Scalable Services organizes the SMW and all internal CLE nodes into tiers as a way to distribute data from the SMW to CLE nodes and aggregate data from CLE nodes to the SMW. The SMW is the server of authority (SoA), and the CLE nodes are tier1, tier2, or tier3 (a node can belong to only one of these tiers). Distribution through the tiers works like this:

- SoA is the server for tier1 nodes.
- 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.

The services outbound from the SMW are NTP (for time synchronization) and LiveUpdates (for enabling `zypper` actions on CLE nodes using repositories shared from the SMW to those nodes). The service inbound to the SMW is LLM (syslog data).

External nodes, such as eLogin nodes, need to use some of the same services as CLE nodes, but they are not included in the Scalable Services structure. Although external nodes can be considered "tier1" because they have a direct network connection to the SMW, they cannot simply be added to the `tier1_groups` setting in the `cray_scalable_services` configuration service, because that setting applies to internal CLE nodes only.

To address this, a new setting (`external_tier1_groups`) has been added to the `cray_scalable_services` config service to identify all external nodes (including eLogin nodes), and the Ansible play for Cray Scalable Services has been revised so that nodes in `external_tier1_groups` behave like the internal CLE nodes in `tier1_groups` with regard to NTP, LLM, and LiveUpdates.

2.8 About eLogin Image and Configuration Management

Image Management

As with releases prior to SMW 8.0.UP06 / CLE 6.0.UP06, the Image Management and Provisioning System (IMPS) is used to create recipes (`recipe`), package collections (`pkgcoll`), repos (`repo`), and image roots (`image create`) on the SMW.

For SMW-managed eLogin nodes, the following has changed:

- **New eLogin recipe.** A new eLogin recipe must be used. Sites with customized eLogin recipes must re-create the custom recipe and add the SMW-managed eLogin recipe as a subrecipe.
- **New command option.** The `image export` command uses a new option, `--format squashfs`, to export an eLogin image as a SquashFS image in `/var/opt/cray/imps/boot_images`.
- **Revised image groups file.** The `cray_image_groups.yaml` file has been changed in two ways:
 - Image specifications now include an `export format` field, which can have as its value any export format supported by the `image export` command. For eLogin images, `export_format` is set to `squashfs`. For most other images, it is set to `cpio`. Do not use a file extension (e.g., `.cpio`) when specifying the destination (`dest`) of an image.

- An image specification for eLogin has been added to the default image group so that the eLogin image can be created when `imgbuilder` is run.
- **New `imgbuilder` behavior.** When `imgbuilder` is run, it calls `image export` with the specified export format option. For an eLogin image, it will create a SquashFS boot image in `/var/opt/cray/imps/boot_images/imagename/imagename.sqsh`.

Configuration Management

The Cray Configuration Management Framework (CMF) comprises the configurator, config set data, IDS (IMPS Distribution System), `cray-ansible`, and Ansible plays. Of these, only config set data and Ansible plays have changed to accommodate SMW-managed eLogin nodes. The CLE config set has two new configuration services (`cray_external_cfgset_exclude` and `cray_kdump`), several configuration services with new settings, and new Ansible plays to consume these new configuration settings.

- **`cray_cfgset_exclude` (new service).** The `cray_cfgset_exclude` configuration service defines what files and directories should be excluded when the config set is delivered to the eLogin node. The eLogin node requests the config set from the external state daemon (`esd`) on the SMW, then `esd` does an `rsync` push, using the excludes assigned to this eLogin node, to deliver the sanitized config set to the eLogin node. The eLogin node never sees the data in `cray_cfgset_exclude`.
- **`cray_kdump` (new service).** The `cray_kdump` configuration service configures the kernel dump tool on eLogin nodes.
- **`cray_storage` (existing service, new settings).** The `cray_storage` configuration service includes settings to define the local storage layout for eLogin nodes, specifying which disks will be used to hold the needed file systems, their file system types, and their file system sizes. This is used during the booting process to prepare local storage for the node.
- **`cray_scalable_services` (existing service, new setting).** The `cray_scalable_services` configuration service includes a setting to define `external_tier1_groups` so that the functionality of Cray Scalable Services can be extended to eLogin nodes. The default value for this list of node groups is a list with a single member: `elogin_nodes`.

Mapping Boot Attributes to Nodes

When booting a node, whether the node is internal or external, it is necessary to specify attributes that are needed for booting the node: what boot image to use, what config set to use, and several other kernel parameters.

For CLE nodes, these kernel parameters, or boot attributes, are managed by the NIMS (Node Image Mapping Service) daemon, `nimsd`, using the `cnode` command line interface. The attributes are stored in the active NIMS map.

For eLogin nodes, boot attributes are managed by the external state daemon, `esd`, using the `enode` command line interface. The attributes are stored on the SMW. After a network boot (PXE boot) to provision the local storage on the node, the attributes are cached on the persistent storage of the node so that they will be available for future disk boots.

- For a disk boot, the boot attributes are stored in the grub menu in `/boot/grub2/grub.cfg` on the eLogin's internal storage.
- For a PXE boot, the boot attributes are stored on the SMW in a node-specific directory underneath `/opt/tftpboot/external` and passed as part of the PXE boot process.

2.9 About the eLogin Boot and Provisioning Process

eLogin Node Booting Process

An eLogin node is booted using one of three processes:

BIOS boot The BIOS boot enables interaction with the console for BIOS and system setup activities. It will pause the boot on the BIOS and System Setup screen for the console of the node.

PXE boot In a PXE boot, the eLogin node is started over the network connection and boots from the kernel and initramfs (boot image) provided by the PXE server. An eLogin node must be PXE booted from the network for an initial deployment.

For an initial deployment, the local storage is prepared with file systems and then the SMW securely transfers the eLogin image root via `rsync`, the PE image root via `rsync`, and the sanitized config set.

For a redeployment using a PXE boot, the code in the initramfs will detect whether a valid eLogin image root (in SquashFS format) and sanitized config set are present on the eLogin local persistent storage, and it will skip the transfer of anything that is a valid copy of the corresponding image root and sanitized config set on the SMW.

Disk boot In a disk boot, the eLogin node is booted from local storage. The SMW does not need to be available for a disk boot.

If power is lost or the eLogin node is manually reset, the default reboot is a disk boot. A disk boot can also be triggered using the `enode` command if there is no desire or need to change the image or config set on the node.

Whether an eLogin node is booted using PXE boot or disk boot, during the boot process, the eLogin node checks whether the assigned boot image is present on local persistent storage and whether the assigned config set is present on local persistent storage. If the boot image or the config set is not available locally, the eLogin node must request that the SMW `esd` daemon push them to the eLogin node.

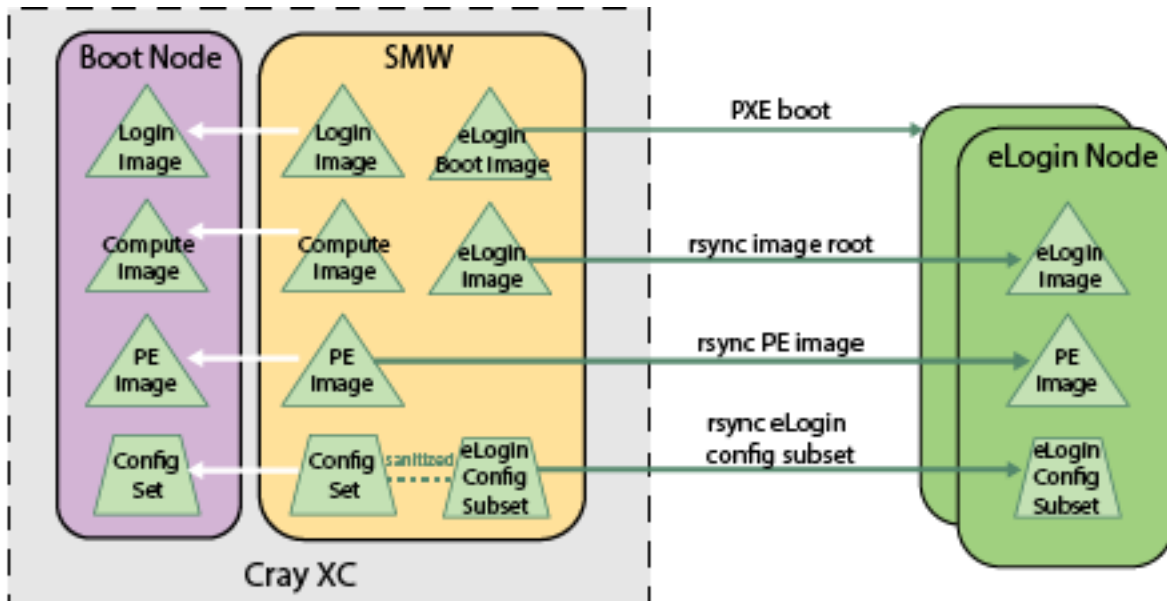
Distribution of eLogin Image and Config Set to eLogin Nodes

An eLogin node is initially provisioned with the node image root and sanitized config set during the initial PXE boot process. However, any time the eLogin node image root and config set are updated on the SMW, they must be pushed to the eLogin node using the `image sqpush` and `cfgset push` commands, respectively. The `cfgset push` command excludes the data specified in `cray_cfgset_exclude` to push a sanitized config set to the eLogin node.

The eLogin node will continue to run with a config set cached on local storage until the system administrator pushes a new (sanitized) config set to the node and runs `cray-ansible` on the node.

When a new image root has been created for an eLogin node and exported into SquashFS format, the system administrator can push that SquashFS-formatted image to the eLogin node, if the node is booted. The image will be cached on persistent storage.

Figure 8. Distribution of Images and Config Set from SMW to eLogin Nodes



Distribution of the PE Image to eLogin Nodes

The PE image root to be used on an eLogin node is specified in the `cray_image_binding` config service. When the eLogin node is first provisioned with an eLogin image root for the operating system, the PE image root will also be transferred to local persistent storage on the node as part of the PXE boot process. Until the PE image root has been transferred to an eLogin node, the node is not ready for users.

For internal CLE nodes, when the PE image root is updated with a new release of the PE software, the `image sqpush` command is used to push the PE image root in SquashFS format to the boot node. For SMW-managed eLogin nodes, use the same command to push the PE image root in SquashFS format to an eLogin node. That command can also be used to push the PE image root to multiple eLogin nodes by adding multiple `-d hostname` arguments or by using a node group.

All image roots are stored under `/var/opt/cray/persistent/image_roots`, which is on the local persistent storage for the eLogin node. A symbolic link is made from `/var/opt/cray/imps/image_roots` to this persistent location. After the PE image is synchronized from the SMW to `/var/opt/cray/imps/image_roots`, the same `image_binding` Ansible play that mounts the PE image root SquashFS for internal CLE nodes is used on the eLogin node. The only difference is that the base location is local storage instead of a network mount.

2.10 About Storage Profiles for eLogin Nodes

Storage profiles define the disk layout and partition information for internal disks on eLogin nodes. The profiles are defined in the `cray_storage` service in the CLE config set that is assigned to each eLogin node.

- Multiple eLogin nodes may use the same storage profile, but only one storage profile can be assigned to a single eLogin node at a time.

- A config set can define multiple storage profiles, which enables administrators to maintain multiple profiles (default, test, production, etc.) in a config set and switch between them as needed.
- A storage profile can be disabled.
- The storage profile assigned to an eLogin node must be enabled before the eLogin node can be booted.

Disk Layout in the eLogin Default Profile

When installed, the Cray-provided `eloin_default` storage profile can be found in the `cray_storage` configuration service. To view the default settings, update a CLE config set (p0 in example) in prepare mode, then view the worksheet and search for

```
cray_storage.settings.storage.profiles.data.eloin_default.
```

```
smw# cfgset update -m prepare p0  
smw# vi /var/opt/cray/imps/config/sets/p0/worksheets/cray_storage_worksheet.yaml
```

eLogin nodes require a certain set of partitions in order to properly function. The `eloin_default` storage profile satisfies the eLogin node partition requirements. It includes two disks: `/dev/sda`, which contains nonpersistent partitions, and `/dev/sdb` for persistent partitions.

/dev/sda Stores nonpersistent data and contains the TMP, WRITELAYER, BOOT, GRUB, and SWAP partitions. This disk layout is set to be repartitioned and its file systems re-created on every boot by default.

/dev/sdb Stores persistent data and contains the CRASH and PERSISTENT partitions. This disk layout is set to be persisted on node boot. The partitions will not be re-created nor will the file systems on the partitions be re-created on each boot by default.

Persistence behavior is handled at the disk level in storage profiles, not the partition level. A disk can be set to have all of its partitions persistent by setting the `persist_on_boot` value to `true` in the storage profile in the `cray_storage` service.

Required Partitions

An eLogin node requires the following partitions to properly function. These partitions are included in the `eloin_default` storage profile provided with the `cray_storage` config service. Any custom storage profiles assigned to eLogin nodes must also contain partitions with these labels.

- | | |
|-------------------|---|
| GRUB | Partition for storing the GRUB bootloader data. Not persistent. Should be at least 1MiB in size with a file system type of ext3. |
| BOOT | Partition for storing kernels, initrds, and GRUB configuration for booting. Minimum size is 1 GiB. (Note binary value. See Prefixes for Binary and Decimal Multiples on page 206.) |
| WRITELAYER | Partition for use with the writeable overlay of the eLogin image. This partition is erased and its file system reformatted on every boot even if <code>persist_on_boot: true</code> is set on the disk it resides on. |
| TMP | Partition for the temp file system. This partition is erased and its file system reformatted on every boot even if <code>persist_on_boot: true</code> is set on the disk it resides on. |
| SWAP | Standard linux swap partition. This partition is erased and its file system reformatted on every boot even if <code>persist_on_boot: true</code> is set on the disk it resides on. |
| CRASH | Partition for storing kdump data. By default, this partition is expected to be persistent across boots. |

PERSISTENT Partition for storing data that should persist between image deployments, such as config sets, security keys, and boot images. By default, this partition is expected to be persistent across boots. Minimum size is 200 GiB (note binary value).

The partitions in the `elogin_default` profile have the Cray-recommended values for the file system type, size, and partition flag fields. These values should be modified to fit the requirements of eLogin nodes in this system.

IMPORTANT: The sum of the sizes of all of the volatile data partitions on the first disk (`/dev/sda`) must be less than the available storage on the first disk. Similarly, the sum of the sizes of all of the persistent data partitions on the second disk (`/dev/sdb`) must be less than the available storage on the second disk.

Update the `cray_storage` config service so that the storage profile assigned to each eLogin node has file system sizes that fit within the available storage on each disk of that node.

The partitions in the `elogin_default` profile also have default values for the partition `mount_point` and `mount_options` fields. These values are NOT configurable currently.

IMPORTANT: The default values provided in the `elogin_default` storage profile are used to mount the file system on each partition, if provided. Users should not mount or use the directories specified by the partitions in the `elogin_default` storage profile.

Managing Partitions and Persistent Data

For nonpersistent disks (`persist_on_boot: false`), changes to the partition configuration in the storage profile are applied during the bringup of the node. Partition sizes, file system types, and partition ordering can be safely modified because all of the partitions are removed and re-created at boot time.

For persistent disks (`persist_on_boot: true`), only the addition of partitions is supported and only if the disk contains adequate space for the new partition(s). Resizing, reordering, and removing partitions are not supported as long as the `persist_on_boot` remains true. Changing file system types on partitions is also not supported on persistent disks.

To reprovision a nonpersistent or persistent disk on a booted eLogin node, see [Manage Partitions and Persistent Data on an eLogin Node](#) on page 155.

Overview of Storage Setup on eLogin Nodes

The initial setup of storage on eLogin nodes follows this procedure:

1. Create a storage profile in a CLE config set that will be assigned to the eLogin nodes. Use the default storage profile provided for eLogin nodes or create a custom storage profile, as needed.
2. Validate the config set to validate the storage profile data. Validation rules specific to eLogin nodes will be applied by `enode` commands later.
3. Assign a storage profile when an eLogin node is created or updated. The profile does not need to exist in the config set when the eLogin node is created; `enode update` can be used to assign the profile at a later time.
4. Validate the eLogin storage profile (validates profile existence, enabled status, and the existence of required partitions), which occurs when one of these commands is run: `enode validate`, `enode boot`, or `enode reboot`.
5. eLogin nodes provision their internal storage, as specified in the assigned storage profile, during node bringup.

2.11 About the External State Daemon and eLogin Node States

The external state daemon, `esd`, resides on the SMW and provides a service to manage external nodes, including eLogin nodes. The `esd` does the following:

- Maintains node registry.
- Maintains node state.
- Maintains the following configuration for eLogin nodes:
 - Console logging configuration
 - DHCPD configuration
 - SMW `/etc/hosts` entries
 - TFTP configuration in `/opt/tftpboot/external/...`
- Performs all node life-cycle tasks: boot, reboot, stage, shutdown, and status check.

Why Node States are Important

Awareness of the state of an eLogin node is important for system administration and for system security. System administrators need to know the state of all eLogin nodes: whether the nodes are powered on or off, in the process of booting, or ready for users to log in and do work. The `esd` uses the state of eLogin nodes to maintain security of the SMW and XC system during the PXE boot of an eLogin node. Depending on the eLogin node state, `esd` opens and closes access between an eLogin node and the SMW. Open access is needed to transfer the following from the SMW to the eLogin now at the proper points in the PXE boot:

- X.509 certificate (so that the SMW will trust the identity of the eLogin node)
- public root SSH key (so that the eLogin node will trust `root@smw` for SSH)
- operating system image root
- PE image root

At other points in the PXE boot, access will be closed so that the eLogin node does not have unfettered access to the SMW. When the node boots from disk, there is no need for the opening and closing of access to the SMW.

The states of an eLogin node are different and more numerous than the states of an internal CLE node that are stored in the HSS database. With internal nodes, it is important to know only whether the node is powered on or off and whether it is ready for access by users. With external nodes such as eLogin nodes, additional states are needed to maintain system security.

eLogin Node States Initiated on the SMW

The following states are initiated on the SMW and require no message from an eLogin node. Note that states are entered at the beginning of the work, and the next state is entered when work for that state begins. Therefore, a state indicates that the work associated with that state is being performed, not that the work for that state has been completed.

Table 5. States Initiated on the SMW

Node State	esd Actions
prepare_exports	esd opens up the security profile for transfer of information to the node. This state occurs only during a PXE boot.
power_on	The <code>enode boot</code> and <code>enode reboot</code> commands cause esd to transition the node to the <code>power_on</code> state.
status_wait	<p>After the <code>power_on</code> state, esd transitions the node to the <code>status_wait</code> state, indicating that it is waiting for communication from the node while the BIOS power-on self test (POST) is finishing.</p> <p>During this state, certificates are retrieved from the SMW for a PXE boot. The state can be advanced only when secure communication can be established between the node and esd. If a node is booting from disk, and the network connection via the management network is missing or misconfigured, the node will continue to boot while esd will remain in <code>status_wait</code> state.</p>
shut_down	The <code>enode shutdown</code> command causes esd to transition the node to the <code>shut_down</code> state. The <code>shut_down</code> state begins with issuing a soft power-off to the node, then waits for a timeout before issuing a hard power-off. If the hard power-off fails, the node will enter the <code>Error</code> state. The <code>shut_down</code> state indicates that the node is in the process of shutting down.
node_off	The <code>node_off</code> state is the result of a successful node shutdown. It should correspond with the power status of the node. However, if an IPMI power command is issued or the physical power button is pressed on the node, the state will not reflect the correct status of the node as shown by the <code>enode status</code> or <code>enode list</code> commands. An <code>enode status</code> command will recheck the power status of the node, even if it is off. If the node is no longer powered off, esd will transition the node to the <code>UNKNOWN</code> state.
UNKNOWN	The <code>UNKNOWN</code> state typically occurs when esd is started on the SMW. esd can check for whether an eLogin node is physically powered off, but otherwise, esd does not yet know the state of the eLogin node.
Error	<p>The esd daemon will not intentionally put a node into the <code>Error</code> state, but any of the other states can transition to this state. If that happens, esd does some cleanup, such as closing any security access.</p> <p>To leave the <code>Error</code> state, the node must be shut down or rebooted. There is no error-recovery command that will enable the node to continue to boot or set the node state to <code>node_up</code>.</p>

eLogin Node States initiated by a Message from an eLogin Node to the SMW

Some state transitions are initiated by an eLogin node sending a message to esd on the SMW. Such messages can be sent by calling dracut scripts during early boot phases or by calling `cray-ansible` in the booted phase.

Dracut scripts in early boot phases

Several of the Cray dracut scripts send a message by calling `/bin/cray/dracut_dispatch_state` with a state payload. The `esd` daemon requires that the eLogin node has a valid certificate for the node running with that host name and IP address. If there is a mismatch, the connection attempt by `dracut_dispatch_state` will be rejected.

When a CLE node is booted, the `/init` script starts running and (among other actions) calls `cray-ansible` in the init phase, switches to `systemd`, which (among other actions) calls `cray-ansible` in the booted phase, and then the node is up.

In contrast, when an eLogin node is booted, the `/init` script is effectively replaced by several dracut scripts (some core, some Cray-enhanced) that are run in the eLogin pre-mount phase. In the pre-mount phase, the environment is prepared so that the necessary image root is on local storage. When the pre-mount phase is complete, the eLogin boot moves into the pre-pivot phase. The pre-pivot phase ensures that everything else needed on the node is present so that the node can pivot from using the small `initrd` image (used in the early part of the boot process) to using storage that was just put onto the disk from the full SquashFS-formatted image root.

cray-ansible in booted phase

Three state transitions occur in multi-user mode (booted phase), and to track those state transitions, messages to `esd` are sent by `cray-ansible`.

States that are initiated by a message from an eLogin node to the SMW are listed in the following table. The order represents a healthy boot. Note that there are fewer states associated with a disk boot than with a PXE boot.

The last state listed in the table, `staging`, is applicable only to nodes that are already booted and are being staged for a later boot.

Table 6. States Initiated by Message from eLogin Node to the SMW

Node State	Occurs During		Node Actions
	PXE Boot	Disk Boot	
States controlled by dracut scripts:			
<code>storage_send</code>	yes	N/A	Node requests access to storage configuration from <code>esd</code> and then transfers the <code>storage.yaml</code> via TFTP. Storage configuration information is stored in the <code>cray_storage</code> config service.
<code>provisioning</code>	yes	N/A	Node applies storage configuration to format the storage for the node.
<code>sync_root</code>	yes	N/A	Node NFS-mounts SMW <code>/var/opt/cray/imps/boot_images/image</code> read-only and copies the SquashFS-formatted image root with the operating system to persistent storage.
<code>mount_root</code>	yes	yes	Node prepares OverlayFS with writable layer and mounts the SquashFS image root read-only.
<code>grub_install</code>	yes	yes	Node installs GRUB2 on BOOT device, which enables future disk boots.
<code>setup_hosts</code>	yes	yes	Node sets up host file to ensure it has enough to continue to the next boot phase (which will be the booted phase).

Node State	Occurs During		Node Actions
	PXE Boot	Disk Boot	
<code>config_sync</code>	yes	N/A	Node prepares OverlayFS with config set directory and starts <code>sshd</code> so <code>esd</code> can push config sets.
<code>config_send_global</code>	yes	N/A	Node requests <code>esd</code> to send the global config set.
<code>config_send_cle</code>	yes	N/A	Node requests <code>esd</code> to send the CLE config set and then stops <code>sshd</code> .
<code>link_cfgset</code>	yes	yes	Node creates links between the persistent storage where the config sets are placed and the running system.
<code>cray_ansible_ininit</code>	yes	yes	Node runs <code>cray-ansible</code> in the init phase of boot to run Ansible plays.
<code>udev_rules</code>	yes	yes	Node runs the <code>udev</code> rules script to properly order the network interfaces.
<code>hostbased_auth</code>	yes	yes	Node prepares host-based authentication.
States controlled by <code>cray-ansible</code> :			
<code>cray_ansible_booted</code>	yes	yes	Node begins running <code>cray-ansible</code> in the booted phase.
<code>image_binding_sync</code>	yes	yes	Node continues in <code>cray-ansible</code> in the booted phase to request <code>esd</code> to transfer from the SMW to persistent storage any SquashFS-formatted images for the <code>cray_image_binding</code> profiles that apply to this node.
<code>node_up</code>	yes	yes	Node finishes in <code>cray-ansible</code> in the booted phase and indicates that everything is up, ready for users to log in.
State applicable only to staging			
<code>staging</code>	N/A	N/A	The <code>enode stage</code> or <code>enode reboot --staged</code> commands cause the node to transition to the <code>staging</code> state from the <code>node_up</code> state. When the node is done staging, it transitions back to <code>node_up</code> .

Checking Node State

To check node state, use the `enode status` command. The output of this command has the following four columns. The fourth column reports state.

- **NODE:** name of the eLogin node
- **PING:** whether the node is pingable (up/down)
- **POWER:** whether the chassis power is on or off
- **STATE:** state of the node

States that are of short duration may be difficult to capture using `enode status`, but all states and state transitions are logged in the `esd` log.

State vs. status. State indicates what processes the node and/or `esd` may be performing for the node. In contrast, status is independent information about the node that is broadly applicable across states, including information about whether the node responds to a `ping` command and whether the node power is on or off.

3 Migrate from CIMS/CDL or from CMC/eLogin

To migrate from CIMS/CDL or CMC/eLogin to SMW-managed eLogin,

1. Open a SFDC case with Cray Customer Service.
2. Use the Cray `smw_enode_migration` tool to gather configuration data.

The `smw_enode_migration` tool and a PDF of instructions for its use, *MIGRATION: ESLOGIN, CDL, ELOGIN TO CLE-6.0.UP06*, are available for download from Customer Service after opening an SFDC case. See SFDC Knowledge Article 6642 for more details.

This tool generates an `inventory.csv` file, which will be used to register eLogin nodes with the SMW later in the process.

3. Follow the installation/configuration process provided in [Install and Configure eLogin](#) on page 39.

The following list shows tasks in that process that are either unique to the migration process or are sufficiently important that they warrant emphasis. Use this list as a checklist to ensure that nothing has been overlooked; do not use this list as a procedure, because many other steps are needed.

Table 7. eLogin Migration Checklist

✓	Task	Notes
	Collect required configuration data. For a migration, some of this data is already in the <code>inventory.csv</code> file, which was generated by the Cray <code>smw_enode_migration</code> tool in step 2.	
	Power off eLogin nodes (unique to migration). On the CMC or CIMS, power off the eLogin nodes so that they are in a known, powered-off state prior to switching the cabling.	
	Switch cabling (unique to migration). Switch cabling of the ToR switch from the CMC or CIMS to the SMW.	
	Update iDRAC. Change iDRAC settings on each eLogin node so that the SMW can communicate with the node's BMC.	
	Use <code>site</code> key for site-user-net. Ensure that the site-user-net network is configured in <code>cray_net</code> with <code>site</code> as the key. That particular key is	

✓	Task	Notes
	<p>expected in certain Ansible plays, and the system will not work unless that key value is used.</p>	
	<p>Migrate any manually added site host data (unique to migration).</p> <p>Any site host entries and other data added directly to <code>/etc/hosts</code> on eLogin nodes in previous releases must now be migrated to <code>hosts_ext.head</code> or <code>hosts_ext.tail</code>. And if Simple Sync was used to place <code>/etc/hosts</code> files with site data on eLogin nodes, remove those files from the Simple Sync directory structure.</p>	
	<p>Reconfigure disks and partitions.</p> <p>Storage for SMW-managed eLogin nodes is configured in storage profiles defined in the <code>cray_storage</code> configuration service. For any eLogin node that was managed by a CMC or CIMS, it is necessary to create/assign a profile that has <code>persist_on_boot</code> set to <code>false</code> for the <code>dev/sdb</code> device so that the correct storage layout can be applied on the first PXE boot.</p> <p>After a successful PXE boot, either change that profile to configure <code>dev/sdb</code> to be persistent, or assign a different profile to the node that already has <code>dev/sdb</code> configured to be persistent.</p>	
	<p>Reconfigure the internal RAID virtual disks.</p> <p>Do not skip the procedure that configures eLogin RAID virtual disks. It is required for migration so that the virtual disks, <code>/dev/sda</code> and <code>/dev/sdb</code>, are reconfigured as needed for the storage profile(s) in <code>cray_storage</code>.</p>	

4 Install and Configure eLogin

All of the software needed to manage an eLogin node from the SMW is installed from the SMW media and the CLE media. No other software needs to be installed for eLogin. However, additional configuration is required.

The following procedures describe how to configure the SMW to manage eLogin nodes and configure eLogin nodes to be managed by the SMW. This includes configuring SMW networks, updating CLE configuration worksheets, configuring eLogin node hardware, enrolling eLogin nodes into the node registry, creating an eLogin image, assigning that image to eLogin nodes, and booting eLogin nodes.

Known Issues that Occur at SMW Reboot or Failover

With this initial release of SMW-managed eLogin, the following issues occur when the SMW is rebooted or when the active SMW of an SMW HA pair fails over and the passive SMW becomes active. Cray expects these issues to be fixed in a future CLE 6.0 release.

After an SMW reboot or failover:

- eLogin nodes are in state UNKNOWN.

Issue: Any eLogin node that was not in state `node_off` when the SMW was rebooted will be assigned state UNKNOWN by `esd` after it restarts.

Workaround: Reboot any eLogin node in state UNKNOWN.

```
smw# enode status elogin1
smw# enode reboot elogin1
```

- ConMan (`conman`) is not running.

Issue: The SMW uses ConMan to provide console logging and interactive console access for eLogin nodes. Currently, ConMan is started automatically when an eLogin node is added to the node registry, but it is not started automatically when the SMW is booted.

Workaround: Manually restart ConMan after rebooting the SMW or after a failover of the active SMW to the passive SMW in an SMW HA configuration.

IMPORTANT: ConMan should be running only on the active SMW. Failure to start it on only the active SMW will result in loss of console logging and console access for eLogin nodes.

```
smw# systemctl restart conman
```

- SMW firewall may not open ports correctly.

Issue: If the `SuSEfirewall2` service starts before the `rpcbind` service as the SMW boots, ports in the firewall that depend on `rpcbind` to help manage them will remain closed. The NFS port, port 2049, is one of the ports that depend on `rpcbind`. If the NFS port is not open, eLogin nodes will be unable to boot because they will be unable to NFS-mount the necessary file systems.

Workaround: Instructions are located in [Ensure that NFS Port is Open in Firewall](#) on page 70.

4.1 Collect Required Configuration Information

Prerequisites

SMW with SMW 8.0.UP06 and CLE 6.0.UP06 software installed.

About this task

This procedure collects the information needed to begin the SMW-managed eLogin installation and configuration process. It applies to both an initial deployment of SMW-managed eLogin and a migration from CIMS/CDL or CMC/eLogin to SMW-managed eLogin.

Procedure

1. Collect SMW and network configuration information.

The following table lists the SMW and network information needed to connect the SMW to eLogin nodes. Enter values in the last column.

Table 8. SMW and Network Configuration Information

Item	Default Value	Value for this System
SMW host name	smw	
external-ipmi-net network	10.6.0.0	
external-ipmi-net netmask	255.255.0.0	
SMW IP address on external-ipmi-net	10.6.1.1	
external-management-net network	10.7.0.0	
external-management-net netmask	255.255.0.0	
SMW IP address on external-management-net	10.7.1.1	
site-user-net network		
site-user-net netmask		
site-user-net DNS servers		
site-user-net DNS domain		
site-user-net NTP servers		
site-user-net default gateway		

2. Collect eLogin node configuration information.

The following table lists the eLogin node information needed to set up each eLogin node for management by the SMW. In a system with multiple eLogin nodes, this information is required for each node: add a "Value for this Node" column for each additional eLogin node to be configured for this system.

- **INITIAL DEPLOYMENT:** Enter a value for each item in each node column.

- **MIGRATION:** Enter a value for each item/column except for "name" and "MAC address." The `inventory.csv` file that was generated by the `smw_enode_migration` tool earlier in the migration already has that information.

The `inventory.csv` file also has the BMC IP address, but it is the wrong address for SMW-managed eLogin. Enter the correct BMC IP address in the table—one that is on the 10.6.0.0 network—for each eLogin node.

Table 9. eLogin Node Configuration Information

Item	Value for this Node
name	
BMC IP address	
BMC user ID	
BMC password	
Boot interface*	
MAC address*	
Number and size of internal storage devices	

* To determine these values, see [Determine Boot Interface and MAC Address](#) on page 41.

4.1.1 Determine Boot Interface and MAC Address

Prerequisites

SMW and network configuration information has been gathered.

About this task

This procedure determines the boot interface and MAC address of an eLogin node.

The boot interface depends on the hardware being used. The boot interface is the first 1GbE interface, per one of the following configurations:

- eth0 on eLogin nodes with the 4x1GbE LOM (LAN on motherboard) network adapter
- eth2 on eLogin nodes with the 2x10GbE+2x1GbE LOM network adapter

Procedure

1. Power up the node.

When the BIOS power-on self-test (POST) process begins, press the **F2** key immediately 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
```

Figure 9. BIOS Config Screen

```

F2 = System Setup
F10 = Lifecycle Controller
F11 = BIOS Boot Manager
F12 = PXE Boot

Phoenix ROM BIOS PLUS Version 1.10 2.0.21
Copyright 1985-1988 Phoenix Technologies Ltd.
Copyright 1990-2014 Dell Inc.
All Rights Reserved

Dell System PowerEdge R820
www.dell.com

Four 2.40 GHz Eight-core Processors, Bus Speed:8.00 GT/s, L2/L3 Cache:2 MB/20 MB
System running at 2.40 GHz
System Memory Size: 512.0 GB, System Memory Speed: 1600 MHz, Voltage: 1.5V

Dell Serial ATA AHCI BIOS Version 1.0.2
Copyright (c) 1988-2014 Dell Inc.

```

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 **System Setup Main Menu** screen appears with the following sub-menus:

```

System BIOS
iDRAC Settings
Device Settings

```

2. Select **Device Settings** from the **System Setup Main Menu**, then press **Enter**.

Figure 10. System Setup Main Menu: Device Settings

```

System Setup
System Setup Main Menu
System BIOS
iDRAC Settings
Device Settings

```

3. Select **Integrated NIC 1 Port N ...** in the **Device Settings** window.

Choose the NIC port number that corresponds to the Ethernet port for the external-management-net network:

- If external-management-net uses the first Ethernet port (eth0), select **Integrated NIC 1 Port 1 ...**
- If external-management-net uses the third Ethernet port (eth2), select **Integrated NIC 1 Port 3 ...**

Figure 11. Device Settings: Integrated NIC Port Number

```

System Setup

Device Settings

RAID Controller in Slot 7: Dell PERC <PERC H310 Adapter> Configuration ...
Integrated NIC 1 Port 1: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 ...
Integrated NIC 1 Port 2: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 ...
Integrated NIC 1 Port 3: Broadcom NetXtreme II 1 Gb Ethernet BCM57800 -...
Integrated NIC 1 Port 4: Broadcom NetXtreme II 1 Gb Ethernet BCM57800 -...

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

```

4. Verify that the correct NIC port number is selected, then press **Enter** to open the **Main Configuration Page**.
5. Identify the **Permanent MAC Address** on the **Main Configuration Page** screen.

The following figure shows an example MAC address.

Figure 12. Integrated NIC Port / Main Configuration Page: MAC Address

```

Integrated NIC 1 Port 1: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 -

Main Configuration Page

Main Configuration Page

Firmware Image Menu
Device Hardware Configuration Menu
MBA Configuration Menu
iSCSI Boot Configuration Menu
FCoE Boot Configuration Menu
Virtualization Mode                <None>
Blink LEDs                          [0 ]
Chip Type                            BCM57800 B0
PCI Device ID                        168A
Bus:Device:Function                 01:00:00
Link Status                          <Disconnected>
Permanent MAC Address                [90:B1:1C:39:EA:38 ]
Virtual MAC Address                  [90:B1:1C:39:EA:38 ]

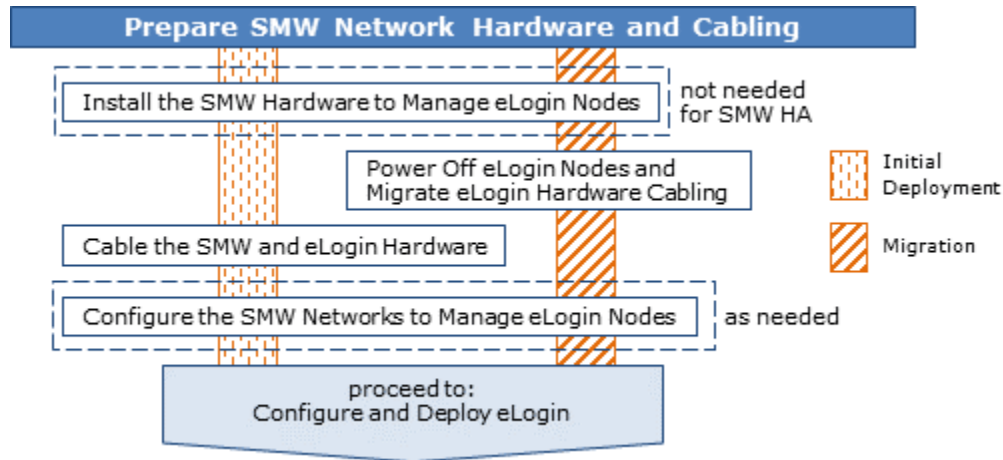
```

6. Record the MAC address and the boot interface in the table in [SMW, Network, and eLogin Configuration Information](#).
7. Press **Esc** to exit to the **Device Settings** menu.
8. Select **No** when prompted with the "Settings have changed" message, then press **Enter**.
9. Press **Esc** to exit the **System Setup Main Menu**.
The **System Setup Main Menu** screen appears.
10. Press **Esc** to exit the **System Setup Main Menu**.

4.2 Prepare SMW Network Hardware and Cabling

Starting with CLE 6.0.UP06, eLogin nodes are managed from the SMW instead of the CMC. The following procedures prepare the network hardware for a stand-alone SMW or an SMW HA pair, configure the SMW networks to manage eLogin nodes, and cable the hardware appropriately.

Figure 13. Visual Guide to Preparing SMW Network Hardware and Cabling



4.2.1 Install the SMW Hardware to Manage eLogin Nodes

Prerequisites

SKIP THIS PROCEDURE IF THE SMW ALREADY HAS ETH6 AND ETH7 INTERFACES.

A stand-alone SMW with:

- no eth6 and eth7 Ethernet interfaces
- available card slot for the addition of a quad port Ethernet card

About this task

To manage eLogin nodes from the SMW, the SMW must have two available network interfaces to connect to external-ipmi-net and external-management-net. This procedure prepares a stand-alone SMW for the installation of this quad port Ethernet card.

Although a stand-alone SMW could be briefly removed from support of an XC system, Cray recommends shutting down the CLE nodes before adjusting the hardware on the SMW to add a quad Ethernet card.

Procedure

1. Check whether the boot node is up.

```
smw# ping -c3 boot
```

2. If the boot node is up, then shut down the CLE system.

```
smw# su - crayadm
crayadm@smw> xtbootsys -s last -a auto.hostname.stop
crayadm@smw> exit
smw#
```

If this site does not have an `auto.hostname.stop` file, use the following command to shut down the system.

```
crayadm@smw> xtbootsys -s last -a auto.xtshutdown
```

3. Shut down the SMW.

```
smw# shutdown -h now;exit
```

4. Install the quad port Ethernet card into an available slot.

5. Power up the SMW.

6. Reboot the CLE nodes.

```
smw# su - crayadm
crayadm@smw> xtbootsys -a auto.hostname.start
```

Further configuration of the SMW to manage eLogin nodes does not impact CLE nodes once they are booted. However, the CLE config set will be modified in a later procedure, and that should not be done while CLE nodes are still in the process of booting.

4.2.2 Power Off eLogin Nodes and Migrate eLogin Hardware Cabling

Prerequisites

FOR MIGRATION ONLY. SKIP THIS PROCEDURE FOR AN INITIAL DEPLOYMENT.

- SMW or SMW HA pair has SMW 8.0.UP06 / CLE 6.0.UP06 software installed and eth6 and eth7 Ethernet interfaces available.
- CMC is connected to one or more eLogin nodes.

About this task

Migrating to SMW-managed eLogin requires a network change in the SMW and a cabling change on the ToR switch connecting the eLogin nodes. The SMW network change was done in the previous procedure. This procedure describes the necessary cabling change for the TOR switch.

The CMC uses the following two Ethernet interfaces to connect to and manage the eLogin nodes:

em2 or en2 (CMC) Interface for the 10.142.x.x network, which is used for provisioning/booting the eLogin nodes.

em3 or en3 (CMC) Interface for the 10.148.x.x network, which is the IPMI network for the eLogin nodes.

The SMW uses the following two Ethernet interfaces to connect to and manage the eLogin nodes:

eth6 (SMW) Interface for the 10.6.x.x network (external-ipmi-net), which is the IPMI network for the eLogin nodes.

eth7 (SMW) Interface for the 10.7.x.x network (external-management-net), which is used for provisioning/booting the eLogin nodes.

When migrating from CMC-managed eLogin to SMW-managed eLogin, sites must disconnect the cables from the CMC interfaces and then connect the cables to the correct SMW interfaces.

IMPORTANT: The SMW and CMC must not be connected to the eLogin ToR switch at the same time. This could cause the eLogin nodes to be randomly rebooted or power cycled.

Procedure

1. Power off the eLogin nodes.

For CMC/eLogin, use the `nova stop` command.

2. Disconnect the CMC em2/en2 interface and place that cable near the SMW eth7 interface, but do not connect it to the SMW yet.
3. Disconnect the CMC em3/en3 interface and place that cable near the SMW eth6 interface, but do not connect it to the SMW yet.
4. Connect the cable near the SMW eth7 interface to that interface.
This cable is the one that was formerly connected to the CMC em2/en2 interface.
5. Connect the cable near the SMW eth6 interface to that interface.
This cable is the one that was formerly connected to the CMC em3/en3 interface.

4.2.3 Cable the SMW and eLogin Hardware

Prerequisites

FOR INITIAL DEPLOYMENT ONLY. SKIP THIS PROCEDURE IF PERFORMING A MIGRATION.

- SMW or SMW HA pair with SMW 8.0.UP06 and CLE 6.0.UP06 software installed and eth6 and eth7 Ethernet interfaces available.
- Ethernet cables for connecting the SMW and eLogin nodes to the ToR switch.

About this task

The SMW uses the following two Ethernet interfaces to connect to and manage the eLogin nodes:

eth6 (SMW) Interface for the 10.6.x.x network (external-ipmi-net), which is the IPMI network for the eLogin nodes.

eth7 (SMW) Interface for the 10.7.x.x network (external-management-net), which is used for provisioning/booting the eLogin nodes.

This procedure cables the SMW eth6 interface and eth7 interface to the eLogin nodes. It applies to both a stand-alone SMW and an SMW HA pair.

Procedure

1. Connect eLogin nodes to a top-of-rack (ToR) switch.
2. Connect an Ethernet cable from the ToR to eth6 on the SMW quad port Ethernet card.
This cable should be connected to the Ethernet switch port configured for the external-ipmi-net.
3. Connect an Ethernet cable from the ToR to eth7 on the SMW quad port Ethernet card.
This cable should be connected to the Ethernet switch port configured for the external-management-net.

4.2.4 Configure the SMW Networks to Manage eLogin Nodes

Prerequisites

- The SMW has the eth6 and eth7 Ethernet ports (interfaces) available and connected to these networks:
 - external-ipmi-net (eth6)
 - external-management-net (eth7)
- The SMW eth6 and eth7 Ethernet interfaces do not yet have the proper IP addresses and netmask assigned.

SKIP THIS PROCEDURE if this system has had a fresh install of SMW 8.0.UP06 / CLE 6.0.UP06. It is unnecessary because the installation of the operating system using the bootable SMW SLES media (Cray-slebase-12-SP3-201709141039.iso or DVD) sets the proper default values for eth6 and eth7.

This procedure applies to both a stand-alone SMW and an SMW HA pair, and to an initial deployment and a migration.

About this task

This procedure configures the SMW eth6 Ethernet interface for use on external-ipmi-net, and the eth7 interface for use on external-management-net.

For a stand-alone SMW, the IP address for eth6 is set to 10.6.1.1, and the IP address for eth7 is set to 10.7.1.1. For SMW HA, the addresses 10.6.1.1 and 10.7.1.1 cannot be set on both smw1 and smw2 at the same time. Instead, the IP addresses on smw1 are set to 10.6.1.2 for eth6 and 10.7.1.2 for eth7, and the IP addresses on smw2 are set to 10.6.1.3 for eth6 and 10.7.1.3 for eth7. The configuration of the virtual IP addresses of 10.6.1.1 and 10.7.1.1 will be done by `SMWHAconfig` such that these virtual IP addresses become resources that are enabled on the active SMW in the SMW HA pair.

Procedure

1. (Conditional) If this is an SMW HA system, place the SMW HA pair in maintenance mode before making any network changes.
 - a. Enable maintenance mode on the active SMW.

```
smw1# maintenance_mode_configure enable
smw1# sleep 240
```
 - b. Check cluster status.

Verify that ALL cluster resources have stopped.

```
smw1# crm_mon -r1
```

2. Use the `yast2` command to configure LAN on the SMW.

For an SMW HA pair, run this command and change network settings on both SMWs.

```
smw# yast2 lan
```

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

```
————— CONFIGURE ETH6 —————
```

3. Select the **eth6** line on the **Overview** tab, then select **Edit**.

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

4. Select **Statically Assigned IP address** on the **Address** tab and enter values for IP address, subnet mask, and host name (including the domain name).
 - IP address for a stand-alone SMW: 10.6.1.1
 - IP address for SMW HA: 10.6.1.2 for smw1, 10.6.1.3 for smw2
 - subnet mask: 255.255.0.0
 - host name for a stand-alone SMW (suggested): smw-net6
 - host names for SMW HA (suggested): smw1-net6, smw2-net6

Note that the host names for the two SMWs in an SMW HA pair should be different.

5. On the **General** tab, ensure that **Activate Device** is set to **At Boot Time**. Then select **Next**.

This will configure `STARTMODE='auto'` correctly in the resultant `ifcfg-eth6` file on the SMW.

```
————— CONFIGURE ETH7 —————
```

6. Select the **eth7** line on the **Overview** tab, then select **Edit**.

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

7. Select **Statically Assigned IP address** on the **Address** tab and enter values for IP address, subnet mask, and host name (including the domain name).
 - IP address for a stand-alone SMW: 10.7.1.1
 - IP address for SMW HA: 10.7.1.2 for smw1, 10.7.1.3 for smw2
 - subnet mask: 255.255.0.0
 - host name for a stand-alone SMW (suggested): smw-net7
 - host names for SMW HA (suggested): smw1-net7, smw2-net7

8. On the **General** tab, ensure that **Activate Device** is set to **At Boot Time**. Then select **Next**.

This will configure `STARTMODE='auto'` correctly in the resultant `ifcfg-eth7` file on the SMW.

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

10. (Conditional) If this is an SMW HA system, take the SMW HA pair out of maintenance mode and wait for the cluster to stabilize.

- a. Disable maintenance mode on the active SMW.

```
smw1# maintenance_mode_configure disable
smw1# sleep 240
```

- b. Check cluster status.

Verify that ALL cluster resources have started.

```
smw1# crm_mon -r1
```

4.3 Configure and Deploy eLogin

4.3.1 Start a Typescript File Before Configuring eLogin Software

About this task

Sites can make as few or as many typescripts as they deem useful. Cray recommends starting a typescript file before configuring eLogin 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 `eLogin`.

6. Change prompt to include a timestamp.

```
smw# PS1="\u@\h:\w \t# "
```

4.3.2 Change the esd Port

Prerequisites

SMW/CLE software is installed and configured.

About this task

The external state daemon (`esd`) on the SMW listens for client nodes on the port specified in the `/etc/opt/cray/esd/esd.ini` file. That port is designated by the variable `esd_port` in that file, and the default value is 8449.

If this site wishes to have `esd` listen on a port other than the default port, change the value of `esd_port`. Otherwise, skip this procedure.

Procedure

1. (Conditional) If the `esd` port will be changed, back up the iptables.

Changing the `esd` port changes the firewall configuration. Cray recommends saving the iptables prior to changing the firewall configuration on the SMW.

```
smw# iptables-save > iptables-before-firewall-changes
```

2. (Optional) Change the port on which `esd` listens.

Edit `/etc/opt/cray/esd/esd.ini` and change the value of `esd_port`.

```
smw# vi /etc/opt/cray/esd/esd.ini
```

```
#  
# Copyright 2017, Cray Inc. All Rights Reserved.  
#  
# esd.ini  
#  
# Initialization file for Cray External Node State Daemon (ESD).  
#  
...  
[esd]  
enode_port = 8448  
enode_endpoint = /esd/v1/node  
  
esd_port = 1234
```

3. (Conditional) If `esd` was started prior to changing `esd_port`, restart `esd` now.

```
smw# systemctl restart esd
```

4.3.3 Start the External State Daemon

Prerequisites

SMW/CLE software is installed and configured.

About this task

The external state daemon (`esd`) must be started before the `enode` command can be used. This is a one-time action that is necessary the first time SMW-managed eLogin is set up. After `esd` is enabled, it will be started automatically during subsequent boots of the SMW. For an SMW HA system, it is also necessary to add eLogin resources to the SMW HA cluster, but that is done at the end of the eLogin configuration and deployment process.

Procedure

1. Enable the `esd` daemon.

```
smw# systemctl enable esd
```

2. Start the `esd` daemon.

```
smw# systemctl start esd
```

4.3.4 Update the Config Set for eLogin

Prerequisites

Successful configuration of SMW `eth6` and `eth7` Ethernet interfaces and connection to the `external-ipmi-net` and `external-management-net`, which connect to the eLogin nodes.

About this task

During the process of booting an eLogin node from the image assigned to the node, configuration settings from the global config set and the CLE config set are applied to the node. This procedure prepares a work area and fresh configuration worksheets so that selected configuration services in the CLE config set can be updated on the SMW in the procedures that follow. This configuration is necessary prior to booting eLogin nodes.

These CLE configuration services are used to configure CLE nodes, and they are modified in the following procedures to configure eLogin nodes as well.

- `cray_firewall` (in both CLE and global config sets)
- `cray_image_binding` (if Cray PE software has been installed)
- `cray_lustre_client` (if using a Lustre file system)
- `cray_net`
- `cray_node_groups`
- `cray_ssh`
- `cray_storage`

These CLE configuration services are used for eLogin nodes only, and they are modified in the following procedures.

- `cray_cfgset_exclude`
- `cray_elogin_lnet` (if using a Lustre file system)
- `cray_elogin_motd`
- `cray_elogin_networking`
- `cray_eproxy`
- `cray_kdump`

Settings in these CLE configuration services apply to both CLE nodes and eLogin nodes. No modification is necessary to support eLogin nodes.

- `cray_auth`
- `cray_liveupdates`
- `cray_local_users`
- `cray_logging`
- `cray_simple_sync`
- `cray_sysenv`
- `cray_time`

Procedure

1. Back up the current CLE config set.

This example uses a CLE config set named `p0`. Substitute the correct name for the current CLE config set used for this system.

```
smw# cfgset create --clone p0 p0-pre-eloginconfig- $\{$ TODAY $\}$ 
```

2. Update and validate the current CLE config set.

This step ensures that the config set is valid prior to making eLogin changes.

```
smw# cfgset update -m prepare p0  
smw# cfgset validate p0
```

3. Copy the fresh configuration worksheets (generated in the previous step) to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea  
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets  
smw# cp *_worksheet.yaml /my/workarea
```

Change to the new work area.

```
smw# cd /my/workarea
```

4.3.4.1 Update `cray_net` for eLogin

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_net` worksheet.

```
smw# vi cray_net_worksheet.yaml
```

2. Define the external-management-net and site-user-net networks.

In the worksheet, add the following lines to define these two new networks.

If the network name contains any hyphens, substitute an underscore for each hyphen when entering the `common_name` key for the network. Use `external_management_net` as the key for the external-management-net network. For the site-user-net network, use `site` as the key to match what is expected by certain Ansible plays.

```
# NOTE: Place additional 'networks' setting entries here, if desired.

cray_net.settings.networks.data.name.external_management_net: null
cray_net.settings.networks.data.external_management_net.description:
  SMW/eLogin External Management Network
cray_net.settings.networks.data.external_management_net.ipv4_network: 10.7.0.0
cray_net.settings.networks.data.external_management_net.ipv4_netmask: 255.255.0.0
cray_net.settings.networks.data.external_management_net.ipv4_gateway: 10.7.1.1
cray_net.settings.networks.data.external_management_net.dns_servers: []
cray_net.settings.networks.data.external_management_net.dns_search: []
cray_net.settings.networks.data.external_management_net.ntp_servers: []

cray_net.settings.networks.data.name.site: null
cray_net.settings.networks.data.site.description: eLogin site-user-net network
cray_net.settings.networks.data.site.ipv4_network: IP_address
cray_net.settings.networks.data.site.ipv4_netmask: IP_address
cray_net.settings.networks.data.site.ipv4_gateway: IP_address
cray_net.settings.networks.data.site.dns_servers: []
cray_net.settings.networks.data.site.dns_search: []
cray_net.settings.networks.data.site.ntp_servers: []
```

3. Define the Ethernet and InfiniBand network interfaces and host names in use on each eLogin node.

In the worksheet, do the following for each eLogin node:

- Add the lines in the example to define a host with three network interfaces.
- Replace the key (`<example_elogin>`) with the host name of the eLogin node (if the host name contains a hyphen, replace it with an underscore, because hyphens are not allowed in the key for a multival setting).
- Set both the `hostid` and `hostname` fields to the host name of the eLogin node (hyphen allowed here).

IMPORTANT: Every host definition must have the `hostid` field configured, because its value is used in Ansible plays. For eLogin nodes, `hostid` is set to the host name of the node. For internal CLE nodes, it is set to the `cname` of the node. For an SMW, it is set to the output of the `hostid` command.

```
# NOTE: Place additional 'hosts' setting entries here, if desired.

cray_net.settings.hosts.data.common_name.<example_elogin>: null
cray_net.settings.hosts.data.<example_elogin>.description: eLogin Node
cray_net.settings.hosts.data.<example_elogin>.hostid: example-elogin
cray_net.settings.hosts.data.<example_elogin>.host_type: ''
cray_net.settings.hosts.data.<example_elogin>.hostname: example-elogin
cray_net.settings.hosts.data.<example_elogin>.aliases: ''

cray_net.settings.hosts.data.<example_elogin>.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.description:
    SMW/eLogin External Management network
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.network: external_management_net
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.ipv4_address: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.bootproto: dhcp
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth0.extra_attributes: []

cray_net.settings.hosts.data.<example_elogin>.interfaces.common_name.eth1: null
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.name: eth1
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.description: Site eth1
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.aliases: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.network: site
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.ipv4_address: IP_address
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.bootproto: static
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.mtu: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth1.extra_attributes: []

cray_net.settings.hosts.data.<example_elogin>.interfaces.common_name.ib0: null
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.description: IB to External Lustre
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.aliases: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.ipv4_address: eLogin_LNet_address
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.mtu: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.ib0.extra_attributes: []
```

4. Change the configured interfaces to match the hardware network adapter of each eLogin node, as needed.

- **4x1GbE LOM**

An eLogin node with a 4x1GbE LOM network adapter uses eth0 and eth1 to connect to the following networks. No further configuration is needed because the previous step already configured those interfaces.

- eth0: external-management-net interface
- eth1: site-user-net network interface

- **2x10GbE+2x1GbE LOM**

An eLogin node with a 2x10GbE+2x1GbE LOM network adapter uses eth2 and eth3 to connect to the following networks. Configure the eth2 and eth3 interfaces in this step.

- eth2: external-management-net interface
- eth3: site-user-net network interface

The lines added to the worksheet in the previous step assume a 4x1GbE LOM network adapter. For all eLogin nodes that have a 2x10GbE+2x1GbE LOM network adapter instead, replace eth0 with eth2 in all lines for the interface to external-management-net, and replace eth1 with eth3 in all lines for the interface to site-user-net (site).

eth2 entry:

```

cray_net.settings.hosts.data.<example_elogin>.interfaces.common_name.eth2: null
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.name: eth2
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.description:
  SMW/eLogin External Management network
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.aliases: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.network: external_management_net
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.ipv4_address: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.bootproto: dhcp
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.mtu: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth2.extra_attributes: []

```

eth3 entry:

```

cray_net.settings.hosts.data.<example_elogin>.interfaces.common_name.eth3: null
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.name: eth3
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.description: Site eth1
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.aliases: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.network: site
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.ipv4_address: IP_address
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.bootproto: static
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.mtu: ''
cray_net.settings.hosts.data.<example_elogin>.interfaces.eth3.extra_attributes: []

```

5. Add IPv6 information to a configured interface, if needed.

Sites that use the IPv6 interface in addition to IPv4 must configure an interface to include both an IPv4 and IPv6 address in the config set. However, the configuration worksheets do not accommodate more than one route. The IPv6 route must be configured using Simple Sync or site Ansible play.

In the worksheet, locate the eLogin node network interface that is on the `site` network (site-user-net). For the `extra_attributes` field of that interface, replace the default empty list with the two lines shown in the following example. Replace `IPV6_ADDR` with the actual IPv6 address to use for this node.

```

cray_net.settings.hosts.data.<example_elogin1>.interfaces.eth1.extra_attributes:
- IPADDR1='IPV6_ADDR'
- PREFIXLEN1='64'

```

The `IPADDR1` and `PREFIXLEN1` entries will be added to the `/etc/sysconfig/network/ifcfg-eth1` file on the eLogin node.

6. Upload the modified `cray_net` worksheet to the config set.

```

smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0

```

4.3.4.2 Add or Migrate Site Data to `/etc/hosts` File

Prerequisites

Host entries have been defined in the `cray_net` configuration service.

About this task

- **MIGRATION:** All of the steps in this procedure are important for sites migrating eLogin nodes to SMW-managed eLogin.
- **INITIAL DEPLOYMENT:** Step 3 provides an opportunity to manually add site host data to a protected area in the `/etc/hosts` file.

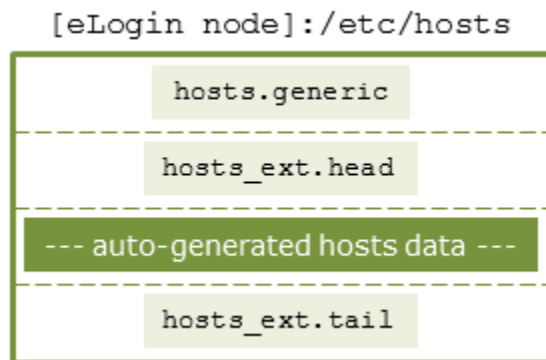
Cray system management software uses Hardware Supervisory System (HSS) and config set data to automatically generate the `hosts` file for internal CLE nodes and the `hosts.external` file for eLogin nodes.

The `hosts` file is installed onto CLE nodes as `/etc/hosts`, and the `hosts.external` file is installed onto eLogin nodes as `/etc/hosts`.

Sites often need to edit the `hosts` and `hosts.external` files to add their own data, but because these files are generated each time the CLE config set is updated, those additions could be overwritten. To address this issue, those files are now composed of four components: the auto-generated hosts data and three additional files that can be modified by sites. The four components of `hosts.external` on the SMW (`/etc/hosts` on an eLogin node) are:

- `hosts.generic`
Contains generic entries from Cray, which can be modified by sites. The contents are placed at the top of the `hosts` file.
- `hosts_ext.head`
Contains site entries that will be placed above the auto-generated hosts data in the `hosts` file on eLogin nodes.
- Auto-generated hosts data
Generated from HSS and configuration data. Note that site entries in this component of the `hosts` file (comments and other changes) are no longer preserved. To add extra labels to a host entry line, add them to the configured aliases list of that host definition in the `cray_net` configuration service.
- `hosts_ext.tail`
Contains site entries that will be placed below the auto-generated hosts data in the `hosts` file on eLogin nodes.

Figure 14. Components of the Hosts File for eLogin Nodes



These component files are located in the config set on the SMW in the following directory:

```
/var/opt/cray/imps/config/sets/<CONFIG_SET>/files/roles/common/etc/
```

If the head and/or tail files are not present when the CLE config set is updated, the missing file(s) will be automatically generated and will be empty. Also, when the config set is updated, if there is no backup hosts file, the old hosts files for that config set will be automatically saved as `hosts.autobk`.

Procedure

1. Look for eLogin-related `/etc/hosts` files in the Simple Sync directory structure.

If Simple Sync was used previously to ensure that manually entered site data was present in `/etc/hosts` on eLogin nodes, then those files may have site data that should be migrated to `hosts_ext.head` and/or `hosts_ext.tail`.

This example lists the files under the `eloin_nodes` node group directory. It may be necessary to look in other Simple Sync directories as well.

```
smw# cd /var/opt/cray/imps/config/sets/p0
smw# ls files/simple_sync/nodegroups/eloin_nodes/files/etc
```

2. Migrate site data to `hosts_ext.head` and `hosts_ext.tail`.

Site host entries and other data entered directly into `hosts.external` or added to `/etc/hosts` files in the Simple Sync directory prior to CLE 6.0.UP06 may need to be migrated to `hosts_ext.head` or `hosts_ext.tail`.

Create the needed head or tail files and enter data into them. For more information and examples, see "About the `/etc/hosts` File" in *XC™ Series Software Installation and Configuration Guide (S-2559)*. If the head and/or tail files are not present when the CLE config set is updated, the missing file(s) will be automatically generated and will be empty.

3. Add new site-specific data and comments to the head and/or tail files, as needed.
4. Remove eLogin-related `/etc/hosts` files from the Simple Sync directory structure.

If there are eLogin-related `/etc/hosts` files in the Simple Sync directory, remove those files now so that they do not interfere with or override the `etc/hosts` file that will be placed on the eLogin nodes at boot time.

4.3.4.3 Update `cray_node_groups` for eLogin

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_node_groups` worksheet.

```
smw# vi cray_node_groups_worksheet.yaml
```

2. Ensure that `cray_node_groups` is enabled.

Uncomment the following line, if it is not already uncommented, and ensure that the value is set to `true`.

```
cray_node_groups.enabled: true
```

3. Add all eLogin nodes to the `eloin_nodes` node group.

Find the following lines and add each eLogin node on a separate line prefixed by a hyphen and space (`-`).

```
# ** 'groups' DATA **
...
```

```
cray_node_groups.settings.groups.data.group_name.elogin_nodes: null
cray_node_groups.settings.groups.data.elogin_nodes.description: Default node
  group which contains the EXTERNAL login nodes for the configured system.
cray_node_groups.settings.groups.data.elogin_nodes.members:
- elogin1
- elogin2
- elogin3
```

Additional node groups can be created to identify subsets of eLogin nodes for this system, such as `elogin_test_nodes` or `elogin_production_nodes`, but Cray recommends adding to the `elogin_nodes` node group ALL of the eLogin nodes related to this CLE config set.

4. Upload the modified `cray_node_groups` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_node_groups_worksheet.yaml' p0
```

4.3.4.4 Update `cray_storage` for eLogin

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_storage` worksheet.

```
smw# vi cray_storage_worksheet.yaml
```

2. Ensure that `cray_storage` is enabled.

Uncomment the following line, if it is not already uncommented, and ensure that the value is set to `true`.

```
cray_storage.enabled: true
```

3. Prepare the storage profile that will be assigned to the eLogin nodes.

The `elogin_default` profile has the Cray-recommended values for the partition file system type and size. To change these, do not modify the default profile. Instead, create a custom storage profile by copying the default profile and then modifying values as needed for the eLogin nodes in this system.

To prepare the storage profile that will be assigned to the eLogin nodes, choose one of the following strategies:

- **Make Temporary Modification to Default Profile for First Boot.** The default profile will work for most systems, but its `persist_on_boot` flag needs to be changed from `true` to `false` for the first PXE boot of an eLogin node (for both initial deployment and migration). If this strategy is used, the flag must be reset to `true` after the first boot so that data on the persistent disk is not lost when the node is booted again (there is a procedure for this later in the process).

To modify the default profile, proceed to step 4 on page 59.

- **Create Default 'First Boot' Profile.** To minimize risk of data loss, Cray recommends this alternative to changing the `persist_on_boot` flag on the default profile. Make a copy of the default profile, name it `eloin_default_first_boot`, and change the `persist_on_boot` flag to `false`. Use the `eloin_default_first_boot` profile only for eLogin nodes that are being provisioned for the first time (or reprovisioned), and use the `eloin_default` profile for already provisioned eLogin nodes. In a case where a site adds eLogin nodes later, and boots the new nodes along with the existing nodes, using different profiles for new and existing will avoid inadvertently wiping disks on existing nodes.

To create a default first-boot profile, proceed to step 5 on page 59.

- **Create Custom Profile.** Custom storage profiles may be needed for a variety of purposes, such as testing, production, and accommodation of smaller disk sizes.

To create a custom storage profile, proceed to step 6 on page 60.

4. Make a temporary modification to the default storage profile for first boot.

Risk of data loss. Be aware that changing the `persist_on_boot` flag in the `eloin_default` storage profile will affect all nodes with `storage_profile` set to `eloin_default` in the node registry. If not all of them should be reprovisioned at the next boot, use the safer alternative in step 5 on page 59.

MIGRATION: If using this storage profile, do NOT skip the substep that changes the `persist_on_boot` flag to `false` on the persistent device. This is mandatory on first boot so that the node can be reprovisioned.



WARNING: To avoid loss of data when reprovisioning a persistent disk, move data to a safe location before rebooting the eLogin node. After rebooting the node and restoring that data to the disk, ensure that the disk is reconfigured as persistent.

a. Uncomment all of the lines between the following two comments.

```
# ** 'storage_profiles' DATA **
...
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

b. Change the device `persist_on_boot` flag to `false` to allow the storage layout to be created on the disk as configured in the storage profile for this first boot.

```
cray_storage.settings.storage_profiles.data.eloin_default.layouts./dev/sdb.persist_on_boot: false
```

This flag will be reset to `true` in a later procedure, after the first successful PXE boot of the eLogin node, so that the node can provide the necessary persistent storage.

Proceed to step 8 on page 62.

5. Create a default "first boot" storage profile.

This profile can be used for the first provisioning or the reprovisioning of an eLogin node.

MIGRATION: When creating this storage profile, do NOT skip the substep that changes the `persist_on_boot` flag to `false` on the persistent device. This is mandatory on first boot so that the node can be reprovisioned.



WARNING: To avoid loss of data when reprovisioning a persistent disk, move data to a safe location before rebooting the eLogin node. After rebooting the node and restoring that data to the disk, ensure that the disk is reconfigured as persistent.

a. Copy the default storage profile (`eloin_default`).

In the worksheet, copy the default storage profile and paste it below

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

- b. Uncomment all lines of the copied profile and replace `eloin_default` in all lines with the name (key) for the new storage profile, like `eloin_default_first_boot`. A shorter placeholder is used in this example so that each setting fits on a single line.

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.

cray_storage.settings.storage_profiles.data.name.<1st_boot>: null
cray_storage.settings.storage_profiles.data.<1st_boot>.enabled: true

cray_storage.settings.storage_profiles.data.<1st_boot>.layouts.device./dev/sda: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partition_type: gpt
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.persist_on_boot: false

cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.label.GRUB: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.GRUB.type: ext3
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.GRUB.size: 1MiB
...
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.label.BOOT: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.BOOT.type: ext3
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.BOOT.size: 2GiB
...
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.label.WRITELAYER: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.WRITELAYER.type: ext4
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.WRITELAYER.size: 20GiB
...
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.label.TMP: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.TMP.type: xfs
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.TMP.size: 256GiB
...
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.label.SWAP: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.SWAP.type: swap
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sda.partitions.SWAP.size: 128GiB
...

cray_storage.settings.storage_profiles.data.<1st_boot>.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.persist_on_boot: true

cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.label.CRASH: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.CRASH.type: ext4
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.CRASH.size: 10GiB
...
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.label.PERSISTENT: null
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.PERSISTENT.type: xfs
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.partitions.PERSISTENT.size: ALL
...
```

- c. Change the `persist_on_boot` flag for `/dev/sdb` to `false` to allow the storage layout to be created on that disk as configured in the storage profile for this first boot.

```
cray_storage.settings.storage_profiles.data.<1st_boot>.layouts./dev/sdb.persist_on_boot: false
```

The node registry `storage_profile` field will be set to `eloin_default_first_boot` for this node in the next procedure. Then after a successful PXE boot of the node, a later procedure will set `storage_profile` to `eloin_default` to ensure that the node is configured for persistent storage.

Proceed to step 8 on page 62.

6. Create a custom profile, if needed.

Requirements for creating a storage profile:

- To function properly, all eLogin nodes must have all of the following partitions with these exact labels:
 - nonpersistent disk: GRUB, BOOT, WRITELAYER, TMP, and SWAP
 - persistent disk: CRASH and PERSISTENT
- The sum of the sizes of all of the volatile data partitions on the first disk (`/dev/sda`) must be less than the available storage on the first disk. Similarly, the sum of the sizes of all of the persistent data partitions on the second disk (`/dev/sdb`) must be less than the available storage on the second disk.
- Two partitions have the following minimum size limits:

- o BOOT must be > 1 GiB (note binary value)
- o PERSISTENT must be > 200 GiB (note binary value)

For more information about binary values, see [Prefixes for Binary and Decimal Multiples](#) on page 206.

Because of these requirements, Cray recommends starting with a copy of the default profile, which has all required partitions and labels.

This step illustrates how to create a custom profile to use for eLogin nodes with devices and partitions that are smaller (to accommodate smaller disk sizes) than those specified in the default storage profile.

- a. Copy the default storage profile (`elogin_default`).

In the worksheet, copy the default storage profile and paste it below the following line:

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

- b. Uncomment all lines of the copied profile and replace `elogin_default` in all lines with the name (key) for the custom storage profile (`elogin_small` in this example).

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

```
cray_storage.settings.storage_profiles.data.<elogin_small>: null
cray_storage.settings.storage_profiles.data.<elogin_small>.enabled: true

cray_storage.settings.storage_profiles.data.<elogin_small>.layouts.device./dev/sda: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partition_type: gpt
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.persist_on_boot: false

cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.label.GRUB: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.GRUB.type: ext3
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.GRUB.size: 1MiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.label.BOOT: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.BOOT.type: ext3
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.BOOT.size: 2GiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.label.WRITELAYER: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.WRITELAYER.type: ext4
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.WRITELAYER.size: 20GiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.label.TMP: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.TMP.type: xfs
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.TMP.size: 256GiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.label.SWAP: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.SWAP.type: swap
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.SWAP.size: 128GiB
...

cray_storage.settings.storage_profiles.data.<elogin_small>.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.persist_on_boot: true

cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.label.CRASH: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.CRASH.type: ext4
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.CRASH.size: 10GiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.label.PERSISTENT: null
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.PERSISTENT.type: xfs
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sdb.partitions.PERSISTENT.size: ALL
...
```

- c. Change values for the partition file system type or size, as needed.

Note that the `mount_point` and `mount_options` fields for a partition are not configurable currently. Do not change the values copied from the `elogin_default` profile.

Make the TMP and SWAP partitions smaller to accommodate a smaller disk.

```
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.TMP.size: 100GiB
...
cray_storage.settings.storage_profiles.data.<elogin_small>.layouts./dev/sda.partitions.SWAP.size: 64GiB
```

7. (Conditional) If a custom storage profile was created, create a matching profile for first boot.

Note that the instructions for creating a custom profile did not include changing the `persist_on_boot` flag to `false`. That profile should be used only for already provisioned nodes. This step creates a companion custom profile (called `ellogin_small_first_boot`) to use for the first boot of an eLogin node.

MIGRATION: When creating this storage profile, do NOT skip the substep that changes the `persist_on_boot` flag to `false` on the persistent device. This is mandatory on first boot so that the node can be reprovisioned.



WARNING: To avoid loss of data when reprovisioning a persistent disk, move data to a safe location before rebooting the eLogin node. After rebooting the node and restoring that data to the disk, ensure that the disk is reconfigured as persistent.

- a. Copy the custom storage profile created in step 6 on page 60.
- b. Change the name (key) for this profile.

Substitute `ellogin_small_first_boot` for `ellogin_small` everywhere in the profile.

- c. Change the device `persist_on_boot` flag to `false` for `/dev/sdb` to allow the storage layout to be created on that disk as configured in the storage profile for this first boot.

```
cray_storage.settings.storage_profiles.data.ellogin_small_first_boot.layouts./dev/sdb.persist_on_boot: false
```

The node registry `storage_profile` field can be set to `ellogin_small_first_boot` for this node in the next procedure. Then after a successful PXE boot of the node, a later procedure will provide an opportunity to set `storage_profile` to `ellogin_small` to ensure that the node has persistent storage.

8. Upload the modified `cray_storage` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_storage_worksheet.yaml' p0
```

4.3.4.5 Update `cray_ssh` for eLogin

About this task

When eLogin nodes are added to the node registry later in the deployment process, the `ssh_host_keys` field of the node registry will be set to one of the following values for each eLogin node:

- `simple_sync`: Sites choosing this value will either accept the automatically generated SSH host keys placed in the Simple Sync directory or replace them there with their own.

Sites choosing `simple_sync` must set `simple_ssh_keys` to `true` in `cray_ssh`.

- `generate`: Sites choosing this value will have eLogin-specific SSH host keys generated by `esd`.

Sites choosing `generate` must set `simple_ssh_keys` to `false` in `cray_ssh`, so that automatic generation of SSH host keys by `cray_ssh` will be disabled.

- `/some/directory/path`: Sites that specify an absolute directory path will supply their own SSH host keys in that location.

Sites specifying a directory path must set `simple_ssh_keys` to `false` in `cray_ssh`.

This procedure updates `cray_ssh` depending on how this site wishes to handle SSH host keys for eLogin nodes.

Procedure

1. Determine the value of the `simple_ssh_keys` setting in the CLE config set assigned to eLogin nodes (p0 in example).

```
smw# cfgset get cray_ssh.settings.sshd.data.simple_ssh_keys p0
```

2. (Conditional) Set `simple_ssh_keys` to `true`.

If the current value is `false`, and `ssh_host_keys` will be set to `simple_sync`, then change the value to `true`.

This example modifies CLE config set p0.

```
smw# cfgset modify --set true cray_ssh.settings.sshd.data.simple_ssh_keys p0
smw# cfgset get cray_ssh.settings.sshd.data.simple_ssh_keys p0
```

3. (Conditional) Set `simple_ssh_keys` to `false`.

If the current value is `true`, and `ssh_host_keys` will be set to `generate` or `directory_path`, then change the value to `false`.

This example modifies CLE config set p0.

```
smw# cfgset modify --set false cray_ssh.settings.sshd.data.simple_ssh_keys p0
smw# cfgset get cray_ssh.settings.sshd.data.simple_ssh_keys p0
```

If either of the `cfgset modify` commands were performed, a setting in the eLogin nodes' CLE config set was modified without running pre- and post-configuration scripts. This is not a problem, because the CLE config set will be updated in a later procedure, ensuring that all configuration scripts are run.

4.3.4.6 Update `cray_cfgset_exclude` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_cfgset_exclude` worksheet.

```
smw# vi cray_cfgset_exclude_worksheet.yaml
```

2. Enable `cray_cfgset_exclude`.

Uncomment the following line and ensure that the value is set to `true`.

```
cray_cfgset_exclude.enabled: true
```

3. Configure the Cray-provided `eloin_security` profile for excluding config set content from transfer to an eLogin node group.

Uncomment the following lines and modify which file paths should be excluded and which node groups are affected, as needed.



CAUTION: Use care when excluding YAML files in the config directory. If an excluded file contains data required by the eLogin Ansible playbooks and roles, those playbooks will either fail or use stale data.

```
# ** 'profiles' DATA **

#cray_cfgset_exclude.settings.profiles.data.profile_name.elogin_security: null
#cray_cfgset_exclude.settings.profiles.data.elogin_security.exclude_content:
#- config/cray_sdb_config.yaml
#- config/cray_drc_config.yaml
#- config/cray_lmt_config.yaml
#- files/roles/common/etc/ssh
#- files/roles/common/root
#- files/roles/munge
#- files/roles/common/etc/opt/cray/xtremoted-agent
#- files/roles/merge_account_files
#- files/simple_sync/common/files/etc/ssh
#- files/simple_sync/common/files/root/.ssh
#- worksheets
#cray_cfgset_exclude.settings.profiles.data.elogin_security.groups:
#- elogin_nodes
```

4. Create custom exclusion profiles, as needed.

Custom profiles may be needed to target different groups of eLogin nodes or for purposes other than security, such as performance or testing. To create a custom profile, copy the `elogin_security` profile and place the copy below the following line.

```
# NOTE: Place additional 'profiles' setting entries here, if desired.
```

Modify which file paths should be excluded and which node groups are affected, as needed.



CAUTION: Use care when excluding YAML files in the config directory. If an excluded file contains data required by the eLogin Ansible playbooks and roles, those playbooks will either fail or use stale data.

5. Upload the modified `cray_cfgset_exclude` worksheet to the config set.

```
smw# cray_cfgset update -w '/my/workarea/cray_cfgset_exclude_worksheet.yaml' p0
```

4.3.4.7 Update `cray_lustre_client` for eLogin

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_lustre_client` worksheet.


```
smw# vi cray_lustre_client_worksheet.yaml
```

2. Ensure that `cray_lustre_client` is enabled.

Uncomment the following line, if it is not already uncommented, and ensure that the value is set to `true`.

```
cray_lustre_client.enabled: true
```

3. Configure the eLogin nodes as Lustre clients, if Lustre is the parallel file system in use on the XC compute nodes.

Add the following lines, if not already there, to define a client mount for each Lustre file system that has one or more eLogin nodes as clients.

This example mounts an external Lustre file system (a Sonexion with a file system called `snx11023`) on `/lus/snx11023` by several node groups, including the `eloin_nodes` node group. 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_x86_64
- login_nodes_aarch64
- compute_nodes_x86_64
- compute_nodes_aarch64
- eloin_nodes

#***** END Service Setting: client_mounts *****
```

4. Upload the modified `cray_lustre_client` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_lustre_client_worksheet.yaml' p0
```

4.3.4.8 Update `cray_eloin_lnet` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

About this task

This procedure is needed only if the Lustre parallel file system will be mounted on the eLogin nodes in this system.

Procedure

1. Edit the `cray_eloin_lnet` worksheet.

```
smw# vi cray_eloin_lnet_worksheet.yaml
```

2. Enable `cray_elogin_lnet`.

Uncomment the following line and set the value to `true`.

```
cray_elogin_lnet.enabled: true
```

3. Add a local LNet setting.

The LNet settings control how eLogin nodes connect to the Lustre server.

Add the following lines.

```
# NOTE: Place additional 'local_lnets' setting entries here, if desired.
cray_elogin_lnet.settings.local_lnets.data.lnet_name.o2ib: null
cray_elogin_lnet.settings.local_lnets.data.o2ib.ip_wildcard: 10.149.*.*
```

4. Upload the modified `cray_elogin_lnet` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_elogin_lnet_worksheet.yaml' p0
```

4.3.4.9 Update `cray_image_binding` for eLogin

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_image_binding` worksheet.

```
smw# vi cray_image_binding_worksheet.yaml
```

2. Ensure that `cray_image_binding` is enabled.

Uncomment the following line, if it is not already uncommented, and ensure that the value is set to `true`.

```
cray_image_binding.enabled: true
```

3. Add eLogin nodes to the PE profile.

To make Cray Programming Environment (PE) software available on eLogin nodes, add the `elogin_nodes` node group to the `client_groups` setting of the PE profile. This will ensure that the PE profile is applied to eLogin nodes.

Find the following lines and add the `elogin_nodes` node group on a separate line prefixed by a hyphen and space (-).

```
# ** 'profiles' DATA **
cray_image_binding.settings.profiles.data.PE.client_groups:
- login_nodes
- compute_nodes
- elogin_nodes
```

4. Ensure that the PE image binding profile is enabled.

Ensure that the following line is uncommented and the value is set to `true`.

```
cray_image_binding.settings.profiles.data.PE.enabled: true
```

5. Upload the modified `cray_image_binding` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_image_binding_worksheet.yaml' p0
```

4.3.4.10 Update `cray_elogin_motd` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_elogin_motd` worksheet.

```
smw# vi cray_elogin_motd_worksheet.yaml
```

2. Enable `cray_elogin_motd`.

Uncomment the following line and ensure that the value is set to `true`.

```
cray_elogin_motd.enabled: true
```

3. Enable or disable automatic creation of `/etc/motd` on each eLogin node.

The `cray_elogin_motd` config service provides the ability to enable or disable the automatic creation of the `/etc/motd` file on each eLogin node. The default is to generate the `/etc/motd` file.

For each eLogin node, add the following lines. The first set of two lines enables automatic generation of `/etc/motd` for a node with name `elogin1`, and the second set disables it for a node with name `elogin2`.

```
# NOTE: Place additional 'elogin_motd' setting entries here, if desired.
cray_elogin_motd.settings.elogin_motd.data.hostname.elogin1: null
cray_elogin_motd.settings.elogin_motd.data.elogin1.cray_managed: true
```

```
cray_elogin_motd.settings.elogin_motd.data.hostname.elogin2: null
cray_elogin_motd.settings.elogin_motd.data.elogin2.cray_managed: false
```

4. Upload the modified `cray_elogin_motd` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_elogin_motd_worksheet.yaml' p0
```

4.3.4.11 Update `cray_elogin_networking` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_elogin_networking` worksheet.

```
smw# vi cray_elogin_networking_worksheet.yaml
```

2. Enable `cray_elogin_networking`.

Uncomment the following line and ensure that the value is set to `true`.

```
cray_elogin_networking.enabled: true
```

3. Configure eLogin networking by specifying the postfix relay used by each eLogin node.

Add the following lines.

```
# NOTE: Place additional 'elogin_networking' setting entries here, if desired.
cray_elogin_networking.settings.elogin_networking.data.hostname.<eLogin-hostname>: null
cray_elogin_networking.settings.elogin_networking.data.<eLogin-hostname>.postfix_relay_host: smw
```

Note that the default value for `postfix_relay_host`, `smw`, refers to the SMW that is managing the eLogin nodes.

4. Upload the modified `cray_elogin_networking` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_elogin_networking_worksheet.yaml' p0
```

4.3.4.12 Update `cray_eproxy` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_eproxy` worksheet.

```
smw# vi cray_eproxy_worksheet.yaml
```

2. Enable `cray_eproxy`.

Uncomment the following line and set the value to `true`.

```
cray_eproxy.enabled: true
```

3. Define a mapping for each eLogin node.

The `cray_eproxy` configuration template maps each eLogin node to an internal login node for running `eproxy` commands. It also defines which commands are wrapped.

IMPORTANT: A mapping must exist for each eLogin node configured in the `cray_net` configuration service; otherwise, the eLogin node will not boot.

Add the following lines.

The example shows a list of three eLogin nodes that will use `login_node_hostname` as their `eproxy` host. Substitute a list of the actual eLogin node(s) that will use that host for this system.

```
# NOTE: Place additional 'eproxy_map' setting entries here, if desired.
cray_eproxy.settings.eproxy_map.data.eproxy_host.<login_node_hostname>: null
cray_eproxy.settings.eproxy_map.data.<login_node_hostname>.e_login_hosts:
- e_login1
- e_login2
- e_login3
```

4. If this system does not have a workload manager, enable `aprun` wrapping.

Uncomment the following line and ensure that the value is set to `true`.

```
cray_eproxy.settings.wrapped.data.aprun: true
```

5. Upload the modified `cray_eproxy` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_eproxy_worksheet.yaml' p0
```

4.3.4.13 Update `cray_kdump` for eLogin Deployment

Prerequisites

A work area has been set up for editing CLE configuration worksheets, and the current directory has been set to that work area.

```
smw# cd /my/workarea
```

Procedure

1. Edit the `cray_kdump` worksheet.

```
smw# vi cray_kdump_worksheet.yaml
```

2. Enable `cray_kdump`.

Uncomment the following line and ensure that the value is set to `true`.

```
cray_kdump.enabled: true
```

3. Find the following `level=basic` settings and uncomment them.

Read the worksheet guidance and change values, if needed.

```
#cray_kdump.settings.kdump_opts.data.kdump_commandline: BOOT_IMAGE=vmlinuz rootfstype=squashfs
# nofb nomodeset vga=normal console=tty0 console=ttyS1,115200n8 pci=bfsort elevator=deadline
```

```
# sysrq=yes reset_devices acpi_no_memhotplug cgroup_disable=memory irqpoll nr_cpus=1
# root=kdump rootflags=bind rd.udev.children-max=8 disable_cpu_apicid=0 rd.neednet=0

#cray_kdump.settings.kdump_opts.data.kdump_commandline_append: ''

#cray_kdump.settings.kdump_opts.data.kdump_verbosity: 31

#cray_kdump.settings.kdump_opts.data.kdump_dumplevel: 27

#cray_kdump.settings.kdump_opts.data.kdump_netconfig: auto
```

4. Upload the modified `cray_kdump` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_kdump_worksheet.yaml' p0
```

4.3.4.14 Ensure that NFS Port is Open in Firewall

Prerequisites

SMW/CLE software is installed and configured.

About this task

If the `SuSEfirewall12` service starts before the `rpcbind` service on the SMW, ports in the firewall that depend on `rpcbind` to help manage them will remain closed. The NFS port, port 2049, is one of the ports that depend on `rpcbind`. If the NFS port is not open, eLogin nodes will be unable to boot because they will be unable to NFS-mount the necessary file systems.

This procedure determines whether all firewall ports that should be open are open, and if they are not, restarts the firewall.

Procedure

1. Search for the NFS port (port 2049).

The status of the NFS port is a good indicator of whether `rpcbind` was present in time to open up dependent ports correctly.

```
smw# iptables -L -n | grep 2049
```

If the search results look like the following, then the NFS port has been opened correctly. No further action is needed. Skip the rest of the procedure.

```
LOG udp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs */ limit: avg 3/min burst 5 ctstate NEW udp dpt:2049 LOG
flags 6 level 4 prefix "SFW2-INMGMT-ACC-RPC "
ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs */ udp dpt:2049
LOG tcp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs */ limit: avg 3/min burst 5 ctstate NEW tcp dpt:2049 LOG
flags 6 level 4 prefix "SFW2-INMGMT-ACC-RPC "
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs */ tcp dpt:2049
LOG udp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs_acl */ limit: avg 3/min burst 5 ctstate NEW udp dpt:2049
LOG flags 6 level 4 prefix "SFW2-INMGMT-ACC-RPC "
ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs_acl */ udp dpt:2049
LOG tcp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs_acl */ limit: avg 3/min burst 5 ctstate NEW tcp dpt:2049
LOG flags 6 level 4 prefix "SFW2-INMGMT-ACC-RPC "
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 /* sfw2.rpc.nfs_acl */ tcp dpt:2049
```

If this search returns no results, then the NFS port has not been opened correctly. Continue to the next step.

2. Restart the firewall.

```
smw# systemctl restart SuSEfirewall12
```

3. Search for the NFS port (port 2049) again to confirm that it has been opened correctly.

```
smw# iptables -L -n | grep 2049
```

The search results should look like the successful search in the first step.

4.3.4.15 Update `cray_firewall` for eLogin

About this task

The Cray firewall configuration services and Ansible plays are designed to make it unnecessary for site system administrators to change the SMW and eLogin firewall configuration. However, a site may wish to make one or both of these changes:

- enable or disable the firewall
- change whether the firewall service in the CLE config set inherits settings from the firewall service in the global config set (when a CLE config service inherits from the global config service, configuration values in the CLE service are ignored and values from the global service are used)

Perform this procedure only if either of these changes are needed for this system.

Procedure

1. (Conditional) If any firewall settings will be changed, back up the iptables.

Cray recommends saving the iptables prior to changing the firewall configuration on the SMW.

```
smw# iptables-save > iptables-before-firewall-changes
```

2. (Conditional) If desired, enable/disable the firewall for the SMW.

- a. Change the `cray_firewall.enabled` setting in the global config set.

To enable the firewall in the global config set and verify it is enabled now:

```
smw# cfgset modify --set true cray_firewall.enabled global
smw# cfgset get cray_firewall.enabled global
true
```

To disable the firewall in the global config set and verify it is disabled now:

```
smw# cfgset modify --set false cray_firewall.enabled global
smw# cfgset get cray_firewall.enabled global
false
```

- b. Update the global config set.

The previous substep modified the global config set without running pre- and post-configuration scripts. This substep ensures that all configuration scripts are run.

```
smw# cfgset update -m prepare global
```

- c. Apply configuration changes on the SMW.

To apply the firewall configuration change immediately, use the following command. Otherwise, the change will be applied the next time the SMW is rebooted.

```
smw# /etc/init.d/cray-ansible start
```

TIP: The next time the SMW is rebooted, use the procedure in [Ensure that NFS Port is Open in Firewall](#) on page 70 to check that ports are opened correctly.

3. (Conditional) If desired, enable/disable/inherit the firewall for all CLE and eLogin nodes.

Note that changes to the firewall configuration service in the CLE config set affect all internal CLE nodes and all eLogin nodes.

- a. Change the `cray_firewall.enabled` setting in the CLE config set, if needed.

To enable the firewall in the CLE config set (p0 in the example) and verify it is enabled now:

```
smw# cfgset modify --set true cray_firewall.enabled p0
smw# cfgset get cray_firewall.enabled p0
true
```

To disable the firewall in the CLE config set (p0 in the example) and verify it is disabled now:

```
smw# cfgset modify --set false cray_firewall.enabled p0
smw# cfgset get cray_firewall.enabled p0
false
```

- b. Change the `cray_firewall.inherit` setting in the CLE config set, if needed.

To set the firewall in the CLE config set (p0 in the example) to inherit from the firewall in the global config set, and verify it inherits now:

```
smw# cfgset modify --set true cray_firewall.inherit p0
smw# cfgset get cray_firewall.inherit p0
true
```

To set the firewall in the CLE config set (p0 in the example) to not inherit from the firewall in the global config set, and verify it does not inherit now:

```
smw# cfgset modify --set false cray_firewall.inherit p0
smw# cfgset get cray_firewall.inherit p0
false
```

The previous substeps modified the CLE config set without running pre- and post-configuration scripts. The CLE config set will be updated in the next step, ensuring that all configuration scripts are run.

4.3.4.16 Update and Validate the CLE Config Set

Prerequisites

CLE config set changes are complete.

About this task

Use this procedure to update and validate all CLE config sets that have been modified.

Procedure

1. Update the CLE config set.


```
smw# cfgset update p0
```

Updating the config set is recommended as good practice when any config service has been changed by importing worksheets. It is necessary when any config service has been changed using `cfgset modify`, because the `cfgset` CLI does not run any pre- or post-configuration scripts.

2. Validate the CLE config set.

```
smw# cfgset validate p0
```

The basic configuration of eLogin in the CLE config set is complete.

4.3.5 Change the eLogin BIOS and iDRAC Settings

Prerequisites

- Access to the console of each eLogin node being configured
- Up-to-date firmware (if needed, see [Update eLogin Firmware](#) on page 140)

About this task

This procedure changes the system setup of a Dell R820 server for use as an eLogin node. Depending on the server model and version of BIOS configuration utility, there may be minor differences in the steps to configure the system. For more information, refer to the Dell documentation for this server.

- **INITIAL DEPLOYMENT:** Because Cray ships systems with most of the installation and configuration completed, some of these steps may have been done already.
- **MIGRATION:** If migrating from eLogin nodes managed by CMC or CIMS, all of the steps are REQUIRED. In particular, change the iDRAC IP address of each eLogin node so that the SMW will be able to communicate with it.

Procedure

1. Power up the node. When the BIOS power-on self-test (POST) process begins, quickly press the **F2** key after the function-key menu appears in the upper-right of the screen.

Figure 15. Dell R820 BIOS Power-On Self-Test Menu Screen



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

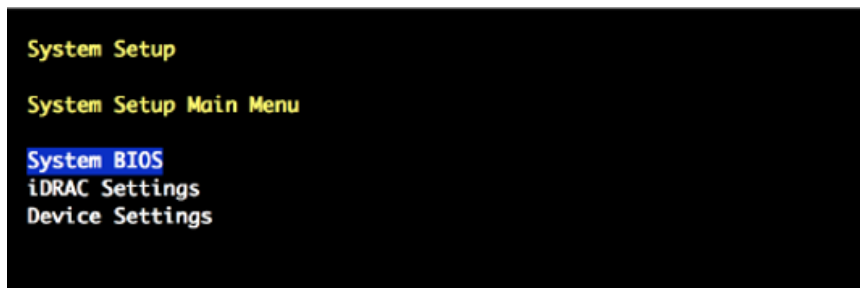
After the POST process completes and all disk and network controllers are initialized, the **Dell System Setup Main Menu** screen appears with the following sub-menus:

- System BIOS
- iDRAC Settings
- Device Settings

————— CHANGE SYSTEM BIOS SETTINGS —————

2. Select **System BIOS** from the **System Setup Main Menu** screen , then press **Enter**.

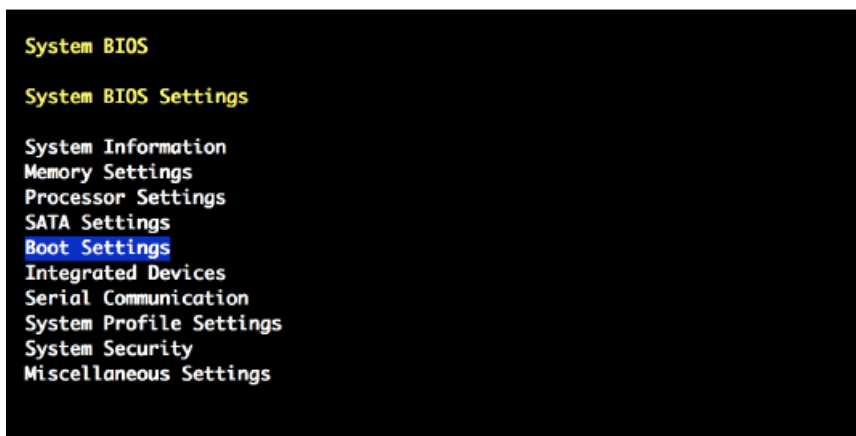
Figure 16. System Setup Main Menu: Select System BIOS



The **System BIOS Settings** menu screen opens.

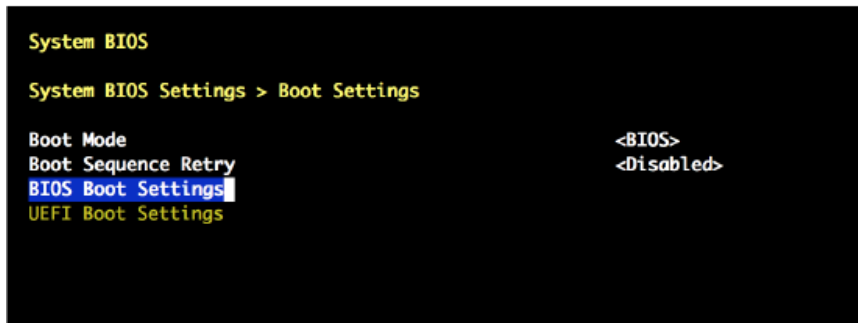
3. Change the BIOS boot settings.
 - a. Select **Boot Settings** from the **System BIOS Settings** screen, then press **Enter**.

Figure 17. System BIOS Settings: Boot Settings



- b. Select **BIOS Boot Settings** from the **Boot Settings** screen, then press **Enter**.

Figure 18. Boot Settings: BIOS Boot Settings



- c. Select **Boot Sequence**, then press **Enter** to view the boot settings.
- d. Change the boot sequence.

Change the boot sequence so that **Integrated NIC** appears last. The boot sequence should be ordered as follows:

- optical (DVD) drive
- hard drive
- Integrated NIC

Figure 19. Bios Boot Settings: Set Boot Sequence

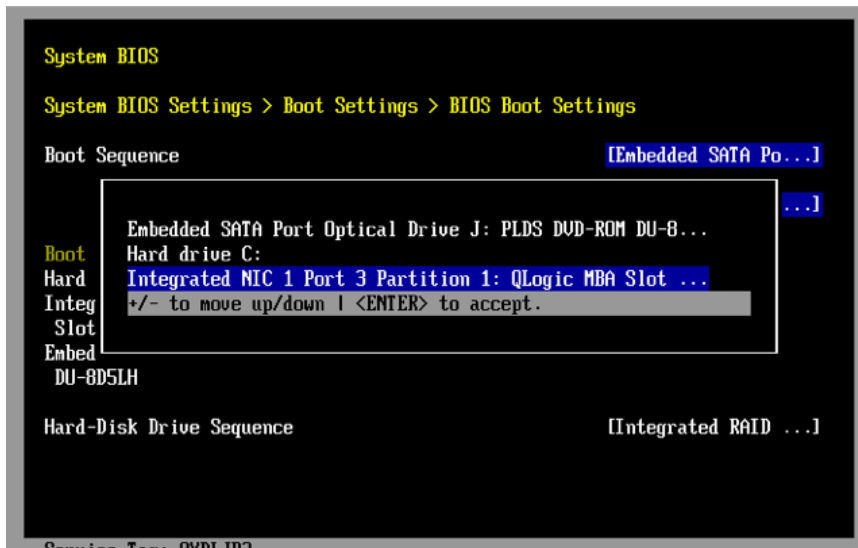
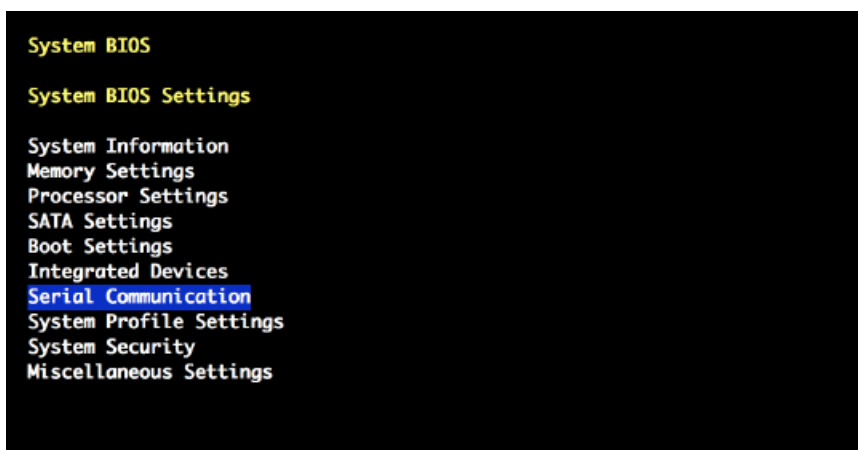


Figure 20. BIOS Boot Settings: Boot Sequence



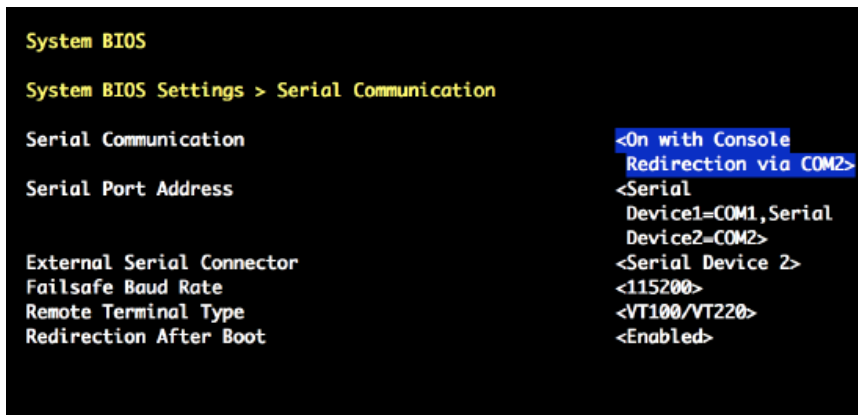
- e. Ensure that the **Integrated NIC Port** is enabled.
 - f. Press **Enter** to return to the **BIOS Boot Settings** screen.
 - g. Press **Escape** to exit **BIOS Boot Settings**.
 - h. Press **Escape** to exit **Boot Settings** and return to the **System BIOS Settings** screen.
4. Change the serial communication settings.
 - a. On the **System BIOS Settings** screen, select **Serial Communication**.

Figure 21. System BIOS Settings: Select Serial Communication



- b. On the **Serial Communication** screen, select **Serial Communication**. A pop-up window displays the available options.
- c. Select **On with Console Redirection via COM2**, then press **Enter**.

Figure 22. Serial Communication: Select Console Redirection via COM2



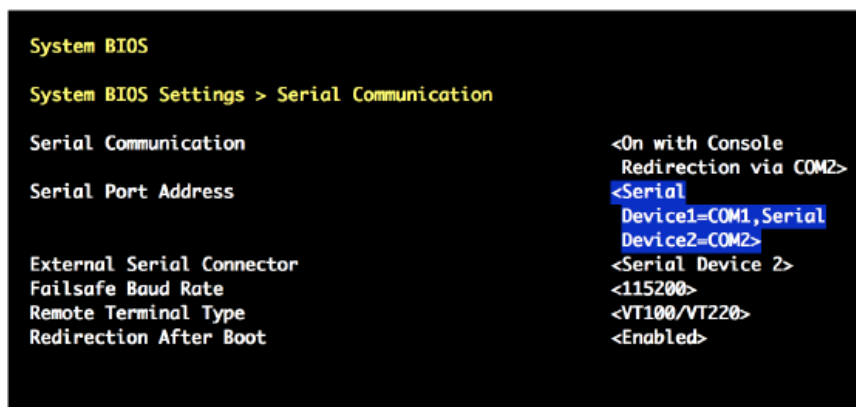
- d. Verify that **Serial Port Address** is set to Serial Device1=COM1, Serial Device2=COM2.

NOTE: This setting enables the remote console. If this setting is incorrect, remote access to the node is not established.

To make any necessary changes to the **Serial Port Address** settings, do the following:

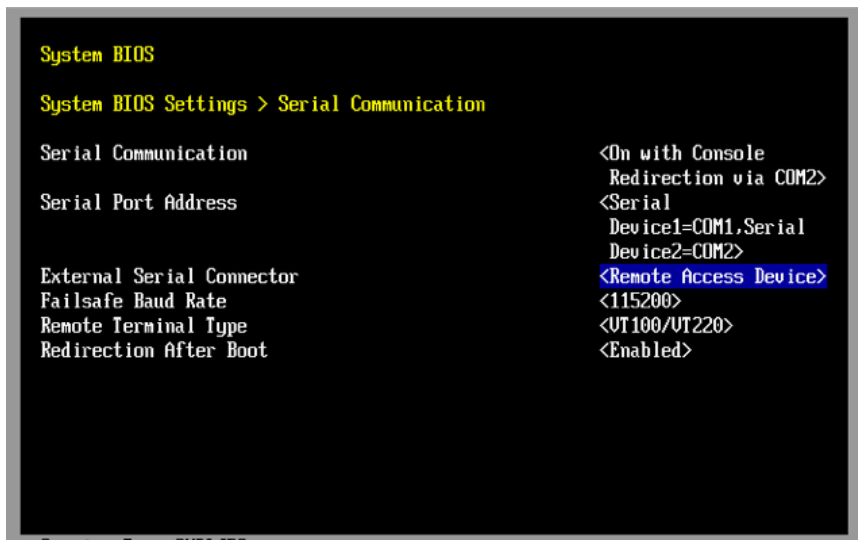
1. Press **Enter** to display the available **Serial Port Address** options.
2. Change the setting to: Serial Device1=COM1, Serial Device2=COM2.

Figure 23. Serial Communication: Serial Port Address



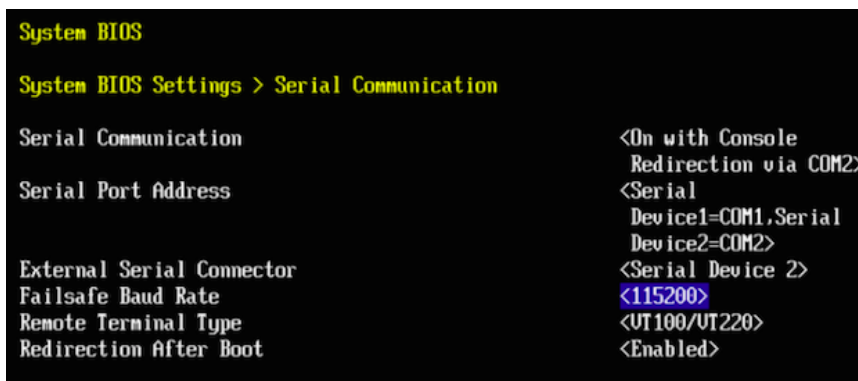
3. Press **Enter** to return to the **Serial Communication** screen.
- e. Select **External Serial Connector**. A pop-up window displays the available options.
- f. Select **Remote Access Device** in the **External Serial Connector** pop-up window, then press **Enter** to return to the previous screen.

Figure 24. External Serial Connector: Select Remote Access Device



- g. Select **Failsafe Baud Rate**. A pop-up window displays the available options.
- h. Select 115200 for the **Failsafe Baud Rate** in the pop up window, and then press **Enter** to return to the previous screen.

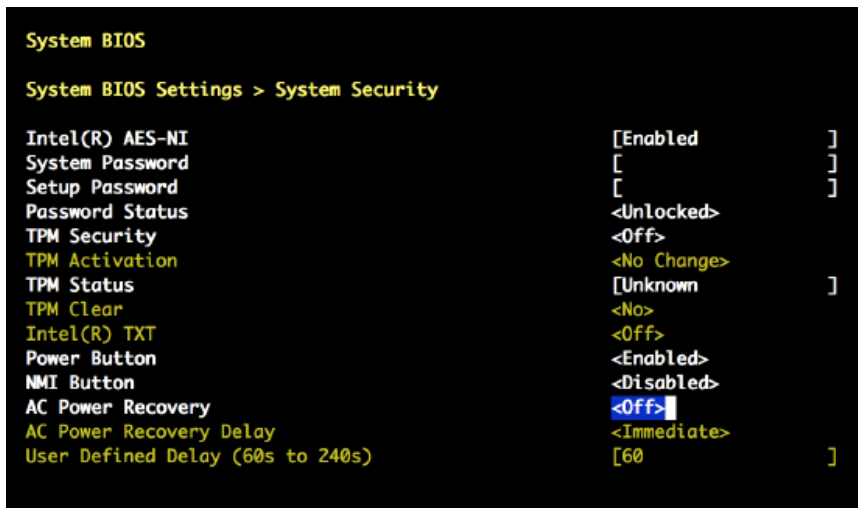
Figure 25. Serial Communication: Select 115200 Failsafe Baud Rate



- i. Press the **Escape** key to exit the **Serial Communication** screen.
 - j. Press the **Escape** key to exit the **System BIOS Settings** screen.
 - k. Press the **Escape** key to exit the **BIOS Settings** screen.
 - l. When the "Settings have changed" message appears, select **Yes** to save changes.
 - m. When the "Settings saved successfully" message appears, select **Ok**.
5. Set AC power recovery.
- a. Set the eLogin node to remain powered off after a system power failure.

Open the **System BIOS Settings** screen and select **System Security**. Select **AC Power Recovery** and set it to **Off** so that the node remains powered off after a system power failure. This will allow the SMW to power up first so that it is operational before all client nodes.

Figure 26. System Security: AC Power Recovery Off

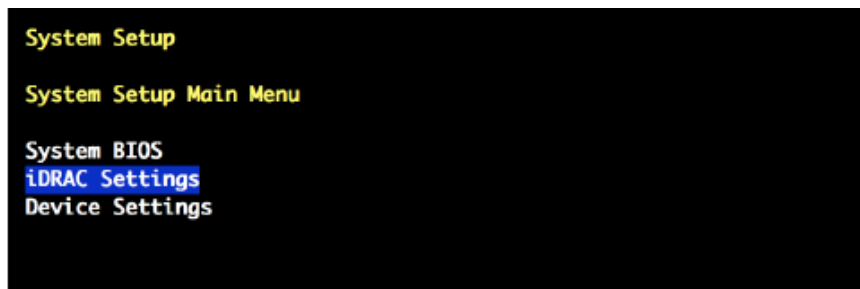


- b. Press the **Escape** key to exit the **System Security** screen.

————— CHANGE iDRAC SETTINGS —————

6. On the **System Setup Main Menu** screen, select **iDRAC Settings**, then press **Enter**.

Figure 27. System Setup Main Menu: iDRAC Settings



7. Select **Network** from **iDRAC Settings** screen, then press **Enter**.

Figure 28. iDRAC Settings: Network

```

iDRAC Settings

iDRAC Settings

iDRAC Settings Version          1.55.55.05
iDRAC Firmware Version         1.57.57 (Build 4)
System Summary
System Event Log
Network
Alerts
Front Panel Security
Virtual Media
vFlash Media
Lifecycle Controller
Power Configuration
Thermal
System Location
User Configuration
Smart Card

```

A long list of network settings is displayed.

8. Change the iDRAC IP address.

Always check to make sure the iDRAC IP address and related settings have the correct values.

MIGRATION: This is especially important when migrating from CMC-managed eLogin. The iDRAC IP address of each eLogin node must be changed so that it can be managed by the SMW.

- a. Scroll to the **IPV4 SETTINGS** list in the **Network** screen using the down-arrow key.

Figure 29. Network IPV4 SETTINGS

```

iDRAC Settings

iDRAC Settings > Network

IPV4 SETTINGS
Enable IPv4                <Enabled>
Enable DHCP                 <Disabled>
Static IP Address          [10.6.0.1]
Static Gateway             [10.6.1.1]
Static Subnet Mask         [255.255.0.0]
Use DHCP to obtain DNS server addresses <Disabled>
Static Preferred DNS Server [0.0.0.0]
Static Alternate DNS Server [0.0.0.0]

IPV6 SETTINGS
Enable IPv6                <Disabled>
Enable Auto-configuration <Enabled>
Static IP Address 1        [::]
Static Prefix Length       [64]

```

- b. Ensure that **Enable IPv4** is enabled.
- c. Ensure that **Enable DHCP** is disabled.
- d. Set **Static IP Address** to 10.6.0.x.

For x, substitute a number between 1 and 100 depending on which eLogin node is being configured.

- e. Set **Static Gateway** to 10.6.1.1.

This must match the IP address of the SMW eth6 interface on the external-ipmi-net network.

9. Change the IPMI settings to enable the Serial Over LAN (SOL) console.
 - a. Scroll to the **IPMI SETTINGS** list in the **Network** screen using the down-arrow key.
 - b. Ensure that **IPMI over LAN** (or **Enable IPMI over LAN**) is enabled.

To change **Enable IPMI over LAN** to **Enabled**, do the following:

1. Select **Enable IPMI over LAN**, then press **Enter**.
2. Select **Enabled** in the pop-up window.

Figure 30. Network IPMI SETTINGS: Enable IPMI over LAN

```

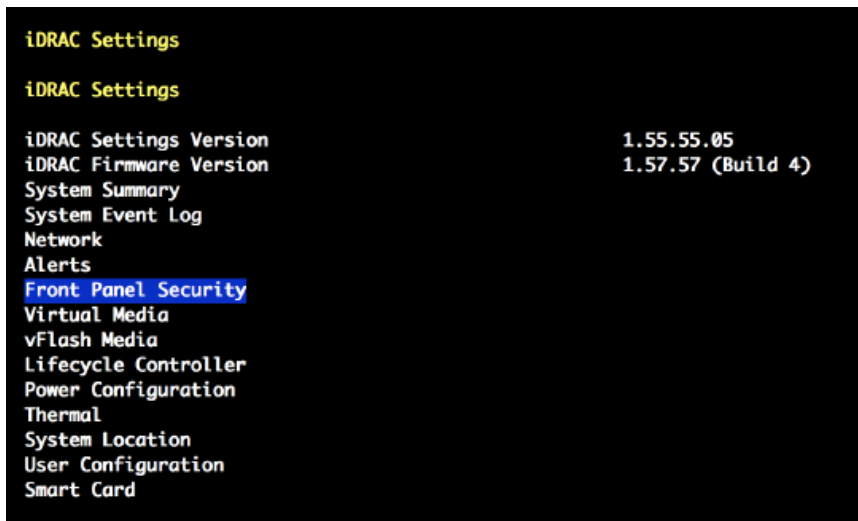
iDRAC Settings
iDRAC Settings > Network

Static IP Address 1      [::          ]
Static Prefix Length    [64         ]
Static Gateway          [::         ]
Link Local Address      ::
Use DHCPv6 to obtain DNS Server address <Disabled>
Static Preferred DNS Server [::         ]
Static Alternate DNS Server [::         ]

IPMI SETTINGS
Enable IPMI Over LAN    <Enabled>
Channel Privilege Level Limit <Administrator>
Encryption Key          [00000000000000000000000000000000]
                        000000000000000000000000]
  
```

3. Press **Enter** to return to the previous screen.
 - c. Press the **Escape** key to exit the **Network** screen, and return to the **iDRAC Settings** menu.
10. Change the LCD configuration to show the host name in the LCD display.
 - a. On the **iDRAC Settings** screen, scroll down using the down-arrow key to **LCD** (or **Front Panel Security**), and then press **Enter**.

Figure 31. iDRAC Settings: Front Panel Security



- b. Select **Set LCD message**. A pop-up window opens.
- c. Select **User-Defined String** in the pop-up window, and then press **Enter**.
- d. Select **User-Defined String** (again), and then press **Enter**. A text pop-up window opens for entering the new string.

Figure 32. Front Panel Security: User Defined String



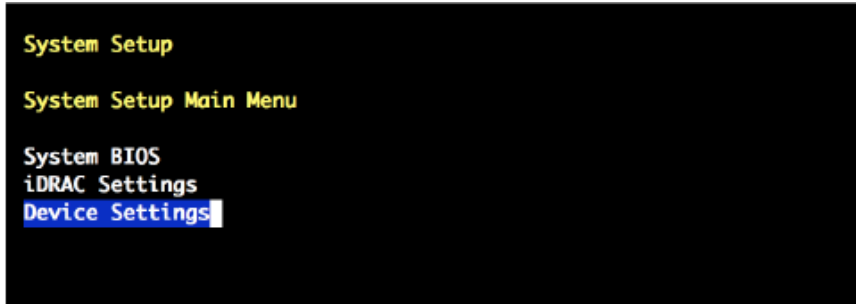
- e. Enter the host name (such as, `ellogin1`) in the text pop-up window.
- f. Press the **Escape** key to exit the **Set LCD message** screen.
- g. Press the **Escape** key to exit the **Network** screen.
- h. Press the **Escape** key to exit the **iDRAC Settings** screen.
- i. When the "Settings have changed" message appears, select **Yes** to save changes.
- j. When the "Settings saved successfully" message appears, select **Ok**, and then **Enter**.

————— CHANGE DEVICE SETTINGS —————

11. Change the device settings so that the node can PXE boot from the SMW management network (external-management-net).

- a. On the **System Setup Main Menu** screen, select **Device Settings**, and then press **Enter**.

Figure 33. System Setup Main Menu: Device Settings

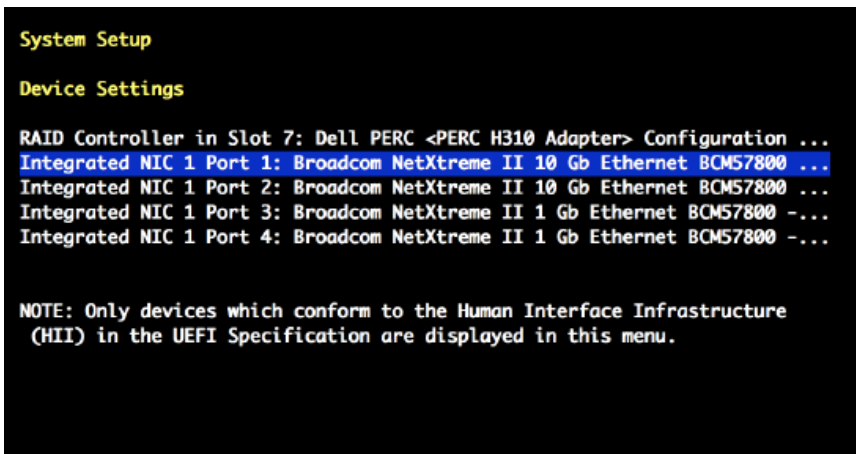


- b. Select **Integrated NIC 1 Port N ...** on the **Device Settings** screen, then press **Enter**. The **Main Configuration Page** opens.

Choose the NIC port number that corresponds to the Ethernet port for the external-management-net network:

- If external-management-net uses the first Ethernet port (eth0), select **Integrated NIC 1 Port 1 ...**
- If external-management-net uses the third Ethernet port (eth2), select **Integrated NIC 1 Port 3 ...**

Figure 34. Main Configuration Page: Select Integrated NIC 1 Port #



PXE booting must be disabled for the other three Ethernet ports.

- c. Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

Figure 35. Main Configuration Page: MBA Configuration Menu

```

Integrated NIC 1 Port 1: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 -
Main Configuration Page
Main Configuration Page
Firmware Image Menu
Device Hardware Configuration Menu
MBA Configuration Menu
iSCSI Boot Configuration Menu
FCoE Boot Configuration Menu
Virtualization Mode          <None>
Blink LEDs                   [0 ]
Chip Type                    BCM57800 B0
PCI Device ID                168A
Bus:Device:Function          01:00:00
Link Status                   <Disconnected>
Permanent MAC Address        [90:B1:1C:39:EA:38 ]
Virtual MAC Address          [90:B1:1C:39:EA:38 ]

```

- d. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, then press **Enter**. A pop-up window displays the available options.
- e. In the pop-up window, use the down-arrow key to highlight **PXE**, then press **Enter**.

Figure 36. MBA Configuration Menu: Legacy Boot Protocol - PXE

```

Integrated NIC 1 Port 1: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 -
Main Configuration Page > MBA Configuration Menu
Main Configuration > MBA Configuration Menu
Broadcom NetXtreme II 10 Gb Ethernet BCM57800 - 90:B1:1C:39:EA:38
Legacy Boot Protocol         <PXE>
Boot Strap Type              <Auto Detect>
Hide Setup Prompt            <Disabled>
Setup Key Stroke             <Ctrl-S>
Banner Message Timeout      [5 ]
Link Speed                   <10 Gbps Full>
Wake On LAN                  <Disabled>
VLAN Mode                    <Disabled>
VLAN ID (1..4094)           [1 ]
Boot Retry Count             <No Retry>

```

- f. Press the **Escape** key to exit the **MBA Configuration Menu** screen.
- g. Verify that **Legacy Boot Protocol** is set to **None** for the other three Ethernet ports. If necessary, change the setting for these three ports by repeating substep **9b**.
- h. Press the **Escape** key to exit the **Device Settings** screen.
- i. When the "Settings have changed" message appears, select **Yes** to save changes.
- j. When the "Settings saved successfully" message appears, select **Ok**, and then **Enter**. The main screen (**System Setup Main Menu**) appears.

12. Save changes and exit.

1. Press **Escape** to exit the **System Setup Main Menu**.
2. Select **Yes** when the utility displays the message "Are you sure you want to exit and reboot?"

The eLogin BIOS and remote access controller configuration is now complete.

13. Power off the node.

Cray recommends powering down the eLogin node prior to registering the node with `esd` on the SMW. This command requires the BMC root password for this node.

```
smw# ipmitool -I lanplus -H 10.6.1.X -U root -P <bmc-root-password> chassis power off
```

4.3.6 Add eLogin Nodes to the Node Registry

Prerequisites

- Required: Information for each eLogin node to be registered, which was gathered earlier in the process (BMC IP address, boot interface, MAC address, etc.).
- Optional: `inventory.csv` file with information for several eLogin nodes, typically used if migrating from CMC/eLogin or CIMS/esLogin.

About this task

The eLogin nodes to be managed by the SMW must be registered, that is, added to the eLogin node registry. The information in the node registry can be manipulated by the `enode create` command with command-line options for the node registry fields. The value of any of the fields can be changed later using the `enode update` command.

MIGRATION: If migrating from a previous method of managing the eLogin node, either CMC/eLogin or CIMS/esLogin, then use the `enode enroll` command to register nodes by importing the `inventory.csv` file. Three of the fields needed to add nodes to the node registry can be imported from the `inventory.csv` file: `node name`, `bmc_ip`, and `mgmt_mac`. The value for BMC IP address must be corrected for SMW-managed eLogin using the `enode update` command after the file has been imported. And because the `inventory.csv` file does not contain all of the information needed to manage eLogin nodes from the SMW, the missing information must be added later using `enode update`.

INITIAL DEPLOYMENT: If this is an initial deployment of eLogin, use the `enode create` command to add nodes to the eLogin node registry. Values for the following fields must be provided on the command line using the associated command-line options.

Table 10. Fields Required by `enode create` that Must be Specified on Command Line

Field	<code>enode create</code> Option	Definition
<code>node_type</code>	<code>--node_type</code>	Node type. For eLogin, set to <code>eLogin</code> . This field can be also be set by the environment variable <code>ENODE_DEFAULT_NODE_TYPE</code> .
<code>bmc_ip</code>	<code>--bmc_ip</code> or <code>-b</code>	IP address of the baseboard management controller (BMC).

Field	<code>enode create</code> Option	Definition
<code>mgmt_ip</code>	<code>--mgmt_ip</code> or <code>-m</code>	IP address of the boot interface, the eLogin interface on the external-management-net network.
<code>mgmt_mac</code>	<code>--mgmt_mac</code> or <code>-a</code>	MAC address of the boot interface, the eLogin interface on the external-management-net network.
<code>bmc_password</code>	<code>--bmc_password</code> or <code>-p</code>	Name of a file containing the user name password used to connect to the BMC. NOTE: If this information is not included on the command line, then <code>enode create</code> prompts for the BMC password and BMC user name for the node.
<code>node</code>	N/A	Name of the eLogin node.

If using the `enode create` command, the following fields are also required, but because they have default values, any that are not provided on the command line will keep the default value. If a default value is not correct for the eLogin node being registered, add the associated command-line option when using `enode create`, or use `enode update` to modify the value later.

Table 11. Fields Required by `enode create` that Have Default Values

Field	<code>enode create</code> Option	Definition
<code>bmc_user</code>	<code>--bmc_user</code> or <code>-u</code>	User name to connect to the BMC. The default value is <code>root</code> .
<code>bootif</code>	<code>--bootif</code>	Management/boot interface. The default value is <code>eth0</code> , which is correct for a 4x1GbE LOM network adapter. If this eLogin node has a 2x10GbE+2x1GbE LOM network adapter instead, then change this value to <code>eth2</code> .
<code>pci</code>	<code>--pci</code>	PCI kernel parameter. The default value is <code>bfsort</code> . It is uncommon to use a different value.
<code>remcon</code>	<code>--remcon</code>	Remote console device setting. The default value is <code>/dev/ttyS1,115200</code> for the path and baud rate of the console.
<code>rootdev</code>	<code>--rootdev</code>	Root storage device. The default value is the first disk, <code>/dev/sda</code> .
<code>ssh_host_keys</code>	<code>--ssh_host_keys</code>	How the node gets its SSH host keys. The default value is <code>simple_sync</code> .

The following fields are not required at the time of eLogin node creation. They can be added with `enode create` then or added later with `enode update`, when the information is known.

Table 12. Fields that Can be Specified with `enode create` or Added Later with `enode update`

Field	<code>enode create</code> Option	Definition
<code>esd_group</code>	<code>--group</code> or <code>-g</code>	The <code>esd</code> group to assign the node. <code>esd_group</code> may be a single value or a comma-separated list of group names.
<code>configset</code>	<code>--configset</code> or <code>-c</code>	CLE config set to assign the node.
<code>storage_profile</code>	<code>--storage_profile</code>	Storage profile in the CLE config set describing the disk setup of the node.
<code>image</code>	<code>--image</code> or <code>-i</code>	Image to assign the node.
<code>kdump_enable</code>	<code>--kdump_enable</code> or <code>-d</code>	Enables <code>kdump</code> functionality on the node.
<code>kdump_high</code>	<code>--kdump_high</code>	High memory value for use with <code>kdump</code> . For example, setting this to <code>256G</code> will result in the kernel parameter <code>crashkernel=256G,high</code> . This is set on the kernel parameter line only when <code>kdump_enable</code> is set.
<code>kdump_low</code>	<code>--kdump_low</code>	Low memory value for use with <code>kdump</code> . For example, setting this to <code>4G</code> will result in the kernel parameter <code>crashkernel=4G,low</code> . This is only set on the kernel parameter line when <code>kdump_enable</code> is set.
<code>param</code>	<code>--parameters</code> or <code>-k</code>	String of kernel parameters to assign the node.



CAUTION: After issuing an `enode create` or `enode update` command, wait at least 5 seconds before issuing an `enode boot` or `enode reboot` command. This delay ensures that modified data in the node registry in memory has been written to the data store on the SMW's disk. Ignoring this short delay may lead to a failed attempt to PXE boot a node.

Procedure

1. Set the default node type for subsequent `enode` commands.

```
smw# export ENODE_DEFAULT_NODE_TYPE=eloin
```

TIP: Add the following line to the `/root/.bash_profile`:

```
export ENODE_DEFAULT_NODE_TYPE=eloin
```

————— ADD NODES TO THE NODE REGISTRY —————

MIGRATION: Continue with step 2 on page 87.

INITIAL DEPLOYMENT: Continue with step 3 on page 88.

2. Register several nodes with the `enode enroll` method.

This example uses an inventory file from a CMC that has only one eLogin node in it, `eloin1`, but several nodes can be registered at one time.

- a. View the contents of the `inventory.csv` file.

```
smw# cat inventory.csv
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUs, ARCH, RAM_MB, DISK_GB, NODE_DESC
eloin1,10.148.0.1,11:22:33:44:55:66,32,x86_64,131072,550,eloin1
```

- b. Edit the inventory file to update the `bmc_ip` value, as needed.

Because the network address range for the `external-ipmi-net` is `10.6.0.0/16`, edit the inventory file to move the `BMC_IP` from the `10.148.0.0/16` network to to `10.6.0.0/16` network.

```
smw# vi inventory.csv

smw# cat inventory.csv
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUs, ARCH, RAM_MB, DISK_GB, NODE_DESC
eloin1,10.6.0.1,11:22:33:44:55:66,32,x86_64,131072,550,eloin1
```

- c. Enroll nodes from the inventory file.

```
smw# enode enroll inventory.csv
Creating the following node:
eloin1
Successfully created ['eloin1'].
```

- d. Add the missing required fields for each node in the inventory file.

```
smw# enode update --set-mgmt_ip 10.7.0.1 \
--set-bmc_password /root/bmc_password eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

When all enrolled nodes have been updated, skip the next step and continue with step 4 on page 88.

3. Register a node with `enode create` method.

Only one node at a time can be registered with `enode create`. This example creates `eloin1`. All of these parameters are required when registering a node.

The BMC password must be supplied. If `--bmc_password /path/to/passwordfile` is not on the command line, then `enode create` will prompt the admin to enter the password for the BMC of the node.

```
smw# enode create --node_type eloin --bmc_ip 10.6.0.1 --mgmt_ip 10.7.0.1 \
--bmc_password /root/bmc_password --mgmt_mac 11:22:33:44:55:66 eloin1
Creating the following node:
eloin1
Successfully created ['eloin1'].
```

Repeat this step for each eLogin node in this system. When all nodes have been registered, continue with the next step.

————— CONFIRM REGISTRY DATA, UPDATE NODES WITH ADDITIONAL DATA —————

4. List the nodes in the registry to confirm the data entered.


```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
elogin1 - - - - 10.6.0.1 10.7.0.1 11:22:33:44:55:66 - UNKNOWN
```

Note that the `configset`, `storage_profile`, `esd_group`, and `image` fields do not yet have values.

5. Update fields required for booting the nodes.

- a. Set the `esd` group for a node.

```
smw# enode update --set-group elogin elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- b. Set the CLE config set that should be applied to this node.

```
smw# enode update --set-configset p0 elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- c. Set the storage profile to be one of the profiles specified in the `cray_storage` config service of the CLE config set that was just assigned to this node.

This example assigns the default storage profile. If a storage profile was created specifically for a first PXE boot, use that profile instead.

```
smw# enode update --set-storage_profile elogin_default elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

Repeat this step for each eLogin node in this system.

————— UPDATE FIELDS THAT HAVE DEFAULT VALUES, IF NEEDED —————

6. Update other fields for the node, if their defaults are not correct for this node.

- a. (Conditional) If the management interface on the eLogin node is not `eth0`, the default for this field, then change the value of `bootif`.

The default value of `eth0` is correct for the 4x1GbE LOM network adapter. If this node has a 2x10GbE +2x1GbE LOM network adapter instead, then set `bootif` to `eth2`.

```
smw# enode update --set-bootif eth2 elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- b. (Conditional) If the boot disk for this eLogin node is not the first disk, the default for this field, then change the value of `rootdev`.

```
smw# enode update --set-rootdev /dev/sdb elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- c. (Conditional) If the console device is not on `/dev/ttyS1` or the baud rate of the console is not 115,200, then change the value of `remcon`, which is set to `ttyS1,115200n8` by default.

```
smw# enode update --set-remcon <disk,baud> elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- d. (Conditional) If the BMC username for this node is not `root`, the default for this field, then change the value of `bmc_username`.

```
smw# enode update --set-bmc_user <other_user> elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- e. (Conditional) If this site wishes to use a non-default method of handling SSH host keys on this node, then change the value of `ssh_host_keys`.

If `ssh_host_keys` is set to `simple_sync` (the default), then `esd` will use the SSH host keys from the config set. If the value is `generate`, then `esd` will generate new SSH host keys. If the value is an absolute path, then `esd` will use the site-supplied SSH host keys from that location.

```
smw# enode update --set_ssh_host_keys generate elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

If setting this to a value other than `simple_sync`, also ensure that:

- SSH host keys are not present in the Simple Sync directory structure for this node (see next step). If present, the PXE boot will fail.
- Automatic generation of SSH host keys has been disabled in `cray_ssh` (see [Update the Config Set for eLogin](#) on page 51).

- f. (Conditional) Remove SSH host keys from Simple Sync directories for an eLogin node, if needed.

If this eLogin node has `ssh_host_keys` set to any non-default value (i.e., not `simple_sync`), then remove any SSH host keys from all Simple Sync directory(ies) that apply to this eLogin node.

SSH host keys for this eLogin node may be present on the SMW in one or more of the following Simple Sync directories:

```
/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/hostname/<name>/files/etc/ssh
/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/nodegroups/<node_group>/files/etc/ssh
/var/opt/cray/imps/config/sets/<config_set>/files/simple_sync/common/files/etc/ssh
```

Repeat this step for each eLogin node in this system.

————— CONFIRM REGISTRY DATA —————

7. List the nodes in the registry to confirm the data entered.

Note that the BMC password is not displayed by the `enode list` command.

Show the most commonly changed fields in the node registry.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
eloin1 p0 eloin_default eloin - 10.6.0.1 10.7.0.1 11:22:33:44:55:66 - UNKNOWN
```

Show ALL fields in the node registry.

```
smw# enode list --fields all
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE ROOTDEV BOOTIF PCI REMCON BMC_USERNAME ENABLE_KDUMP KDUMP_HIGH
KDUMP_LOW SSH_HOST_KEYS
eloin1 p0 eloin_default eloin - 10.6.0.1 10.7.0.1 90:B1:1C:3A:0A:1B -
UNKNOWN /dev/sda eth0 pci=bfsort ttyS1,115200n8 root False - - generate
```

4.3.7 Configure the eLogin RAID Virtual Disks

Prerequisites

- SMW/eLogin network hardware is installed and configured.
- The required disk configuration has been defined in `cray_storage` in the CLE config set.
- The eLogin node has been enrolled in the node registry.

This procedure applies to both a migration and an initial eLogin deployment. It is required for a migration because the RAID virtual disks, `/dev/sda` and `/dev/sdb`, must be reconfigured for SMW-managed eLogin.

About this task

For the SMW-managed eLogin software to function correctly, the eLogin RAID must be configured to have these two disks:

- `/dev/sda`: volatile storage
- `/dev/sdb`: persistent storage

This procedure configures the internal RAID controller with two virtual disks to be presented to the eLogin node's operating system. The two virtual disks must be named `sda` and `sdb`. The `sda` disk is used for volatile storage and configured with these partitions: GRUB, BOOT, WRITELAYER, TMP, and SWAP. The `sdb` disk is used for persistent storage and configured with these partitions: CRASH and PERSISTENT.

This procedure includes detailed steps for the Dell R820 server using the PERC H310 Mini BIOS Configuration Utility 3.00-0020. Depending on the server model and version of RAID configuration utility, there could be minor differences in the steps to configure this server. For more information, refer to the documentation for the Dell PERC controller or server RAID controller software.

Procedure

1. Connect to the console of the eLogin node.

There are two methods of connecting to the console:

- Method 1: Physically connect to the console of the eLogin node.
- Method 2: Connect to the console using `conman`.

In a separate window on the SMW, run this command, substituting the name of this eLogin node for *eLogin1*:

```
smw# conman -j eLogin1
```

2. Restart the eLogin node.

This will restart or start the node to enable access to boot configuration menus.

Substitute the name of this eLogin node for *eLogin1*.


```
smw# enode reboot --bios eLogin1
```

3. Open the BIOS RAID configuration screen.

Press **Ctrl-R** within 5 seconds of seeing the following screen.

NOTE: The BIOS RAID configuration screens may appear different if accessing from `conman` versus the physical console, but the functionality is the same.

Figure 37. Initial Boot Menu for BIOS RAID Configuration: eLogin



```
F2 = System Setup
F10 = Lifecycle Controller
F11 = Boot Manager
F12 = PXE Boot

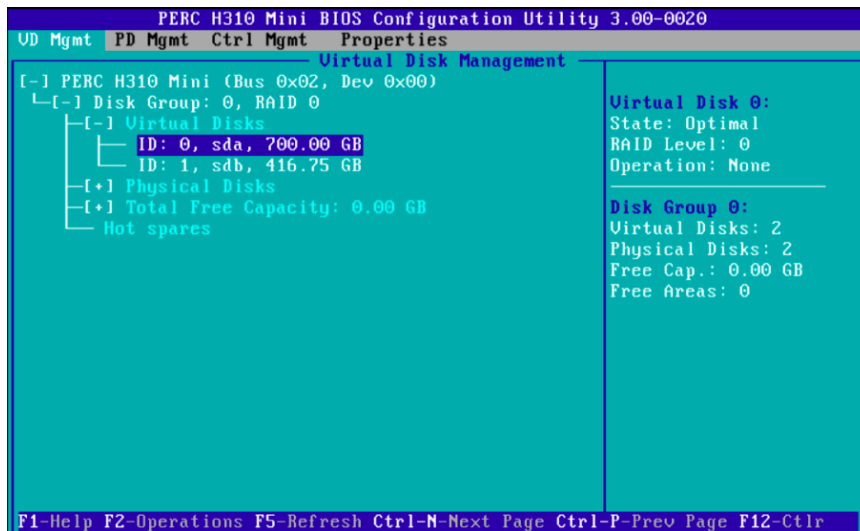
QLogic Ethernet Boot Agent
Copyright (C) 2015 QLogic Corporation
All rights reserved.
Press Ctrl-S to enter Configuration Menu

Initializing Serial ATA devices...
Port J: HL-DT-ST DVD-ROM DU90M

PowerEdge Expandable RAID Controller BIOS
Copyright(c) 2015 Avago Technologies
Press <Ctrl><R> to Run Configuration Utility
```

The RAID configuration screen opens.

Figure 38. RAID Configuration Screen: eLogin

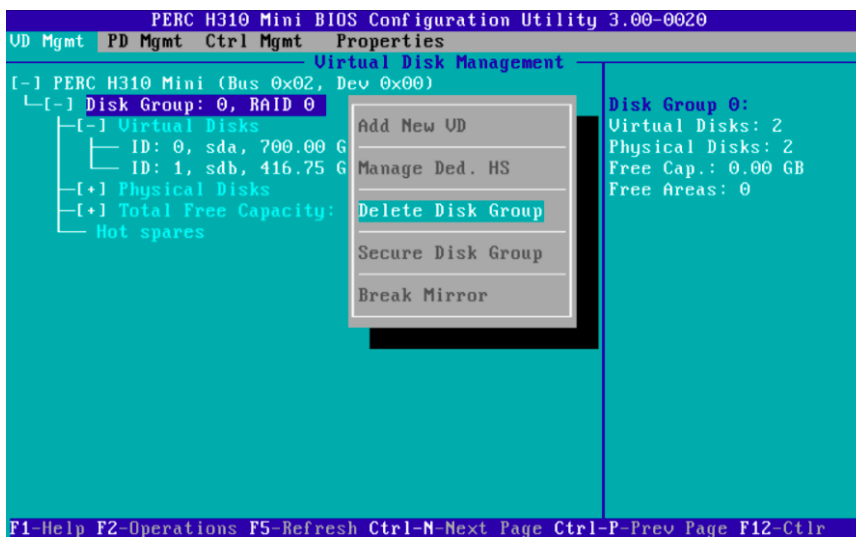


4. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration, as defined in `cray_storage` in the CLE config set. If there are none to delete, skip this step.

Occasionally disks are not viewable by the OS after RAID reconfiguration. This may be caused by residual metadata on the disk from the previous RAID configuration. To clear the metadata, remove the disks from any RAID configuration, and then initialize the disks. After initialization completes, reconfigure the disks as part of the RAID. This clears any pre-existing metadata and allows the OS to see the devices.

- a. Select the disk.
- b. Press **F2** key to get a list of operations.
- c. Select **Delete Disk Group** and press **Enter**.

Figure 39. Delete Disk Group: eLogin BIOS RAID Setup

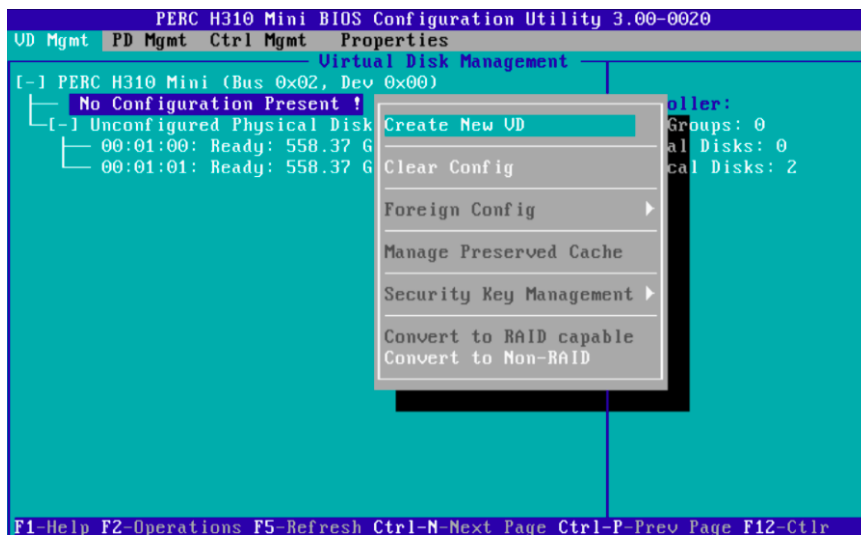


- d. Confirm the selection **Yes**, and press return.



5. Create a new virtual disk A.
 - a. In the virtual disk management window (**VD Mgmt**), navigate to **No Configuration Present !** using the keyboard up/down arrows.
 - b. Press the **F2** key to access the disk creation menu.
 - c. Select **Create New VD** from the menu.

Figure 40. Create Virtual Disk A: eLogin BIOS RAID



The **Create New VD** window opens.

6. Move the cursor to select the disk ID in the **Create New VD** window, and then press spacebar on keyboard to add disk to RAID.
7. Set the RAID Level to **RAID 0**.
8. Set **VD Size** and **VD Name** for virtual disk A.

- a. Set the **VD Size** for virtual disk A to **700 GB** of disk space.

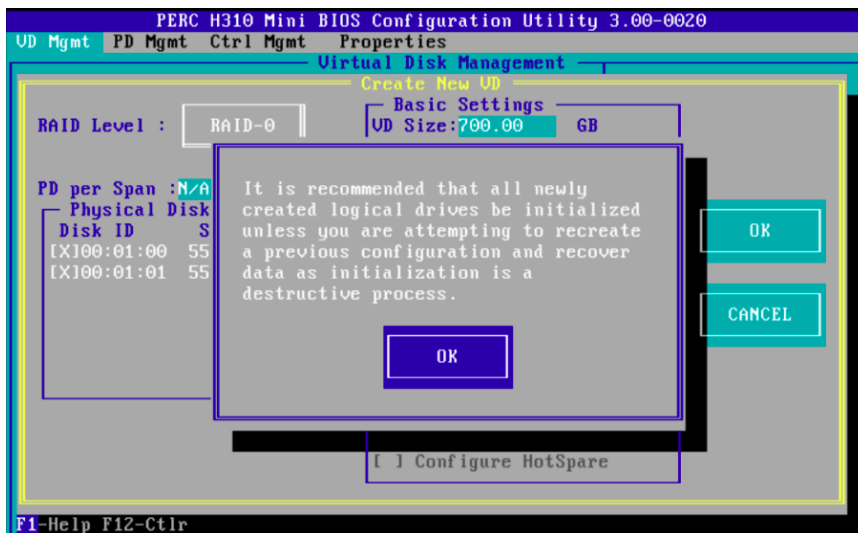
IMPORTANT: 700 GB is sufficient to accommodate the partition sizes specified in the default storage profile for eLogin nodes, `eLogin_default`, which is defined in the `cray_storage` configuration service. If those sizes were increased for this eLogin node, increase the **VD Size** accordingly.

- b. Set the **VD Name** to `sda`.

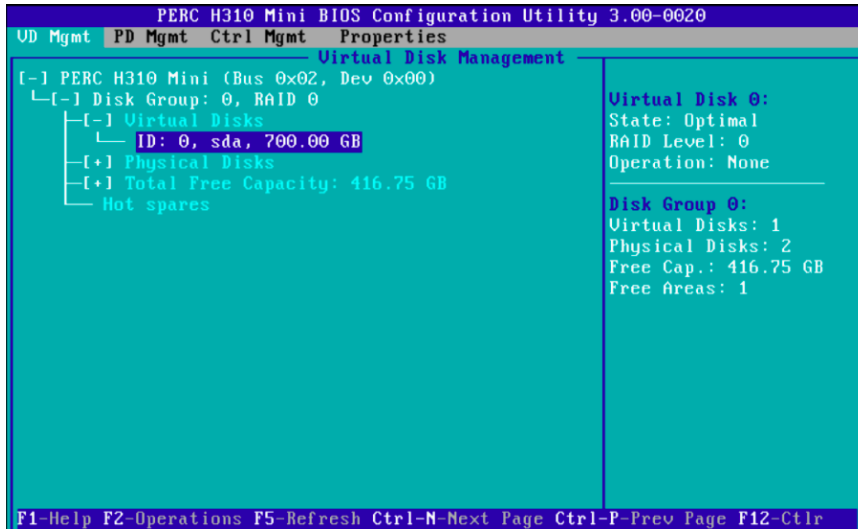
Figure 41. Disk Size and Name Setting for Virtual Disk A: eLogin



- c. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Virtual disk `sda` is now created.



9. Initialize virtual disk A (sda) using **Fast Initialization**.

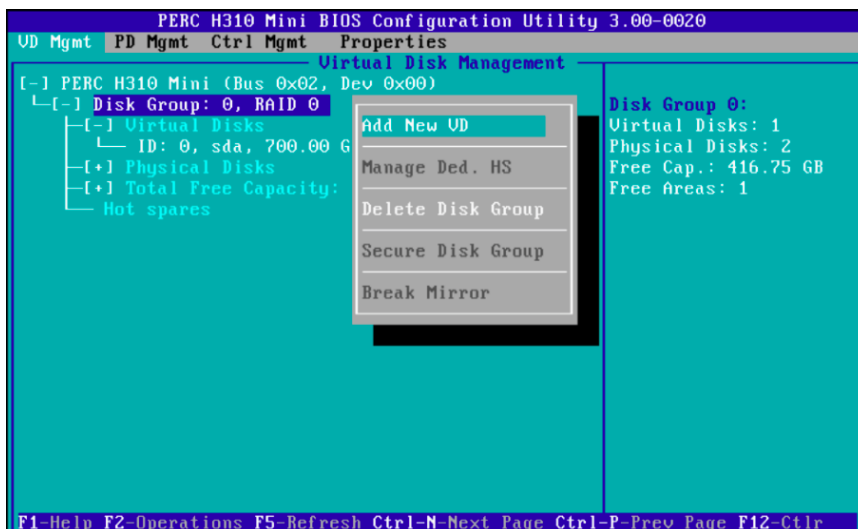
- Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- In the **Initialization** submenu, select **Fast Initialization**.

A pop-up window will be displayed, indicating that the virtual disk has been initialized.

10. Create a new virtual disk B.

- In the **Virtual Disk Management** window, navigate to **Disk Group: 0, RAID 0** using the keyboard up/down arrows.
- Press **F2** to access the disk creation menu.
- Select **Add New VD**.

Figure 42. Create New Virtual Disk B: eLogin BIOS RAID



The **Add VD in Disk Group 0** window opens.

- d. In the window, set the **VD Name** to **sdb**, and verify that the **VD Size** is set to the remaining disk space.

Figure 43. Disk Size and Name Setting for Virtual Disk B: eLogin

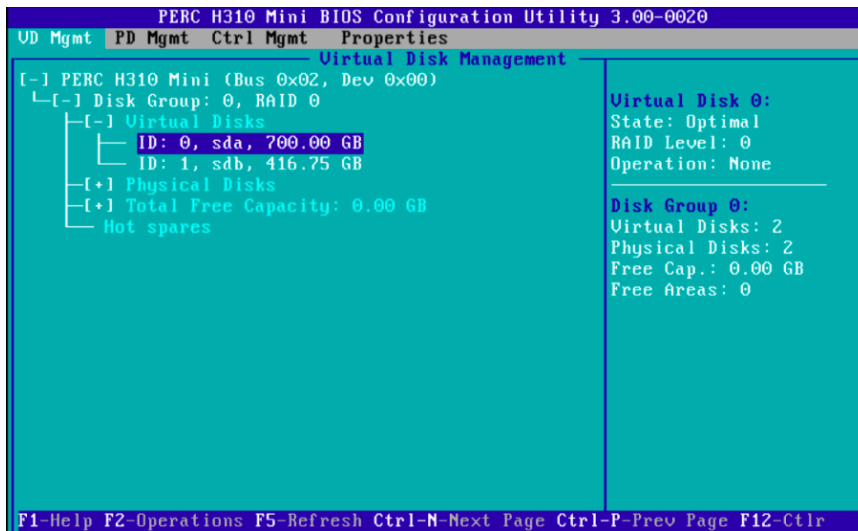


- e. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Two virtual disks are now available.

Figure 44. Two Virtual Disks Available: eLogin BIOS RAID



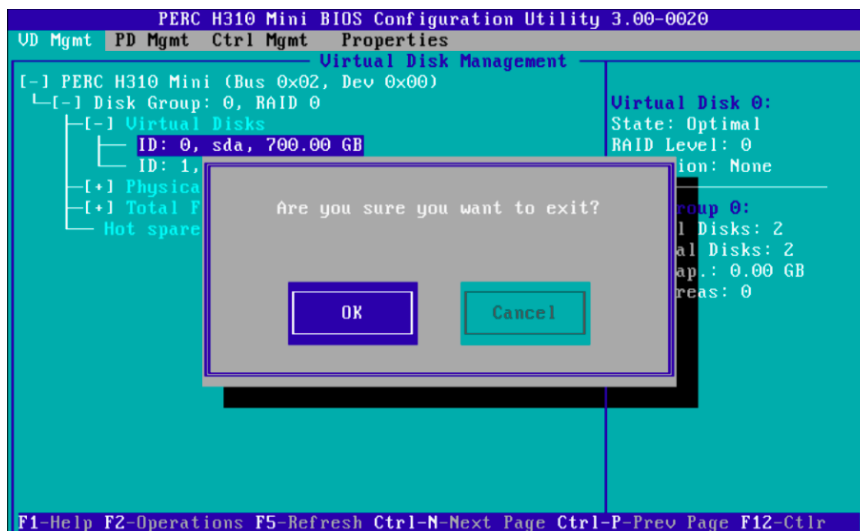
11. Initialize virtual disk B (sdb) using Fast Initialization.

- Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- In the **Initialization** submenu, select **Fast Initialization**.

A pop-up window will be displayed, indicating that the virtual disk has been initialized.

12. Press Esc on the keyboard to exit the BIOS configuration, and then select OK to confirm exit from the BIOS Configuration Utility.

Figure 45. Exit BIOS Configuration: eLogin



The BIOS configuration utility screen is now closed.

13. Press Ctrl+Alt+Delete from the keyboard to reboot the node.

4.3.8 Create an eLogin Image

Prerequisites

SMW/CLE software is installed and configured.

About this task

This procedure creates images for the eLogin nodes using the `image create` command. To create images using `imgbuilder`, see [Create eLogin Images with imgbuilder](#) on page 133. If using `imgbuilder`, the recipes for eLogin nodes can be used to create image roots and export the image roots into the proper format for booting at the same time as other recipes are used to build images.

The recipe used to build a boot image for an eLogin node should be closely matched to the recipe used to build internal login and compute node boot images.

- If the internal nodes are using `tmpfs` recipes, which do not have "large" as part of their name (e.g., `login_cle_6.0up06_sles_12sp3_ari` and `compute_cle_6.0up06_sles_12sp3_ari`), then choose the smaller eLogin recipe: `eloin-smw_cle_6.0up06_sles_12sp3_ari`.
- If the internal nodes are using `netroot` recipes, which have "large" as part of their name (e.g., `login-large_cle_6.0up06_sles_12sp3_ari` and `compute-large_cle_6.0up06_sles_12sp3_ari`), then choose the larger eLogin recipe: `eloin-smw-large_cle_6.0up06_sles_12sp3_ari`.

Procedure

1. Select an eLogin recipe.

There are two types of recipes, the "eloin-smw" recipe and the "eloin-smw-large" recipe. This documentation uses the "eloin-smw-large" recipe in all examples.

Use the "eloin-smw" recipe only if:

- the compute nodes and login nodes are using `tmpfs` images ("login" and "compute") instead of `netroot` images ("login-large" and "compute-large")
- there are specific size constraints for the eLogin image
- the image is intended for test purposes

2. **Optional:** Create a custom eLogin recipe if additional packages are required.

- a. Create a new image recipe with a custom name (using the "custom" prefix).

```
smw# recipe create custom-eloin-smw-large_cle_6.0up06_sles_12sp3_ari
```

- b. Add `eloin-smw-large_cle_6.0up06_sles_12sp3_ari` as a sub-recipe.

```
smw# recipe update -i eloin-smw-large_cle_6.0up06_sles_12sp3_ari \
custom-eloin-smw-large_cle_6.0up06_sles_12sp3_ari
```

- c. Add any additional packages, package collections, `postbuild_copy`, or `postbuild_chroot` information to this custom recipe before building an image root from it.

See "Install Third-Party Software with a Custom Image Recipe" procedure (under "Modify an Installed System" section) in the *XC™ Series System Administration Guide*.

3. Build the eLogin image.

For custom recipes:

```
smw# image create -r custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari \
custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

For "elogin-smw" recipes:

```
smw# image create -r elogin-smw-large_cle_6.0up06_sles_12sp3_ari \
elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

Cray recommends appending a date stamp, such as "YYYYMMDD", to images created with `image create`. For example, if generating `elogin-smw-large_cle_6.0up01_sles_12_ari` on June 1st, 2017, the image should be named `elogin-smw-large_cle_6.0up01_sles_12_ari_20170601`. If `imgbuilder` is used to create the image, it will add date stamps automatically.

4.3.9 Export an eLogin Image

Prerequisites

- SMW/CLE software is installed and configured.
- An eLogin image root has been created from a recipe.

About this task

The `image export` command has a `format` option to produce a SquashFS image from an image root. Two SquashFS images are needed for the eLogin node, both the operating system image and the PE image with Programming Environment (PE) software.

Procedure

Export the eLogin image root into a SquashFS-formatted image for booting the eLogin node.

```
smw# image export --format squashfs \
custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

This image name in this example uses the naming convention suggested in the [Create an eLogin Image](#) on page 99 procedure.

This command creates a directory under `/var/opt/cray/imps/boot_images` with the name of the image, and it creates a SquashFS file and an `.imps_Image_metadata` file in that directory.

4.3.10 Assign Image to eLogin Nodes

Prerequisites

- SMW/CLE software is installed and configured.
- An eLogin image root has been created from a recipe.
- An eLogin boot image in SquashFS format has been created from an image root.

About this task

Each eLogin must be assigned an image before it can PXE boot to provision that image to the internal storage of the eLogin node. The first step in this procedure shows how to assign an image to a single eLogin node. The second step shows how to assign an image to several eLogin nodes at the same time. Use one or both steps, as needed.

Procedure

1. Assign an image to a single eLogin node.

This example assigns a custom eLogin image to the node *ellogin1*.

```
smw# enode update \  
-i custom-ellogin-smw-large_cle_6.0up06_sles_12sp3-YYYYMMDD ellogin1
```

2. Assign an image to all of the eLogin nodes in an ESD group.

This example assigns a custom eLogin image to all eLogin nodes in the ESD group *ellogin*.

```
smw# enode update \  
-i custom-ellogin-smw-large_cle_6.0up06_sles_12sp3-YYYYMMDD -g ellogin
```

4.3.11 Export a PE Image

Prerequisites

- SMW/CLE software is installed and configured on the SMW.
- The Programming Environment (PE) image root has been created from the PE recipe, and PE software has been installed. This is the same PE image used for CLE nodes.

About this task

The `image export` command has a format option to produce a SquashFS image from an image root. Two SquashFS images are needed for the eLogin node: the operating system image and the PE image with PE software. This procedure exports the PE image root into a SquashFS-formatted image in preparation for booting the eLogin node.

Procedure

1. Find the name of the PE image recipe installed on the SMW, and set an environment variable for it.

```
smw# image list | grep ^pe  
pe_compute_cle_6.0up06_sles_12sp3  
smw# export PEIMAGE=$(image list | grep ^pe)  
smw# echo $PEIMAGE  
pe_compute_cle_6.0up06_sles_12sp3
```

2. Determine whether a PE image has already been exported on this SMW.

The first command looks for the existence of a directory with that image name under the boot images directory. Such a directory is created when an image is exported. The second command looks for a

SquashFS file and metadata file within that directory, which would indicate that the export of the PE image as a SquashFS image completed successfully.

```
smw# ls -d /var/opt/cray/imps/boot_images/$PEIMAGE
/var/opt/cray/imps/boot_images/pe_compute_cle_6.0up06_sles_12sp3
smw# ls /var/opt/cray/imps/boot_images/$PEIMAGE
.imps_Image_metadata squashfs
```

The example output shows the existence of a directory containing a SquashFS file. If there is no such directory and file, the PE image has not been exported yet.

3. If there is no SquashFS PE image, export it now.

```
smw# image export --format squashfs $PEIMAGE
```

This will create a directory under `/var/opt/cray/imps/boot_images` with the name of the image, and it will create a SquashFS file and an `.imps_Image_metadata` file in that directory.

4. Ensure that the PE image binding profile is enabled in the CLE config set.

The examples use CLE config set `p0`. Substitute the correct name of the CLE config set used for this system.

- a. Determine whether the PE image binding profile is enabled.

```
smw# cfgset get cray_image_binding.settings.profiles.data.PE.enabled p0
true
```

- b. If the PE profile is not yet enabled, enable it now.

```
smw# cfgset modify -s true \
cray_image_binding.settings.profiles.data.PE.enabled p0
```

5. Ensure that the existing exported PE image name matches the image name in the PE image binding profile.

The examples use CLE config set `p0`. Substitute the correct name of the CLE config set used for this system.

- a. Determine whether the PE image name matches.

```
smw# cfgset get cray_image_binding.settings.profiles.data.PE.image p0
pe_compute_cle_6.0up06_sles_12sp3
```

- b. If the PE image does not match, modify the PE image name in `cray_image_binding` now.

```
smw# cfgset modify -s $PEIMAGE \
cray_image_binding.settings.profiles.data.PE.image p0
```

6. Ensure that the PE image binding profile is applied to eLogin nodes.

- a. Determine whether eLogin nodes are configured in the `e_login_nodes` node group in the `cray_node_groups` config service.

```
smw# cfgset get cray_node_groups.settings.groups.data.e_login_nodes.members p0
e_login1
```

- b. Add any eLogin node(s) not present in the `e_login_nodes` node group.

```
smw# cfgset modify -a e_login2 \
cray_node_groups.settings.groups.data.e_login_nodes.members p0
```

```
smw# cfgset get cray_node_groups.settings.groups.data.elogin_nodes.members p0
elogin1
elogin2
```

- c. Determine whether the PE image binding profile includes the `elogin_nodes` node group in its `client_groups` list.

```
smw# cfgset get cray_image_binding.settings.profiles.data.PE.client_groups
p0
login_nodes
compute_nodes
```

- d. If `elogin_nodes` is not in the `client_groups` list, add it.

```
smw# cfgset modify -a elogin_nodes \
cray_image_binding.settings.profiles.data.PE.client_groups p0
```

```
smw# cfgset get \
cray_image_binding.settings.profiles.data.PE.client_groups p0
login_nodes
compute_nodes
elogin_nodes
```

7. Update the config set if it was modified.

If any of the `cfgset modify` commands in previous steps were necessary, they modified the CLE config set without running pre- and post-configuration scripts, which marks the config set invalid. This step uses `cfgset update` in prepare mode (no user interaction) to ensure that all configuration scripts are run. Replace `p0` with the name of the CLE config set that was modified in previous steps.

```
smw# cfgset update -m prepare p0
```

4.3.12 Add eLogin Resources to the SMW HA Cluster

Prerequisites

- eLogin nodes have completed their first PXE boot successfully.
- The eLogin node registry or CLE config set has been modified to ensure persistent storage for the eLogin nodes.

About this task

NOTE: This procedure applies to SMW HA systems only.

The external state daemon (`esd`) was enabled and started manually early in the eLogin configuration and deployment process so that the `enode` command could be used. For an SMW HA system, eLogin resources must also be added to the SMW HA cluster so that `esd` will be started automatically during subsequent boots of the SMW. This is a one-time action that is necessary only the first time SMW-managed eLogin is set up for an SMW HA system.

This procedure assumes familiarity with *XC™ Series SMW HA Administration Guide (S-2551)* and *XC™ Series SMW HA Installation Guide (S-0044)*.

Procedure

1. Check cluster status.

```
smw1# crm_mon -r1

Stack: unknown
Current DC: smw1 (version unknown) - partition with quorum
Last updated: Thu Feb  8 16:43:30 2018
Last change: Thu Feb  8 08:33:26 2018 by hacluster via crmd on smw1

2 nodes configured
33 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

ClusterIP      (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP1     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP2     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP3     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP4     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP5     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
ClusterTimeSync (ocf::smw:ClusterTimeSync):    Started smw1
HSSDaemonMonitor (ocf::smw:HSSDaemonMonitor):   Started smw1
<snip>
Resource Group: SystemGroup
  NFSServer (systemd:nfsserver):    Started smw1
  EnableRsyslog (ocf::smw:EnableRsyslog):  Started smw1
  syslog.socket (systemd:syslog.socket):  Started smw1
Clone Set: clo_PostgreSQL [PostgreSQL]
  Started: [ smw1 smw2 ]
```

If all of the cluster resources are not running as expected, please refer to *XC™ Series SMW HA Administration Guide (S-2551)*.

2. Log into the host name of the active SMW.

```
node# ssh smw
```

3. Add eLogin resources to the cluster using SMWHAconfig.

```
smw# cd /opt/cray/ha-smw/default/hainst
smw# ./SMWHAconfig --update --add_elogin
```

The SMWHAconfig command puts the cluster into maintenance mode.

4. Exit maintenance mode and wait for the cluster to stabilize.

```
smw# maintenance_mode_configure disable
smw# sleep 240
```

5. Check cluster status.

```
smw1# crm_mon -r1
```


Verify that the cluster has started all resources, including the newly added eLogin resources.

```
Stack: unknown
Current DC: smw1 (version unknown) - partition with quorum
Last updated: Fri Feb 9 14:38:17 2018
Last change: Fri Feb 9 06:28:13 2018 by hacluster via crmd on smw1

2 nodes configured
36 resources configured

Online: [ smw1 smw2 ]

Full list of resources:

ClusterIP      (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP1     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP2     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP3     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP4     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP5     (ocf::heartbeat:IPaddr2):      Started smw1
ClusterMonitor (ocf::smw:ClusterMonitor):      Started smw1
ClusterTimeSync (ocf::smw:ClusterTimeSync):    Started smw1
HSSDaemonMonitor (ocf::smw:HSSDaemonMonitor):   Started smw1
<snip>
Resource Group: SystemGroup
  NFSServer    (systemd:nfsserver):           Started smw1
  EnableRsyslog (ocf::smw:EnableRsyslog):      Started smw1
  syslog.socket (systemd:syslog.socket):       Started smw1
Clone Set: clo_PostgreSQL [PostgreSQL]
  Started: [ smw1 smw2 ]
ClusterIP6      (ocf::heartbeat:IPaddr2):      Started smw1
ClusterIP7      (ocf::heartbeat:IPaddr2):      Started smw1
esd              (systemd:esd.service):        Started smw1
```

If all of the cluster resources are not running as expected, please refer to *XC™ Series SMW HA Administration Guide (S-2551)*.

4.3.13 Provision eLogin Nodes

Prerequisites

- The node registry must have these required fields for each node, and they must be set as described:
 - The BMC username and BMC password must match what has been set on the hardware node BIOS firmware.
 - The value of BOOTIF must be the same as the node's Ethernet interface that is connected to the external-management-net network.
 - The CLE config set assigned to the node must have been created.
 - The storage profile assigned to the node must exist in the CLE config set.
 - The BOOT partition in the storage profile assigned to the node should match the value of ROOTDEV for the node.
 - The image assigned to the node must exist in SquashFS format.

- If the PE profile in `cray_image_binding` is enabled for the node, then the PE image must exist in SquashFS format.

About this task

The first time that an eLogin node is to be booted from the SMW, it must be PXE booted. The PXE boot process will provision the node with new information:

- Transfer the kernel and `initrd` to the node.
- Transfer X.509 certificates and SSH keys to the node.
- Prepare local storage on the node and make file systems from the storage profile assigned to the node.
- Transfer new global and CLE config sets to the node.
- Transfer the operating system image to the node.
- Transfer the PE image to the node if the PE profile is enabled for the node.
- Prepare GRUB2 on the node for a disk boot.

Procedure

1. Confirm the node has been assigned the information required for booting the node.

Note that the BMC password is not displayed by the `enode list` command.

Show the most commonly changed fields in the node registry.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
elogin1 p0 elogin_default elogin - 10.6.0.1 10.7.0.1 11:22:33:44:55:66 - UNKNOWN
```

Show ALL fields in the node registry.

```
smw# enode list --fields all
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE ROOTDEV BOOTIF PCI REMCON BMC_USERNAME ENABLE_KDUMP KDUMP_HIGH
KDUMP_LOW
elogin1 p0 elogin_default elogin - 10.6.0.1 10.7.0.1 90:B1:1C:3A:0A:1B -
UNKNOWN /dev/sda eth0 pci=bfsort ttyS1,115200n8 root False - -
```

2. PXE boot a single node.

- a. Begin the PXE boot.

```
smw# enode boot --pxe elogin1
```

- b. Start ConMan in another window to interact with the console terminal.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

Trouble? If the text is garbled, there may be a problem with the REMCON setting for the node, such as a bad baud rate. Or there may be a BIOS communication issue that requires a connection to the iDRAC via another method, such as using a web browser on the SMW to the BMC IP address of the node, or connecting to the node with a physical monitor, keyboard, and mouse.

- c. Check the node status as it powers on and begins the PXE boot.

```
smw# enode status elogin1
```

A successful boot will take several minutes. During that time, check the status of the node several times to ensure it is making state transitions during the PXE booting process. For more information about possible states, see [About the External State Daemon and eLogin Node States](#) on page 32. A successful boot will end with the node in the `node_up` state.

When a successful PXE boot had completed, the node is in the `node_up` state, which means it is ready for users to log in and do work. The next boot of the node can be a disk boot.

To complete this initial deployment or migration of eLogin, continue to the next procedure: [Modify Config Set or Node Registry After Initial PXE Boot](#) on page 107

4.3.14 Modify Config Set or Node Registry After Initial PXE Boot

Prerequisites

The eLogin node has completed an initial PXE boot to provision all of the local storage on the node.

About this task

The initial boot of an SMW-managed eLogin node must be a PXE boot. During this boot, the persistent storage (typically `/dev/sdb`) is partitioned for the file systems that will be on this disk.

After the initial PXE boot has completed, one of the following must be done:

- If the eLogin node will continue to use the same storage profile, then the CLE config set must be updated to `set persist_on_boot: true` for the disk with persistent storage in that profile.
- If the eLogin node was assigned a storage profile created specifically for a first PXE boot, then the node registry must be updated to assign that node a different storage profile (one that has `persist_on_boot: true` set for the disk with persistent storage) for subsequent boots.

This will ensure that subsequent boots do not remove and re-create the disk partitions and file systems on that disk.

Procedure

1. Identify which disk contains the persistent storage.

Search the CLE config set for the mount point `/var/opt/cray/persistent` to determine which disk has that file system. Use one of the following commands.

- To find the persistent storage device for a particular storage profile:

```
smw# cfgset search -t /var/opt/cray/persistent p0 | grep elogin_default
cray_storage.settings.storage_profiles.data.elogin_default.layouts./dev/
sdb.partitions.PERSISTENT.mount_point: /var/opt/cray/persistent
```

The output shows that persistent storage is on `/dev/sdb` in the `elgin_default` storage profile in the CLE config set `p0`.

- To find the persistent storage device for all storage profiles in a config set:

```
smw# cfgset search -t mount_point -s cray_storage p0 | grep PERSISTENT
cray_storage.settings.storage_profiles.data.elogin_default.layouts./dev/
sdb.partitions.PERSISTENT.mount_point: /var/opt/cray/persistent
cray_storage.settings.storage_profiles.data.elogin_default_first_boot.layouts./dev/
sdb.partitions.PERSISTENT.mount_point: /var/opt/cray/persistent
cray_storage.settings.storage_profiles.data.elogin_gpf.layouts./dev/
sdb.partitions.PERSISTENT.mount_point: /var/opt/cray/persistent
```

The output shows that persistent storage is on /dev/sdb in both the `elogin_default` and `elogin_gpf` storage profiles in the CLE config set `p0`.

UPDATE CLE CONFIG SET

Do the steps in this section only if the eLogin node will continue to use the same storage profile as it was booted with.

- (Conditional) If the eLogin node did not use a storage profile specific to a first boot, then change the `persist_on_boot` setting for the disk that contains the persistent storage for the storage profile used to boot the node.

Set `persist_on_boot` to `true` and confirm it has been set.

```
smw# cfgset modify -s true \  
cray_storage.settings.storage_profiles.data.elogin_default.layouts./dev/sdb.persist_on_boot p0
```

```
smw# cfgset get \  
cray_storage.settings.storage_profiles.data.elogin_default.layouts./dev/sdb.persist_on_boot p0  
true
```

To change this setting for a CLE config set other than `p0`, a storage profile other than `elogin_default`, or a disk other than `/dev/sdb`, use the same commands with the appropriate substitutions.

- (Conditional) If the config set was modified, update the config set.

The command in the previous step modified the config set without running pre- and post-configuration scripts, so that config set was marked invalid. This step uses `cfgset update` in prepare mode (no user interaction) to ensure that all configuration scripts are run.

```
smw# cfgset update -m prepare p0
```

- Push the updated config set to the eLogin node.

If the eLogin node will be PXE booted or staged right after this config set update, it is unnecessary to push the config set to the node. Otherwise, do this step to avoid the possibility of booting the node from disk before it sees this change, which would result in the disk being wiped and any data stored there being lost.

In this example, CLE config set `p0` is pushed to an eLogin node named `elogin1`.

```
smw# cfgset push -d elogin1 p0
```

Skip the rest of this procedure.

UPDATE NODE REGISTRY

Do the steps in this section only if the eLogin node was assigned a storage profile created specifically for a first PXE boot.

- (Conditional) If the eLogin node was assigned a storage profile specific to a first boot, then update the node registry for that node.

This example assumes that a first-boot storage profile was created in an earlier procedure and the default storage profile was not changed, so that the `persist_on_boot` flag on `/dev/sdb` is still set to `true` in the default profile.

To check the value of that flag in the `eloin_default` profile:

```
smw# cfgset get \  
cray_storage.settings.storage_profiles.data.eloin_default.layouts./dev/sdb.persist_on_boot p0  
true
```

In this example, an eLogin node named `eloin1` was assigned the first-boot profile `eloin_default_first_boot` prior to boot, and now that node will be assigned the `eloin_default` profile so that `/dev/sdb` will be persistent through subsequent boots.

```
smw# enode update --set-storage_profile eloin_default eloin1
```

The initial deployment of SMW-managed eLogin or migration to SMW-managed eLogin is now complete.

5 Manage eLogin Nodes

- Manipulate the node registry on the SMW
 - [Register eLogin Nodes](#) on page 110
 - [Enroll eLogin Nodes](#) on page 114
 - [List eLogin Nodes](#) on page 115
 - [Delete eLogin Nodes](#) on page 116
 - [Update eLogin Nodes](#) on page 117
- Manage node life cycle
 - [Check eLogin Node Status](#) on page 122
 - [Boot eLogin Nodes](#) on page 123
 - [Shutdown eLogin Nodes](#) on page 126
 - [Reboot eLogin Nodes](#) on page 127
- Create eLogin images
 - [Create an eLogin Image](#) on page 99
 - [Export an eLogin Image](#) on page 100
 - [Create eLogin Images with imgbuilder](#) on page 133
- Deploy images to eLogin nodes
 - [Push eLogin Image Root to eLogin Node](#) on page 135
 - [Push PE Image Root to eLogin Node](#) on page 136
- Transfer the config set and apply changes to eLogin nodes
 - [Push Config Set to eLogin Node](#) on page 137
 - [Run cray-ansible on eLogin Node](#) on page 138

5.1 Manipulate eLogin Node Registry

5.1.1 Register eLogin Nodes

Prerequisites

SWM/CLE software is installed.

Access to a booted eLogin node is required to determine the proper values for `kdump_low` and `kdump_high`. These values are only required when `kdump` is enabled.

About this task

The `enode create` command is used to register a single eLogin node. The `enode enroll` command is used to create more than one node at a time when migrating data from a CMC/eLogin `inventory.csv` file (see [Enroll eLogin Nodes](#) on page 114).

Any parameters not set when the node is created can be set or changed later with `enode update` (see [Update eLogin Nodes](#) on page 117).

NOTICE: After issuing an `enode create` or `enode update` command, wait at least 5 seconds before issuing an `enode boot` or `enode reboot` command. This delay ensures that modified data in the node registry in memory has been written to the data store on the SMW's disk. Ignoring this short delay may lead to a failed attempt to PXE boot a node.

The following parameters are available for creating an eLogin node.

Table 13. `enode create` Parameters

Command Line Option	Description
<code>-node_type</code>	Specify node type (may also be set by <code>ENODE_DEFAULT_NODE_TYPE</code> environment variable)
<code>-g, --group</code>	Set <code>esd</code> group(s)
<code>-c, --configset</code>	Set CLE config set
<code>--storage_profile</code>	Set <code>storage_profile</code> from the CLE config set
<code>-i, --image</code>	Set boot image
<code>-b, --bmc_ip</code>	Set <code>bmc_ip</code>
<code>-u, --bmc_username</code>	Set <code>bmc_username</code>
<code>-p, --bmc_password</code>	File containing node's <code>bmc_password</code>
<code>-m, --mgmt_ip</code>	Set <code>mgmt_ip</code>
<code>-a, --mgmt_mac</code>	Set <code>mgmt_mac</code>
<code>--rootdev</code>	Set root device for selected nodes
<code>--bootif</code>	Set boot interface for selected nodes
<code>--remcon</code>	Set remote console for selected nodes
<code>--pci</code>	Set PCI kernel parameter for selected nodes
<code>-d, --kdump_enable</code>	Enable <code>kdump</code> functionality. Requires <code>kdump_high</code> and <code>kdump_low</code>
<code>--kdump_high</code>	Set <code>kdump</code> high memory kernel parameter value
<code>--kdump_low</code>	Set <code>kdump</code> low memory kernel parameter value

Command Line Option	Description
<code>-k, --parameters</code>	Set space-delimited kernel parameter key or key-value pairs
<code>--ssh_host_keys</code>	Set how SSH host keys are handled. Defaults to <code>simple_sync</code>

Procedure

Minimum Parameters for BIOS Boot

1. Create a node with the minimum number of parameters for a BIOS boot.

- a. Create the node.

The BMC password must be supplied. If `--bmc_password /path/to/passwordfile` is not on the command line, then there will be an interactive prompt to enter the password for the BMC of the node.

```
smw# enode create --node_type elogin --bmc_ip 10.6.0.1 --mgmt_ip 10.7.0.1 \
--bmc_password /root/bmc_password --mgmt_mac 11:22:33:44:55:66 elogin1
Creating the following node:
eloin1
Successfully created ['eloin1'].
```

- b. If the node does not have the administrative BMC user set to be `root`, add the `--bmc_user` parameter.

```
smw# enode update --set-bmc_user root elogin1
Updating the following node:
eloin1
Successfully updated ['eloin1'].
```

- c. If the console device is not on `/dev/ttyS1` or the baud rate of the console is not 115,200, change the `remcon` parameter.

```
smw# enode update --set-remcon ttyS1,115200n8 elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

This is enough information to communicate with the node for a BIOS boot. The node will have to be augmented with more parameters using `enode update` before it has enough information to be able to PXE boot.

Minimum Parameters for PXE Boot

2. Create a node using minimum number of parameters for a PXE boot.

```
smw# enode create --node_type elogin --bmc_ip 10.6.0.1 --mgmt_ip 10.7.0.1 \
--bmc_password /root/bmc_password --mgmt_mac 11:22:33:44:55:66 \
--configset p0 --storage_profile elogin_default \
--image elogin-smw-large_cle_6.0.UP06-build6.0.6055_sles_12sp3-created20171120
Creating the following node:
eloin1
Successfully created ['eloin1'].
```


- a. If the node does not have the administrative BMC user set to be `root`, add the `--bmc_user` parameter.

```
smw# enode update --set-bmc_user root elogin1
Updating the following node:
elogin1
Successfully updated ['elogin1'].
```

- b. If the management interface on the node is not `eth0`, change the `bootif` parameter.

The default value is `eth0` is correct for a 4x1GbE LOM network adapter. Set this to `eth2` for a node with a 2x10GbE+2x1GbE LOM network adapter.

```
smw# enode update --set-bootif eth2 elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- c. If the boot disk for the node is not the first disk, change the `rootdev` parameter.

```
smw# enode update --set-rootdev /dev/sdb elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- d. If the console device is not on `/dev/ttyS1` or the baud rate of the console is not 115,200, change the `remcon` parameter.

```
smw# enode update --set-remcon ttyS1,115200n8 elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- e. If the PCI options for the node is not `bfsort`, change the `pci` parameter.

```
smw# enode update --set-pci bfsort elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

- f. If this site wishes to use a non-default method of handling SSH host keys on this node, then change the value of `ssh_host_keys`.

If `ssh_host_keys` is set to `simple_sync` (the default), then `esd` will use the SSH host keys from the eLogin node's config set. If the value is `generate`, then `esd` will generate new SSH host keys. If the value is an absolute path, then `esd` will use the site-supplied SSH host keys from that location.

```
smw# enode update --set_ssh_host_keys generate elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

If setting this to a value other than `simple_sync`, also ensure that:

- SSH host keys are not present in the Simple Sync directory structure for this node. If present, the PXE boot will fail.
- Automatic generation of SSH host keys has been disabled in `cray_ssh` (set `simple_ssh_keys` to `false`).

All Possible Parameters

3. Create a node using all possible parameters.

```
smw# enode create --node_type elogin --bmc_ip 10.6.0.1 --mgmt_ip 10.7.0.1 \
--bmc_user root --bmc_password /root/bmc_password --mgmt_mac 11:22:33:44:55:66 \
--configset p0 --storage_profile elogin_default \
--image elogin-smw-large_cle 6.0.UP06-build6.0.6055_sles_12sp3-created20171120 \
--bootif eth0 --rootdev /dev/sda --remcon ttyS1,115200n8 --pci bfsort \
--group elogin --parameters DEBUG=true elogin1 --ssh_host_keys generate \
--kdump_enable --kdump_high=512M --kdump_low=256M
Creating the following node:
elogin1
Successfully created ['elogin1'].
```

`DEBUG=true` is an example of a kernel parameter. This flag will stop the boot at predefined breakpoints in the dracut scripts during the booting process. It is not recommended for production use but may aid in troubleshooting. See [Boot the eLogin Node with the DEBUG Shell](#) on page 200.

4. Add a group to any node already created to allow a group of nodes to be operated upon by the other `enode` commands.

```
smw# enode update --set-group elogin elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
```

5.1.2 Enroll eLogin Nodes

Prerequisites

SWM/CLE software is installed.

About this task

The `enode enroll` command will create nodes from an `inventory.csv` file (as used on the CMC). The data used from the `inventory.csv` file is the node name, `bmc_ip`, and `mgmt_mac`. All remaining data for these nodes must be added using the `enode update` subcommand.

Procedure

1. Enroll several node from an `inventory.csv` file.

This inventory file from a CMC contains only one eLogin node (`elogin1`), but several can be enrolled at one time.

```
smw# cat inventory.csv
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUs, ARCH, RAM_MB, DISK_GB, NODE_DESC
elogin1,10.148.0.1,11:22:33:44:55:66,32,x86_64,131072,550,elogin1
elogin2,10.148.0.2,11:22:33:44:55:77,32,x86_64,131072,550,elogin2
```

- a. Edit the file to move the `bmc_ip` to the external-ipmi-net network range.

Since the network address range for the external-ipmi-net is 10.6.0.0/16, this file should be edited to move the `bmc_ip` from the 10.148.0.0/16 network to to 10.6.0.0/16 network.

```
smw# vi inventory.csv
smw# cat inventory.csv
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUs, ARCH, RAM_MB, DISK_GB, NODE_DESC
ellogin1,10.6.0.1,11:22:33:44:55:66,32,x86_64,131072,550,ellogin1
ellogin2,10.6.0.2,11:22:33:44:55:77,32,x86_64,131072,550,ellogin2
```

```
smw# enode enroll inventory.csv
Creating the following node:
ellogin1
ellogin2
Successfully created ['ellogin1','ellogin2'].
```

2. Add the missing required fields for each node in the `inventory.csv` file.

```
smw# enode update --set-mgmt_ip 10.7.0.1 --
bmc_password /root/bmc_password ellogin1
Updating the following node(s):
ellogin1
Successfully updated ['ellogin1']
smw# enode update --set-mgmt_ip 10.7.0.2 --
bmc_password /root/bmc_password ellogin2
Updating the following node(s):
ellogin2
Successfully updated ['ellogin2']
```

3. List the nodes in the registry to confirm the data entered.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_
↳MAC PARAMETERS STATE
ellogin1 - - - - 10.6.0.1 10.7.0.1
↳11:22:33:44:55:66 - UNKNOWN
ellogin2 - - - - 10.6.0.2 10.7.0.2
↳11:22:33:44:55:77 - UNKNOWN
```

Add any other needed information about the node with `enode update`, see [Update eLogin Nodes](#) on page 117.

5.1.3 List eLogin Nodes

Prerequisites

One or more nodes are in the node registry (see [Register eLogin Nodes](#) on page 110).

About this task

The `enode list` command can display any information about nodes in the node registry (except the `bmc_password`).

Procedure

1. Display the default fields from the node registry.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
elogin1 p0 elogin_default elogin - 10.6.0.1 10.7.0.111:22:33:44:55:66 - UNKNOWN
```

2. Display the non-default fields in the node registry.

```
smw# enode list --fields
name,bmc_username,rootdev,bootif,remcon,pci,enable_kdump,kdump_high,kdump_low
NAME BMC_USERNAME ROOTDEV BOOTIF REMCON PCI ENABLE_KDUMP KDUMP_HIGH KDUMP_LOW
elogin1 root /dev/sda eth0 ttyS1,115200n8 pci=bfsort False - -
```

3. Display all of the fields in the node registry.

```
smw# enode list --fields all
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
↳ PARAMETERS STATE ROOTDEV BOOTIF PCI REMCON
↳ BMC_USERNAME KDUMP_ENABLE KDUMP_HIGH KDUMP_LOW
elogin1 p0 elogin_default elogin - 10.6.0.1 10.7.0.1
↳ 90:B1:1C:3A:0A:1B - UNKNOWN /dev/sda eth0 pci=bfsort ttyS1,
↳ 115200n8 root False - -
```

5.1.4 Delete eLogin Nodes

Prerequisites

- One or more nodes are in the node registry.
- Nodes must be in `node_off` state or node deletion will fail.

About this task

Nodes can be removed from the node registry using `enode delete`.

The following changes are made on the SMW when a node is deleted:

- The node host name and IP address on the external-management-net network are removed from the `/etc/hosts` file.
- The X.509 certificates and SSH host keys are removed.
- The DHCP configuration for the node is removed.
- The `/opt/tftpboot/external` directory structure for the node is removed.

No files are removed from the storage on the node. Deleting the node does not affect the current status of the node. So if the node is up and running, it will continue to run even after it has been deleted.

Procedure

1. Delete a single node.

```
smw# enode delete elogin1
Deleting the following node(s):
elogin1
Successfully deleted ['elogin1']
```

`enode delete` will not delete nodes from the registry unless they are in the `node_off` state. `enode delete` can be forced with the `-f` or `--force` option.

2. Delete a list of nodes.

- a. Check the nodes to be deleted.

```
smw# enode list --filter esd_group=ellogin --fields name,esd_group
NAME ESD_GROUP
ellogin1 ellogin
ellogin2 ellogin
ellogin3 ellogin
```

- b. Delete the nodes.

```
smw# enode delete ellogin1 ellogin2 ellogin3
Deleting the following node(s):
ellogin1
ellogin2
ellogin3
Successfully deleted ['ellogin1','ellogin2','ellogin3']
```

`enode delete` will not delete nodes from the registry unless they are in the `node_off` state. `enode delete` can be forced with the `-f` or `--force` option.

5.1.5 Update eLogin Nodes

Prerequisites

- Nodes must already be in the node registry (see [Register eLogin Nodes](#) on page 110)

About this task

The eLogin nodes to be managed by the SMW must be added to the eLogin node registry. The information in the node registry for previously created nodes can be manipulated by the `enode update` command with command-line options for the node registry fields.

To make configuration changes to eLogin nodes, use `enode update` (instead of `enode delete` and `enode create`).

If changing the image, config set, kernel options, or other fields, the node can be staged so that the changes take effect on the next disk boot. For more information, see [Stage an eLogin Node](#) on page 134.

The following fields can be updated after the node creation with `enode update`. All fields must be accurate for a successful boot.

Table 14. Fields that Can be Modified with `enode update`

Field	<code>enode update</code> Option	Definition
group	<code>--set-group</code> or <code>-g</code>	ESD group to assign the node. <code>group</code> may be a single value or a comma-separated list of group names.

Field	enode update Option	Definition
	--unset-group or -G	Clears group(s) from the node.
configset	--set-configset or -c	CLE config set to assign the node.
	--unset-configset or -C	Clear CLE config set for the node.
storage_profile	--set-storage_profile	Storage profile in the CLE config set describing the disk setup of the node.
	--unset-storage_profile	Clear storage profile for the node.
image	--set-image or -i	Boot image to assign the node.
	--unset-image or -I	Clear boot image from the node.
bmc_ip	--set-bmc_ip or -b	IP address of the baseboard management controller (BMC)
	--unset-bmc_ip or -B	Restores bmc_ip to default value (0.0.0.0)
bmc_username	--set-bmc_user or -u	User name to connect to the BMC
	--unset-bmc_user or -U	Resets bmc_username to default value (root)
bmc_password	--set-bmc_password or -p	Name of a file containing the password used to connect to the BMC or a value to prompt for the password (if left as an empty value, the user is prompted for a the new password)
mgmt_ip	--set-mgmt_ip or -m	IP address of the boot interface, the eLogin interface on the external-management-net network
	--unset-mgmt_ip or -M	Restores mgmt_ip to default value (0.0.0.0)
mgmt_mac	--set-mgmt_mac or -a	MAC address of the boot interface, the eLogin interface on the external-management-net network
	--unset-mgmt_mac or -A	Restores mgmt_mac to default value (00:00:00:00:00:00)
rootdev	--set-rootdev	Sets the root storage device (default value is the first disk)
	--unset-rootdev	Resets rootdev to default value (first disk).

Field	enode update Option	Definition
bootif	--set-bootif	Management/boot interface. The default value is <code>eth0</code> , which is correct for a 4x1GbE LOM network adapter (if this eLogin node has a 2x10GbE+2x1GbE LOM network adapter instead, then change this value to <code>eth2</code>)
	--unset-bootif	Resets <code>bootif</code> to the default value (<code>eth0</code>)
remcon	--set-remcon	Remote console device setting for the path and baud rate of the console.
	--unset-remcom	Resets <code>remcon</code> to the default value (<code>/dev/ttyS1,115200</code>) for the path and baud rate of the console
pci	--set-pci	PCI kernel parameter. The default value is <code>bfsort</code>
	--unset-pci	Resets <code>pci</code> to the default value (<code>bfsort</code>)
kdump_enable	<code>kdump_enable</code>	Enables the kdump functionality on the node
	--unset-kdump_enable	Disables the kdump functionality on the node
kdump_high	--set-kdump_high	High memory value for use with kdump (for example, setting this to 256G will result in the kernel parameter <code>crashkernel=256G,high</code> . This is set on the kernel parameter line only when <code>enable_kdump</code> is set)
	--unset-kdump_high	Unsets the <code>kdump_high</code> parameter for the node
kdump_low	--set-kdump_low	Low memory value for use with kdump (for example, setting this to 4G will result in the kernel parameter <code>crashkernel=4G,low</code>). This is only set on the kernel parameter line when <code>enable_kdump</code> is set

Field	enode update Option	Definition
	<code>--unset-kdump_low</code>	Unsets the <code>kdump_low</code> parameter for the node
ssh_host_keys	<code>--set-ssh_host_keys</code>	Set how SSH host keys are handled. The default value is <code>simple_sync</code>
	<code>--unset-ssh_host_keys</code>	Restore SSH host key handling to the default (<code>simple_sync</code>)
parameter	<code>--set-parameter</code> or <code>-k</code>	String of kernel parameters to assign the node
	<code>--unset-parameter</code> or <code>-K</code>	Removes kernel parameters from the node



CAUTION: After issuing an `enode create` or `enode update` command, wait at least 5 seconds before issuing an `enode boot` or `enode reboot` command. This delay ensures that modified data in the node registry in memory has been written to the data store on the SMW's disk. Ignoring this short delay may lead to a failed attempt to PXE boot a node.

Procedure

- List the nodes in the registry to confirm the data entered.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
eloin1 - - - 10.6.0.1 10.7.0.1 11:22:33:44:55:66 - UNKNOWN
```

Note that the `configset`, `storage_profile`, `group`, and `image` fields do not yet have values.

- Update fields required for booting the nodes.
 - Set the CLE config set that should be applied to this node.

```
smw# enode update --set-configset p0 eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- Set the storage profile to be one of the profiles specified in the `cray_storage` config service of the CLE config set that was just assigned to this node.

```
smw# enode update --set-storage_profile eloin_default eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- Update the `image` field to use a new image.


```
smw# enode update --set-image image_name elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

Repeat this step for each eLogin node in this system.

3. Update other fields for the node, if their defaults are not correct.

- a. If the management interface on the node is not correct, then change the value of `bootif`.

The default value of `eth0` is correct for the 4x-1GbE LOM device. If this device is 2x-10GbE / 2x-1GbE LOM, then set `bootif` to `eth2`.

```
smw# enode update --set-bootif eth2 elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- b. If the boot disk for the node is not the first disk, then change the value of `rootdev`.

```
smw# enode update --set-rootdev /dev/sdb elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- c. If the console device is not on `/dev/ttyS1` or the baud rate of the console is not 115,200, then change the value of `remcon`.

```
smw# enode update --set-remcon ttyS1,115200n8 elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- d. If the BMC username for the node is not `root`, then change the value of `bmc_username`.

```
smw# enode update --set-bmc_user root elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- e. If the PCI kernel parameter for the node is not `bfsort`, then change the `pci` parameter.

```
smw# enode update --set-pci bfsort elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- f. If this site wishes to use a non-default method of handling SSH host keys on this node, then change the value of `ssh_host_keys`.

If `ssh_host_keys` is set to `simple_sync` (the default), then `esd` will use the SSH host keys from the eLogin node's config set. If the value is `generate`, then `esd` will generate new SSH host keys. If the value is an absolute path, then `esd` will use the site-supplied SSH host keys from that location.

```
smw# enode update --set-ssh_host_keys generate elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

If setting this to a value other than `simple_sync`, also ensure that:

- SSH host keys are not present in the Simple Sync directory structure for this node. If present, the PXE boot will fail.
- Automatic generation of SSH host keys has been disabled in `cray_ssh` (set `simple_ssh_keys` to `false`).

4. Set the group to assign to a node.

```
smw# enode update --set-group elogin elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

5. To update `kdump_enable`, `kdump_high`, and `kdump_low` fields, go to [Enable and Start kdump](#) on page 190.

6. Update the `parameter` field to enable or disable debug.

See [Boot the eLogin Node with the DEBUG Shell](#) on page 200 for more information.

```
smw# enode update --set-parameter parameters elogin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

7. List the nodes in the registry to confirm the data entered.

Note that the BMC password is not displayed by the `enode list` command.

Show the most commonly changed fields in the node registry.

```
smw# enode list
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE
eloin1 p0 eloin1_default elogin - 10.6.0.1 10.7.0.1 11:22:33:44:55:66 - UNKNOWN
```

Show all fields in the node registry.

```
smw# enode list --fields all
NAME CONFIGSET STORAGE_PROFILE ESD_GROUP IMAGE BMC_IP MGMT_IP MGMT_MAC
PARAMETERS STATE ROOTDEV BOOTIF PCI REMCON BMC_USERNAME ENABLE_KDUMP KDUMP_HIGH
KDUMP_LOW SSH_HOST_KEYS
eloin1 p0 eloin1_default elogin - 10.6.0.1 10.7.0.1 90:B1:1C:3A:0A:1B -
UNKNOWN /dev/sda eth0 pci=bfsort ttyS1,115200n8 root False - - generate
```

If the image, config set, kernel options, or other fields were changed, the node can be staged so that the changes take effect on the next disk boot. For more information, see [Stage an eLogin Node](#) on page 134.

5.2 Manage eLogin Node Life Cycle

5.2.1 Check eLogin Node Status

Prerequisites

The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 110)

About this task

Even if the node is powered down, `enode status` will still attempt to ping the target using the IP address on the external-management-net network. This may cause the command to appear to stall for several seconds while it waits for a ping command to timeout.



CAUTION:

Procedure

1. Check the status for a single node.

```
smw# enode status elogin1
NODE PING POWER STATE
elogin1 DOWN Chassis Power is off node_off
```

If the chassis power is `off` and the PING status is `UP`, this may indicate that the management IP address intended for the node to use is already in use by another node.

If the bmc cannot be contacted, the command will pause for a long time (about 30 seconds) and return an unknown power state. This may indicate a network issue or that the ip/username/password are misconfigured for the bmc.

2. Check the status for a list of nodes.

```
smw# enode status elogin1 elogin2 elogin3
NODE PING POWER STATE
elogin1 DOWN Chassis Power is off node_off
elogin2 UP Chassis Power is on node_on
elogin3 DOWN Chassis Power is on status_wait
```

3. Use the `-l` or `--long` option to view a list of errors.

```
smw# enode status --long elogin1
NODE PING POWER STATE
  elogin1 Down Chassis Power is off Error
  State prior to error: node_off
  elogin1 ERROR 1: Failed to transition. Self state: node_off desired state:
node_off; Error: exception: Failed to transition to state: node_off, Next state
node_off not valid from current state node_off
  elogin1 ERROR 2: Encountered an error during shutdown
```

5.2.2 Boot eLogin Nodes

Prerequisites

- Node registry has required fields for each node (see [Register eLogin Nodes](#) on page 110)

- CLE config set has been created
- Storage profile for each node exists in the CLE config set
- Image assigned to the node exists in SquashFS format
- If the PE profile in `cray_image-binding` is enabled for the node and PE image exists in SquashFS format
- Node(s) being booted is in `node_off` state
- The node hardware BMC device is configured
- The network connections between the SMW and the node for the `external-ipmi-net` and `external-managementnet` networks are cabled
- Two physical or virtual disks are available on the node for a PXE boot to transfer information to the node's local persistent storage

About this task

eLogin nodes are booted with the `enode boot` command. There are different ways to boot:

- BIOS boot
 - can boot a single node or group of nodes
 - adjust BIOS settings, system settings, iDRAC settings, or other device settings for network or RAID configuration
- PXE boot
 - can boot single node or group of nodes
 - provisions new config sets, image, and PE image to the node
- Boot from disk
 - can boot single node or group of nodes
 - default boot method

The state of the node will change during the booting process. The state can be checked with the `enode status` command (see [Check eLogin Node Status](#) on page 122).

If the node is not in the `node_off` state, use the `enode shutdown` command to turn the node off (see [Shutdown eLogin Nodes](#) on page 126).

After a PXE boot or a BIOS boot is issued to the node's BMC, the `next_boot` setting in the BMC is adjusted so that it will boot from disk the next time it boots.

Procedure

—————BIOS BOOT—————

1. Boot a single node to BIOS.

This gives the opportunity to interact with the console of the node to adjust BIOS settings, system settings, iDRAC settings, or other device settings for network or RAID configuration.



CAUTION: Upon exiting BIOS, the node will proceed with the boot order specified in BIOS. If PXE appears before disk boot, the node may be reprovisioned (resulting in data loss) and put into an error state.

```
smw# enode boot --bios elogin1
```

- a. Start ConMan in another window to interact with the console terminal.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it powers on and begins the BIOS initialization.

```
smw# enode status elogin1
```

```
—————PXE BOOT—————
```

2. Boot a single node to begin the PXE boot process.

This will use PXE boot to:

- transfer the kernel and initrd to the node
- transfer X.509 certificates and SSH keys to the node
- prepare local storage on the node and make file systems from the storage profile assigned to the node
- push new global and CLE config sets to the node
- transfer the operating system image to the node
- push the PE image to the node if the PE profile is enabled for the node

```
smw# enode boot --pxe elogin1
```

- a. Start ConMan in another window to interact with the console terminal.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it powers on and begins the PXE boot.

```
smw# enode status elogin1
```

3. Boot a group of nodes to begin PXE boot process.

This will use PXE boot to:

- transfer the kernel and initrd to the node
- transfer X.509 certificates and SSH keys to the node
- prepare local storage on the node and make filesystems from the storage profile assigned to the node
- push new global and CLE config sets to the node

- transfer the operating system image to the node
 - push the PE image to the node if the PE profile is enabled for the node
- a. Boot using a list of nodes.

```
smw# enode boot --pxe elogin1 elogin2 elogin3
```

- b. Check the status on the nodes as they power on and begin the PXE boot.
This will show the stage of the boot and any errors that occur.

```
smw# enode status elogin1 elogin2 elogin3
```

```
—————BOOT FROM DISK—————
```

4. Boot a single node from disk.

```
smw# enode boot elogin1
```

This is the default type of boot. The following example shows the `--disk` command line option, which is optional.

```
smw# enode boot --disk elogin1
```

- a. Start ConMan in another window to interact with the console terminal.
This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it powers on and begins the PXE boot.

```
smw# enode status elogin1
```

5. Boot a group of nodes from disk.

- a. Boot using a list of nodes.

```
smw# enode boot elogin1 elogin2 elogin3
```

- b. Check the status on the nodes as they power on and begin to boot.
This will show the stage of the boot and any errors that occur.

```
smw# enode status elogin1 elogin2 elogin3
```

If the node is shutting down and an `enode boot` command is issued, the command will be rejected.

5.2.3 Shutdown eLogin Nodes

Prerequisites

- Node registry has required fields for each node.

About this task

`enode shutdown` will first attempt a graceful shutdown of the operating system. If the node does not respond within the timeout period, the power will be turned off for the node.

Procedure

1. Shut down a single node.

```
smw# enode shutdown elogin1
```

- a. Start ConMan in another window to watch the console messages as the node shuts down.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

- b. Check the node status as it shuts down.

```
smw# enode status elogin1
```

Any error state on the node is cleared and the node is set to `node_off`.

2. Shut down a group of nodes.

```
smw# enode shutdown elogin1 elogin2 elogin3
```

- a. Start ConMan in another window to watch the console messages as the node shuts down.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

- b. Check the node status as it shuts down.

```
smw# enode status elogin1 elogin2 elogin3
```

Any error state on the node is cleared and the node is set to `node_off`.

Special cases:

- If the node is in the process of shutting down and a reboot command is issued, the node will power back on and proceed with the type of boot specified by the reboot command.
- If the node is in the process of shutting down from either a shutdown command or a reboot command, and multiple reboot commands are issued, the last reboot command will specify if the boot type is BIOS, PXE or disk.
- If the node is shutting down and a boot command is issued, the command will be rejected.

5.2.4 Reboot eLogin Nodes

Prerequisites

- Node registry has required fields for each node
- CLE config set has been created
- Storage profile for each node exists in the CLE config set
- Image assigned to the node exists in SquashFS format
- If the PE profile in `cray_image-binding` is enabled for the node, then the PE image must exist in SquashFS format

About this task

eLogin nodes are rebooted with the `enode reboot` command. There are different ways to reboot:

- BIOS Reboot
 - can reboot single node or group of nodes
 - adjust BIOS settings, system settings, iDRAC settings, or other device settings for network or RAID configuration
- PXE Reboot
 - can reboot single node or group of nodes
 - provisions new config sets, image, and PE image to the node
- Reboot from Disk
 - can reboot single node or group of nodes
 - default reboot method

A staged reboot is also supported, but the behavior is different than the reboot methods described here. For information about the staged reboot, see [Stage an eLogin Node](#) on page 134.

`enode reboot` will first attempt a graceful shutdown of the operating system. If the node does not respond within the timeout period, the power will be turned off for the node. Then the node will begin the booting process using the selected reboot method (disk, PXE, staged, BIOS).

Procedure

—————BIOS REBOOT—————

1. Reboot a single node to BIOS.

This gives the opportunity to interact with the console of the node to adjust BIOS settings, system settings, iDRAC settings, or other device settings for network or RAID configuration.



CAUTION: Upon exiting BIOS, the node will proceed with the boot order specified in BIOS. If PXE appears before disk boot, the node may be reprovisioned (resulting in data loss) and put into an error state.

```
smw# enode reboot --bios elogin1
```


- a. Start ConMan in another window to interact with the console terminal.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it shuts down, powers on, and begins the BIOS boot.

```
smw# enode status elogin1
```

```
—————PXE REBOOT—————
```

2. Reboot a single node to begin the PXE boot process.

This will use PXE boot to:

- transfer the kernel and initrd to the node
- transfer X.509 certificates and SSH keys to the node
- prepare local storage on the node and make file systems from the storage profile assigned to the node
- transfer new global and CLE config sets to the node
- transfer the operating system image to the node
- transfer the PE image to the node if the PE profile is enabled for the node

```
smw# enode reboot --pxe elogin1
```

- a. Start ConMan in another window to interact with the console terminal.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it shuts down, powers on, and begins the PXE boot.

```
smw# enode status elogin1
```

3. Reboot a group of nodes to begin PXE boot process.

This will use PXE boot to:

- transfer the kernel and initrd to the node
- transfer X.509 certificates and SSH keys to the node
- prepare local storage on the node and make filesystems from the storage profile assigned to the node
- transfer new global and CLE config sets to the node
- transfer the operating system image to the node

- transfer the PE image to the node if the PE profile is enabled for the node
- a. Reboot using a list of nodes.

```
smw# enode reboot --pxe elogin1 elogin2 elogin3
```

- b. Check the status on the nodes as they power on and begin the PXE boot.

```
smw# enode status elogin1 elogin2 elogin3
```

```
—————REBOOT FROM DISK—————
```

4. Reboot a single node from disk.

```
smw# enode reboot elogin1
```

This is the default type of reboot. The following example shows the `--disk` command line option, which is optional.

```
smw# enode reboot --disk elogin1
```

- a. Start ConMan in another window to interact with the console terminal.

This is an optional step used to monitor the boot.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the `remcon` setting for the node with a bad baud rate. Or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

- b. Check the node status as it shuts down, powers on, and begins the PXE boot.

```
smw# enode status elogin1
```

5. Reboot a group of nodes from disk.

- a. Reboot using a list of nodes.

```
smw# enode reboot elogin1 elogin2 elogin3
```

- b. Check the status on the nodes as they shut down, power on, and begin to boot.

```
smw# enode status elogin1 elogin2 elogin3
```

Special cases:

- If the node is in the process of booting, the `enode reboot` command will attempt to shut the node down and reboot it.
- If the node is in the process of shutting down and an `enode reboot` command is issued, the node will power back on and proceed with the type of boot specified by the `enode reboot` command.
- If the node is in the process of shutting down from either an `enode shutdown` command or an `enode reboot` command, and multiple `enode reboot` commands are issued, the last `enode reboot` command will specify if the boot type is BIOS, PXE or disk.

- If the node is off and an `enode reboot` command is issued, it will proceed as if a boot command was issued. This will clear errors.

5.3 Create eLogin Images

5.3.1 Create an eLogin Image

Prerequisites

SMW/CLE software is installed and configured.

About this task

This procedure creates images for the eLogin nodes using the `image create` command. To create images using `imgbuilder`, see [Create eLogin Images with imgbuilder](#) on page 133. If using `imgbuilder`, the recipes for eLogin nodes can be used to create image roots and export the image roots into the proper format for booting at the same time as other recipes are used to build images.

The recipe used to build a boot image for an eLogin node should be closely matched to the recipe used to build internal login and compute node boot images.

- If the internal nodes are using `tmpfs` recipes, which do not have "large" as part of their name (e.g., `login_cle_6.0up06_sles_12sp3_ari` and `compute_cle_6.0up06_sles_12sp3_ari`), then choose the smaller eLogin recipe: `eloin-smw_cle_6.0up06_sles_12sp3_ari`.
- If the internal nodes are using `netroot` recipes, which have "large" as part of their name (e.g., `login-large_cle_6.0up06_sles_12sp3_ari` and `compute-large_cle_6.0up06_sles_12sp3_ari`), then choose the larger eLogin recipe: `eloin-smw-large_cle_6.0up06_sles_12sp3_ari`.

Procedure

1. Select an eLogin recipe.

There are two types of recipes, the "eloin-smw" recipe and the "eloin-smw-large" recipe. This documentation uses the "eloin-smw-large" recipe in all examples.

Use the "eloin-smw" recipe only if:

- the compute nodes and login nodes are using `tmpfs` images ("login" and "compute") instead of `netroot` images ("login-large" and "compute-large")
- there are specific size constraints for the eLogin image
- the image is intended for test purposes

2. **Optional:** Create a custom eLogin recipe if additional packages are required.

- a. Create a new image recipe with a custom name (using the "custom" prefix).

```
smw# recipe create custom-eloin-smw-large_cle_6.0up06_sles_12sp3_ari
```

- b. Add `elogin-smw-large_cle_6.0up06_sles_12sp3_ari` as a sub-recipe.

```
smw# recipe update -i elogin-smw-large_cle_6.0up06_sles_12sp3_ari \
custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari
```

- c. Add any additional packages, package collections, `postbuild_copy`, or `postbuild_chroot` information to this custom recipe before building an image root from it.

See "Install Third-Party Software with a Custom Image Recipe" procedure (under "Modify an Installed System section) in the *XC™ Series System Administration Guide*.

3. Build the eLogin image.

For custom recipes:

```
smw# image create -r custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari \
custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

For "elogin-smw" recipes:

```
smw# image create -r elogin-smw-large_cle_6.0up06_sles_12sp3_ari \
elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

Cray recommends appending a date stamp, such as "YYYYMMDD", to images created with `image create`. For example, if generating `elogin-smw-large_cle_6.0up01_sles_12_ari` on June 1st, 2017, the image should be named `elogin-smw-large_cle_6.0up01_sles_12_ari_20170601`. If `imgbuilder` is used to create the image, it will add date stamps automatically.

5.3.2 Export an eLogin Image

Prerequisites

- SMW/CLE software is installed and configured.
- An eLogin image root has been created from a recipe.

About this task

The `image export` command has a `format` option to produce a SquashFS image from an image root. Two SquashFS images are needed for the eLogin node, both the operating system image and the PE image with Programming Environment (PE) software.

Procedure

Export the eLogin image root into a SquashFS-formatted image for booting the eLogin node.

```
smw# image export --format squashfs \
custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari-YYYYMMDD
```

This image name in this example uses the naming convention suggested in the [Create an eLogin Image](#) on page 99 procedure.

This command creates a directory under `/var/opt/cray/imps/boot_images` with the name of the image, and it creates a SquashFS file and an `.imps_Image_metadata` file in that directory.

5.3.3 Create eLogin Images with imgbuilder

Prerequisites

- SMW/CLE software is installed and configured
- Connection to the SMW via `ssh`

About this task

The `imgbuilder` command builds the recipes into image roots and then optionally exports them to boot images of the desired format. For CLE nodes the format for boot images is `cpio`. For eLogin nodes the format for boot images is SquashFS.

The older format in this file for CLE nodes where the `dest` parameter ended in `.cpio` should be changed to the newer format which includes the `export_format` parameter.

Old format for CLE boot images in `cray_image_groups.yaml`

```
- recipe: "admin_cle_6.0up06_sles_12sp3_x86-64_ari"
  dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}.cpio"
  nims_group: "admin"
```

New format for CLE boot images in `cray_image_groups.yaml` removes `.cpio` from `dest` and adds the new `export_format`:

```
- recipe: "admin_cle_6.0up06_sles_12sp3_ari"
  dest: "admin{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}"
  export_format: "cpio"
  nims_group: "admin"
```

New format for eLogin boot images in `cray_image_groups.yaml`:

The `eloin` image is a companion to `tmpfs` style images for CLE login and compute nodes.

```
- recipe: "eloin-smw_cle_6.0up06_sles_12sp3_ari"
  dest: "eloin-smw{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}"
  export_format: "squashfs"
```

The "eloin-large" image is a companion to `netroot` images for CLE login and compute nodes.

```
- recipe: "eloin-large-smw_cle_6.0up06_sles_12sp3_ari"
  dest: "eloin-smw-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}"
  export_format: "squashfs"
```

Procedure

1. Edit `cray_image_groups.yaml` to include eLogin images.

This step is needed for systems which were not freshly installed with SMW 8.0.UP06/CLE 6.0.UP06.

```
smw# vi /var/opt/cray/imps/config/sets/global/config/cray_image_groups.yaml
```

This example adds the "eloin-smw-large" recipe to the `default` group of recipes to build.

```
cray_image_groups:
  default:
  ...
  - recipe: "eloin-smw-large_cle_6.0up06_sles_12sp3_ari"
    dest: "eloin-smw-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}"
    export_format: "squashfs"
```

This example adds a customized "custom-elogin-smw-large" recipe to the `default` group of recipes to build.

```
cray_image_groups:
  default:
  ...
  - recipe: "custom-elogin-smw-large_cle_6.0up06_sles_12sp3_ari"
    dest: "custom-elogin-smw-large{note}_cle_{cle_release}-build{cle_build}{patch}_sles_12sp3-created{date}"
    export_format: "squashfs"
```

2. Run `imgbuilder` to create the desired set of images.

```
smw# imgbuilder --map
```

This will create an image with a name similar to "elogin-smw-large_cle_6.0.UP06-build6.0.6055_sles_12sp3-created20171120".

The `-map` option will cause `imgbuilder` to call the `cnode update` command to assign new CLE images to nodes in the NIMS map. There is no similar option to have `imgbuilder` assign images to eLogin nodes with `enode update`, so that must be done with a separate command.

3. Assign newly built image to the desired eLogin nodes.

When assigning an image to eLogin nodes, the full path to the image is not required, nor is any SquashFS suffix, just the name of the image root that was built by `imgbuilder` from the recipe.

```
smw# enode update -i \  
elogin-smw-large_cle_6.0.UP06-build6.0.6055_sles_12sp3-created20171120 \  
elogin1 elogin2
```

5.4 Stage an eLogin Node

Prerequisites

- SMW/CLE software is installed and configured
- eLogin nodes have been configured with `enode` and the CLE config set
- Image root for eLogin node has been prepared from an eLogin recipe
- All associated images to be pushed are exported in the SquashFS format
- The eLogin nodes are in the `node_up` state

About this task

Staging a node will make the following changes on the eLogin node:

- Push the current global config set to the eLogin node
- Push the eLogin node's config set to the eLogin node
- Push the eLogin node's image root to the eLogin node
- Push the eLogin node's PE image root to the eLogin node
- Update the GRUB boot entry to use the image kernel and initramfs, config set, and current kernel options

Each time a node is staged a new GRUB boot entry is added named with a timestamp. The new grub boot entry is set as the default. The grub entries and boot disk are cleared when the node is booted using the PXE method

Booting using an older GRUB boot entry will use the most recently staged config set.

Procedure

1. Connect to the SMW.

```
# ssh root@smw
```

2. Start staging the eLogin node.

`enode stage` stages the node(s) for next boot. `enode reboot --staged` stages the node(s) and immediately reboots the node(s). Below are four options for staging either single or multiple nodes with or without an automatic reboot.

- Stage a single node.

```
smw# enode stage elogin1
```

- Stage a node with an automatic reboot using the disk method.

```
smw# enode reboot --staged elogin1
```

Once rebooted, the node will be using the staged kernel, kernel options, and the staged configuration will be applied.

- Stage several nodes at once.

```
smw# enode stage elogin1 elogin2
```

- Stage several nodes with an automatic reboot.

```
smw# enode reboot --staged elogin1 elogin2
```

Once staging is complete, the next time the node is booted using the disk method, it will be using the latest staged image, config set, and kernel options.

The `enode stage` command will return once the nodes have started staging and the staging process will occur asynchronously. While the nodes are staging the nodes will be in the staging state. When staging is complete, the nodes will return to the `node_up` state. If an error occurs during staging, the node will go to the `Error` state.

If there are config set changes that need to be applied to the node after staging is complete, see [Run *cray-ansible* on eLogin Node](#) on page 138.

5.5 Deploy Images to eLogin Nodes

5.5.1 Push eLogin Image Root to eLogin Node

Prerequisites

- SMW/CLE software is installed

- eLogin nodes are configured with `enode` and in the CLE config set
- The image root for eLogin node is prepared from an eLogin recipe
- eLogin nodes are in the `node_up` state

About this task

ATTENTION: Cray recommends pushing the eLogin image root by staging a node (see [Stage an eLogin Node](#) on page 134). `enode stage` sends the eLogin image root, PE images, global config sets, and CLE config sets to the node. Staging also updates the kernel boot parameters in a new GRUB boot entry.

The image root containing the operating system will be transferred to the eLogin node during a PXE boot, but can also be transferred to a booted node.

Procedure

1. Connect to the SMW.

```
# ssh root@smw
```

2. Push the eLogin image root to the eLogin node.

The eLogin image root is built on the SMW installation and is also cached on the eLogin node for accessibility in the circumstance where the SMW is not available.

- a. Push the image root to a single eLogin node.

```
smw# image push -d elogin1 elogin_image
```

- b. Push the image root to multiple nodes.

Pushing to several eLogin nodes can be done with a single command which uses the CLE config set `CLE_config_set`, the node group within that CLE config set `node_group`, and the name of the image. The node group `elogin_nodes` is normally defined to be all eLogin nodes related to a CLE config set.

```
smw# image push -r CLE_config_set -g node_group elogin_image
```

The estimated time to complete this process is about 5 minutes, depending on the size of the eLogin image root and the speed of the networking link between the SMW and the eLogin node.

5.5.2 Push PE Image Root to eLogin Node

Prerequisites

- SMW/CLE software is installed and configured
- eLogin nodes are configured with `enode` and in the CLE config set
- PE software is installed into a PE image root

About this task

ATTENTION: Cray recommends pushing the eLogin image root by staging a node (see [Stage an eLogin Node](#) on page 134). `enode stage` sends the eLogin image root, PE images, global config sets, and CLE config sets to the node. Staging also updates the kernel boot parameters in a new GRUB boot entry.

The Programming Environment (PE) software is installed into a PE image root on the SMW following directions in the *XC™ Series Software Installation and Configuration Guide* or the *Cray® Programming Environments Installation Guide*. There are frequent releases of the PE software which may be added to the PE image root for a given CLE release. This procedure shows how to push a newly updated PE image root to an eLogin node without requiring a PXE boot of the node to transfer the content.

Procedure

1. Connect to the SMW.

```
# ssh root@smw
```

2. Push the PE image root to the eLogin node.

The PE image root is prepared during the installation of PE software and is also cached on the eLogin node for accessibility in the circumstance when the SMW is not available.

- a. Push the image root to a single eLogin node.

```
smw# image push --format squashfs --prefer-existing -d elogin1 PE_image
```

- b. Push the image root to multiple nodes.

Pushing to several eLogin nodes can be done with a single command which uses the CLE config set `CLE_config_set`, the node group within that CLE config set `node_group`, and the name of the image. The node group `eloin1_nodes` is normally defined to be all eLogin nodes related to a CLE config set.

```
smw# image push --format squashfs --prefer-existing -r CLE_config_set -g
node_group PE_image
```

The estimated time to complete this process is at least 10 minutes, depending on the size of the image root and the speed of the networking link between the SMW and the eLogin node.

3. Run `cray-ansible` on the eLogin node.

```
smw# ssh elogin1 /etc/init.d/cray-ansible start
```

This will rerun the `cray_image_binding` play which makes the new PE image available for use.

5.6 Transfer Config Set and Apply Changes to eLogin Nodes

5.6.1 Push Config Set to eLogin Node

Prerequisites

- SMW/CLE software is installed and configured.
- eLogin node has been configured with `enode` and is in the CLE config set.
- eLogin node is in the `node_up` state

About this task

This procedure is for manually pushing the config set an eLogin node. The config set is automatically pushed to the eLogin node during a staged boot. For more information on staging a node, see [Stage an eLogin Node](#) on page 134.

Procedure

1. Connect to the SMW node.

```
# ssh root@smw
```

2. Push the config set to the eLogin node.

The config set was generated during CLE installation and then modified in the "Update the Config Set for eLogin" procedure in the *XC™ Series SMW-managed eLogin Installation Guide*.

```
smw# cfgset push -d elogin1 global
smw# cfgset push -d elogin1 config_set
```

- a. Push the config set to multiple nodes.

Pushing to several eLogin nodes can be done with a single command which uses the CLE config set `config_set`, the node group within that CLE config set `node_group`, and the name of the config set being pushed. When the config set being pushed is a CLE config set, the node group can be found within that CLE config set.

```
smw# cfgset push -r ref_config_set -g node_group global
smw# cfgset push -g node_group config_set
```

If the changes in the config set need to be applied to the node immediately, see [Run cray-ansible on eLogin Node](#) on page 138.

5.6.2 Run cray-ansible on eLogin Node

Prerequisites

- SMW/CLE software is installed and configured
- eLogin nodes have been configured with `enode` and in the CLE config set
- The global config set or CLE config set has been pushed to the eLogin node after some change has been made

About this task

When the global config set or CLE config set has been changed and pushed to an eLogin node, to have those changes take effect on the node either the node needs to be rebooted or `cray-ansible` needs to be run on the node.

Procedure

1. Connect to the SMW node.

```
# ssh root@smw
```

2. Run `cray-ansible` on the eLogin nodes.

There are two ways to run `cray-ansible` on the eLogin nodes.

- a. Run `cray-ansible` directly on node.

```
smw# ssh elogin  
elogin# /etc/init.d/cray-ansible start
```

- b. Run `cray-ansible` on several nodes with `pdsh -w`.

```
smw# module load pdsh  
smw# pdsh -w elogin[1-3] /etc/init.d/cray-ansible start
```

6 eLogin Configuration Options

This section includes optional configuration procedures that can be done after the initial eLogin deployment.

- [Update eLogin Firmware](#) on page 140
- [Configure Log Rotation for eLogin Logs on the SMW](#) on page 149
- [Change the Firewall Configuration](#) on page 151
- [Manage Partitions and Persistent Data on an eLogin Node](#) on page 155
- [Enable LiveUpdates Support for eLogin Nodes](#) on page 162
- Optional network configurations
 - [Configure an IPv4 Interface to Include IPv6 Address](#) on page 166
 - [Configure Tagged VLANs for eLogin](#) on page 168
 - [Configure Bonded Interfaces for eLogin](#) on page 170
 - [Configure Bonded VLAN Interfaces for eLogin](#) on page 173
- [Configure SSDs on eLogin Nodes](#) on page 177

6.1 Update eLogin Firmware

Prerequisites

Connection to the Internet.

About this task

This procedure checks for the latest Dell firmware (BIOS, NIC, iDRAC, RAID, etc.) and updates firmware for an eLogin node. The steps show examples for a Dell R710 servers and Dell iDRAC6 Enterprise. Depending on the version of server and iDRAC, there may be minor step differences for the eLogin nodes at this site.

Procedure

1. From a web browser, navigate to the IP of the iDRAC for this eLogin node.
For example, `https://my-elogin-idrac.us.cray.com`.
2. Log in with `root` credentials.

Figure 46. Dell R710 / iDRAC6: Integrated Dell Remote Access Controller Login Screen

DELL INTEGRATED DELL REMOTE
ACCESS CONTROLLER 6 - ENTERPRISE

Login ?

Type in Username and Password, and then click Submit.

Username: **Password:**

Domain:

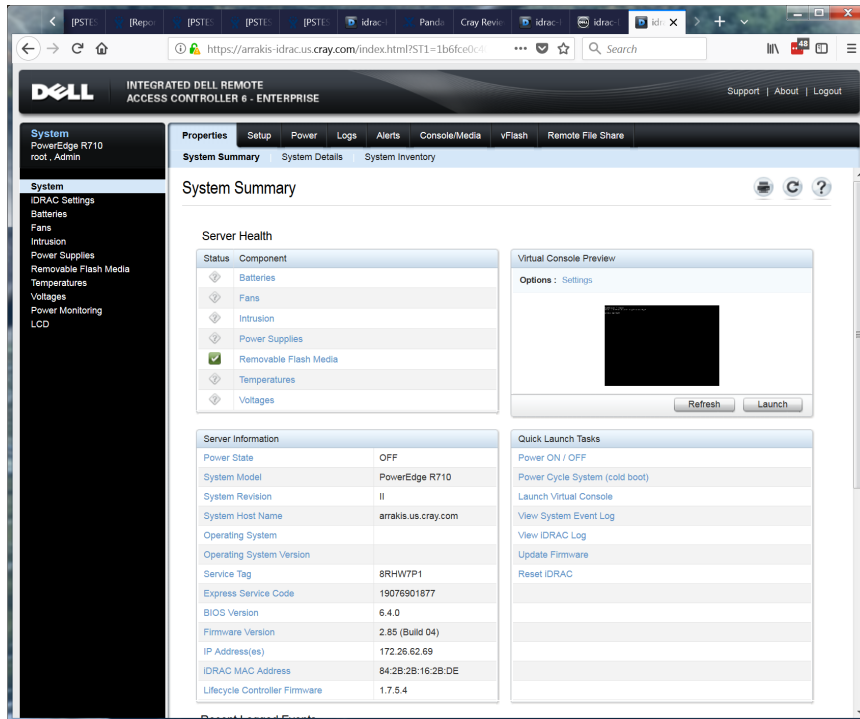
Support | About

Note that it is normal to see one or more security warnings appear at this point, advising that connection to this website is untrusted or that this application may be a security risk. Select **Continue** or **Run** to continue with this procedure.

3. Launch the virtual console.

On the iDRAC6 screen, in the **Properties > System Summary** tab, under **Quick Launch Tasks**, select **Launch Virtual Console**.

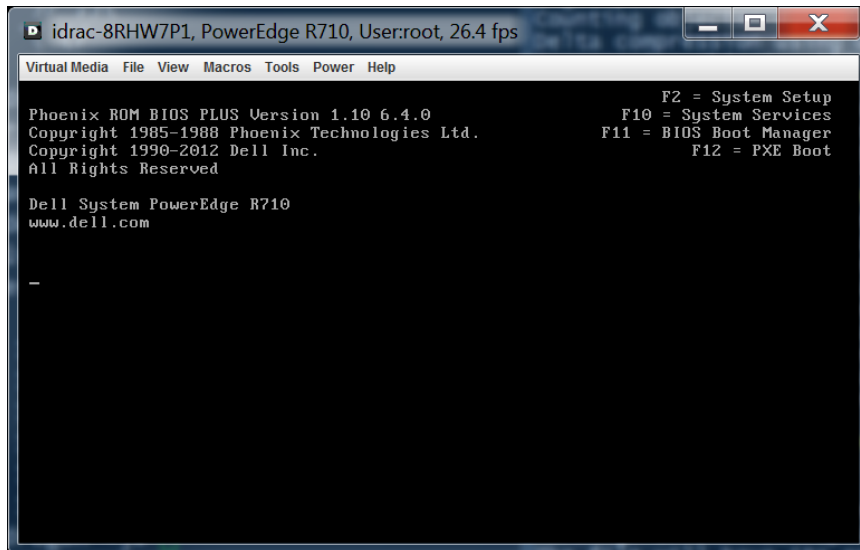
Figure 47. Dell R710 / iDRAC6: Launch Virtual Console



4. Power up or reboot the eLogin node.

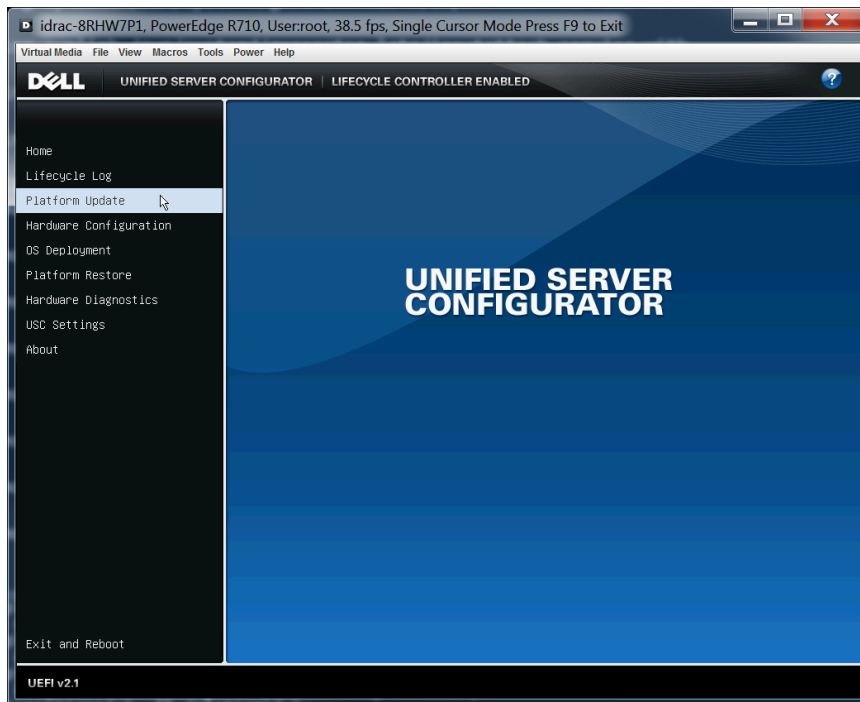
When the BIOS/POST screen appears, press **F10** to enter **System Services**.

Figure 48. Dell R710 / iDRAC6: BIOS Screen



Entering **System Services** starts up the Unified Server Configurator (USC), and eventually the **Unified Server Configurator** screen appears.

Figure 49. Dell R710 / iDRAC6: Unified Server Configurator

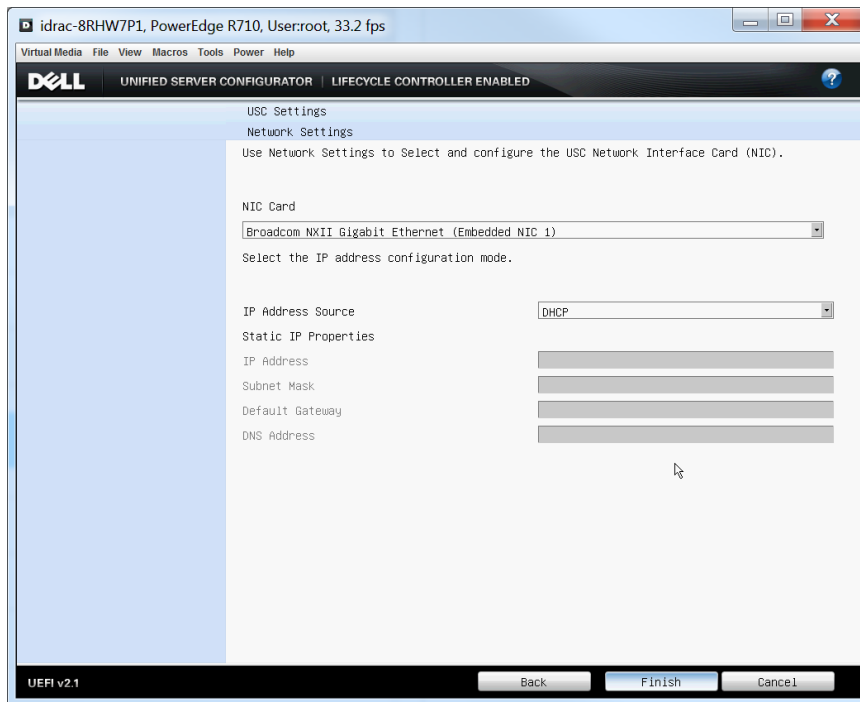


5. Check the network settings to ensure they are correct.

In the **Unified Server Configurator** screen, select **USC Settings**, and then on the **USC Settings** page, select **Network Settings**.

In the **Network Settings** screen, select the network interface card that has external access (NIC1 in the example). Set **IP Address Source** to **DHCP** or fill in the appropriate addresses under **Static IP Properties**. Select **Finish**.

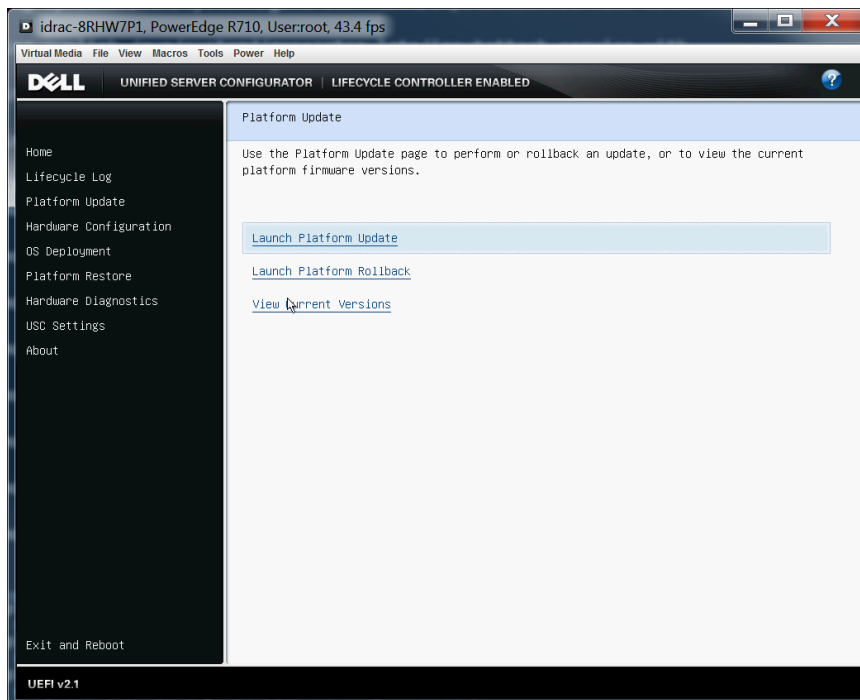
Figure 50. Dell R710 / iDRAC6: Network Settings



6. Launch the platform update.

In the **Unified Server Configurator** screen, select **Platform Update**, and then on the **Platform Update** page, select **Launch Platform Update**.

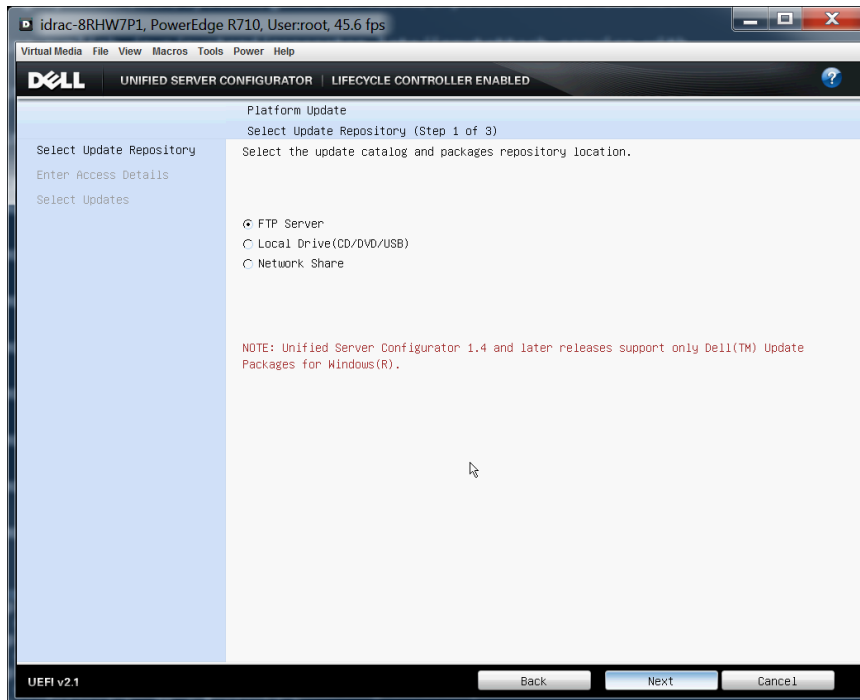
Figure 51. Dell R710 / iDRAC6: Platform Update



7. Select an update repository.

In the **Platform Update / Select Update Repository** screen, select **FTP Server** and then select **Next**.

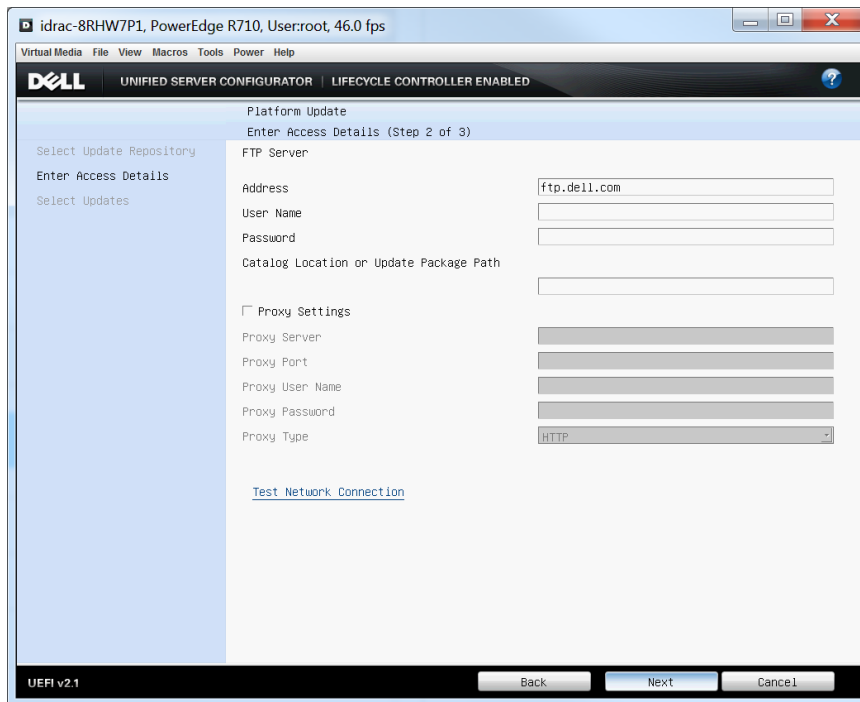
Figure 52. Dell R710 / iDRAC6: Platform Update / Select Update Repository



8. Enter access details.

In the **Platform Update / Enter Access Details** screen, select **Proxy Settings** and enter information in that section, if applicable. If using Dell FTP (ftp.dell.com), the **User Name**, **Password**, and **Update Package Path** can be left blank.

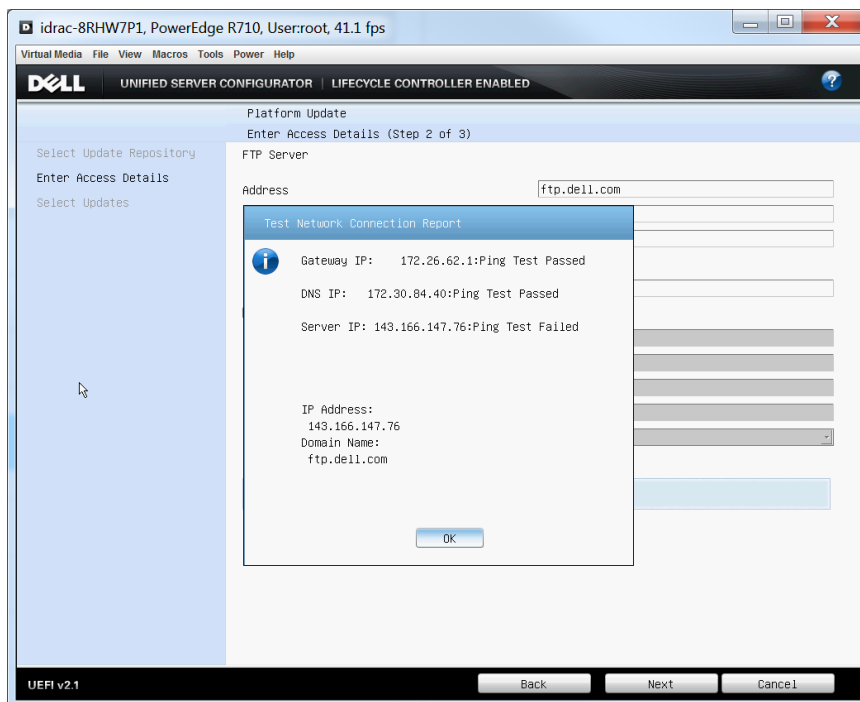
Figure 53. Dell R710 / iDRAC6: Platform Update / Enter Access Details



9. Check the network connection.

Click **Test Network Connection** near the bottom of the **Platform Update / Enter Access Details** screen.

Figure 54. Dell R710 / iDRAC6: Test Network Connection Report

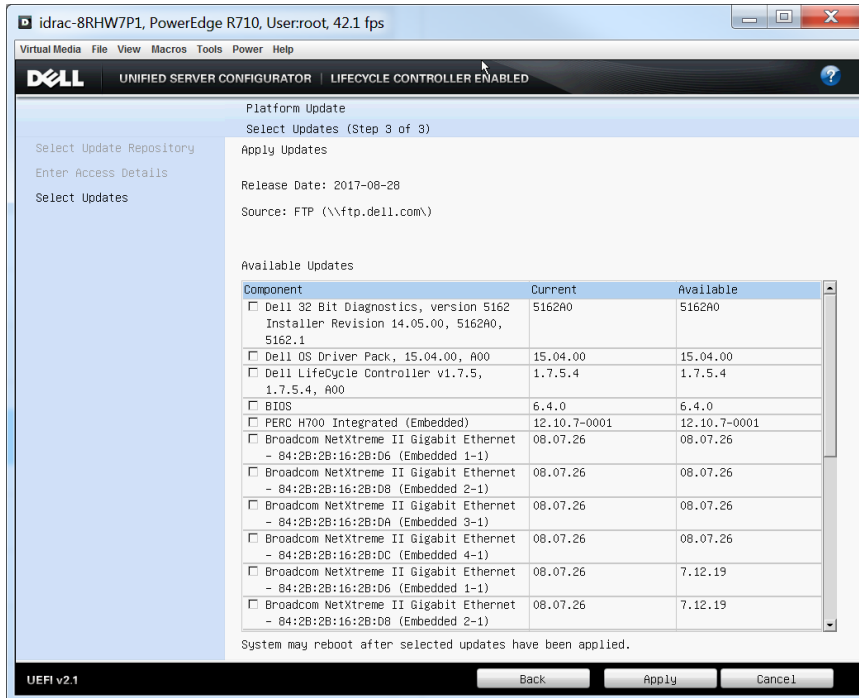


Select OK to dismiss the report, and then select **Next**.

10. Select updates.

When the catalog is finished downloading, the **Platform Update / Select Updates** screen appears. A comparison of the **Current** column and the **Available** column shows that the example eLogin node has the latest firmware installed, so no updates are necessary.

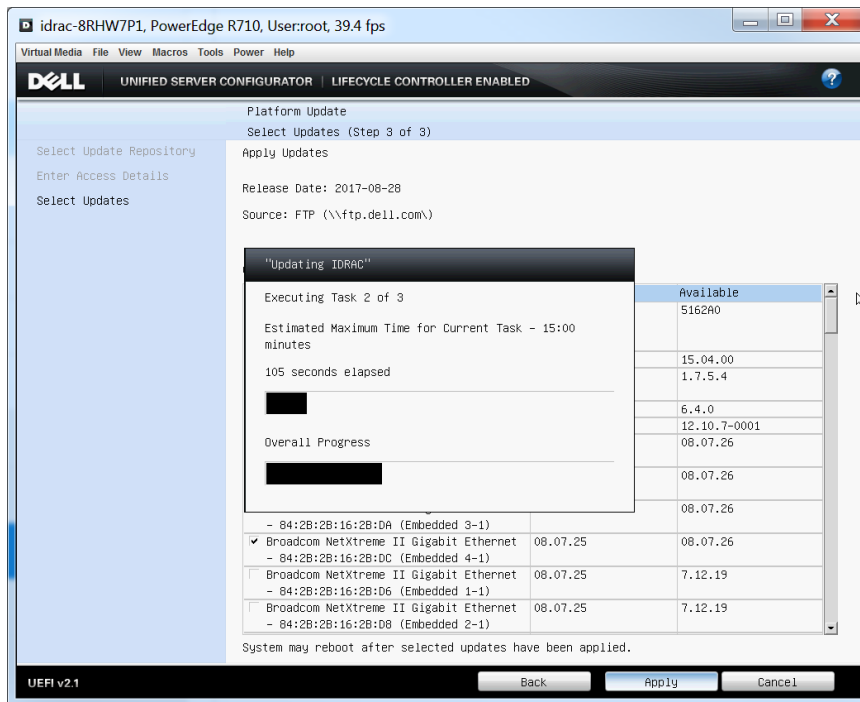
Figure 55. Dell R710 / iDRAC6: Platform Update / Select Updates



If the eLogin node at this site is not up to date, select the components that need to be updated, and then select **Apply**. The server will begin downloading and installing the firmware. It will reboot as necessary.

NOTE: If the iDRAC is one of the components being updated, the current session will be disconnected and it will be necessary to log in and launch the virtual console again (repeat steps 1 through 3) to proceed.

Figure 56. Dell R710 / iDRAC6: Apply Updates



When all firmware updates are complete, select **Cancel** in the **Platform Update / Select Updates** screen. This returns to the **Platform Update** page and the main menu of the Unified Server Configurator.

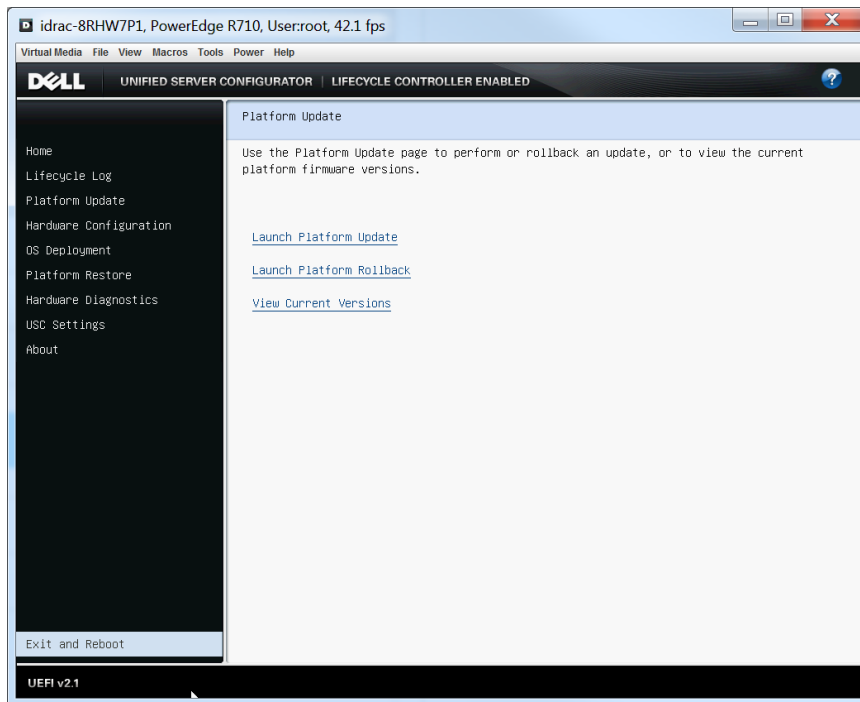
11. Verify platform updates.

To verify that all platform updates have been applied, return to the **Platform Update / Select Updates** screen. To do this, go back to step 6 on page 144. There is no need to re-enter settings, such as FTP and proxy data.

12. Exit the Unified Server Configurator and reboot the node.

Select **Exit and Reboot** from the main menu (at bottom left). Do NOT press **Ctrl-Alt-Delete**, because that will reboot back to **System Services**.

Figure 57. Dell R710 / iDRAC6: Exit from USC and Reboot



6.2 Configure Log Rotation for eLogin Logs on the SMW

Prerequisites

- eLogin is fully installed.
- The system is booted.

About this task

This procedure provides a set of `logrotate` and `xttrim` files to manage (rotate, compress, and delete) the eLogin log files on the SMW in a similar manner as LLM currently manages log files under `/var/opt/cray/log`.

Procedure

1. Configure `logrotate` files.
 - a. Create `/etc/logrotate-esd.conf` with the following contents.

```
# see "man logrotate" for details
# rotate log files weekly
# weekly

# keep 4 weeks worth of backlogs
# rotate 4
```

```

# create new (empty) log files after rotating old ones
# create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
# compress

# comment these to switch compression to use gzip or another
# compression scheme
compresscmd /usr/bin/gzip
uncompresscmd /usr/bin/gunzip

# RPM packages drop log rotation information into this directory
include /etc/logrotate-esd.d

```

- b. Create `/etc/logrotate-esd.d/esd` with the following contents.

This `/etc/logrotate-esd.d/esd` file rolls all the logs with a date stamp extension and sends a HUP (a signal hangup) to `command` to restart ConMan daemon logging. This config file handles `esd`, `esd-uwsgi`, `conman`, and all the node ConMan logs under `/var/opt/cray/log/external/conman`. This log rolling method is consistent with the way the nightly LLM log rolling occurs.

```

# Example logrotate file for esd, esd-uwsgi, enode and conman
#
# xtrim can be used to handle the compression and deletion of old log files.

/var/opt/cray/log/external/esd.log {
    daily
    dateext
    missingok
    copytruncate
    rotate 100
}

/var/opt/cray/log/external/esd-uwsgi.log {
    daily
    dateext
    missingok
    copytruncate
    rotate 100
}

/var/opt/cray/log/external/enode.log {
    daily
    dateext
    missingok
    copytruncate
    rotate 100
}

/var/opt/cray/log/external/conman/*.console {
    daily
    dateext
    missingok
    copytruncate
    rotate 100
}

```

```

/var/opt/cray/log/external/conman.log {
    daily
    dateext
    missingok
    rotate 100
    postrotate
        /usr/bin/killall -HUP conmand
    endscript
}

```

2. Add the following entries to the `/etc/opt/cray/llm/xttrim.conf` file.

This example specifies 30 days before log compression and 90 days before log deletion. Change those values as appropriate for this system. To disable log deletion, replace 90 with 0.

```

/var/opt/cray/log/external 30 90
/var/opt/cray/log/external/conman 30 90

```

3. Add the following crontab entry for the `root` user to schedule eLogin log rotation.

```

# esd log rotation via logrotate
0 0 * * * /usr/sbin/logrotate /etc/logrotate-esd.conf

```

This crontab entry may get overwritten during an SMW software update, and so it may need to be added again after the update.

4. After one day, check the eLogin log directory to make sure logs have been rotated as desired.

```

smw# ls -lR /var/opt/cray/log/external
.:
total 9712
drwxr-xr-x 2 root    root      4096 Jan  9 00:00 conman
-rw-r--r-- 1 root    root        567 Jan  8 08:01 conman.log-20180108
-rw-r--r-- 1 root    root        627 Jan  8 15:44 conman.log-20180109
-rw-r--r-- 1 crayadm crayadm      0 Jan  9 00:00 enode.log
-rw-r--r-- 1 crayadm crayadm   5446 Jan  8 08:01 enode.log-20180108
-rw-r--r-- 1 crayadm crayadm 117254 Jan  9 00:00 enode.log-20180109
-rw-r----- 1 root    root          0 Jan  9 00:00 esd-uwsgi.log
-rw-r----- 1 root    root  9483477 Jan  8 08:01 esd-uwsgi.log-20180108
-rw-r----- 1 root    root   21149 Jan  9 00:00 esd-uwsgi.log-20180109
-rw-r--r-- 1 root    root          0 Jan  9 00:00 esd.log
-rw-r--r-- 1 root    root    3692 Jan  8 08:01 esd.log-20180108
-rw-r--r-- 1 root    root  219666 Jan  9 00:00 esd.log-20180109
drwxr-xr-x 2 root    root        56 Dec 14 12:18 test-esd
-rw-r----- 1 root    root   69453 Jan  8 15:44 test-esd-uwsgi.log

```

6.3 Change the Firewall Configuration

Prerequisites

- SMW/CLE software is installed and configured.
- eLogin nodes are deployed.

About this task

The Cray firewall configurations services and Ansible plays are designed to make it unnecessary for site system administrators to change the SMW and eLogin firewall configuration. However, there are several basic changes a site may wish to make:

- Enable or disable a firewall by using the configurator to update the global or CLE `cray_firewall` configuration service.
- Change whether CLE and eLogin nodes inherit firewall settings from the SMW by using the configurator to update the CLE `cray_firewall` configuration service.
- Change the port that the external state daemon (`esd`) listens on by editing the `esd.ini` file.

This procedure provides examples of how to make these basic firewall changes on a booted system with eLogin nodes deployed. For general information about SMW and eLogin firewalls, see [About the Firewall for SMW and eLogin Nodes](#) on page 19.

Procedure

————— CHECK FIREWALL PORTS —————

1. Check firewall ports.

The NFS port in the firewall must be open so that eLogin nodes can NFS-mount the necessary file systems. Use the procedure in [Ensure that NFS Port is Open in Firewall](#) on page 70.

————— MAKE A BACKUP OF THE IPTABLES —————

2. Save iptables.

Cray recommends saving the iptables prior to changing the firewall configuration on the SMW or eLogin nodes.

```
smw# iptables-save > iptables-before-firewall-changes
```

```
eLogin# iptables-save > iptables-before-firewall-changes
```

————— CHANGE THE FIREWALL CONFIGURATION SERVICE —————

3. Enable/disable the firewall for the SMW.
 - a. Change the `cray_firewall.enabled` setting in the global config set.

To enable the firewall in the global config set:

```
smw# cfgset modify --set true cray_firewall.enabled global
smw# cfgset get cray_firewall.enabled global
```

To disable the firewall in the global config set:

```
smw# cfgset modify --set false cray_firewall.enabled global
smw# cfgset get cray_firewall.enabled global
```

- b. Update the global config set.

The previous substep modified the config set without running pre- and post-configuration scripts. This substep ensures that all configuration scripts are run.

```
smw# cfgset update -m prepare global
```

4. Enable/disable/inherit the firewall for all CLE and eLogin nodes.

Note that changes to the firewall configuration service in the CLE config set affect all internal CLE nodes and all eLogin nodes.

a. Change the `cray_firewall.enabled` setting in the CLE config set, if needed.

To enable the firewall in the CLE config set (p0 in the example):

```
smw# cfgset modify --set true cray_firewall.enabled p0
smw# cfgset get cray_firewall.enabled p0
```

To disable the firewall in the CLE config set (p0 in the example):

```
smw# cfgset modify --set false cray_firewall.enabled p0
smw# cfgset get cray_firewall.enabled p0
```

b. Change the `cray_firewall.inherit` setting in the CLE config set, if needed.

To set the firewall in the CLE config set (p0 in the example) to inherit from the firewall in the global config set:

```
smw# cfgset modify --set true cray_firewall.inherit p0
smw# cfgset get cray_firewall.inherit p0
```

To set the firewall in the CLE config set (p0 in the example) to not inherit from the firewall in the global config set:

```
smw# cfgset modify --set false cray_firewall.inherit p0
smw# cfgset get cray_firewall.inherit p0
```

c. Update the CLE config set.

The previous substeps modified the CLE config set without running pre- and post-configuration scripts. This substep ensures that all configuration scripts are run.

If a CLE config set other than p0 was modified, substitute the correct config set in this command.

```
smw# cfgset update -m prepare p0
```

————— APPLY FIREWALL CONFIGURATION SERVICE CHANGES —————

If the firewall configuration for a node is changed, the changes are applied at the next boot of the node.

To apply changes immediately, use one of the following steps.

5. Apply firewall config set changes immediately on the SMW.

A firewall config set change is one of the following: enable the firewall, disable the firewall, or set the CLE (and eLogin) firewall config settings to inherit from the global firewall config settings. To apply a firewall config set change on the SMW, run Ansible.

Run all Ansible plays (recommended):

```
smw# /etc/init.d/cray-ansible start
```

Or run only this eLogin-SMW firewall play:

```
smw# ansible-playbook -v /etc/ansible/elogin_smw_firewall.yaml
```

6. Apply firewall config set changes immediately on an eLogin node.

A firewall config set change is one of the following: enable the firewall, disable the firewall, or set the CLE (and eLogin) firewall config settings to inherit from the global firewall config settings. To apply a firewall config set change on an eLogin node, push the config set from the SMW and then run Ansible.

a. Push the config set to one or more eLogin nodes.

Push to a single eLogin node :

```
smw# cfgset push -d my_elogin p0
```

Or push to an eLogin node group:

```
smw# cfgset push -g my_elogin_nodes p0
```

b. Run Ansible plays on the eLogin node.

Run all Ansible plays on the eLogin node (recommended):

```
elogin# /etc/init.d/cray-ansible start
```

Or run only this eLogin-SMW firewall play on the eLogin node:

```
elogin# ansible-playbook -v /etc/ansible/elogin_smw_firewall.yaml
```

————— CHANGE THE FIREWALL PORT FOR ESD —————

7. Change the port on which `esd` listens.

The `esd` daemon listens for client nodes on the port specified in the `/etc/opt/cray/esd/esd.ini` file. That port is designated by the variable `esd_port` in that file, and the default value is 8449. Edit the file and change this value to have `esd` listen on a different port.

```
smw# vi /etc/opt/cray/esd/esd.ini
```

```
#
# Copyright 2017, Cray Inc. All Rights Reserved.
#
# esd.ini
#
# Initialization file for Cray External Node State Daemon (ESD).
#
...
[esd]
enode_port = 8448
enode_endpoint = /esd/v1/node

esd_port = 1234
```

————— APPLY FIREWALL PORT CHANGES —————

If the firewall port on which `esd` listens has been changed, `esd` must be restarted (if it was started prior to the change) and Ansible plays must be re-run on the SMW. To maintain contact with `esd`, the eLogin nodes must be rebooted, because the `esd` port is communicated to eLogin nodes through kernel parameters at boot time. If they are not rebooted, they will be unable to report status back to `esd` on the SMW, and if `esd` sends a request and gets no response, it may put the nodes into an error state.



CAUTION: If the port on which `esd` listens is changed, a booted eLogin node will be unable to communicate status to the `esd` daemon until it is rebooted with `enode reboot --pxe`.

8. Apply firewall `esd` port (the port on which `esd` listens) changes immediately.

a. Start or restart `esd`.

If `esd` was not started prior to the port change:

```
smw# systemctl start esd
```

If `esd` was started prior to the port change:

```
smw# systemctl restart esd
```

b. Run Ansible plays on the SMW.

Run all Ansible plays (recommended):

```
smw# /etc/init.d/cray-ansible start
```

Or run only this eLogin-SMW firewall play.

```
smw# ansible-playbook -v /etc/ansible/elogin_smw_firewall.yaml
```

c. Reboot all eLogin nodes.

Do a PXE reboot so that the kernel parameter indicating the new `esd` port is transferred from the SMW.

```
smw# enode reboot --pxe elogin1 elogin2 elogin3
```

————— VERIFY THE CHANGES —————

9. Verify the applied firewall changes.

After applying firewall changes on the SMW or any eLogin nodes, save the iptables again and compare with the previously saved iptables file to verify the changes had the desired effect.

On the SMW:

```
smw# iptables-save > iptables-after-firewall-changes
```

On an eLogin node:

```
elogin# iptables-save > iptables-after-firewall-changes
```

6.4 Manage Partitions and Persistent Data on an eLogin Node

Storage profiles define the disk layout and partition information for internal disks on eLogin nodes. The profiles are defined in the `cray_storage` service in the CLE config set that is assigned to each eLogin node. Storage profile changes are applied when the node is PXE booted or rebooted using `enode reboot --staged`. When necessary, storage profiles can be changed and applied on a running system. The following two procedures describe how to make and apply changes in the CLE config set. If it is necessary to change the configuration of virtual disk `sda` or `sdb`, see [Configure the eLogin RAID Virtual Disks](#) on page 91.

- Nonpersistent Disks

For nonpersistent disks (devices with `persist_on_boot: false`), ALL of the partitions are removed and re-created at boot time. The following changes are supported:

- Add or remove partitions
- Change partition size
- Change partition file system type
- Change partition ordering

- Persistent Disks

For persistent disks (devices with `persist_on_boot: true`), no partitions are removed and re-created at boot time. Only the following change is supported:

- Add partitions (only if the disk contains adequate space for the new partitions)

Partition size, partition file system type, and partition ordering cannot be changed as long as the `persist_on_boot` field remains set to `true`. Removal of partitions is also not supported on persistent disks.

To reprovision a nonpersistent disk, simply make the changes to the storage profile in the CLE config set assigned to that eLogin node and either PXE boot the node or reboot it with the `--staged` option.

To reprovision a persistent disk, it is necessary to first set it to nonpersistent, make any other storage profile changes, reboot the node with the new storage layout, then reset the disk to persistent. Note that ALL DATA WILL BE LOST in the process.



WARNING: To avoid loss of data when reprovisioning a persistent disk, move data to a safe location before rebooting the eLogin node.

6.4.1 Reprovision a Persistent Disk on an eLogin Node

Prerequisites

eLogin node is booted.

About this task

To reprovision a persistent disk with a new partition layout, that disk must be reconfigured as nonpersistent. This can be done by creating a new storage profile for that node with the desired layout and `persist_on_boot` set to `false`.

The new partition scheme will be created on the eLogin node after rebooting the node; however ALL DATA WILL BE LOST in the process. If data that resides on the disk needs to be retained, move the data to a safe location before rebooting the node, and copy it back after the node successfully provisions.

To make the disk persistent again, set `persist_on_boot: true` in the new storage profile after the node has rebooted, so that subsequent reboots do not repartition the disk and cause data loss.



WARNING: To avoid loss of data when reprovisioning a persistent disk, move data to a safe location before rebooting the eLogin node. After rebooting the node and restoring that data to the disk, ensure that the disk is reconfigured as persistent.

This procedure safely reprovisions a persistent disk on an eLogin node (`ellogin1` in the example commands).

Procedure

1. Copy data from the persistent disk to a safe location somewhere off that eLogin node.
2. Prepare configuration worksheets for editing.
 - a. Generate a set of configuration worksheets with the current CLE configuration data.

This example uses the existing CLE config set `p0`.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets
smw# cp *_worksheet.yaml /my/workarea
```

- c. Change to the new work area.

```
smw# cd /my/workarea
```

3. Edit the `cray_storage` configuration worksheet to add a storage profile.

```
smw# vi cray_storage_worksheet.yaml
```

4. Add a new storage profile.

Copy `ellogin_default` (or another storage profile with a layout similar to the desired layout), then change the persistent disk (device) to be nonpersistent and make other changes, as needed.

- a. Copy the storage profile.

In the worksheet, copy the default storage profile and paste it below this line:

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

- b. Replace the name (key) of the copied profile with the key for the new storage profile (`new_ellogin` in this example).

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

```

cray_storage.settings.storage_profiles.data.new_elogin: null
cray_storage.settings.storage_profiles.data.new_elogin.enabled: true

cray_storage.settings.storage_profiles.data.new_elogin.layouts.device./dev/sda: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.persist_on_boot: false

cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.GRUB: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.GRUB.type: ext3
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.GRUB.size: 1MiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.BOOT: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.BOOT.type: ext3
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.BOOT.size: 2GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.WRITELAYER: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.WRITELAYER.type: ext4
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.WRITELAYER.size: 20GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.TMP: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.TMP.type: xfs
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.TMP.size: 256GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.SWAP: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.SWAP.type: swap
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.SWAP.size: 128GiB
...

cray_storage.settings.storage_profiles.data.new_elogin.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.persist_on_boot: true

cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.label.CRASH: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.CRASH.type: ext4
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.CRASH.size: 10GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.label.PERSISTENT: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.PERSISTENT.type: xfs
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.PERSISTENT.size: ALL
...

```

c. Change the `persist_on_boot` flag to `false` for the `/dev/sdb` disk.

```

cray_storage.settings.storage_profiles.data.new_elogin.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.persist_on_boot: false

```

d. Make the desired changes to this storage profile.

Because the disk is temporarily nonpersistent, partitions can be added, removed, resized, reordered, or have their file system type changed. Make the desired reprovisioning changes now, bearing in mind the following requirements:

- To function properly, all eLogin nodes must have all of the following partitions with these exact labels:
 - nonpersistent disk: GRUB, BOOT, WRITELAYER, TMP, and SWAP
 - persistent disk: CRASH and PERSISTENT
- The sum of the sizes of all of the volatile data partitions on the first disk (`/dev/sda`) must be less than the available storage on the first disk. Similarly, the sum of the sizes of all of the persistent data partitions on the second disk (`/dev/sdb`) must be less than the available storage on the second disk.
- Two partitions have the following minimum size limits:
 - BOOT must be `> 1 GiB` (note binary value)
 - PERSISTENT must be `> 200 GiB` (note binary value)

If it is necessary to change the configuration of virtual disk `sda` or `sdb`, see [Configure the eLogin RAID Virtual Disks](#) on page 91.

For more information about binary values, see [Prefixes for Binary and Decimal Multiples](#) on page 206.

5. Upload modified `cray_storage` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_storage_worksheet.yaml' p0
```

6. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

7. Validate the config set.

```
smw# cfgset validate p0
```

8. Assign the new storage profile to the eLogin node.

```
smw# enode update --set-storage_profile new_elogin elogin1
```

9. Reboot the eLogin node.

This example reboots an eLogin node named *elogin1*.

```
smw# enode reboot --pxe elogin1
```

10. Verify the changes to the storage layout.

a. On the SMW, determine if the node is finished booting.

In this example, the eLogin node is *elogin1*.

```
smw# enode status elogin1
```

The eLogin node has finished booting if its status is `node_up`.

b. On the eLogin node, verify that the desired partitions exist with the expected sizes.

```
elogin# df
```

11. Change the formerly persistent disk, which was temporarily made nonpersistent, to be persistent again.

This example uses the *new_elogin* storage profile. Substitute the actual storage profile name for this system.

```
smw# cfgset modify --set true \  
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.persist_on_boot p0
```

12. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

13. Validate the config set.

```
smw# cfgset validate p0
```

14. Push the config set to the eLogin node.

```
smw# cfgset push -d elogin1 p0
```

15. Move the data copied from the persistent disk (in the first step) back to the eLogin node.

6.4.2 Reprovision a Nonpersistent Disk on an eLogin Node

Prerequisites

eLogin node is booted.

About this task

To reprovision a nonpersistent disk on an eLogin node, the disk must be reconfigured and the node rebooted (with either the `--pxe` or `--staged` option). To reconfigure the disk, either modify the storage profile in the CLE config set assigned to that eLogin node, or create a new storage profile in that config set and assign it to the node.

Procedure

1. Prepare configuration worksheets for editing.

- a. Generate a set of configuration worksheets with the current CLE configuration data.

This example uses the existing CLE config set `p0`.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets
smw# cp *_worksheet.yaml /my/workarea
```

- c. Change to the new work area.

```
smw# cd /my/workarea
```

2. Edit the `cray_storage` configuration worksheet to add or change a storage profile.

```
smw# vi cray_storage_worksheet.yaml
```

3. If changing an existing storage profile, make the desired changes to the nonpersistent disk of that profile.

Because the disk is nonpersistent, partitions can be added, removed, resized, reordered, or have their file system type changed. Make the desired reprovisioning changes now, bearing in mind the following requirements:

- To function properly, all eLogin nodes must have all of the following partitions with these exact labels:
 - nonpersistent disk: GRUB, BOOT, WRITELAYER, TMP, and SWAP
 - persistent disk: CRASH and PERSISTENT
- The sum of the sizes of all of the volatile data partitions on the first disk (`/dev/sda`) must be less than the available storage on the first disk. Similarly, the sum of the sizes of all of the persistent data partitions on the second disk (`/dev/sdb`) must be less than the available storage on the second disk.

- Two partitions have the following minimum size limits:
 - BOOT must be > 1 GiB (note binary value)
 - PERSISTENT must be > 200 GiB (note binary value)

If it is necessary to change the configuration of virtual disk `sda` or `sdb`, see [Configure the eLogin RAID Virtual Disks](#) on page 91.

For more information about binary values, see [Prefixes for Binary and Decimal Multiples](#) on page 206.

4. If creating a new storage profile, copy `elogin_default` or another storage profile, then make the desired changes to the nonpersistent disk.

- a. Copy the storage profile.

In the worksheet, copy the default storage profile and paste it below this line.

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

- b. Replace the name (key) of the copied profile with the key for the new storage profile (`new_elogin` in this example).

```
# NOTE: Place additional 'storage_profiles' setting entries here, if desired.
```

```
cray_storage.settings.storage_profiles.data.new_elogin: null
cray_storage.settings.storage_profiles.data.new_elogin.enabled: true

cray_storage.settings.storage_profiles.data.new_elogin.layouts.device./dev/sda: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.persist_on_boot: false

cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.GRUB: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.GRUB.type: ext3
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.GRUB.size: 1MiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.BOOT: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.BOOT.type: ext3
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.BOOT.size: 2GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.WRITELAYER: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.WRITELAYER.type: ext4
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.WRITELAYER.size: 20GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.TMP: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.TMP.type: xfs
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.TMP.size: 256GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.label.SWAP: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.SWAP.type: swap
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sda.partitions.SWAP.size: 128GiB
...

cray_storage.settings.storage_profiles.data.new_elogin.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.persist_on_boot: true

cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.label.CRASH: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.CRASH.type: ext4
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.CRASH.size: 10GiB
...
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.label.PERSISTENT: null
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.PERSISTENT.type: xfs
cray_storage.settings.storage_profiles.data.new_elogin.layouts./dev/sdb.partitions.PERSISTENT.size: ALL
...
```

- c. Make the desired changes to the nonpersistent disk of this new storage profile.

Because the disk is nonpersistent, partitions can be added, removed, resized, reordered, or have their file system type changed. Make the desired reprovisioning changes now, bearing in mind the requirements listed at the beginning of this procedure.:

5. Upload modified `cray_storage` worksheet to the config set.

```
smw# cfgset update -w '/my/workarea/cray_storage_worksheet.yaml' p0
```

6. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

7. Validate the config set.

```
smw# cfgset validate p0
```

8. (Conditional) If a new storage profile was created, assign the new storage profile to the eLogin node.

```
smw# enode update --set-storage_profile new_elogin elogin1
```

9. Reboot the eLogin node.

This example reboots an eLogin node named *ellogin1*.

```
smw# enode reboot --pxe elogin1
```

10. Verify the changes to the storage layout.

- a. On the SMW, determine if the node is finished booting.

In this example, the eLogin node is *ellogin1*.

```
smw# enode status elogin1
```

The eLogin node has finished booting if its status is `node_up`.

- b. On the eLogin node, verify that the desired partitions exist with the expected sizes.

```
ellogin# df
```

6.5 Enable LiveUpdates Support for eLogin Nodes

About this task

The Cray LiveUpdates service is a repository content distribution mechanism designed to route IMPS repository content information from a central location to client nodes. The LiveUpdates service provides client nodes the ability to dynamically install update packages or install new software using the distribution-native package manager, `zypper`.

LiveUpdates is disabled by default for eLogin nodes due to security concerns. With LiveUpdates, eLogin nodes dynamically pull content from the SMW. Although the use of RPM and package managers is limited to users with root access, the distribution of RPM content is over HTTP. This effectively allows any user to pull package content but not install it locally without elevated privileges.

IMPORTANT: Software Environment Congruence

If LiveUpdates is used to maintain and update internal CLE nodes, eLogin environments must also be kept up-to-date. If LiveUpdates is configured for eLogin use, then the same package manager update commands can be used to accomplish this. If not, a new eLogin image must be created on the SMW and then deployed and booted on a eLogin node.

On eLogin nodes, the booted config set determines whether the LiveUpdates service is enabled. If enabled, local repositories are configured to reference the upstream repositories of origin on the SMW.

The LiveUpdates service is already registered with the SMW and eLogin firewalls.

This procedure ensures that LiveUpdates is enabled in the config sets and the package manager invocations function correctly. It also provides an example command to run LiveUpdates on an eLogin node, if the system administrator chooses to do so.

Procedure

1. Ensure that LiveUpdates is enabled in the config sets for an eLogin node.

Cray recommends setting `cray_liveupdates.inherit` to `true` in the CLE config set and setting `cray_liveupdates.enabled` to `true` in the global config set. Then it is easy to disable LiveUpdates by setting `cray_liveupdates.enabled` to `false` in the global config set, as needed.

- a. Determine whether `cray_liveupdates` is enabled in both the global and CLE config sets.

This example checks CLE config set `p0`.

```
smw# cfmset get cray_liveupdates.enabled global
true
smw# cfmset get cray_liveupdates.enabled p0
false
```

- b. If `cray_liveupdates` in the CLE config set is not enabled, determine whether it inherits from the global config set.

```
smw# cfmset get cray_liveupdates.inherit p0
true
```

In this example for CLE config set `p0`, `cray_liveupdates` inherits from global, which is enabled. Therefore LiveUpdates is enabled for eLogin.

- c. If `cray_liveupdates` is not enabled in the global config set, enable it.

```
smw# cfmset modify --set true cray_liveupdates.enabled global
smw# cfmset update -m prepare global
```

- d. If `cray_liveupdates` is not enabled in the CLE config set, and it does not inherit from the global config set, enable it.

```
smw# cfmset modify --set true cray_liveupdates.enabled p0
smw# cfmset update -m prepare p0
```

2. Test to ensure the repositories are available via HTTP from an eLogin node.

```
eLogin# wget -O - http://smw:2526/repos
--2018-01-30 13:58:50-- http://smw:2526/repos
Resolving smw (smw)... 10.7.1.1
Connecting to smw (smw)|10.7.1.1|:2526... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://smw:2526/repos/ [following]
--2018-01-30 13:58:50-- http://smw:2526/repos/
Reusing existing connection to smw:2526.
```

```
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'STDOUT'
```

```
[<=>] 0 --.-K/s
<html>
<head><title>Index of /repos/</title></head>
<body bgcolor="white">
<h1>Index of /repos/</h1><hr><pre><a href="..">../</a>
<a href="centos_6.5/">centos_6.5/</a> 26-Jan-2018 15:44 -
<a href="centos_6.5_updates/">centos_6.5_updates/</a> 26-Jan-2018 15:44 -
<a href="centos_6.5_x86-64/">centos_6.5_x86-64/</a> 03-Jan-2018 18:14 -
<a href="centos_6.5_x86-64_updates/">centos_6.5_x86-64_updates/</a> 03-Jan-2018 18:14 -
<a href="common_cle_6.0up05_centos_6.5_x86-64_ari/">common_cle_6.0up05_centos_6.5_x86-64_ari/</a> 15-Oct-2017
22:11 -
<a href="common_cle_6.0up05_centos_6.5_x86-64_ari_updates/">common_cle_6.0up05_centos_6.5_x86-64_ari_updates/</a>
15-Oct-2017 22:11 -
<a href="common_cle_6.0up05_sles_12sp3_x86-64_ari/">common_cle_6.0up05_sles_12sp3_x86-64_ari/</a> 15-Oct-2017
22:11 -
<a href="common_cle_6.0up05_sles_12sp3_x86-64_ari_updates/">common_cle_6.0up05_sles_12sp3_x86-64_ari_updates/</a>
15-Oct-2017 22:11 -
<snip>
<a href="smw_8.0up06_sles_12sp3_x86-64/">smw_8.0up06_sles_12sp3_x86-64/</a> 03-Jan-2018 18:14 -
<a href="smw_8.0up06_sles_12sp3_x86-64_updates/">smw_8.0up06_sles_12sp3_x86-64_updates/</a> 03-Jan-2018
18:14 -
<a href="snapshots/">snapshots/</a> 15-Oct-2017 22:02 -
</pre><hr></body>
</html>
[<=>] 14,644
--.-K/s in 0s
2018-01-30 13:58:50 (92.4 MB/s) - written to stdout [14644]
```

3. Test the commands to list and view the repos.

```
eloin# zypper lr
Repository priorities are without effect. All enabled repositories share the same priority.
```

#	Alias	Name
1	common_cle_6.0up06_sles_12sp3_ari	common_cle_6.0up06_sles_12sp3_ari
Yes	() No No	
2	common_cle_6.0up06_sles_12sp3_ari_updates	common_cle_6.0up06_sles_12sp3_ari_updates
Yes	() No No	
3	cray_sle_common_12sp3_updates	cray_sle_common_12sp3_updates
Yes	() No No	
4	lustre-2.7_cle_6.0up06_sles_12sp3_ari	lustre-2.7_cle_6.0up06_sles_12sp3_ari
Yes	() No No	
5	lustre-2.7_cle_6.0up06_sles_12sp3_ari_updates	lustre-2.7_cle_6.0up06_sles_12sp3_ari_updates
Yes	() No No	
6	passthrough-common_cle_6.0up06_sles_12sp3	passthrough-common_cle_6.0up06_sles_12sp3
Yes	() No No	
7	passthrough-common_cle_6.0up06_sles_12sp3_updates	passthrough-common_cle_6.0up06_sles_12sp3_updates
Yes	() No No	
8	sle-module-legacy_12	sle-module-legacy_12
Yes	() No No	
9	sle-module-legacy_12_updates	sle-module-legacy_12_updates
Yes	() No No	
10	sle-sdk_12sp3	sle-sdk_12sp3
Yes	() No No	
11	sle-sdk_12sp3_updates	sle-sdk_12sp3_updates
Yes	() No No	
12	sle-server_12sp3	sle-server_12sp3
Yes	() No No	
13	sle-server_12sp3_updates	sle-server_12sp3_updates
Yes	() No No	
14	sle-we_12sp3	sle-we_12sp3
Yes	() No No	
15	sle-we_12sp3_updates	sle-we_12sp3_updates
Yes	() No No	

```
eloin# zypper pa -r common_cle_6.0up06_sles_12sp3_ari | head -n 20
Loading repository data...
Reading installed packages...
S | Repository | Name | Version | Arch
-----+-----+-----+-----+-----
i+ | common_cle_6.0up06_sles_12sp3_ari | QConvergeConsoleCLI | 1.1.03-49 |
x86_64
| common_cle_6.0up06_sles_12sp3_ari | cray-aeld | 1.3.1-6.0.6.0_2.90_ge24aa27.ari |
x86_64
| common_cle_6.0up06_sles_12sp3_ari | cray-aeld-imps_ansible | 1.3.1-6.0.6.0_2.90_ge24aa27.ari |
x86_64
```

```

i+ | common_cle_6.0up06_sles_12sp3_ari | cray-aeld-man | 1.3.1-6.0.6.0_2.90_ge24aa27.ari |
x86_64
i | common_cle_6.0up06_sles_12sp3_ari | cray-aeld-switch | 1.3.1-6.0.6.0_2.90_ge24aa27.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alps | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alps-boot | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alps-compute | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alps-imps_ansible | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alps-imps_configurator | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
i+ | common_cle_6.0up06_sles_12sp3_ari | cray-alps-man | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
i | common_cle_6.0up06_sles_12sp3_ari | cray-alps-switch | 6.6.0-6.0.6.0_35.9_gd0a1ab9.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alpscomm | 1.3.4-6.0.6.0_5.43_g8f4elf0.ari |
x86_64
  | common_cle_6.0up06_sles_12sp3_ari | cray-alpscomm-compute | 1.3.4-6.0.6.0_5.43_g8f4elf0.ari |
x86_64
i+ | common_cle_6.0up06_sles_12sp3_ari | cray-alpscomm-man | 1.3.4-6.0.6.0_5.43_g8f4elf0.ari |
x86_64
i | common_cle_6.0up06_sles_12sp3_ari | cray-alpscomm-switch | 1.3.4-6.0.6.0_5.43_g8f4elf0.ari |
x86_64

```

4. Search for the availability of a package on an eLogin node.

The `zypper search` command searches for the availability of a given package in the configured repositories. It differs from `zypper lr` in that it builds repository cache information the first time it is run in a boot session. However, this command does not install or update anything.

```

eloin# zypper search cray-imps-api
Building repository 'common_cle_6.0up06_sles_12sp3_ari' cache .....[done]
Building repository 'common_cle_6.0up06_sles_12sp3_ari_updates' cache .....[done]
Building repository 'cray_sle_common_12sp3_updates' cache .....[done]
Building repository 'lustre-2.7_cle_6.0up06_sles_12sp3_ari' cache .....[done]
Building repository 'lustre-2.7_cle_6.0up06_sles_12sp3_ari_updates' cache .....[done]
Building repository 'passthrough-common_cle_6.0up06_sles_12sp3' cache .....[done]
Building repository 'passthrough-common_cle_6.0up06_sles_12sp3_updates' cache .....[done]
Building repository 'sle-module-legacy_12' cache .....[done]
Building repository 'sle-module-legacy_12_updates' cache .....[done]
Building repository 'sle-sdk_12sp3' cache .....[done]
Building repository 'sle-sdk_12sp3_updates' cache .....[done]
Building repository 'sle-server_12sp3' cache .....[done]
Building repository 'sle-server_12sp3_updates' cache .....[done]
Building repository 'sle-we_12sp3' cache .....[done]
Building repository 'sle-we_12sp3_updates' cache .....[done]
Loading repository data...
Reading installed packages...

S | Name | Summary | Type
-----+-----+-----+-----
i+ | cray-imps-api | Contains API level routines of IMPS as a package level organization | package

```

5. From the SMW, start the update on an eLogin node (eloin1 in the example).



CAUTION: This command is potentially disruptive to the users of the eLogin node. It will update everything that is installed that has updates available. Run this command on a live eLogin node only if there are new packages in the repos on the SMW that the system administrator would like to update on the node.

Note that this type of update is only temporary and will disappear when the node is rebooted. If the admin wants more permanent changes, build a new image on the SMW and stage or boot that image on the eLogin node.

```
smw# ssh eloin1 "zypper --non-interactive up"
```

Repeat this step for each eLogin node in this system, as needed.

6.6 Optional Network Configurations

6.6.1 Configure an IPv4 Interface to Include IPv6 Address

Prerequisites

eLogin is fully installed and configured with an IPv4 interface.

About this task

Sites that use the IPv6 interface in addition to IPv4 must configure eLogin nodes for IPv6 in the CLE config set for eLogin nodes. This procedure configures an interface to include both an IPv4 and IPv6 address in the config set.

However, the configuration worksheets do not accommodate more than one route. The IPv6 route must be configured using Simple Sync or site Ansible play.

Procedure

1. Prepare configuration worksheets for editing.

- a. Generate a set of configuration worksheets with the current CLE configuration data.

This example uses the existing CLE config set p0.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets
smw# cp *_worksheet.yaml /my/workarea
```

- c. Change to the new work area.

```
smw# cd /my/workarea
```

2. Edit the `cray_net` configuration worksheet to add IPv6 information for an eLogin node.

In worksheet, locate the eLogin node network interface that is on the `site` network (`site-user-net`). For the `extra_attributes` field of that interface, replace the default empty list with the two lines shown in the following example. Replace `<elogin1>` with the correct host name for this eLogin node, and replace `IPV6_ADDR` with the actual IPv6 address to use for this system.

```
smw# vi cray_net_worksheet.yaml
```

```
cray_net.settings.hosts.data.<elogin1>.interfaces.eth1.extra_attributes:
- IPADDR1='IPV6_ADDR'
- PREFIXLEN1='64'
```

The `IPADDR1` and `PREFIXLEN1` entries will be added to the `/etc/sysconfig/network/ifcfg-eth1` file on the node.

IMPORTANT: The configuration worksheets do not accommodate more than one route, so the IPv6 route must be configured using Simple Sync or site Ansible play.

3. Upload the modified `cray_net` worksheet to the config set (`p0` in this example).

```
smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0
```

4. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

5. Validate the config set.

```
smw# cfgset validate p0
```

6. Push the config set to the eLogin node.

This example uses CLE config set `p0` and eLogin node `ellogin1`.

```
smw# cfgset push -d ellogin1 p0
```

Or use this command to push the CLE config set to all eLogin nodes in the specified node group (`my_node_group` in this example).

```
smw# cfgset push -g my_node_group p0
```

7. Apply config set changes on the eLogin node.

These changes can be applied immediately by running `cray-ansible` on the node (`ellogin1` in the example) while the node is booted, and then verifying that the interfaces come up with the expected configuration. This should also be tested by rebooting the node.

```
ellogin1# /etc/init.d/cray-ansible start
```

8. Verify the interface comes up with the expected configuration.

9. Reboot the eLogin node (`ellogin1` in the example).

On the SMW, reboot the eLogin node from disk.

```
smw# enode reboot --disk ellogin1
```

Or, if the SMW is not available, on that node, reboot the eLogin node from disk (default, when rebooting on the node).

```
ellogin1# enode reboot
```

10. Check the settings for this eLogin node interface in the CLE config set.

```
smw# cfgset search -t ellogin1 -s cray_net p0
```

11. Verify the interface comes up with the expected configuration.

6.6.2 Configure Tagged VLANs for eLogin

Prerequisites

eLogin is fully installed.

About this task

Virtual local area networks (VLAN) are used to divide a physical network into several broadcast domains. This procedure describes how to configure tagged-packet type VLANs for eLogin in the `cray_net` configuration service.

Procedure

1. Prepare configuration worksheets for editing.
 - a. Generate a set of configuration worksheets with the current CLE configuration data.

This example uses the existing CLE config set `p0`.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets
smw# cp *_worksheet.yaml /my/workarea
```

- c. Change to the new work area.

```
smw# cd /my/workarea
```

2. Edit the `cray_net` worksheet and add the following data to configure VLAN interfaces.

```
smw# cray_net_worksheet.yaml
```

Locate the section of the worksheet where the interfaces for the eLogin node are defined, and insert or modify the data there.

NOTE: This example configures the `eth4` interface on the tagged VLAN named `eth4.1234`, for an eLogin node named `elogin1`.

```
# ** 'hosts' DATA **
...
cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth4: null
cray_net.settings.hosts.data.elogin1.interfaces.eth4.name: eth4
cray_net.settings.hosts.data.elogin1.interfaces.eth4.description:
  Ethernet connecting network node to customer network
cray_net.settings.hosts.data.interfaces.eth4.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4.network: site
cray_net.settings.hosts.data.elogin1.interfaces.eth4.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth4.ipv4_secondary_addresses: []
```



```

cray_net.settings.hosts.data.elogin1.interfaces.eth4.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.eth4.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.eth4.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.params: ''
#cray_net.settings.hosts.data.elogin1.interfaces.eth4.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth4_1234: null
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.name: eth4.1234
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.description:
    Ethernet connecting the network
cray_net.settings.hosts.data.interfaces.eth4_1234.vlan_id: '1234'
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.vlan_etherdevice: eth4
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.network: site
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.ipv4_address: 10.236.1.190
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.extra_attributes:
- USERCONTROL='no'
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.params: ''
#cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.unmanaged_interface: false

```

The worksheet edit is now complete.

3. Upload the modified `cray_net` worksheet to the config set (`p0` in this example).

```
smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0
```

4. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

5. Validate the config set.

```
smw# cfgset validate p0
```

6. Push the config set to the eLogin node.

This example uses CLE config set `p0` and eLogin node `elogin1`.

```
smw# cfgset push -d elogin1 p0
```

Or use this command to push the CLE config set to all eLogin nodes in the specified node group (`my_node_group` in this example).

```
smw# cfgset push -g my_node_group p0
```

7. Apply config set changes on the eLogin node.

These changes can be applied immediately by running `cray-ansible` on the node (`elogin1` in the example) while the node is booted, and then verifying that the interfaces come up with the expected configuration. This should also be tested by rebooting the node.

```
elogin1# /etc/init.d/cray-ansible start
```

8. Verify the interface comes up with the expected configuration.

9. Reboot the eLogin node (*ellogin1* in the example).

On the SMW, reboot the eLogin node from disk.

```
smw# enode reboot --disk ellogin1
```

Or, if the SMW is not available, on that node, reboot the eLogin node from disk (default, when rebooting on the node).

```
ellogin1# enode reboot
```

10. Check the settings for this eLogin node interface in the CLE config set.

```
smw# cfgset search -t ellogin1 -s cray_net p0
```

11. Verify the interface comes up with the expected configuration.

6.6.3 Configure Bonded Interfaces for eLogin

Prerequisites

eLogin is fully installed.

About this task

This procedure configures bonded interfaces for an eLogin node.

Procedure

1. Prepare configuration worksheets for editing.

a. Generate a set of configuration worksheets with the current CLE configuration data.

This example uses the existing CLE config set *p0*.

```
smw# cfgset update -m prepare --no-scripts p0
```

b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called */my/workarea*. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea  
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets  
smw# cp *_worksheet.yaml /my/workarea
```

c. Change to the new work area.

```
smw# cd /my/workarea
```

2. Edit the *cray_net* worksheet.

```
smw# vi cray_net_worksheet.yaml
```

3. Add network definitions.

Locate the section of the worksheet where networks are defined, and insert the following data there.

In the following example, a customer network named `net_224` is defined, which will be assigned to the bonding master interface. Then a "dummy" network named `slave_eth` is defined, which will be assigned to the physical interfaces that will be bonded to the bonding master.

```
# NOTE: Place additional 'networks' setting entries here, if desired.
cray_net.settings.networks.data.name.net_224: null
cray_net.settings.networks.data.net_224.description: Customer 10.0.224.0 network
cray_net.settings.networks.data.net_224.ipv4_network: 10.0.224.0
cray_net.settings.networks.data.net_224.ipv4_netmask: 255.255.255.0
cray_net.settings.networks.data.net_224.ipv4_broadcast: ''
cray_net.settings.networks.data.net_224.ipv4_gateway: 10.0.224.1
cray_net.settings.networks.data.net_224.dns_servers:
- 10.0.146.10
- 10.0.199.10
cray_net.settings.networks.data.net_224.dns_search:
- site-example.net
cray_net.settings.networks.data.net_224.ntp_servers:
- 10.0.6.48
- 10.0.42.217
#cray_net.settings.networks.data.net_224.fw_external: false

cray_net.settings.networks.data.name.slave_eth: null
cray_net.settings.networks.data.slave_eth.description: Disabled eth device for bond
cray_net.settings.networks.data.slave_eth.ipv4_network: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_netmask: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_broadcast: ''
cray_net.settings.networks.data.slave_eth.ipv4_gateway: ''
cray_net.settings.networks.data.slave_eth.dns_servers: []
cray_net.settings.networks.data.slave_eth.dns_search: []
cray_net.settings.networks.data.slave_eth.ntp_servers: []
#cray_net.settings.networks.data.slave_eth.fw_external: false
```

4. Add host interface definitions.

Locate the section of the worksheet where the interfaces for the eLogin node are defined, and insert or modify the data there.

In the example, the `bond0` interface is defined as the bonding master for two physical interfaces, `eth0` and `eth2`, and the `net_224` network is assigned to it. Then the `eth0` and `eth2` interfaces are defined with the disabled `slave_eth` network assigned to them.

The bonding options set for `bond0` are as follows:

- Traffic to `bond0` can come over either the `eth0` or `eth2` interface based on the `xmit_hash_policy` value.
- The `mode` parameter is set for load balancing between the bonded interfaces, which provides fault tolerance.
- `miimon` is a traffic monitoring setting to prevent network degradation in case of network issues.

```
# ** 'hosts' DATA **
...
cray_net.settings.hosts.data.elogin1.interfaces.common_name.bond0: null
cray_net.settings.hosts.data.elogin1.interfaces.bond0.name: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.description: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bonding_slaves:
- eth0
- eth2
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bonding_module_opts:
  miimon=100 mode=balance-xor use_carrier=1 xmit_hash_policy=layer2+3
cray_net.settings.hosts.data.elogin1.interfaces.bond0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.network: net_224
cray_net.settings.hosts.data.elogin1.interfaces.bond0.ipv4_address: 10.0.224.63
```

```

cray_net.settings.hosts.data.elogin1.interfaces.bond0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mtu: '1500'
cray_net.settings.hosts.data.elogin1.interfaces.bond0.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.elogin1.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.description: Slave eth0
cray_net.settings.hosts.data.interfaces.eth0.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.bonding_slaves: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth0.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth0.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth2: null
cray_net.settings.hosts.data.elogin1.interfaces.eth2.name: eth2
cray_net.settings.hosts.data.elogin1.interfaces.eth2.description: Slave eth2
cray_net.settings.hosts.data.interfaces.eth2.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.bonding_slaves: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth2.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth2.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth2.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.unmanaged_interface: false

```

The worksheet edit is now complete.

- Upload the modified `cray_net` worksheet to the config set (`p0` in this example).

```
smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0
```

- Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

- Validate the config set.

```
smw# cfgset validate p0
```

- Push the config set to the eLogin node.

This example uses CLE config set `p0` and eLogin node `elogin1`.

```
smw# cfgset push -d elogin1 p0
```

Or use this command to push the CLE config set to all eLogin nodes in the specified node group (*my_node_group* in this example).

```
smw# cfgset push -g my_node_group p0
```

9. Apply config set changes on the eLogin node.

These changes can be applied immediately by running `cray-ansible` on the node (*elogin1* in the example) while the node is booted, and then verifying that the interfaces come up with the expected configuration. This should also be tested by rebooting the node.

```
elogin1# /etc/init.d/cray-ansible start
```

10. Verify the interface comes up with the expected configuration.

11. Reboot the eLogin node (*elogin1* in the example).

On the SMW, reboot the eLogin node from disk.

```
smw# enode reboot --disk elogin1
```

Or, if the SMW is not available, on that node, reboot the eLogin node from disk (default, when rebooting on the node).

```
elogin1# enode reboot
```

12. Check the settings for this eLogin node interface in the CLE config set.

```
smw# cfgset search -t elogin1 -s cray_net p0
```

13. Verify the interface comes up with the expected configuration.

6.6.4 Configure Bonded VLAN Interfaces for eLogin

Prerequisites

eLogin is fully installed.

About this task

This procedure configures bonded VLAN interfaces for an eLogin node.

Procedure

1. Prepare configuration worksheets for editing.
 - a. Generate a set of configuration worksheets with the current CLE configuration data.
This example uses the existing CLE config set *p0*.

```
smw# cfgset update -m prepare --no-scripts p0
```

- b. Copy the CLE worksheets to a work area for editing.

This example makes a directory called `/my/workarea`. Use a suitable work area directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cd /var/opt/cray/imps/config/sets/p0/worksheets
smw# cp *_worksheet.yaml /my/workarea
```

- c. Change to the new work area.

```
smw# cd /my/workarea
```

2. Edit the `cray_net` worksheet.

```
smw# vi cray_net_worksheet.yaml
```

3. Add network definitions.

Locate the section of the worksheet where networks are defined, and insert the following data there.

In the following example, a customer network named `net_224` is defined, which will be assigned to the VLAN interface. Then a "dummy" network named `slave_eth` is defined, which will be assigned to the interfaces that will be bonded together.

```
# NOTE: Place additional 'networks' setting entries here, if desired.
cray_net.settings.networks.data.name.net_224: null
cray_net.settings.networks.data.net_224.description: Customer 10.0.224.0 network
cray_net.settings.networks.data.net_224.ipv4_network: 10.0.224.0
cray_net.settings.networks.data.net_224.ipv4_netmask: 255.255.255.0
cray_net.settings.networks.data.net_224.ipv4_broadcast: ''
cray_net.settings.networks.data.net_224.ipv4_gateway: 10.0.224.1
cray_net.settings.networks.data.net_224.dns_servers:
- 10.0.146.10
- 10.0.199.10
cray_net.settings.networks.data.net_224.dns_search:
- site-example.net
cray_net.settings.networks.data.net_224.ntp_servers:
- 10.0.6.48
- 10.0.42.217
#cray_net.settings.networks.data.net_224.fw_external: false

cray_net.settings.networks.data.name.slave_eth: null
cray_net.settings.networks.data.slave_eth.description: Disabled eth device for bond
cray_net.settings.networks.data.slave_eth.ipv4_network: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_netmask: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_broadcast: ''
cray_net.settings.networks.data.slave_eth.ipv4_gateway: ''
cray_net.settings.networks.data.slave_eth.dns_servers: []
cray_net.settings.networks.data.slave_eth.dns_search: []
cray_net.settings.networks.data.slave_eth.ntp_servers: []
#cray_net.settings.networks.data.slave_eth.fw_external: false
```

4. Add host interface definitions.

Locate the section of the worksheet where the interfaces for the eLogin node are defined, and insert or modify the data there.

In the example, the `bond0` interface is defined with no network assigned to it and two other interfaces (`eth0` and `eth2`) bound to it. The `eth0` and `eth2` interfaces are defined with the disabled `slave_eth` network assigned to them. Finally, a the `bond_224` interface is defined with the `net_224` network assigned to it and `bond0` as the device used for the VLAN.

The bonding options set for `bond0` are as follows:

- Traffic to `bond0` can come over either the `eth0` or `eth2` interface based on the `xmit_hash_policy` value.

- The `mode` parameter is set for load balancing between the bonded interfaces, which provides fault tolerance.
- `miimon` is a traffic monitoring setting to prevent network degradation in case of network issues.

```
# ** 'hosts' DATA **
...
cray_net.settings.hosts.data.elogin1.interfaces.common_name.bond0: null
cray_net.settings.hosts.data.elogin1.interfaces.bond0.name: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.description: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bonding_slaves:
- eth0
- eth2
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bonding_module_opts:
  miimon=100 mode=balance-xor use_carrier=1 xmit_hash_policy=layer2+3
cray_net.settings.hosts.data.elogin1.interfaces.bond0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.network: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mtu: '9000'
cray_net.settings.hosts.data.elogin1.interfaces.bond0.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth0: null
cray_net.settings.hosts.data.elogin1.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.description: Slave eth0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.bonding_slaves: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth0.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth0.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth2: null
cray_net.settings.hosts.data.elogin1.interfaces.eth2.name: eth2
cray_net.settings.hosts.data.elogin1.interfaces.eth2.description: Slave eth2
cray_net.settings.hosts.data.elogin1.interfaces.eth2.vlan_id: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.vlan_etherdevice: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.bonding_slaves: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth2.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth2.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth2.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.bond0_224: null
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.name: bond0.224
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.description:
  bond0 and vlan 244
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.vlan_id: '224'
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.vlan_etherdevice: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.bonding_slaves: []
```

```

cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.network: net_224
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.ipv4_address: 10.0.224.63
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.ipv4_secondary_addresses: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.extra_attributes:
- USERCONTROL='no'
- REORDER_HDR='no'
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.unmanaged_interface: false

```

The worksheet edit is now complete.

5. Upload the modified `cray_net` worksheet to the config set (`p0` in this example).

```
smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0
```

6. Update the CLE config set.

```
smw# cfgset update p0
```

This update runs all pre-configuration and post-configuration scripts. It is good practice to update the config set when any config services have been changed by importing worksheets.

7. Validate the config set.

```
smw# cfgset validate p0
```

8. Push the config set to the eLogin node.

This example uses CLE config set `p0` and eLogin node `elogin1`.

```
smw# cfgset push -d elogin1 p0
```

Or use this command to push the CLE config set to all eLogin nodes in the specified node group (`my_node_group` in this example).

```
smw# cfgset push -g my_node_group p0
```

9. Apply config set changes on the eLogin node.

These changes can be applied immediately by running `cray-ansible` on the node (`elogin1` in the example) while the node is booted, and then verifying that the interfaces come up with the expected configuration. This should also be tested by rebooting the node.

```
elogin1# /etc/init.d/cray-ansible start
```

10. Verify the interface comes up with the expected configuration.

11. Reboot the eLogin node (`elogin1` in the example).

On the SMW, reboot the eLogin node from disk.

```
smw# enode reboot --disk elogin1
```

Or, if the SMW is not available, on that node, reboot the eLogin node from disk (default, when rebooting on the node).


```
elogin1# enode reboot
```

12. Check the settings for this eLogin node interface in the CLE config set.

```
smw# cfgset search -t elogin1 -s cray_net p0
```

13. Verify the interface comes up with the expected configuration.

6.7 Configure SSDs on eLogin Nodes

Prerequisites

- Solid-state storage devices (SSD) have been installed on this eLogin node.
- Connection to the console of the eLogin node, using either a physical connection or ConMan (`conman`).

About this task

This procedure describes how to configure internal drives as RAID 0 virtual disks `sda` and `sdb`, and an SSD as a RAID 0 virtual disk with device name `sdc`. The steps ensure that the system drive does not inadvertently move onto the SSD. If this site wants the SSD to be on a virtual disk other than `sdc`, contact a Cray service representative for help.

When an SSD is installed, specific configuration setup is required because of the way the eLogin node BIOS discovers storage devices. The eLogin node, by default, uses the Dell Power Edge Expandable RAID Controller (PERC) to manage disk drives. During discovery, the Dell PERC separates non-RAID devices from RAID devices. The non-RAID devices are presented to the operating system first, followed by the RAID devices.

If this site has installed SSDs, and the SSDs have been discovered prior to the RAID devices, then the SSDs will likely have the device name `sda` and `sdb`. RAID devices discovered after the SSDs would then be named `sdc` and `sdd` because the `sda` and `sdb` names were already taken. That causes a problem for the eLogin installation process, because the RAID devices are required to be devices named `sda` and `sdb`. To solve that problem, the SSDs must be removed to free up the device names `sda` and `sdb`, and the RAID devices must then be configured without the SSDs installed. After the RAID devices `sda` and `sdb` are configured, then the SSDs can be safely installed and configured on the system without those devices taking the `sda` and `sdb` device names.

The following list summarizes this procedure. **Read and understand the entire procedure before attempting to perform it.**

1. Physically remove the SSDs from the eLogin node.
2. Reconfigure the eLogin RAID to default settings, without SSDs present.
3. When saving settings and exiting the RAID utility, power off the node before it can reboot.
4. With node powered off, reinstall the SSDs.
5. Configure the SSDs as RAID devices.

The images used in this procedure are examples only. Depending on the server model and version of RAID configuration utility, there could be minor differences in the screens and steps to configure this node.

Procedure

1. Physically remove all SSDs from the system.
2. Boot the eLogin hardware. On startup of the eLogin node, press **Ctrl-R** when prompted to enter RAID setup. Press **Ctrl-R** within 5 seconds of seeing the following screen.

NOTE: The BIOS RAID configuration screens may appear different if accessing from `conman` versus the physical console, but the functionality is the same.

Figure 58. Initial Boot Menu for BIOS RAID Configuration: eLogin

```

F2 = System Setup
F10 = Lifecycle Controller
F11 = Boot Manager
F12 = PXE Boot

QLogic Ethernet Boot Agent
Copyright (C) 2015 QLogic Corporation
All rights reserved.
Press Ctrl-S to enter Configuration Menu

Initializing Serial ATA devices...
Port J: HL-DT-ST DVD-ROM DU90M

PowerEdge Expandable RAID Controller BIOS
Copyright(c) 2015 Avago Technologies
Press <Ctrl><R> to Run Configuration Utility

```

The RAID configuration screen opens.

Figure 59. RAID Configuration Screen: eLogin

```

PERC H310 Mini BIOS Configuration Utility 3.00-0020
UD Mgmt PD Mgmt Ctrl Mgmt Properties
Virtual Disk Management
[-] PERC H310 Mini (Bus 0x02, Dev 0x00)
  [-] Disk Group: 0, RAID 0
    [-] Virtual Disks
      ID: 0, sda, 700.00 GB
      ID: 1, sdb, 416.75 GB
    [+1] Physical Disks
    [+1] Total Free Capacity: 0.00 GB
    Hot spares

Virtual Disk 0:
State: Optimal
RAID Level: 0
Operation: None

Disk Group 0:
Virtual Disks: 2
Physical Disks: 2
Free Cap.: 0.00 GB
Free Areas: 0

F1-Help F2-Operations F5-Refresh Ctrl-N-Next Page Ctrl-P-Prev Page F12-Ctrl

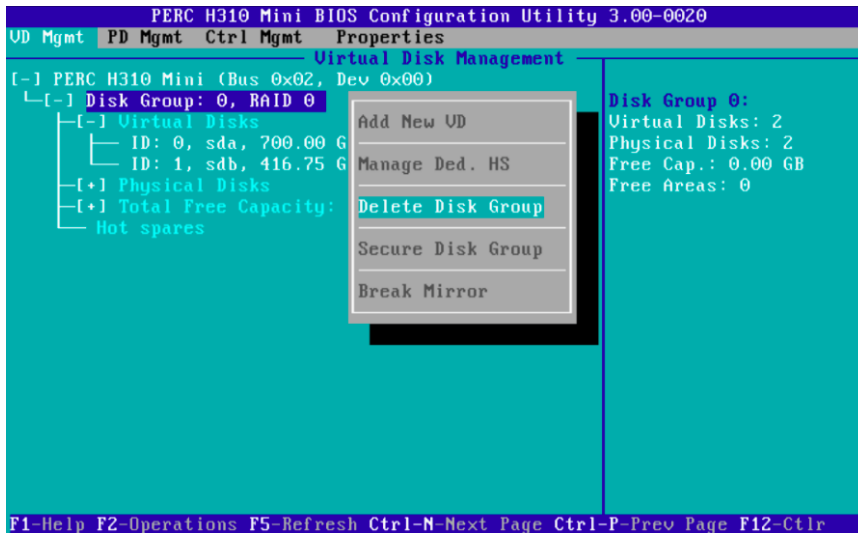
```

3. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration. Otherwise, skip this step.

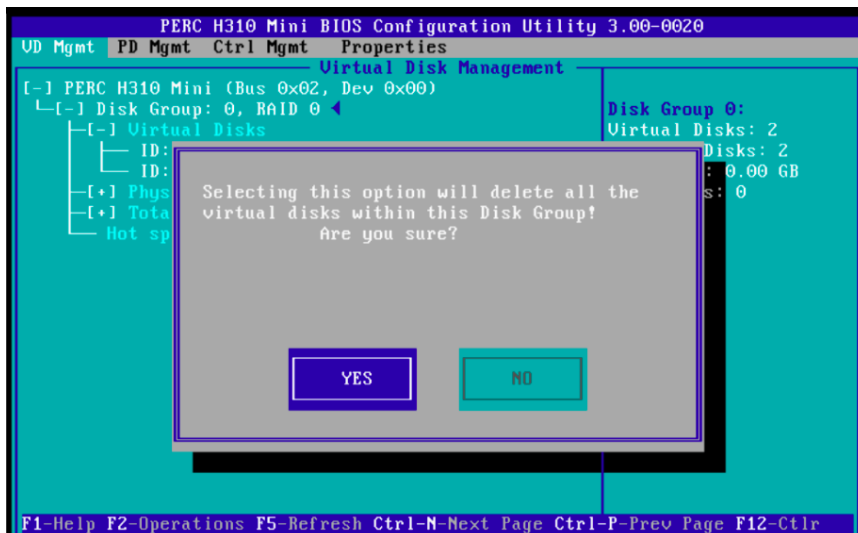
Occasionally disks are not viewable by the OS after RAID reconfiguration. This may be caused by residual metadata on the disk from the previous RAID configuration. To clear the metadata, remove the disks from any RAID configuration, and then initialize the disks. After initialization completes, reconfigure the disks as part of the RAID. This clears any pre-existing metadata and allows the OS to see the devices.

- a. Select the disk.
- b. Press **F2** key to get a list of operations.
- c. Select **Delete Disk Group** and press **Enter**.

Figure 60. Delete Disk Group: eLogin BIOS RAID Setup



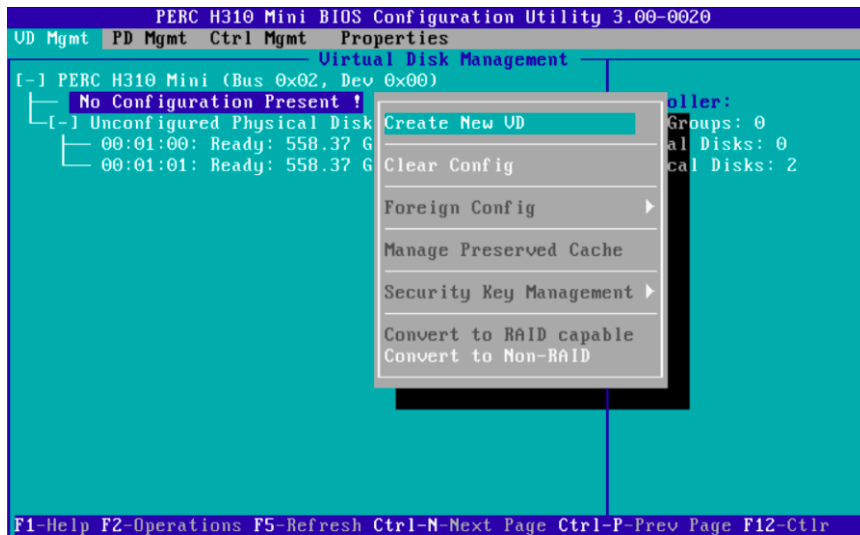
- d. Confirm the selection **Yes**, and press return.



4. Create a new virtual disk A.
 - a. In the virtual disk management window (**VD Mgmt**), navigate to **No Configuration Present !** using the keyboard up/down arrows.
 - b. Press the **F2** key to access the disk creation menu.

- c. Select **Create New VD** from the menu.

Figure 61. Create Virtual Disk A: eLogin BIOS RAID

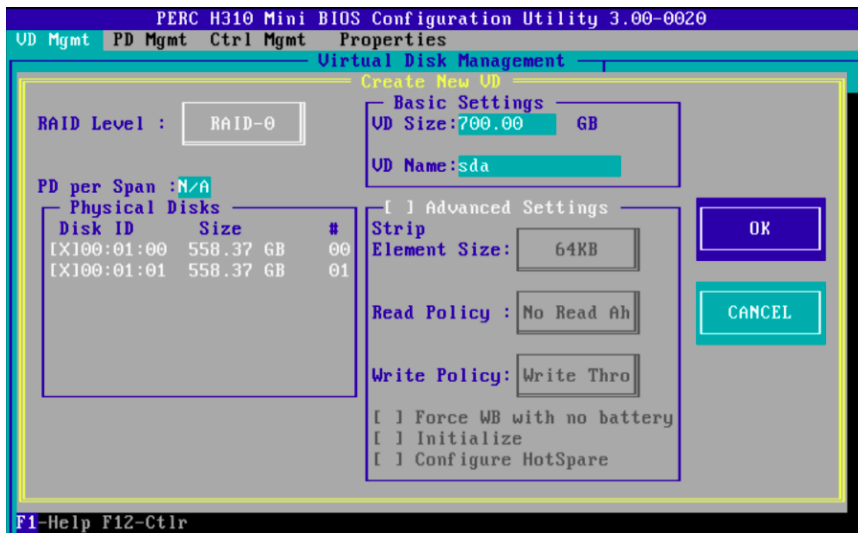


The **Create New VD** window opens.

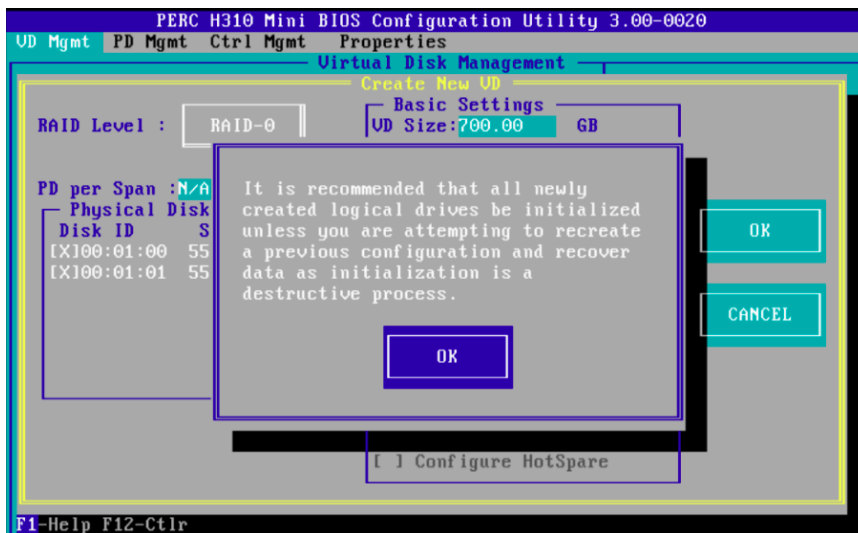
5. Move the cursor to select the disk ID in the **Create New VD** window, and then press spacebar on the keyboard to add disk to RAID.
6. Set the RAID Level to **RAID 0**.
7. Set **VD Size** and **VD Name** for virtual disk A.
 - a. Set the **VD Size** for virtual disk A to **700 GB** of disk space.

IMPORTANT: 700 GB is sufficient to accommodate the partition sizes specified in the default storage profile for eLogin nodes, `e_login_default`, which is defined in the `cray_storage` configuration service. If those sizes were increased for this eLogin node, increase the **VD Size** accordingly.
 - b. Set the **VD Name** to `sda`.

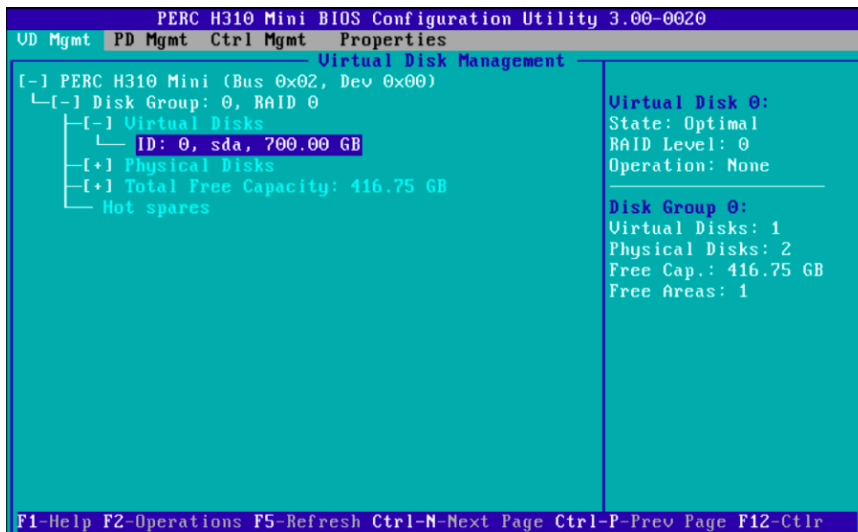
Figure 62. Disk Size and Name Setting for Virtual Disk A: eLogin



- c. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Virtual disk sda is now created.



8. Initialize virtual disk A using **Fast Initialization**.

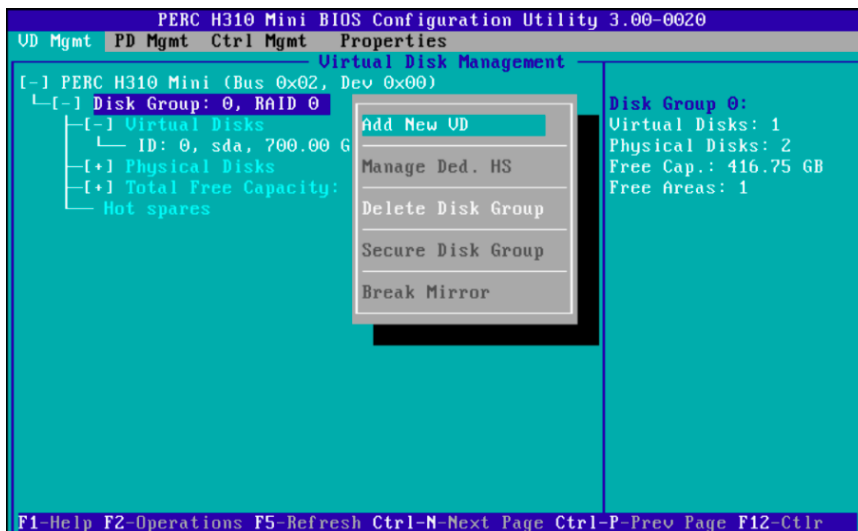
- Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- In the **Initialization** submenu, select **Fast Initialization**.

A pop-up window will be displayed, indicating that the virtual disk has been initialized.

9. Create a new virtual disk B.

- In the **Virtual Disk Management** window, navigate to **Disk Group: 0, RAID 0** using the keyboard up/down arrows.
- Press **F2** to access the disk creation menu.
- Select **Add New VD**.

Figure 63. Create New Virtual Disk B: eLogin BIOS RAID



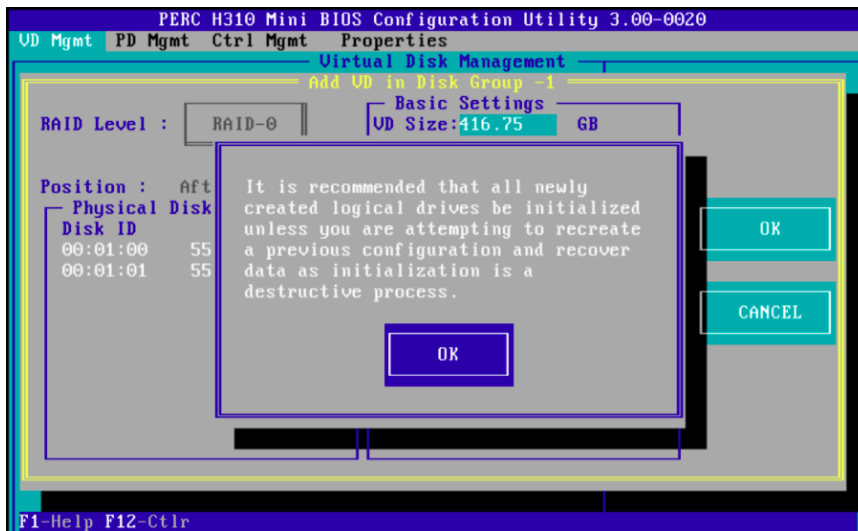
The **Add VD in Disk Group 0** window opens.

- d. In the window, set the **VD Name** to **sdb**, and verify that the **VD Size** is set to the remaining disk space.

Figure 64. Disk Size and Name Setting for Virtual Disk B: eLogin

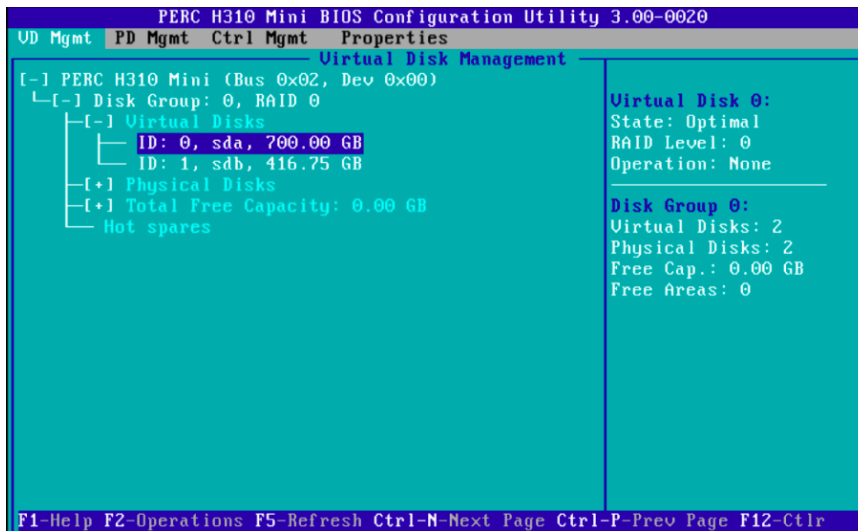


- e. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Two virtual disks are now available.

Figure 65. Two Virtual Disks Available: eLogin BIOS RAID



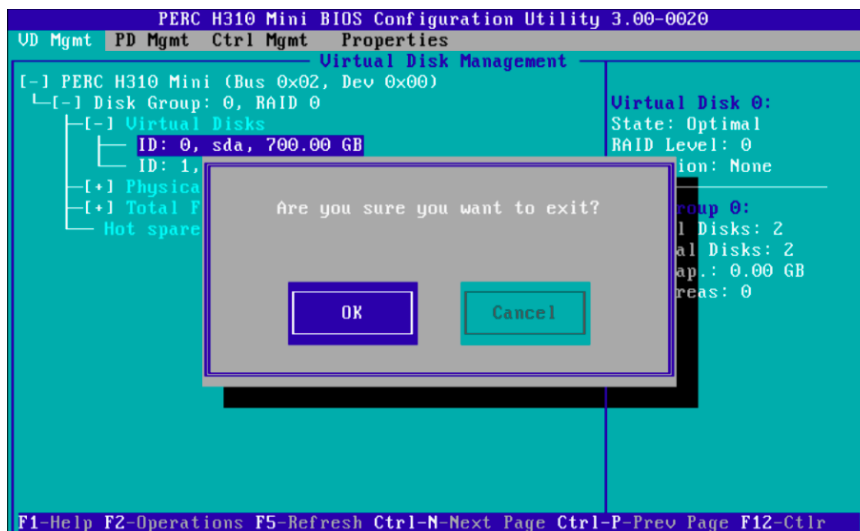
10. Initialize virtual disk B using Fast Initialization.

- Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- In the **Initialization** submenu, select **Fast Initialization**.

A pop-up window will be displayed, indicating that the virtual disk has been initialized.

11. Press **Esc on the keyboard to exit the BIOS configuration, and then select **OK** to confirm exit from the BIOS Configuration Utility.**

Figure 66. Exit BIOS Configuration: eLogin



The BIOS configuration utility screen is now closed.

12. Press **Ctrl+Alt+Delete** from the keyboard to safely save the new BIOS settings, and then power off the system.

IMPORTANT: Normally, after saving the new BIOS settings, the system continues with a reboot. But in this case, power off the system before it can reboot. Do not let it reboot at this point.

13. With the system powered off, plug all SSDs into the system.
14. Reboot the system and press **Ctrl-R** (when prompted) to enter the BIOS Raid Configuration Utility.
15. Configure all SSDs as **RAID** devices (instead of non-RAID devices).
16. Configure all SSDs as **RAID0** without an active partner.

7 Gather Diagnostics for eLogin

7.1 Access the eLogin Console

Prerequisites

The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 110).

Procedure

1. Attach to the console with ConMan using the name of the eLogin node.

```
smw# conman -j elogin1
```

ConMan takes over, putting the user into a serial-over-LAN console session via IPMI with the node. All keystrokes are forwarded to the node.

2. View the node console log.

ConMan logs the console output to: `/var/opt/cray/log/external/conman/console.elogin1`.

```
smw# tail /var/opt/cray/log/external/conman/console.elogin1
```

3. **Trouble?** If the ConMan utility is not working properly or the text is garbled, try the following troubleshooting.
 - a. Disconnect from the console by typing `&` and try to attach to the console again.

```
smw# conman -j elogin1
```

If the problem persists, proceed to the next step.

There may be a problem with the `remcon` setting for the node with a bad baud rate.

- b. If the log files are working but the interactive terminal access is not, check to see if the `remcon` parameter is correct.

The default value is `/dev/ttyS1,115200` for the path and baud rate of the console.

Change the `remcon` parameter with `enode update`.

```
smw# enode list --fields remcon elogin1
smw# enode update --unset-remcon elogin1
Updating the following node(s):
elogin1
Successfully updated ['elogin1']
smw# enode update --set-remcon ttyS1,115200n8 elogin1
```

```
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

If the problem persists, proceed to the next step.

There may be a BIOS communication issue.

- c. If there seems to be a communication issue, connect to the iDRAC virtual console (see [Use the iDRAC](#) on page 202).

If the problem persists or you are unable to connect to the iDRAC remotely, proceed to the next step.

- d. Connect a monitor, keyboard, and mouse to the physical node.

7.2 The journalctl Command

systemd (on both the SMW and eLogin nodes) forgoes traditional logging mechanisms, and instead stores the following messages in a custom database:

- syslogd messages
- Kernel log messages
- Initial RAM disk and early boot messages
- Messages written to stderr/stdout for all services

Access to the information in that database is through the journalctl tool.

The command `journalctl -a` displays all kernel messages and other available information.

```
eloin1# journalctl -a
-- Logs begin at Mon 2017-11-13 17:45:11 CST, end at Wed 2017-11-15 16:35:14 CST.
--
Nov 13 17:45:11 eloin1 systemd-journald[2602]: Runtime journal (/run/log/journal/)
is
↳currently using 8.0M.
Maximum allowed usage is set to 4.0G.
Leaving at least 4.0G free (of currently
↳available 47.1G of space).
Enforced usage limit is thus 4.0G, of
↳which 3.9G are still available.
Nov 13 17:45:11 eloin1 kernel: Initializing cgroup subsys cpuset
Nov 13 17:45:11 eloin1 kernel: Initializing cgroup subsys cpu
Nov 13 17:45:11 eloin1 kernel: Initializing cgroup subsys cpuacct
Nov 13 17:45:11 eloin1 kernel: Linux version 4.4.73-5-default (geeko@buildhost)
(gcc
↳version 4.8.5 (SUSE Linux) ) #1 SMP Tue Jul 4 15:33:39 UTC 2017 (b7ce4e4)
Nov 13 17:45:11 eloin1 kernel: Command line: initrd=initrd imagename=htg.test.
↳20171113.sqsh cfg_set=p0 storage_profile=eloin1_default nfserver=10.7.1.1
smw_mgmt_
↳ip=10.7.1.1 esd_port=8449 es
```

The command `journalctl -f` function is similar to `tail -f`, displaying updates as they happen. For example, `journalctl -f /usr/sbin/ntpd monitors ntpd-related messages`. Any system daemons that produce output visible to `journalctl` can be filtered similarly.

```

eLogin# journalctl -f /usr/sbin/ntpd
-- Logs begin at Mon 2017-11-13 17:45:11 CST. --
Nov 13 17:49:08 eLogin ntpd[7706]: ntpd 4.2.8p10@1.3728-o Thu May 18 14:01:20 UTC
↳2017 (1): Starting
Nov 13 17:49:08 eLogin ntpd[7706]: Command line: /usr/sbin/ntpd -p /var/run/ntp/
ntpd.
↳pid -g -u ntp:ntp -c /etc/ntp.conf
Nov 13 17:49:08 eLogin ntpd[7712]: proto: precision = 0.142 usec (-23)
Nov 13 17:49:08 eLogin ntpd[7712]: restrict 0.0.0.0: KOD does nothing without
LIMITED.
Nov 13 17:49:08 eLogin ntpd[7712]: restrict ::: KOD does nothing without LIMITED.
Nov 13 17:49:08 eLogin ntpd[7712]: switching logging to file /var/log/ntp

```

7.3 Log File Locations

Log files on the SMW

In addition to the log files on the SMW for SMW and CLE described in other documentation, there are some specific log files of interest for the SMW-managed eLogin. All logs from the `enode` command and the `esd` and `conman` daemons will be under `/var/opt/cray/log/external`.

`/var/opt/cray/log/smwmessages-YYYYMMDD`

Many daemons log to the `smwmessages` file. The `dhcpd` (Dynamic Host Configuration Protocol Server) daemon will log messages to this file. The `dhcpd` daemon will log startup messages which may indicate problems with the DHCP configuration. As each node begins a PXE boot process, `dhcpd` will log the DHCPDISCOVER, DHCPDOFFER, DHCPDREQUEST, and DHCPDACK messages.

If there is an incorrect MAC address assigned to a node in the node registry, then when the node begins to PXE boot, a DHCPDISCOVER message with the MAC address of the node will be logged. If there is no response with DHCPDOFFER from `dhcpd` on the SMW with the management IP address (`mgmt_ip`) of the node then the node may have an incorrect `mgmt_mac` in the node registry. Use the `enode update --set-mgmt_mac` command with the correct MAC address for the node's interface on the external-management-net.

If the node begins the PXE boot process, but no DHCPDISCOVER message is logged by `dhcpd`, then there may be an Ethernet cabling or Ethernet switch problem between the node and the SMW.

`/var/log/atftpd/atftp.log`

The `atftpd` (Trivial File Transfer Protocol Server) daemon logs all TFTP transfers from files in the `/opt/tftpboot` directory structure. All files for the eLogin nodes will be under the relative path of `external/<nodename>` in this log file. This includes the messages from the PXE boot process which transfer the kernel (`vmlinux`), kernel parameters (`default`), and `initrd`, as well as the `storage.yaml` file for a node's storage profile.

`/var/log/conman.log`

The `conman` (ConMan) daemon logs its activities as it manages the consoles of nodes.

`/var/opt/cray/log/external/conman/console.<nodename>`

The `conman` daemon will storage all console messages from a particular node into `/var/opt/cray/log/external/conman/<nodename>.log` where `<nodename>` is the host name of that node.

`/var/opt/cray/log/external/enode.log`

Every invocation of the `enode` command will log to the `enode.log` file. The information includes command line arguments to `enode` as well as debugging messages that show interaction with the `esd` daemon.

`/var/opt/cray/log/external/esd.log`

The `esd` daemon logs all actions and debugging messages to the `esd.log` file. This includes interaction with the `enode` command and interactions, including state transitions, with the nodes being managed by `esd`.

`/var/opt/cray/log/external/esd-uwsgi.log`

The log for the uwsgi connection of the rest api to the nginx server.

`/var/log/nginx`

The `nginx` HTTP proxy daemon has both an `access.log` and an `error.log`. The `nginx` daemon is used by other software components on the SMW as well as uwsgi for `esd`.

Log files on the eLogin node

Log files on the eLogin node provide local information for each node. Many system services log to their standard Linux locations in `/var/log`. Most log files are only visible for the user `root`.

`/var/log/messages`

System log message files are located in `/var/log/messages` directory. The message files contain helpful information about the state of the system. Once a node has started `systemd`, the contents of `/var/log/messages` on the node will be collected.

`/root/.boot.log`

As the eLogin node boots, messages from the early `dracut` scripts are logged to `/root/.boot.log` in the `initramfs` and then transferred to the writable layer after pivoting from the `initrd` to the `SquashFS` image. If a boot fails in one of the `dracut` steps, this log file may have more information to diagnose the problem. This file will have some messages which were not sent to the console.

`/var/opt/cray/log/ansible`

All logs from running the `cray-ansible` command at boot time or interactively after the node has been booted are in `/var/opt/cray/log/ansible`.

`/var/log/dracut_stat.log`

When the eLogin node sends state messages to `esd` on the SMW during a boot, those messages and responses are stored in the `dracut_stat.log` file.

7.3.1 Ansible Logs

There are log files on the eLogin node that track work done when `cray-ansible` runs Ansible plays during installation and configuration of the system.

`/var/opt/cray/log/ansible/ansible-init`

Initial configuration of the system before `systemd` startup when `cray-ansible` runs in the init phase.

`/var/opt/cray/log/ansible/file-changelog-init`

Files changed by any Ansible plays called by `cray-ansible` in the init phase. Ansible writes change logs for most files changed by the Ansible modules affecting files: `acl`, `assemble`, `blockinfile`, `copy`, `fetch`, `file`, `find`, `ini_file`, `lineinfile`, `patch`, `replace`, `stat`, `synchronize`, `template`, `unarchive`, `xtattr`.

Initial configuration of the system before `systemd` startup when `cray-ansible` runs in the init phase.

`/var/opt/cray/log/ansible/ansible-booted`

Configuration of the system during `systemd` startup when `cray-ansible` runs in the booted phase.

`/var/opt/cray/log/ansible/file-changelog-booted`

Files changed by any Ansible plays called by `cray-ansible` in the booted phase. Ansible writes changelogs for most files changed by the Ansible modules affecting files: `acl`, `assemble`, `blockinfile`, `copy`, `fetch`, `file`, `find`, `ini_file`, `lineinfile`, `patch`, `replace`, `stat`, `synchronize`, `template`, `unarchive`, `xtattr`.

7.4 Collect Dump and Log Information for eLogin

7.4.1 Enable and Start `kdump`

Prerequisites

- Required: eLogin node is configured according to *XC™ Series SMW-managed eLogin Installation Guide*
- Required: root privileges on both the eLogin node and SMW
- Recommended: ConMan console utility is configured for the eLogin node on the SMW and used to follow the `kdump` console messages.

About this task



CAUTION:

- Critical Failure
- Make sure `kdump_low` and `kdump_high` are set at reasonable amounts of memory. If `kdump_low` is set too low, it will cause a critical failure when the `kdump` capture kernel is booted. The `kdump_low` and `kdump_high` values should always be tested and verified by performing an administrator

triggered panic test. When any system hardware is modified, the `kdump_high` and `kdump_low` values should be reviewed and tested for accuracy.

Procedure

1. Identify the eLogin node on which to configure and start the `kdump` service.
 - a. List the available eLogin nodes configured on the SMW.

```
smw# enode list
NAME          CONFIGSET      STORAGE_PROFILE  ESD_GROUP
IMAGE
MGMT_IP      MGMT_MAC      PARAMETERS      STATE           BMC_IP
smw_cle_6.0.UP06-build6.0.6191_sles_12sp3-created20180103  10.6.1.10
10.7.0.1     F8:BC:12:3B:25:70 -                cray_ansible_booted
smw_cle_6.0.UP06-build6.0.6191_sles_12sp3-created20180103  10.6.1.11
10.7.0.2     F8:BC:12:3B:5E:AC -                cray_ansible_booted
smw_cle_6.0.UP06-build6.0.6079_sles_12sp3-created20171219  10.6.1.12
10.7.0.3     18:66:DA:87:7F:92 -                cray_ansible_booted
smw_cle_6.0.UP06-build6.0.6191_sles_12sp3-created20180103  10.6.1.13
10.7.0.4     18:66:DA:EF:9F:28 -                node_up
smw_cle_6.0.UP06-build6.0.6191_sles_12sp3-created20180103  10.6.1.14
10.7.0.5     F8:BC:12:3B:40:44 -                node_up
```

- b. List the eLogin nodes along with the `kdump_enable` status, the `kdump_high` value and the `kdump_low` value configured on each.

```
smw# enode list --fields name,kdump_enable,kdump_low,kdump_high
NAME          KDUMP_ENABLE  KDUMP_LOW  KDUMP_HIGH
elogin1      False         -          -
elogin2      False         -          -
elogin3      False         -          -
elogin4      False         -          -
elogin5      False         -          -
```

In this example, the `kdump` service has not been defined on any available eLogin nodes.

2. From the eLogin node, determine the amount of memory to reserve on the eLogin node for the `kdump` service.
 - a. Find the recommended amount of high and low memory to reserve for `kdump`.

```
elogin2# kdumptool calibrate
Total: 65490
Low: 72
High: 116
MinLow: 72
MaxLow: 3281
MinHigh: 0
MaxHigh: 62208
```

All values from `kdumptool` are in MB.

- b. Calculate the amount of low memory (memory below 4GB) to reserve for kdump.

Use the following formula:

$$\text{SIZE_LOW} = (\text{Recommendation} * \text{RAM_in_TB}) + (\text{Adjustment})\text{M}$$

- *Recommendation*: Use the recommendation from previous step using `kdumpool calibrate`.
- *RAM_in_TB*: Use the value of the node's RAM in TB. (Round up to the nearest TB.)
- *Adjustment*: In order to ensure that sufficient low memory is reserved, add an arbitrary amount of memory, (in this example 40MB) up to 256MB.

Below is an example of this calculation:

$$\text{SIZE_LOW} = (\text{Recommendation} * \text{RAM_in_TB}) + (\text{Adjustment})\text{M}$$

$$\text{SIZE_LOW} = (72 * 3) + (40)\text{M}$$

$$\text{SIZE_LOW} = 256\text{M}$$

Testing has shown that reserving insufficient low memory results in the capture kernel's panic at boot time and the loss of the ability to capture memory. The following kernel panic message appears:

```
2017-11-29 11:03:38 [ 8.893214] ---[ end Kernel panic - not syncing: Can
not allocate SWIOTLB buffer earlier and can't now provide you with the DMA
bounce buffer
earlier and can't now provide you with the DMA bounce buffer
```

Reserve a larger amount of kdump low memory to resolve this problem.

- c. Calculate the amount of high memory (above 4GB) to reserve for kdump.

Use the following formula:

$$\text{SIZE_HIGH} = (\text{Recommendation} * \text{RAM_in_TB}) + (\text{LUNs}/2)\text{M}$$

- *Recommendation*: Use the recommendation from previous step using `kdumpool calibrate`.
- *RAM_in_TB*: Use the value of the node's RAM in TB. (Round up to the nearest TB.)
- *LUNs*: The maximum number of LUN kernel paths that expected to ever exist on the system. Exclude multipath devices from this number, as these are ignored.

Below is an example of this calculation:

$$\text{SIZE_HIGH} = (\text{Recommendation} * \text{RAM_in_TB}) + (\text{LUNs}/2)\text{M}$$

$$\text{SIZE_HIGH} = (116 * 3) + (6 / 2)\text{M}$$

$$\text{SIZE_HIGH} = 351\text{M}$$

In order to ensure that sufficient high memory is reserved, increase this value to 512M.

3. From the SMW, define amounts of high and low memory in the eLogin node definition.
- a. Update the node to set `kdump_enable`, `kdump_high`, and `kdump_low`.


```
smw# enode update --set-kdump_high=512M --set-kdump_low=256M --set-
kdump_enable elogin2
Updating the following node(s):
eloin2
Successfully updated ['eloin2']
```

- b. Verify the kdump parameters are set and accurate.

```
smw# enode list --fields name,kdump_enable,kdump_low,kdump_high
NAME          KDUMP_ENABLE  KDUMP_LOW  KDUMP_HIGH
eloin1        False         -          -
eloin2        True          256M       512M
eloin3        False         -          -
eloin4        False         -          -
eloin5        False         -          -
```

- c. Shutdown and reboot the node to ensure the kernel parameters for kdump memory reservation are passed in and the kdump memory is reserved.

```
smw# enode shutdown elogin2
Shutting down the following node(s):
eloin2
['eloin2']: All node(s) started shutdown process.
smw# enode status elogin2
NODE          PING  POWER          STATE
eloin2  Down  Chassis Power is off  node_off
smw# enode boot --pxe elogin2
Booting the following node(s) using mode: pxe
eloin2
['eloin2']: All node(s) started boot process.
```

- d. Verify the kdump memory is reserved on the eLogin node.

```
eloin2# cat /proc/cmdline > /tmp/cmdline
eloin2# vi /tmp/cmdline
```

Verify the following kernel parameters are listed:

```
crashkernel=512M,high crashkernel=256M,low
```

kdump creates the following dump directories and files for dump analysis.

```
eloin2# ls -al /var/crash
total 44
drwxr-xr-x 8 root root 4096 Dec 22 10:49 .
drwxr-xr-x 1 root root 4096 Jan  4 19:33 ..
drwxr-xr-x 2 root root 4096 Dec 21 12:19 2017-12-21-18:18
drwxr-xr-x 2 root root 4096 Dec 22 08:23 2017-12-22-14:22
drwxr-xr-x 2 root root 4096 Dec 22 08:42 2017-12-22-14:41
drwxr-xr-x 2 root root 4096 Dec 22 09:30 2017-12-22-15:30
drwxr-xr-x 2 root root 4096 Dec 22 10:50 2017-12-22-16:49
drwx----- 2 root root 16384 Nov 28 10:20 lost+found
eloin2# ls -al /var/crash/2017-12-22-16\:49
total 1859216
drwxr-xr-x 2 root root 4096 Dec 22 10:50 .
drwxr-xr-x 8 root root 4096 Dec 22 10:49 ..
-rw-r--r-- 1 root root 191 Dec 22 10:50 README.txt
-rw-r--r-- 1 root root 3237477 Dec 22 10:50 System.map-4.4.73-5-default
-rw----- 1 root root 95669 Dec 22 10:49 dmesg.txt
```

```
-rw----- 1 root root 1893587037 Dec 22 10:50 vmcore
-rw-r--r-- 1 root root 6890106 Dec 22 10:50 vmlinux-4.4.73-5-default.gz
```

The kdump service is now configured and ready to capture a vmcore file in the event of a kernel panic. When a panic occurs, the kdump capture kernel is booted and the kdump initrd has the required tools to capture memory to a vmcore file. kdump will create a `GMT-timestamp` directory on the `/var/crash` partition and copy all debugging evidence to that directory.

4.  **CAUTION:** This will stop all processes currently running.

Trigger a kernel panic to test the kdump configuration.

```
eLogin2# echo c > /proc/sysrq-trigger
```

This will cause the kernel to panic. The kdump capture kernel will boot and create the kdump vmcore file. When that has completed, the system will reboot to the original production kernel.

IMPORTANT: Cray recommends testing the configuration to ensure that `kdump_low` and `kdump_high` are set at reasonable values. `kdump_high` cannot be set above the max limit of memory on the system. If `kdump_low` is set too low, it will cause the kernel panic to trigger another kernel panic. Any failure due to `kdump_low` or `kdump_high` values will most likely present after the kdump service starts.

ATTENTION: After triggering kdump, `enode status` will show the node in an `Error` state. This can safely be ignored. The node is functionally in a `node_up` state and ready for use; it just appears to be in an `Error` state.

7.4.2 Configure and Run edumpsys

Prerequisites

SMW/CLE software is installed (which ensures that `xtumpsys`, `esd`, and Python 2.7 are available).

About this task

To help diagnose problems with eLogin nodes, Cray provides `edumpsys`, a functionality that enables administrators to use the Cray `xtumpsys` tool to collect eLogin logs and dumps from the SMW and eLogin nodes. If a targeted eLogin node has no dump to collect, `edumpsys` enables an administrator to trigger a kdump on that node.

`edumpsys` comprises the following:

- eLogin-related `xtumpsys` plugins, which are executed in the order indicated:
 1. eLogin Base
 2. eLogin Data Capture
 3. eLogin Gather
 4. eLogin Kdump
- A scenario file, `edumpsys.conf`, which configures those plugins and tells `xtumpsys` to execute only those plugins when collecting logs and dumps for an eLogin node. Specify the full path of this file with the `--config-file` option when invoking `xtumpsys`.

- A data capture file, `eLogin-data-capture.ini`, which defines the file globs to be collected and the commands to be executed on the SMW (targeted by default) and the targeted eLogin nodes.

The `edumpsys` functionality is provided in a separate RPM that is installed in a different location than the RPM for `xtumpsys` (by default), to prevent any interference with the use of `xtumpsys` for internal CLE nodes.

This procedure includes steps to customize `edumpsys` for this system, collect and view logs and `kdumps` on eLogin nodes, perform a workaround if there is no boot session ID, and trigger a `kdump` if an eLogin node has no `kdumps` to collect. Some of the steps in this procedure may not be needed.

Procedure

————— CUSTOMIZE EDUMPSYS FOR THIS SYSTEM —————

1. Customize eLogin data capture for this system.

To change what data is captured for this system, either edit the default data capture file, as shown in the example, or create a new data capture file. If creating a new data capture file, ensure that the scenario file can find it by editing the scenario file (`edumpsys.conf`) and specifying the new file path there.

```
smw# vi /etc/opt/cray/edumpsys/conf/eLogin-data-capture.ini
```

The following portion of the data capture file shows the default for an external node. Note that the `iptables` commands are commented out for security reasons.

Copy and paste this section for each eLogin node in this system. Replace `External Node` with the name of an eLogin node, and add/remove files and commands, as needed.

```
...
[External Node]
files =
/.imps_Image_metadata
/root/.boot.log
/var/log/dracut_stat.log
/var/opt/cray/log/ansible/*
/var/log/messages*
commands =
cat /proc/cmdline
cat /proc/cpuinfo
cat /proc/meminfo
cat /proc/filesystems
dmidecode
systemctl status kdump
#iptables commands are disabled by default for security reasons
#iptables -L input_MGMT
#iptables -L forward_MGMT
dmesg
journalctl --no-pager
...
```

2. Customize the scenario file for this system.

To make changes, edit this file and follow the guidance provided in the file.

```
smw# vi /etc/opt/cray/edumpsys/config/edumpsys.conf
```

- a. Customize the log window time for this system, as needed.

All files and kdump files that contain a time stamp in the file name will be analyzed to see if they fit into the `xtdumpsys` log window. This ensures that only the most recent files and kdump files are collected, which prevents extraneous data from being collected and reduces the time it takes to collect the data.

- b. If a new data capture file was created in the first step, specify the new location in the scenario file to ensure it can be found.
- c. Customize other settings for this system, as needed.

In addition to log window time and location of the data capture file, the following settings can be customized:

- log window length
- log window enable/disable
- which plugins to run
- individual plugin settings:
 - timeouts
 - SSH usernames
 - SSH timeouts
 - kdump collection directory

COLLECT AND VIEW LOGS AND KDUMPS

3. Collect logs and kdump files on one or more eLogin nodes.

The edumpsys use of `xtdumpsys` requires the following:

- Root access (running as `crayadm` is not supported).
- A scenario file, specified with the `--config-file` option.
- Target eLogin nodes, specified with the `--add` option.
- A boot session ID in `/opt/tftpboot/SESSION-ID.p0`, and a log directory for that boot session in `/var/opt/cray/log/` (e.g., `/var/opt/cray/log/p0-20171220t094412`).

```
smw# xtdumpsys -r "reason for dump" \
--config-file /etc/opt/cray/edumpsys/config/edumpsys.conf \
--add elogin1 elogin2
```

Trouble? If the output contains a line like this, then that node has no kdump files.

```
WARNING: eLogin Kdump: No kdump files found on elogin2
```

If it also has a line like this, then the `kdump` utility has not been enabled on that node.

```
WARNING: eLogin Kdump: NOTE: kdump must be enabled on a node before triggering.
```

- If the node has no kdump files, but the `kdump` utility is enabled, go to step 5 on page 197.
- If the node has no kdump files, and the `kdump` utility has NOT been enabled, do steps 2–5 in [Enable and Start kdump](#) on page 190, and then return to this procedure and go to step 5 on page 197.

4. View the edumpsys dump from the directory provided at the end of the `xtdumpsys` output.

The directory name will have this format: `/var/opt/cray/dump/p0-SESSION-ID-DUMP-TIME/`

edumpsys data is contained in the `edumpsys/` directory within that main dump directory, and each node gets its own directory within the `edumpsys/` directory.

```
smw# tree -ah /var/opt/cray/dump/p0-20171221t080455-1712211853/edumpsys/
/var/opt/cray/dump/p0-20171221t080455-1712211853/edumpsys/
├── [ 56] elogin2
│   ├── [582K] elogin_cmds.out
│   └── [4.0K] files
│       ├── [ 28K] .boot.log
│       ├── [1.2K] .imps_Image_metadata
│       ├── [228K] ansible-booted
│       ├── [263K] ansible-booted.1
│       ├── [244K] ansible-init
│       ├── [  0] ansible-init.1
│       ├── [1.7K] dracut_stat.log
│       ├── [ 19K] file-changelog-booted
│       ├── [  0] file-changelog-booted.1
│       ├── [ 34K] file-changelog-booted.yaml
│       ├── [  0] file-changelog-booted.yaml.1
│       ├── [ 19K] file-changelog-init
│       ├── [ 36K] file-changelog-init.yaml
│       └── [454K] messages
├── [ 37] kdumps
│   └── [1.7G] 2017-12-21-18:18.tar.gz
├── [7.9K] smw_cmds.out
├── [4.0K] smw_files
├── [5.4M] access.log
├── [ 12K] atftp.log
├── [4.0K] conman
│   ├── [140K] .console.elogin2.swp
│   ├── [341K] console.elogin
│   ├── [ 25M] console.elogin1
│   ├── [ 17M] console.elogin2
│   ├── [6.6M] console.elogin3
│   ├── [9.3M] console.elogin4
│   ├── [9.0M] console.elogin5
│   └── [135K] dwel_nooverlay_driver
├── [4.3K] conman.log
├── [ 94M] enode.log
├── [191K] error.log
├── [8.6M] esd-uwsgi.log
├── [ 77M] esd.log
└── [3.9M] smwmessages-20171221
```

TRIGGER A KDUMP ON AN ELOGIN NODE

If the following prerequisites are true, use the steps in this section to trigger a kdump on an eLogin node.

- `xtumpsys` has been run on an eLogin node.
- There are no kdumps to collect on that eLogin node.
- The `kdump` utility has been enabled on that eLogin node. (If the `kdump` utility has NOT been enabled, do steps 2–5 in [Enable and Start kdump](#) on page 190, and then return here and continue with the following steps.)

5. Trigger `kdump` on an eLogin node.

```

smw# xtdumpsys -r "trigger a kdump on elogin2" \
--config-file /etc/opt/cray/edumpsys/config/edumpsys.conf \
--add elogin2
--conf trigger_kdump=1
...
INFO: eLogin Kdump: starting thread (timeout: 1800s)
INFO: eLogin Kdump: kdump_trigger option detected
INFO: eLogin Kdump: Running 'ssh -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -tt -x -l root
elogin2 'echo "###start";find /var/crash -mindepth 1 -maxdepth 1 -type d ;echo
"###end"'
INFO: eLogin Kdump: RC=0
WARNING: eLogin Kdump: No kumps found on elogin2.
INFO: eLogin Kdump: Triggering kdump on elogin2...
INFO: eLogin Kdump: Running 'ssh -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -tt -x -l root
elogin2 'echo c > /proc/sysrq-trigger'
INFO: eLogin Kdump:
INFO: eLogin Kdump: kumps have been triggered on the following nodes: elogin2
INFO: eLogin Kdump:
INFO: eLogin Kdump: The nodes will now dump and then reboot. Once the nodes are
rebooted, you can then re-run xtdumpsys without '--
config trigger_kdump=1' to collect the kumps. Use the enode command to
determine when the nodes have rebooted.
INFO: eLogin Kdump: thread finished
INFO: eLogin Kdump: Finished in 11729 ms
INFO:
#####
INFO: # Your dump is available in /var/opt/cray/dump/
p0-20171221t080455-1712211850 #
INFO:
#####

```

This step triggers `kdump` on the eLogin node and reboots the node. However, `xtdumpsys` does not wait for `kdump` to finish, so it will not be able to collect a `kdump` for this node. The next step is necessary for collecting the `kdump` created by this step.

- When the node has completed its reboot, collect logs and the new `kdump` on the eLogin node.

```

smw# xtdumpsys -r "collect the kdump on elogin2" \
--config-file /etc/opt/cray/edumpsys/config/edumpsys.conf \
--add elogin2
...
INFO: eLogin Kdump: starting thread (timeout: 1800s)
INFO: eLogin Kdump: Running 'ssh -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -tt -x -l root
elogin2 'echo "###start";find /var/crash -mindepth 1 -maxdepth 1 -type d ;echo
"###end"'
INFO: eLogin Kdump: RC=0
INFO: eLogin Kdump: Attempting to retrieve the following kumps from elogin2: /
var/crash/2017-12-21-18:18
INFO: eLogin Kdump: Running 'ssh -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -tt -x -l root
elogin2 'tar -zcvf /var/crash/2017-12-21-18:18.tar.gz /var/crash/
2017-12-21-18:18'
INFO: eLogin Kdump: RC=0
INFO: eLogin Kdump: Running 'scp -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -r 'root@orionelogin4:/
var/crash/2017-12-21-18:18.tar.gz' /var/opt/cray/dump/
p0-20171221t080455-1712211853/edumpsys/elogin2/kumps'
INFO: eLogin Kdump: RC=0

```

```
INFO: eLogin Kdump: Running 'ssh -v -o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o ConnectTimeout=5 -tt -x -l root
eLogin2 'rm -r /var/crash/2017-12-21-18:18.tar.gz''
INFO: eLogin Kdump: RC=0
INFO: eLogin Kdump: thread finished
INFO: eLogin Kdump: Finished in 74822 ms
INFO:
#####
INFO: # Your dump is available in /var/opt/cray/dump/
p0-20171221t080455-1712211853 #
INFO:
#####
```

8 Troubleshoot eLogin Issues

This section contains a collection of procedures to use for troubleshooting eLogin issues that may arise. They include techniques for troubleshooting boot problems and disk space issues. These procedures assume familiarity with eLogin diagnostic tools.

8.1 Boot the eLogin Node with the DEBUG Shell

Prerequisites

- Node is in a `node_off` state.
- The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 110)
- CLE config set has been created
- Storage profile assigned to the node exists in the CLE config set
- Image assigned to the node exists in SquashFS format
- PE profile in `cray_image_binding` is enabled for the node
- PE image exists in SquashFS format

Booting a node should be done only when the node power is turned off and the node is in the `node_off` state. The `enode shutdown` command will put the node into this state.

About this task

Each of the boot options can operate on a single node or a group of nodes.

Procedure

PXE Boot

1. Set the DEBUG shell and boot the node.
 - To PXE boot, go to step 2 on page 200
 - To boot from disk, go to step 3 on page 201
2. Set the DEBUG shell and PXE boot the node.
 - a. Update the node to enable the DEBUG shell.

```
smw# enode update --add-parameter DEBUG=true elogin1
```


- b. Boot a single node to begin the PXE boot process.

This will use PXE boot to:

- transfer the kernel and initrd to the node,
- transfer X.509 certificates and SSH keys to the node
- prepare local storage on the node and make file systems from the storage profile assigned to the node
- transfer new global and CLE config sets to the node
- transfer the operating system image to the node
- transfer the PE image to the node if the PE profile is enabled for the node

```
smw# enode boot --pxe elogin1
```

3. Set the DEBUG shell and boot the node from disk.

- a. Update the node to enable the DEBUG shell.

```
smw# enode update --add-parameter DEBUG=true elogin1
```

- b. Stage the node with an automatic reboot from disk.

```
smw# enode reboot --staged elogin1
```

4. Start ConMan in another window to interact with the node's console terminal.

```
smw# conman -j elogin1
```

Once the node completes Power On Self Test (POST), text should appear in this window.

If the text is garbled, there may be a problem with the remcon setting for the node with a bad baud rate or there may be BIOS communication issue which requires a connection to the iDRAC via another method (see [Use the iDRAC](#) on page 202).

5. Disable DEBUG shell for next boot.

```
smw# enode update --remove-parameter DEBUG=true elogin1
```

6. Reboot the node.

- Stage the node with an automatic reboot from disk.

```
smw# enode reboot --staged elogin1
```

- Stage the node so the change will apply with the next disk boot.

```
smw# enode stage elogin1
```

- PXE boot the node.

```
smw# enode reboot --pxe elogin1
```

8.2 Troubleshoot Disk Space Issues

This procedure describes how to free up disk space on eLogin nodes if they run out of space. An eLogin node has two disks, `/dev/sda` and `/dev/sdb`, which are partitioned according to the storage profile assigned to that node. Storage profiles are defined in the `storage_profiles` setting of the `cray_storage` service in the CLE config set. The name of the storage profile can be viewed with `enode list` and updated with `enode update --set-storage_profile`.

By default `/dev/sda` is partitioned into five partitions with the labels `GRUB`, `BOOT`, `TMP`, `WRITELAYER`, and `SWAP`. These partitions are configured in a layout with `persist_on_boot` set to `false`, so they will be overwritten each time the eLogin node is PXE booted. When the node is booted from disk, the `TMP`, `WRITELAYER`, and `SWAP` partitions are all cleared, but the `BOOT` and `GRUB` partitions are not cleared. This means there should never be a need to manually free up space on the `TMP`, `WRITELAYER`, and `SWAP` partitions of `/dev/sda`. However, the `BOOT` partition may fill up if the node is staged via `enode stage` and rebooted from disk many times without ever PXE booting the node. If this occurs, the safest way to clear space is to perform a PXE boot of the node. This will clear all old image data from `BOOT` and will remove all other boot options from the `GRUB` boot menu. Only the data for the image which was PXE booted will remain in the `BOOT` partition.

By default `/dev/sdb` is partitioned into two partitions with the labels `PERSISTENT` and `CRASH`, which are mounted at `/var/opt/cray/persistent` and `/var/crash`, respectively. All data which is no longer wanted can be freely removed from the `CRASH` partition if it becomes full. It is relatively safe to remove data in the `PERSISTENT` partition, but the following data should not be removed or else rebooting the node from disk will no longer be possible:

- The global config set at `/var/opt/cray/imps/config/sets/global`.
- The CLE config set currently configured to boot on the node, e.g. `/var/opt/cray/imps/config/sets/p0`.
- The operating system image currently configured to boot on the node. This will be a directory in `/var/opt/cray/imps/image_roots`.
- The PE image which is configured in the node's current CLE config set to be mounted on the eLogin nodes

In addition, any OS images, PE images, and config sets which are configured in the `GRUB` boot loader configuration by previous `enode stage` operations should not be removed. If images which are referenced by the `GRUB` configuration are removed, those `GRUB` menu entries will no longer be functional. All previous staged eLogin images can be removed from the `GRUB` boot loader configuration by performing a PXE boot of the eLogin node. If the currently configured CLE image, CLE config set, global config set, or PE image is removed from the node, everything can be re-synchronized to the node's `PERSISTENT` partition based on the values currently configured in `enode` by performing a PXE boot or reboot of the node via `enode boot --pxe elogin_node` or `enode reboot --pxe elogin_node`.

8.3 Use the iDRAC

Prerequisites

This procedure assumes an integrated Dell Remote Access Controller (iDRAC) has been set up for use with the node.

About this task

An iDRAC enables remote management of a node. This procedure describes how to access the node console through the iDRAC.

Procedure

1. Bring up a web browser.
2. Go to: `https://cray-drac`, where `cray-drac` is the name assigned to the iDRAC during setup. The iDRAC login screen appears.
3. Enter the account user name and password set up in the iDRAC setup procedure.
The **System Summary** window appears.
4. Select **Submit**.
5. To access the SMW console, select the **Console Media** tab.
The **Virtual Console and Virtual Media** window appears.

6. Select **Launch Virtual Console**.

TIP: By default, the console window has two cursors: one for the console and one for the administrator's window environment. To switch to single-cursor mode, select **Tools**, then **Single Cursor**. This single cursor will not move outside the console window. To exit single-cursor mode, press the **F9** key.

TIP: To log out of the virtual console, kill the window or select **File**, then **Exit**. The web browser is still logged into the iDRAC.

For detailed information, see the iDRAC documentation at: <http://www.dell.com/support>.

9 Supplemental Information

9.1 Glossary

Term	Definition
ACL	access control list Permit or deny traffic based on MAC and/or IP addresses using a filter containing some criteria to match (examine IP, TCP, or UDP packets) and an action to take (permit or deny).
BMC	baseboard management controller Device used for out-of-band management of a commodity server, such as the Dell iDRAC.
CDL	Cray Development and Login Former name for an eLogin node, or external login node.
CentOS	Operating system (OS) provided by CentOS/RedHat.
CIMS	Cray Integrated Management Server Management node running Bright Computing software and Cray ESM software to manage a CDL node.
CLE	Cray Linux Environment Operating system that runs on Cray XC series nodes.
CMC	Cray Management Controller Management node running OpenStack and CSMS software to manage an eLogin node.
CMF	Configuration Management Framework Comprises the config set on the SMW, the IDS process to distribute the config set to nodes, and <code>cray-ansible</code> and the Ansible plays that run on nodes to apply configuration changes.
CSMS	Cray System Management Software
DHCP	dynamic host configuration protocol
eLogin	External login node used for application development, job submission, and access to data in a parallel file system such as Lustre or GPFS.
enode	Command line interface for <code>esd</code> , which enables system administrators to manage external node information and perform actions on external nodes.

Term	Definition
esd	external state daemon Daemon running on the SMW that holds state information for external nodes.
esLogin	external services login Former name for an eLogin node, or external login node.
ESM	external services management
external node	Any node not directly connected to the HSN. Most external nodes have local storage that holds the operating system or some other persistent storage.
GPFS	General Parallel File System
HSN	high speed network The Aries interconnect on Cray XC series systems.
iDRAC	integrated Dell Remote Access Controller The BMC device for Dell computers. The iDRAC supports IPMI connections.
IDS	IMPS Distribution Service
IMPS	Image Management and Provisioning System Uses prescriptive recipes (and package collections and repositories) to build image roots and boot images for nodes.
internal node	Any node directly connected to the HSN. Most internal nodes have no local storage, so they use a memory-based file system to hold the operating system. The boot and SDB nodes do have local storage, but not for their operating system.
IPMI	Intelligent Platform Management Interface Protocol used to communicate with a BMC device on a node to remotely control the node and access the environmental state of the node.
LAN	local area network
LLM	lightweight log manager Uses rsyslog to transfer syslog data from nodes to the SMW and optionally to a site log host.
LOM	LAN on motherboard
NFS	network file system
NIC	network interface controller
NIMS	Node Image Mapping Service Stores information needed to boot a node (boot image, config set, and kernel parameters) and provides this information to the booting process for CLE nodes.
PE	Cray Programming Environment Used for application development.

Term	Definition
PXE boot	preboot execution environment (sometimes pronounced as pixie) Specification that describes a standardized client-server environment that boots a software assembly, retrieved from a network, on PXE-enabled clients. On the client side it requires only a PXE-capable NIC, and uses a small set of industry-standard network protocols, such as DHCP and TFTP.
SLES	SUSE Linux Enterprise Server
SMW	System Management Workstation The management node for an XC series system running CLE on the XC nodes.
SOL	IPMI serial-over-LAN
TCP	transmission control protocol
TFTP	trivial file transfer protocol
ToR	top of rack A ToR switch connects nodes that are all in the same rack.
UDP	user datagram protocol
VLAN	virtual local area network
YaST	yet another setup tool A Linux operating system setup and configuration tool that is part of the SUSE Linux Enterprise distribution.

9.2 Prefixes for Binary and Decimal Multiples

The International System of Units (SI) prefixes and symbols (e.g., kilo-, Mega-, Giga-) are often used interchangeably (and incorrectly) for decimal and binary values. This misuse not only causes confusion and errors, but the errors compound as the numbers increase. In terms of storage, this can cause significant problems. For example, consider that a kilobyte (10^3) of data is only 24 bytes less than 2^{10} bytes of data. Although this difference may be of little consequence, the table below demonstrates how the differences increase and become significant.

To alleviate the confusion, the International Electrotechnical Commission (IEC) adopted a standard of prefixes for binary multiples for use in information technology. The table below compares the SI and IEC prefixes, symbols, and values.

SI decimal vs IEC binary prefixes for multiples					
SI decimal standard			IEC binary standard		
Prefix (Symbol)	Power	Value	Value	Power	Prefix (Symbol)
kilo- (kB)	10^3	1000	1024	2^{10}	kibi- (KiB)
mega- (MB)	10^6	1000000	1048576	2^{20}	mebi- (MiB)

SI decimal vs IEC binary prefixes for multiples					
SI decimal standard			IEC binary standard		
Prefix (Symbol)	Power	Value	Value	Power	Prefix (Symbol)
giga- (GB)	10^9	1000000000	1073741824	2^{30}	gibi- (GiB)
tera- (TB)	10^{12}	1000000000000	1099511627776	2^{40}	tebi- (TiB)
peta- (PB)	10^{15}	1000000000000000	1125899906842624	2^{50}	pebi- (PiB)
exa- (EB)	10^{18}	1000000000000000000	1152921504606846976	2^{60}	exbi- (EiB)
zetta- (ZB)	10^{21}	1000000000000000000000	1180591620717411303424	2^{70}	zebi- (ZiB)
yotta- (YB)	10^{24}	1000000000000000000000000	1208925819614629174706176	2^{80}	yobi- (YiB)

For a detailed explanation, including a historical perspective, see <http://physics.nist.gov/cuu/Units/binary.html>.