



XC™ Series SMW-managed eLogin Administration Guide

(CLE 6.0.UP07)

S-3021

Contents

1 About the XC Series SMW-managed eLogin Administration Guide (S-3021).....	5
2 About the SMW-managed eLogin System.....	7
2.1 About eLogin Network Architecture.....	7
2.2 About eLogin Security.....	10
2.3 About the Firewall for SMW and eLogin Nodes.....	12
2.4 About the eLogin Node Registry and Node Enrollment.....	15
2.5 About eLogin and Cray Scalable Services.....	16
2.6 About eLogin Image and Configuration Management.....	17
2.7 About the eLogin Boot and Provisioning Process.....	18
2.8 About Storage Profiles for eLogin Nodes.....	20
2.9 About the External State Daemon and eLogin Node States.....	22
2.10 About eLogin and Simple Sync.....	26
3 SMW, Network, and eLogin Configuration Information.....	29
3.1 Determine Boot Interface and MAC Address.....	30
4 Manipulate Node Registry.....	33
4.1 Register eLogin Nodes.....	33
4.2 Enroll eLogin Nodes.....	36
4.3 List eLogin Nodes.....	38
4.4 Delete eLogin Nodes.....	38
4.5 Update eLogin Nodes.....	39
5 Manage Node Life Cycle.....	46
5.1 Check eLogin Node Status.....	46
5.2 Boot eLogin Nodes.....	47
5.3 Shutdown eLogin Nodes.....	50
5.4 Reboot eLogin Nodes.....	51
6 Create eLogin Images.....	55
6.1 Create an eLogin Image.....	55
6.2 Export an eLogin Image.....	56
6.3 Create eLogin Images with imgbuilder.....	57
6.4 Assign Image to eLogin Nodes.....	58
7 Stage an eLogin Node.....	60
8 Deploy eLogin Images.....	62
8.1 Push eLogin Image Root to eLogin Node.....	62
8.2 Push PE Image Root to eLogin Node.....	63
9 Config Set Transfer.....	65

9.1 Push Config Set to eLogin Node.....	65
9.2 Run cray-ansible on eLogin Node.....	66
10 Validate an eLogin Node.....	67
11 Enable SMW HA Management of eLogin.....	70
11.1 Disable SMW HA Management of eLogin	72
12 Hardware Configuration.....	75
12.1 Change the eLogin BIOS and iDRAC Settings.....	75
12.2 Configure SSDs on eLogin Nodes.....	86
13 Manage Partitions and Persistent Data on an eLogin Node.....	96
13.1 Reprovision a Persistent Disk on an eLogin Node.....	96
13.2 Reprovision a Nonpersistent Disk on an eLogin Node.....	100
13.3 Configure the eLogin RAID Virtual Disks.....	103
14 File System Configuration.....	117
14.1 AutoFS.....	117
14.2 Connect eLogin Nodes to a Lustre File System.....	117
15 Other Configuration Options.....	119
15.1 Change the Firewall Configuration.....	119
15.1.1 Ensure that NFS Port is Open in Firewall.....	123
15.2 PBS License Server Configuration.....	124
15.3 User Authentication.....	124
15.4 Configure Passwordless SSH.....	124
15.5 Configure eProxy to Wrap Reduced Set of Commands.....	125
15.6 Use eProxy Utility.....	127
16 Diagnostics.....	129
16.1 Access the eLogin Console.....	129
16.2 The journalctl Command.....	130
16.3 Log File Locations.....	131
16.3.1 Ansible Logs.....	132
16.4 Enable and Start kdump.....	133
16.5 Analyze KDUMP vmcore Files.....	137
16.6 Configure and Run edumpsys.....	140
17 Troubleshooting.....	146
17.1 Boot the eLogin Node with the DEBUG Shell.....	146
17.2 Use the iDRAC.....	147
17.3 Troubleshoot Disk Space Issues.....	148
17.4 Disable the Intel TOC Watchdog Timer on eLogin Nodes.....	149
18 Supplemental Information.....	151
18.1 Prefixes for Binary and Decimal Multiples.....	151

18.2 Glossary.....	151
--------------------	-----

1 About the XC Series SMW-managed eLogin Administration Guide (S-3021)

The *XC™ Series SMW-managed eLogin Administration Guide (S-3021)* provides information and procedures to manage the 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 does not include eLogin installation procedures; for those see *XC™ Series SMW-managed eLogin Installation Guide (S-3020)*.

New for CLE 6.0 UP07

The `--hard` argument was added in the [Shutdown eLogin Nodes](#) on page 50 and [Reboot eLogin Nodes](#) on page 51 procedures to preform a hard shutdown.

Table 1. Record of Revision

Publication Title	Date	Release
<i>XC™ Series SMW-managed eLogin Administration Guide (CLE 6.0 UP07) S-3021</i>	12 July 2018	CLE 6.0 UP07 / SMW 8.0 UP07
<i>XC™ Series SMW-managed eLogin Administration Guide (CLE 6.0 UP06 Rev B) S-3021</i>	June 2018	CLE 6.0 UP06 / SMW 8.0 UP06
<i>XC™ Series SMW-managed eLogin Administration Guide (CLE 6.0 UP06 Rev A) S-3021</i>	28 Mar 2018	CLE 6.0 UP06 / SMW 8.0 UP06
<i>XC™ Series SMW-managed eLogin Administration Guide (CLE 6.0 UP06) S-3021</i> NOTE: S-3021 supersedes S-2570. There will be no revisions of S-2570 in this or future releases.	01 Mar 2018	CLE 6.0 UP06 / SMW 8.0 UP06
<i>XC Series eLogin Administration Guide (CLE 6.0 UP05) S-2570</i>	05 Oct 2017	CLE 6.0 UP05 / CSMS 1.1.4

Scope and Audience

This publication is written for experienced system administrators.

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 GUI Window , GUI element , cascading menu (Ctrl → Alt → Delete), or key strokes (press Enter).
\ (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).

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, ClusterStor, 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.

2 About the SMW-managed eLogin System

2.1 About eLogin Network Architecture

eLogin Networks

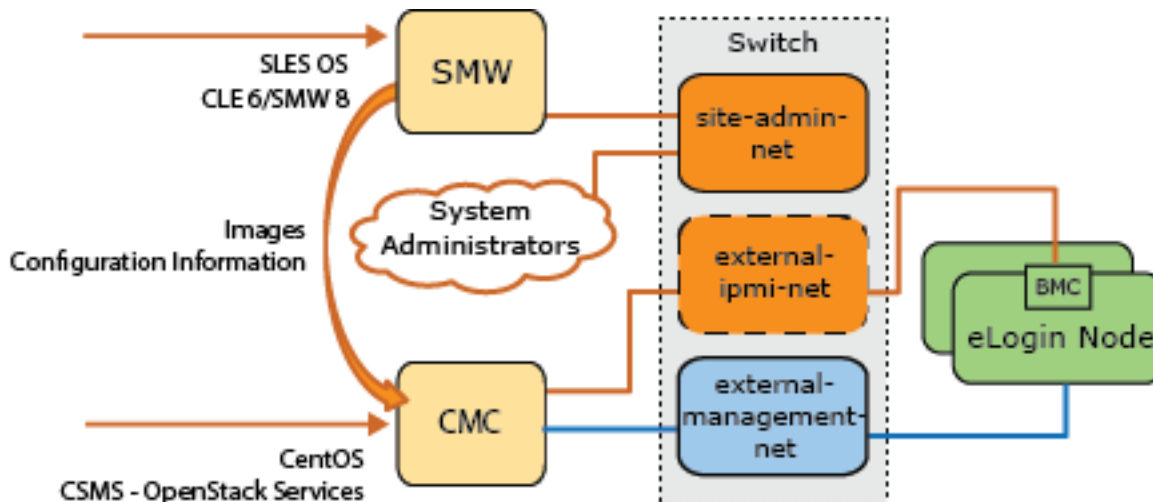
The following networks are used to connect eLogin nodes to the SMW, users, and the Cray XC system.

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 site-admin-net, but instead be connected to a separate network with tighter access control.
external-ipmi-net	<p>External management network that connects the SMW to the eLogin IPMI devices. This network enables remote console and power management of eLogin nodes.</p> <p>The dedicated IPMI device port of each eLogin node must be connected to the IPMI network.</p>
external-management-net	<p>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.</p> <p>The first 1GbE device of each eLogin node must be connected to external-management-net. 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.</p>
site-user-net	<p>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.</p> <p>IMPORTANT: The site-user-net network must be configured as <code>site</code> in the network section of the <code>cray_net</code> configuration service.</p> <p>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.</p>
IB Net	Internal Infiniband® network used for high-speed Lustre LNet traffic.

Connection of eLogin Nodes to the SMW

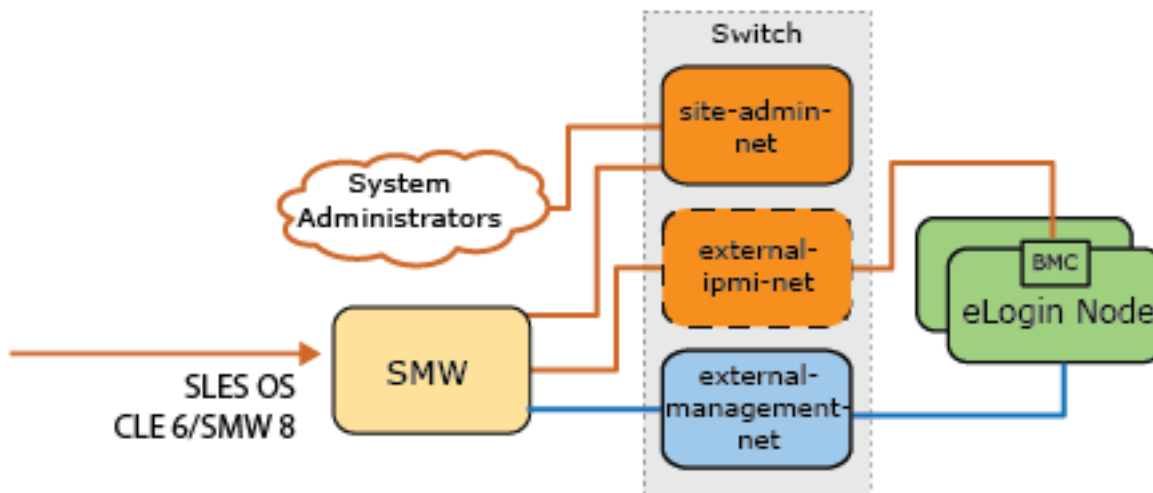
In CLE 6.0 releases prior to UP06, the SMW was used to prepare image and configuration data for eLogin nodes and push that data to the CMC (Cray Management Controller), which was used to provision and manage eLogin nodes. The following figure shows the network topology connecting SMW, CMC, and eLogin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



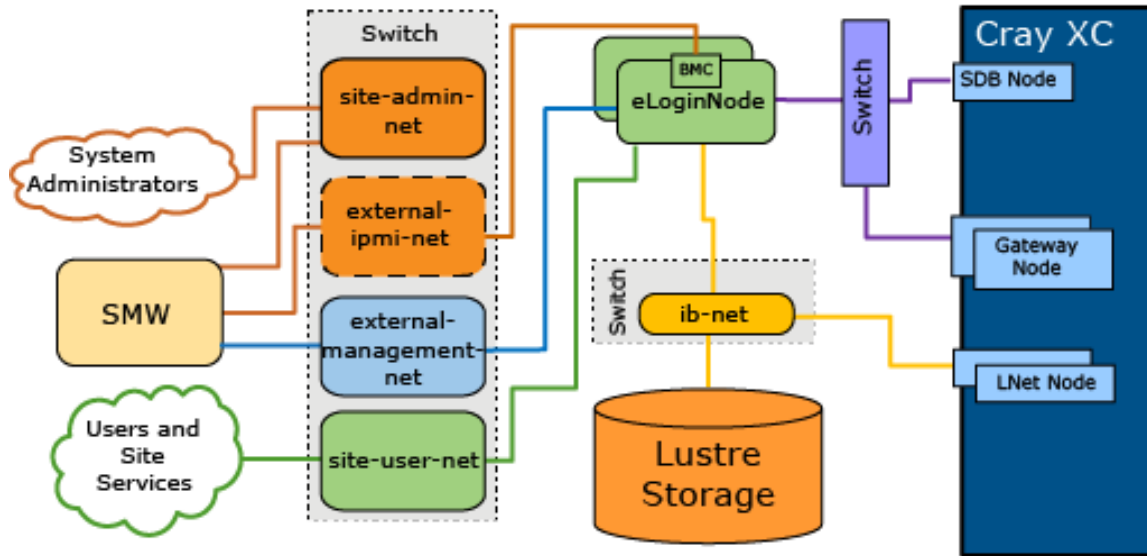
Connection of eLogin Nodes to the XC System

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.

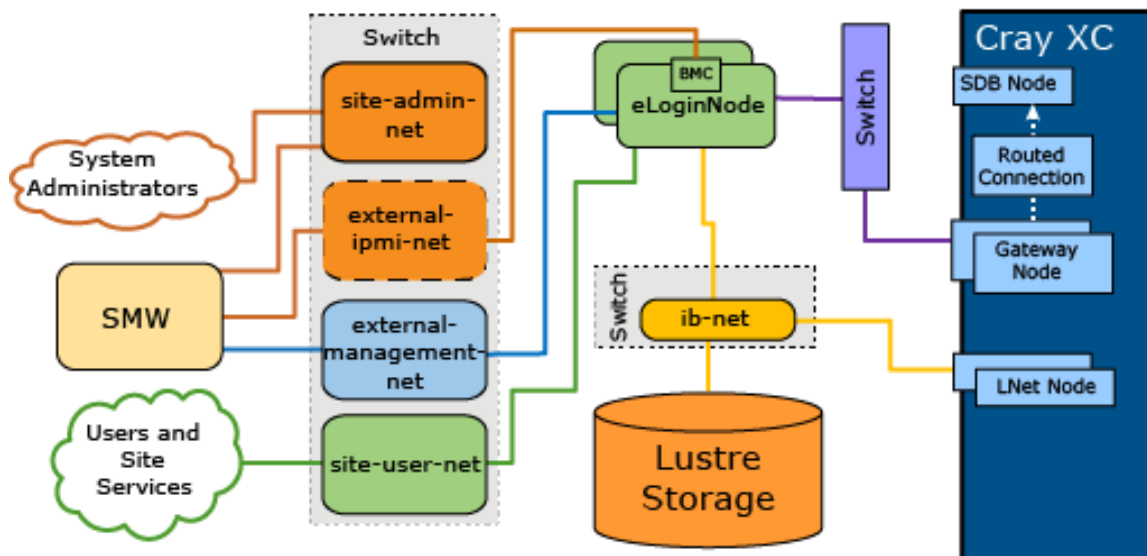
- **eLogin Nodes Connected to SDB Node and to Gateway Node via Switch.** Jobs are submitted from an eLogin node directly to the SDB node.

Figure 3. eLogin Connected to SDB and Gateway via Switch



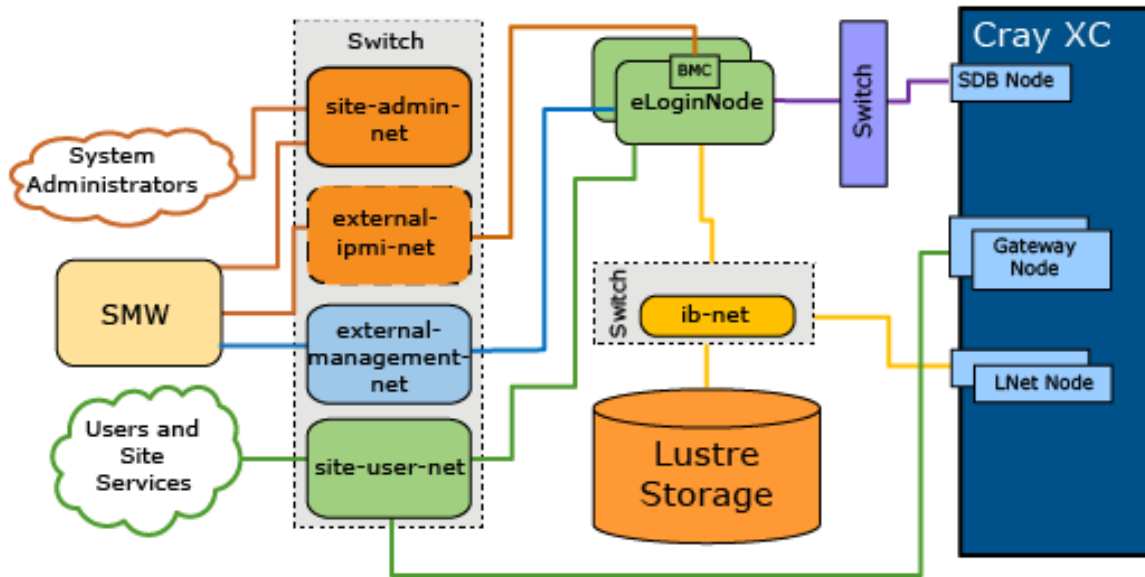
- **eLogin Nodes Connected to SDB Node via Routed Connection from Gateway Node and to Gateway Node via Switch.** Jobs are submitted from an eLogin node through the gateway node to the SDB node.

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



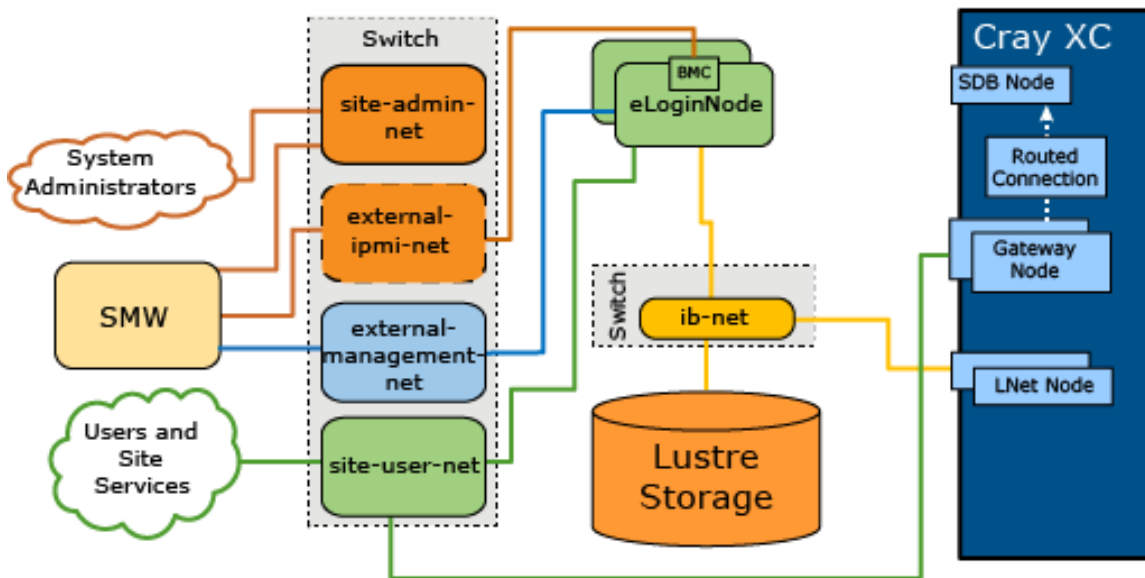
- **eLogin Nodes Connected to SDB Node via Switch and to Gateway Node via Site User Net.** Jobs are submitted from an eLogin node directly to the SDB node. Users access the gateway node directly from the site user net.

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

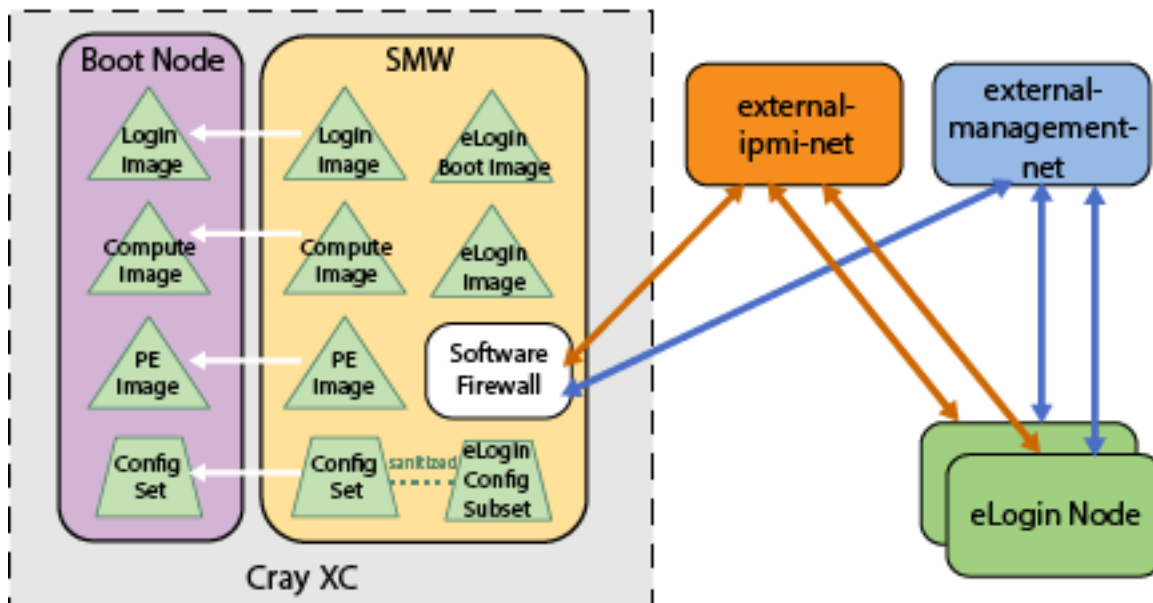
Security for a Cray XC system with SMW-managed eLogin nodes is implemented through network security, data security, and console security.

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).

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.

About disabling the firewall. A site that disables the `cray_firewall` configuration service is assuming responsibility for managing the firewall for the XC system and all eLogin nodes. A site that chooses to manage the firewall directly must ensure that certain ports are opened on eLogin nodes so that they can receive information from the SMW. For more information, see the section below titled [Site-Managed Firewall Considerations](#).

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 2. 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 3. 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
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 119.

Site-Managed Firewall Considerations

Sites that disable `cray_firewall` so that they can directly manage the firewall on a Cray XC system and eLogin nodes must ensure that the following ports are opened on eLogin node to enable it to receive information from the SMW:

- `ntp`
- `sshd`
- `nfs-client`

The following excerpt from the `/etc/sysconfig/SuSEfirewall12` configuration file on an eLogin node shows an example configuration.

```
FW_DEV_EXT="eth3"
FW_CONFIGURATIONS_EXT="sshd"
FW_ZONES="MGMT"
FW_ZONE_DEFAULT=''
FW_LOAD_MODULES="nf_conntrack_ipv4"
FW_DEV_MGMT="eth2"
FW_CONFIGURATIONS_MGMT="ntp sshd nfs-client"
```

The `eth3` interface is the interface out to the site network (site-user-net) for users. It has only `sshd` opened because the only thing users can do is log in, and this is in the EXT (external) untrusted zone. The MGMT zone needs to allow in `ntp` for time synchronization, `sshd` so an admin can `ssh` in from the SMW, and `nfs-client` so that it can receive NFS communication from the SMW because it is NFS-mounting file systems from the SMW.

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.

The data in the node registry is written to `/var/opt/cray/imps/esd/node_info` on the boot RAID, so in the case of an SMW HA system, the information is migrated between SMWs during a failover. Note that this registry data is not part of a snapshot, so like config sets, the same content will be available from all snapshots.

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 `conman` 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 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.

2.6 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: `ellogin_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.

2.7 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.

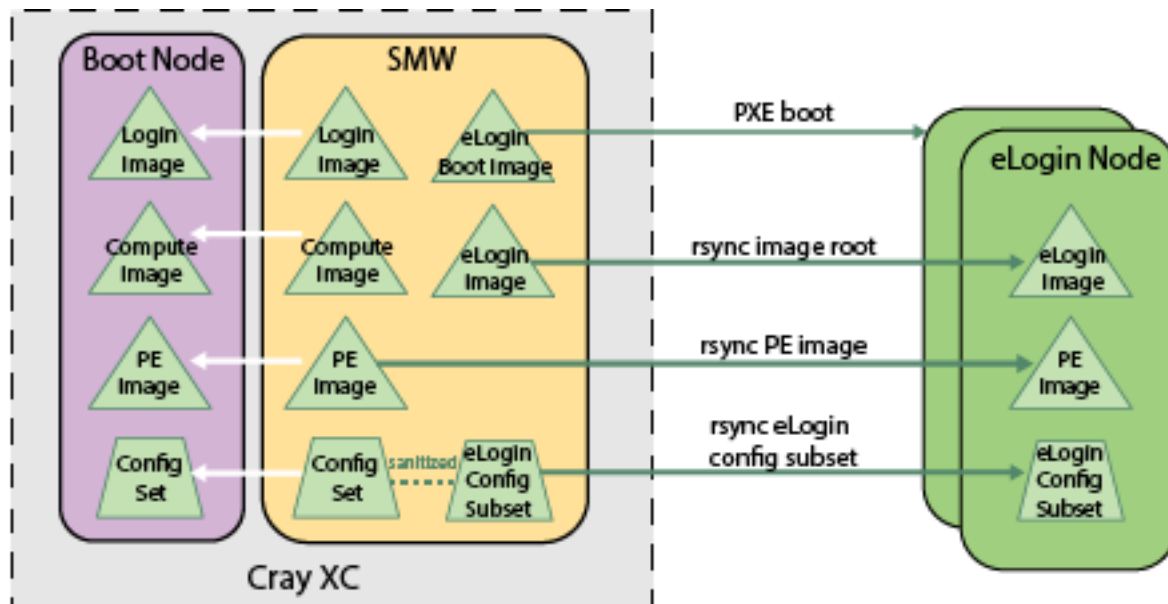
Distribution of eLogin Image and Config Set to eLogin Nodes

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 squash` 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.

2.8 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, use the following command.

```
smw# cfgset search -t eloin_default p0
```

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 <code>ext3</code> .
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 151.)
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 `eloin_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 `eloin_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 `eloin_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 `eloin_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 96.

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.

2.9 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 4. 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	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. 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.
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.

Node State	esd Actions
Error	<p>The <code>esd</code> 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, <code>esd</code> 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 5. 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:			
storage_send	yes	N/A	Node requests access to storage configuration from esd and then transfers the storage.yaml via TFTP. Storage

Node State	Occurs During		Node Actions
	PXE Boot	Disk Boot	
			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).
<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			

Node State	Occurs During		Node Actions
	PXE Boot	Disk Boot	
staging	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.

2.10 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.

<code>./common/files/</code>	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 <hostname> directory and the files directory.
<code>./nodegroups/<node_group_name>/files/</code>	Targets all nodes in the specified node group. The directories for this nodegroups directory are automatically stubbed out when the config set is updated after node groups are defined and configured in the <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 “`eloin_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)*.

3 SMW, Network, and eLogin Configuration Information

SMW and Network Configuration Information

The following table lists the SMW and network information needed to connect the SMW to eLogin nodes.

Table 6. 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		

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.

Table 7. eLogin Node Configuration Information

Item	Value for this Node
name	
BMC IP address	
BMC user ID	

Item	Value for this Node
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 30.

3.1 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

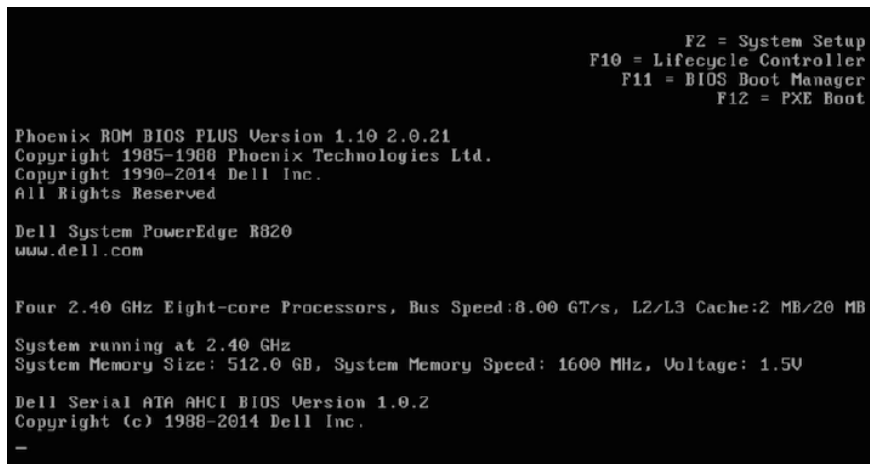
Procedure

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



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

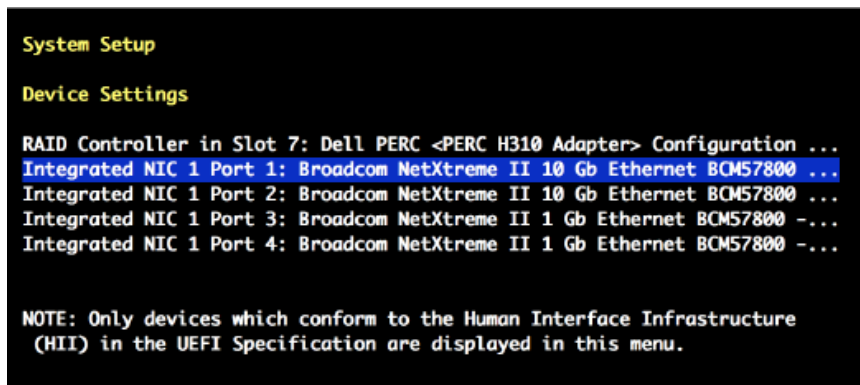


3. Select **Integrated NIC 1 Port N ...** in the **Device Settings** window.

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 ...**

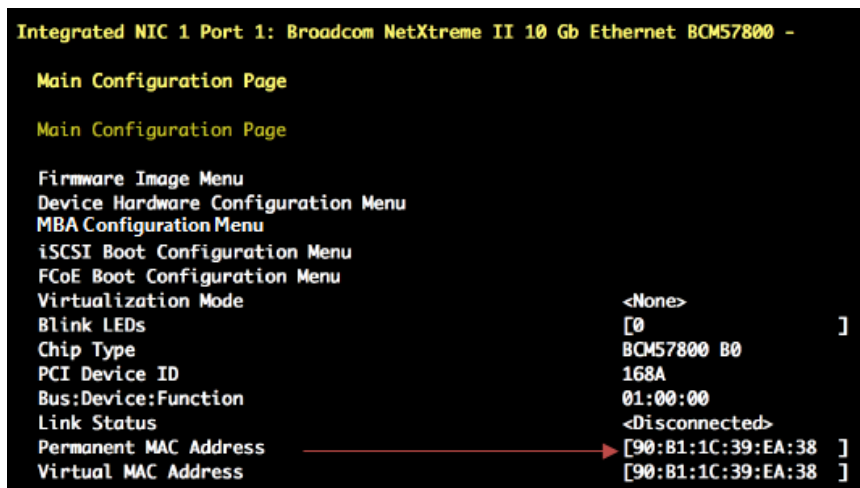
Figure 11. Device Settings: Integrated NIC Port Number



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



6. Record the MAC address and the boot interface in the table in [SMW, Network, and eLogin Configuration Information](#) on page 29.
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 Manipulate Node Registry

4.1 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 36).

Any parameters not set when the node is created can be set or changed later with `enode update` (see [Update eLogin Nodes](#) on page 39).

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 8. `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_group(s)</code>
<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>

Command Line Option	Description
-p, --bmc_password	File containing node's bmc_password
-m, --mgmt_ip	Set mgmt_ip
-a, --mgmt_mac	Set mgmt_mac
--rootdev	Set root device for selected nodes
--bootif	Set boot interface for selected nodes
--remcon	Set remote console for selected nodes
--pci	Set PCI kernel parameter for selected nodes
-d, --kdump_enable	Enable kdump functionality. Requires kdump_high and kdump_low
--kdump_high	Set kdump high memory kernel parameter value
--kdump_low	Set kdump low memory kernel parameter value
-k, --parameters	Set space-delimited kernel parameter key or key-value pairs
--ssh_host_keys	Set how SSH host keys are handled. Defaults to simple_sync

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:
elogin1
Successfully created ['elogin1'].
```

- 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:
elogin1
Successfully updated ['elogin1'].
```

- 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 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 \
--configset p0 --storage_profile eloin_default \
--image eloin-smw-large_cle_6.0.UP07-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 eloin1
Updating the following node:
eloin1
Successfully updated ['eloin1'].
```

- 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 eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- c. If the boot disk for the node is not the first disk, change the `rootdev` parameter.

```
smw# enode update --set-rootdev /dev/sdb eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- 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 eloin1
Updating the following node(s):
eloin1
Successfully updated ['eloin1']
```

- e. If the PCI options for the node is not `bfsort`, change the `pci` parameter.

```
smw# enode update --set-pci bfsort eloin1
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):
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.UP07-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 146.

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']
```

4.2 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 (`ellogin1`), 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
ellogin1,10.148.0.1,11:22:33:44:55:66,32,x86_64,131072,550,ellogin1
ellogin2,10.148.0.2,11:22:33:44:55:77,32,x86_64,131072,550,ellogin2
```

- 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 39.

4.3 List eLogin Nodes

Prerequisites

One or more nodes are in the node registry (see [Register eLogin Nodes](#) on page 33).

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
eloin1 p0 eloin_default eloin - 10.6.0.1 10.7.0.111:22:33:44:55:66 - UNKNOWN
```

2. Display specific, 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
eloin1 root /dev/sda eth0 ttyS1,115200n8 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 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 bfsort ttyS1,115200n8 root False - - simple_sync
```

4.4 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=elogin --fields name,esd_group
NAME ESD_GROUP
elogin1 elogin
elogin2 elogin
elogin3 elogin
```

b. Delete the nodes.

```
smw# enode delete elogin1 elogin2 elogin3
Deleting the following node(s):
elogin1
elogin2
elogin3
Successfully deleted ['elogin1','elogin2','elogin3']
```

`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.

4.5 Update eLogin Nodes

Prerequisites

- Nodes must already be in the node registry (see [Register eLogin Nodes](#) on page 33)

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 60.

The following fields can be updated after the node creation with `enode update`. All fields must be accurate for a successful boot.

Table 9. 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. group may be a single value or a comma-separated list of group names.
	<code>--unset-group</code> or <code>-G</code>	Clears group(s) from the node.
configset	<code>--set-configset</code> or <code>-c</code>	CLE config set to assign the node.
	<code>--unset-configset</code> or <code>-C</code>	Clear CLE config set for the node.
storage_profile	<code>--set-storage_profile</code>	Storage profile in the CLE config set describing the disk setup of the node.
	<code>--unset-storage_profile</code>	Clear storage profile for the node.
image	<code>--set-image</code> or <code>-i</code>	Boot image to assign the node.
	<code>--unset-image</code> or <code>-I</code>	Clear boot image from the node.
bmc_ip	<code>--set-bmc_ip</code> or <code>-b</code>	IP address of the baseboard management controller (BMC)
	<code>--unset-bmc_ip</code> or <code>-B</code>	Restores <code>bmc_ip</code> to default value (0.0.0.0)
bmc_username	<code>--set-bmc_user</code> or <code>-u</code>	User name to connect to the BMC
	<code>--unset-bmc_user</code> or <code>-U</code>	Resets <code>bmc_username</code> to default value (root)

Field	enode update Option	Definition
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).
bootif	--set-bootif	Management/boot interface. The default value is eth0, 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 eth2(
	--unset-bootif	Resets bootif to the default value (eth0)
remcon	--set-remcon	Remote console device setting for the path and baud rate of the console.
	--unset-remcom	Resets remcon to the default value (/dev/ttyS1,115200) for the path and baud rate of the console
pci	--set-pci	PCI kernel parameter. The default value is bfsort
	--unset-pci	Resets pci to the default value (bfsort)

Field	enode update Option	Definition
kdump_enable	kdump_enable	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
	--unset-kdump_low	Unsets the <code>kdump_low</code> parameter for the node
ssh_host_keys	--set-ssh_host_keys	Set how SSH host keys are handled. The default value is <code>simple_sync</code>
	--unset-ssh_host_keys	Restore SSH host key handling to the default (<code>simple_sync</code>)
parameters	--set-parameter or -k	String of kernel parameters to assign the node
	--unset-parameter or -K	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

1. 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.

2. Update fields required for booting the nodes.

- a. 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']
```

- b. 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']
```

- c. Update the image field to use a new image.

```
smw# enode update --set-image image_name eloin1
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 eloin1
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 eloin1
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 eloin1
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):
elogin1
Successfully updated ['elogin1']
```

- 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):
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`).

4. Set the group to assign to a node.

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

5. To update `kdump_enable`, `kdump_high`, and `kdump_low` fields, go to [Enable and Start kdump](#) on page 133.

6. Update the `parameters` field to enable or disable debug.

See [Boot the eLogin Node with the DEBUG Shell](#) on page 146 for more information.

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

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 eloin-smw-
large_cle_6.0.up07_sles_12sp3-20180520 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 KDUMP_ENABLE KDUMP_HIGH
KDUMP_LOW SSH_HOST_KEYS
eloin1 p0 eloin_default eloin eloin-smw-
large_cle_6.0.up07_sles_12sp3-20180520 10.6.0.1 10.7.0.1 90:B1:1C:3A:0A:1B -
UNKNOWN /dev/sda eth0 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 60.

5 Manage Node Life Cycle

5.1 Check eLogin Node Status

Prerequisites

The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 33)

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
```

```

eloin1 Down Chassis Power is off Error
State prior to error: node_off
eloin1 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
eloin1 ERROR 2: Encountered an error during shutdown

```

5.2 Boot eLogin Nodes

Prerequisites

- Node registry has required fields for each node (see [Register eLogin Nodes](#) on page 33)
- 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 46).

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 50).

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 147).

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 147).

- 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 147).

- 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.

5.3 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 (180 seconds), the power will be turned off for the node. Add the `--hard` argument for a hard shutdown.

Procedure

1. Shut down a single node.

```
smw# enode shutdown elogin1
```

`enode shutdown` will first attempt a graceful shutdown of the operating system. If the node does not respond within the timeout period (180 seconds), the power will be turned off for the node.

To perform a hard shutdown, rather than waiting for the timeout, use the following command:

```
smw# enode shutdown --hard 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
```

`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.

To perform a hard shutdown, rather than waiting for the timeout, use the following command:

```
smw# enode shutdown --hard elogin1 elogin2 elogin3
```

- a. Start ConMan in another window to watch the console messages as the node shuts down.

This is an optional (but recommended) 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.

5.4 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 60.

`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).

NOTE: To perform a hard reboot of the node, add the `--hard` argument.

```
smw# enode reboot --type_of_reboot --hard elogin1
```

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 147).

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 147).

- 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 147).

- 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.

6 Create eLogin Images

6.1 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 57. 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_6.0.up07_sles_12sp3_ari` and `compute_cle_6.0.up07_sles_12sp3_ari`), then choose the smaller eLogin recipe: `eloin-smw_cle_6.0.up07_sles_12sp3_ari`.
- If the internal nodes are using `netroot` recipes, which have “large” as part of their name (e.g., `login-large_cle_6.0.up07_sles_12sp3_ari` and `compute-large_cle_6.0.up07_sles_12sp3_ari`), then choose the larger eLogin recipe: `eloin-smw-large_cle_6.0.up07_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-elogin-smw-large_cle_6.0.up07_sles_12sp3_ari
```

- b. Add `elogin-smw-large_cle_6.0.up07_sles_12sp3_ari` as a sub-recipe.

```
smw# recipe update -i elogin-smw-large_cle_6.0.up07_sles_12sp3_ari \
custom-elogin-smw-large_cle_6.0.up07_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 (S-2393)*.

3. Build the eLogin image.

For custom recipes:

```
smw# image create -r custom-elogin-smw-large_cle_6.0.up07_sles_12sp3_ari \
custom-elogin-smw-large_cle_6.0.up07_sles_12sp3-YYYYMMDD
```

For "elogin-smw" recipes:

```
smw# image create -r elogin-smw-large_cle_6.0.up07_sles_12sp3_ari \
elogin-smw-large_cle_6.0.up07_sles_12sp3-YYYYMMDD
```

Cray recommends appending a date stamp, such as "YYYYMMDD", to images created with `image create`. For example, if generating an image from the recipe `elogin-smw-large_cle_6.0.up07_sles_12_ari` on June 1, 2018, the image should be named `elogin-smw-large_cle_6.0.up07_sles_12_ari_20180601`. If `imgbuilder` is used to create the image, it will add date stamps automatically.

6.2 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 (the eLogin 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_6.0.up07_sles_12sp3-YYYYMMDD
```

This image name in this example uses the naming convention suggested in [Create an eLogin Image](#) on page 55.

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.

6.3 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 and a new placeholder for the release in both recipe and image name.

Old format for CLE boot images in `cray_image_groups.yaml`

```
- recipe: "admin_cle_6.0up07_sles_12sp3_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_{cle_release_lowercase}_sles_12sp3_ari"
  dest: "admin{note}_cle_{cle_release_lowercase}-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 eLogin image is a companion to `tmpfs` style images for CLE login and compute nodes.

```
- recipe: "eloin-smw_cle_{cle_release_lowercase}_sles_12sp3_ari"
  dest: "eloin-smw{note}_cle_{cle_release_lowercase}-build{cle_build}{patch}_sles_12sp3-created{date}"
  export_format: "squashfs"
```

The "eLogin-large" image is a companion to `netroot` images for CLE login and compute nodes.

```
- recipe: "eloin-large-smw_cle_{cle_release_lowercase}_sles_12sp3_ari"
  dest: "eloin-smw-large{note}_cle_{cle_release_lowercase}-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.UP07 / CLE 6.0.UP07.

```
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_{cle_release_lowercase}_sles_12sp3_ari"
    dest: "eloin-smw-large{note}_cle_{cle_release_lowercase}-build{cle_build}{patch}_sles_12sp3-created{date}"
    export_format: "squashfs"
```

This example adds a customized "custom-eloin-smw-large" recipe to the `default` group of recipes to build.

```
cray_image_groups:
  default:
  ...
  - recipe: "custom-eloin-smw-large_cle_{cle_release_lowercase}_sles_12sp3_ari"
    dest: "custom-eloin-smw-large{note}_cle_{cle_release_lowercase}-build{cle_build}{patch}_sles_12sp3-
created{date}"
    export_format: "squashfs"
```

2. Run `imgbuilder` to create and export the desired set of images.

```
smw# imgbuilder --map
```

This will create an image with a name similar to "eloin-smw-large_cle_6.0.up07-build6.0.7128_sles_12sp3-created20180724" and export it as a SquashFS file.

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.

6.4 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 `eloin1`.

```
smw# enode update \  
-i custom-eloin-smw-large_cle_6.0.up07_sles_12sp3-YYYYMMDD eloin1
```

2. Assign an image to multiple eLogin nodes.

This example assigns a custom eLogin image to a space-separated list of two eLogin nodes: *ellogin1* and *ellogin2*.

```
smw# enode update \  
-i custom-ellogin-smw-large_cle_6.0.up07_sles_12sp3-YYYYMMDD ellogin1 ellogin2
```

7 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 66.

8 Deploy eLogin Images

8.1 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 60). `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.

8.2 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 PE image root by staging a node (see [Stage an eLogin Node](#) on page 60). `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 reboot of the node to transfer the content.

Procedure

1. Export the new PE image to squashfs on the SMW.

```
smw# image export --format squashfs PE_image
```

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 `elogin_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 the `elogin_image_binding.yaml` Ansible playbook on the node to re-mount and bind the image.

```
smw# ssh elogin1 "ansible-playbook /etc/ansible/elogin_image_binding.yaml"
```


9 Config Set Transfer

9.1 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 60.

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 66.

9.2 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
```

10 Validate an eLogin Node

Prerequisites

The eLogin node is in state `node_up` after completion of an initial deployment, migration, update, or other change to the node.

About this task

This procedure tests basic eLogin node functionality and should be performed by a system administrator (as `root` or `crayadm`) to ensure that the eLogin node is ready to be released to users. A user other than `root` or `crayadm` (`user@eloin>`) can also run these commands.

By first setting up passwordless Secure Shell (SSH), a user can run commands without entering a password.

Procedure

1. Log in to the eLogin node.

2. Generate an SSH key pair.

```
crayadm@eloin> ssh-keygen
```

3. Add the key pair to the `.ssh/authorized_keys` file on the login node of the Cray XC system.

NOTE: This step is performed on the internal login node, not on the eLogin node. All other steps are performed on the eLogin node.

```
crayadm@login_hostname> ssh-copy-id eloin_name
```

4. Test the eProxy utility.

```
crayadm@eloin> cnselect
20-27,32-43,48-51,60-63
crayadm@eloin> xtproadmin
```

NID	(HEX)	NODENAME	TYPE	STATUS	MODE
1	0x1	c0-0c0s0n1	service	up	interactive
2	0x2	c0-0c0s0n2	service	up	interactive
5	0x5	c0-0c0s1n1	service	up	interactive
6	0x6	c0-0c0s1n2	service	up	interactive
20	0x14	c0-0c0s5n0	compute	up	interactive
21	0x15	c0-0c0s5n1	compute	up	interactive
22	0x16	c0-0c0s5n2	compute	up	interactive

```
...
crayadm@eloin> xtnodestat
```

C0-0	
n3	;; ;;X; ;

```

n2 SS    ;;S;;;X;  ;
n1 SS    ;;S;;;X;  ;
c0n0     ;;   ;;X;  ;
s0123456789abcdef

```

Legend:

nonexistent node	S	service node
; free interactive compute node	-	free batch compute node
A allocated (idle) compute or ccm node	?	suspect compute node
W waiting or non-running job	X	down compute node
Y down or admin down service node	Z	admin down compute node

Available compute nodes: 28 interactive, 0 batch

5. Test the aprun command (if no workload manager is configured on the system).

```

crayadm@ellogin> aprun hostname
nid00020
Application 21221 resources: utime ~0s, stime ~0s, Rss ~4256, inblocks ~0,
outblocks ~0

```

6. Test PBS or Moab/TORQUE (if installed on the system).

```

crayadm@ellogin> pbsnodes -a
percival-pl_305
  Mom = nid00008,nid00043
  ntype = PBS
  state = free
  pcpus = 8
  resv_enable = True
  sharing = force_exclhost
  resources_available.arch = XT
  resources_available.host = percival-pl_305
  resources_available.mem = 67108864kb
  ...
crayadm@ellogin> qstat

```

Job id	Name	User	Time Use	S	Queue
2034657.sdb	STDIN	crayadm	00:00:00	R	workq

```

crayadm@ellogin> qsub -I
qsub: waiting for job 2034657.sdb to start
qsub: job 2034657.sdb ready

```

7. Test Slurm (if installed on the system).

```

crayadm@ellogin> squeue

```

JOBID	USER	ACCOUNT	NAME	ST	REASON	START_TIME	TIME	TIME_LEFT
131669	xmp	(null)	testMPI	R	None	10:37:09	1:36	
8:24	5	10						
131543	ymp	(null)	sst	R	None	09:27:32	1:11:13	
2:48:47	2	32						
131534	c90	(null)	bash	R	None	09:23:31	1:15:14	
4:44:46	1	24						

```

crayadm@ellogin> sinfo

```

PARTITION	AVAIL	JOB_SIZE	TIMELIMIT	CPUS	S:C:T	NODES	STATE	NODELIST
workq*	up	1-infini	infinite	48	2:12:2	1	drained	nid00022

```
workq*      up      1-infini   infinite    48 2:12:2      5 mixed
nid000[13-15,20-21]
workq*      up      1-infini   infinite    48 2:12:2      5 allocated
nid000[08-12]
workq*      up      1-infini   infinite    48 2:12:2     41 idle
nid000[23-63]

crayadm@eloin> salloc
salloc: Granted job allocation 131674
```

11 Enable SMW HA Management of eLogin

Prerequisites

This procedure assumes the following:

- 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.

About this task

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. Stop the HA software from managing the cluster resources.

```
smw# maintenance_mode_configure enable
```

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.

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 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

5. 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 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
6. Click **OK** after all of the **Network Settings** have been prepared.
7. Exit maintenance mode and wait for the cluster to stabilize.

```
smw# maintenance_mode_configure disable
smw# sleep 300
```

8. Check cluster status.

```
smw1# crm status
```

```
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)*.

9. Log into the host name of the active SMW.

```
node# ssh smw
```

10. 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.

11. Exit maintenance mode and wait for the cluster to stabilize.

```
smw# maintenance_mode_configure disable
smw# sleep 300
```

12. Check cluster status.

```
smw1# crm status
```

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).

11.1 Disable SMW HA Management of eLogin

Prerequisites

SMW HA is configured to manage the eLogin nodes (see [Enable SMW HA Management of eLogin](#) on page 70).

Procedure

1. Check cluster status.

```
smw1# crm status
```

Note that eLogin resources appear in the `crm status` output.

```
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)*.

2. Log into the host name of the active SMW.

```
node# ssh smw
```

3. Remove eLogin resource from the cluster using `SMWHAconfig`.

```
smw# cd /opt/cray/ha-smw/default/hainst
smw# ./SMWHAconfig --update --remove_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 300
```

5. Check cluster status.

```
smw1# crm status
```

The eLogin resources (ClusterIP6, ClusterIP7, esd, and Resource Group eLoginGroup) should no longer appear in the `crm status` output.

```
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)*.

12 Hardware Configuration

12.1 Change the eLogin BIOS and iDRAC Settings

Prerequisites

- Access to the console of each eLogin node being configured

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 13. 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**.

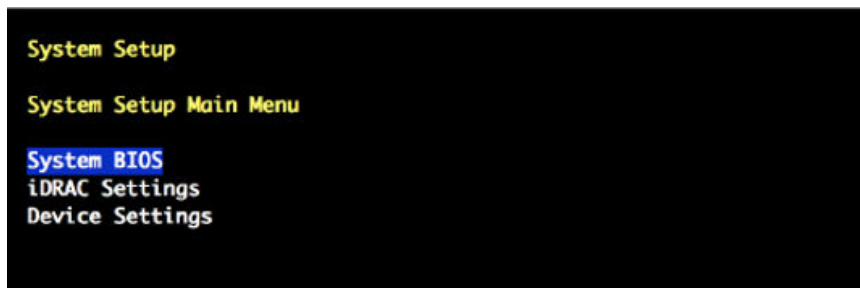
After the POST process completes and all disk and network controllers are initialized, the **Dell System Setup Main Menu** screen appears with the following sub-menus:

- System BIOS
- iDRAC Settings
- Device Settings

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

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

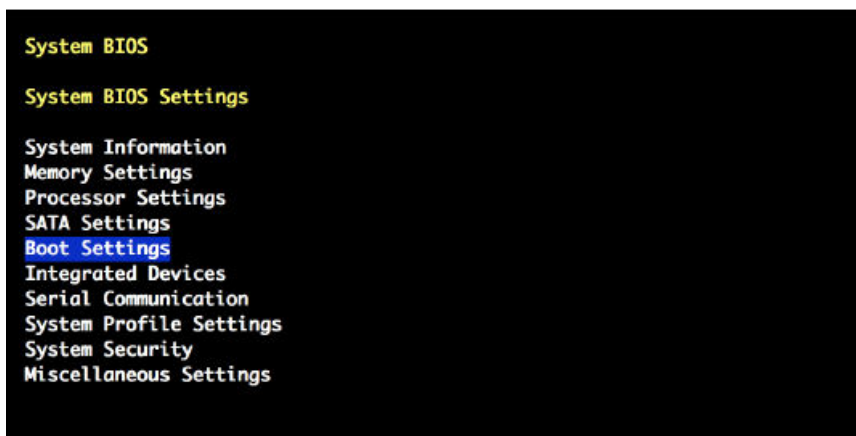
Figure 14. System Setup Main Menu: Select System BIOS



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

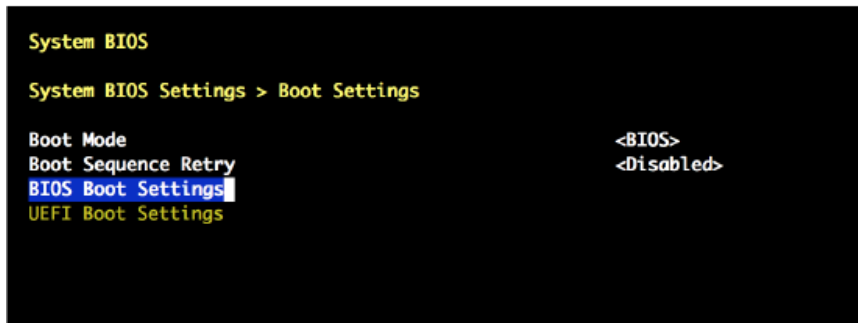
3. Change the BIOS boot settings.
 - a. Select **Boot Settings** from the **System BIOS Settings** screen, then press **Enter**.

Figure 15. System BIOS Settings: Boot Settings



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

Figure 16. 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 17. Bios Boot Settings: Set Boot Sequence

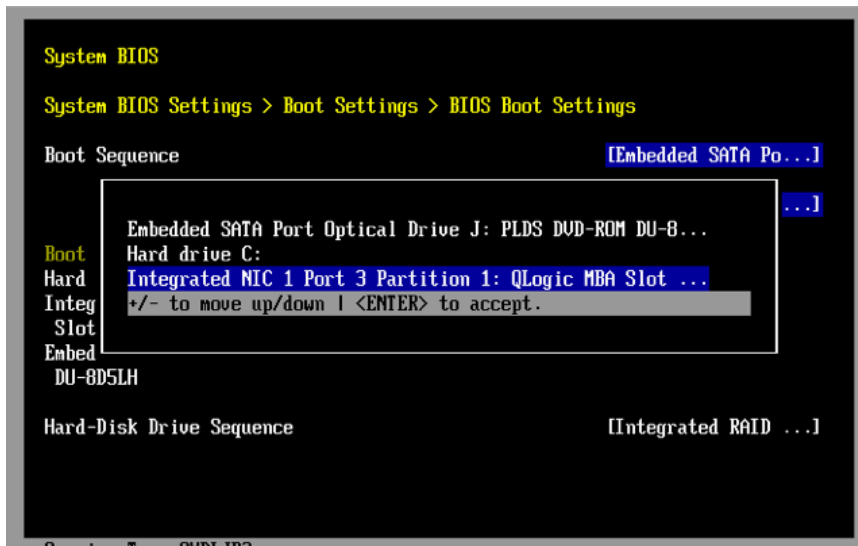
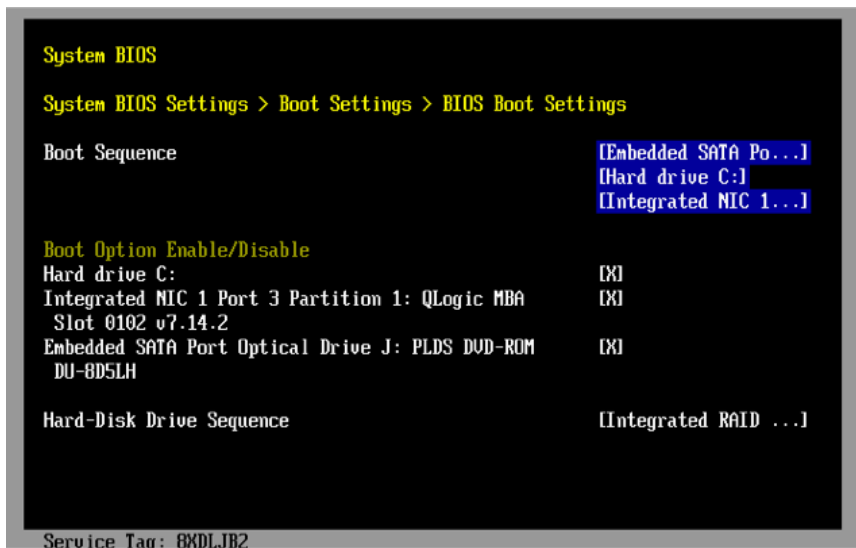
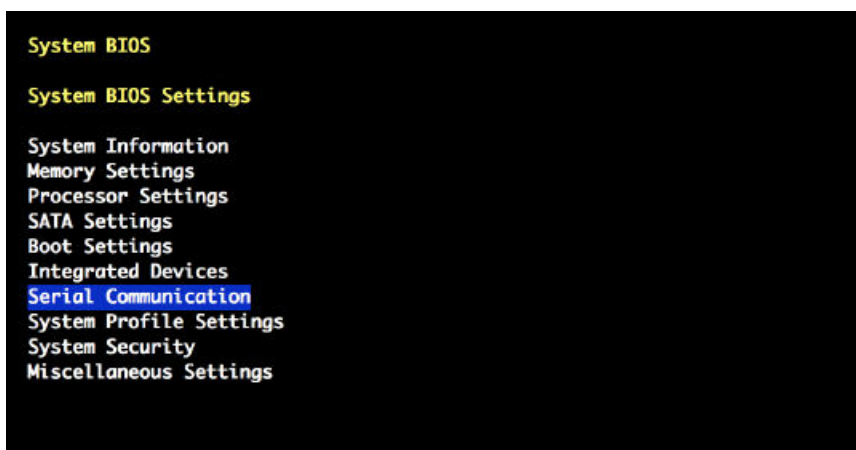


Figure 18. BIOS Boot Settings: Boot Sequence



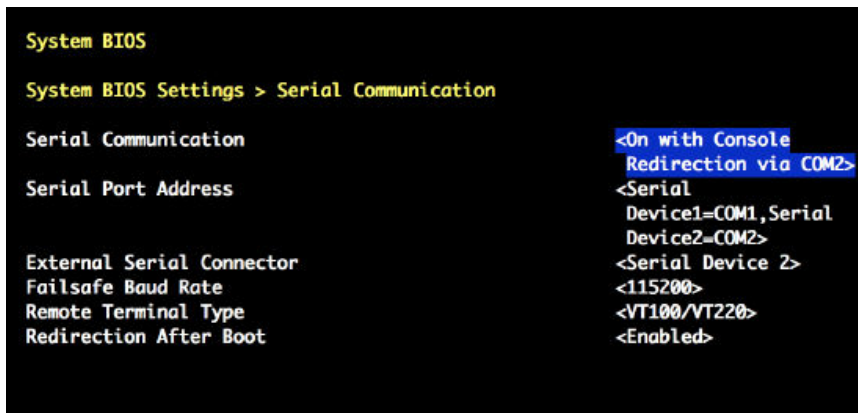
- e. Ensure that the **Integrated NIC Port** is enabled.
 - f. Press **Enter** to return to the **BIOS Boot Settings** screen.
 - g. Press **Escape** to exit **BIOS Boot Settings**.
 - h. Press **Escape** to exit **Boot Settings** and return to the **System BIOS Settings** screen.
4. Change the serial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

Figure 19. 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 20. Serial Communication: Select Console Redirection via COM2



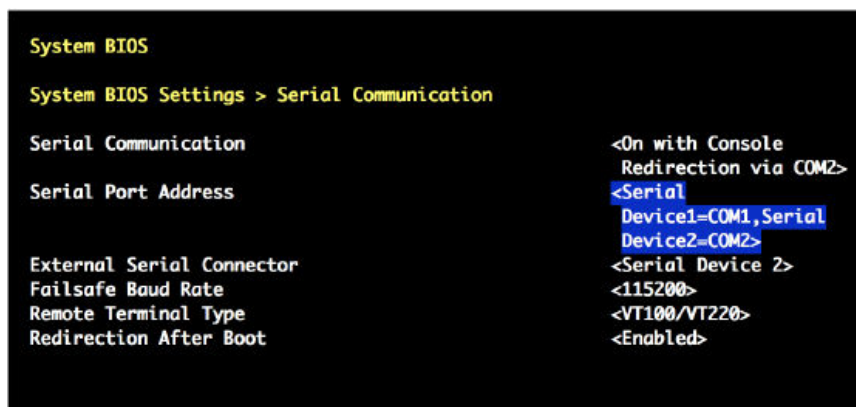
- d. Verify that **Serial Port Address** is set to Serial Device1=COM1, Serial Device2=COM2.

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:

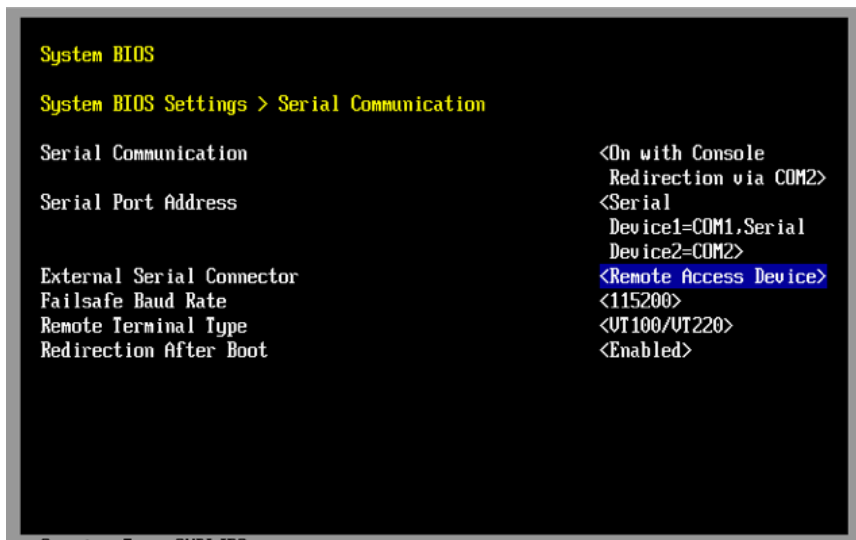
1. Press **Enter** to display the available **Serial Port Address** options.
2. Change the setting to: Serial Device1=COM1, Serial Device2=COM2.

Figure 21. Serial Communication: Serial Port Address



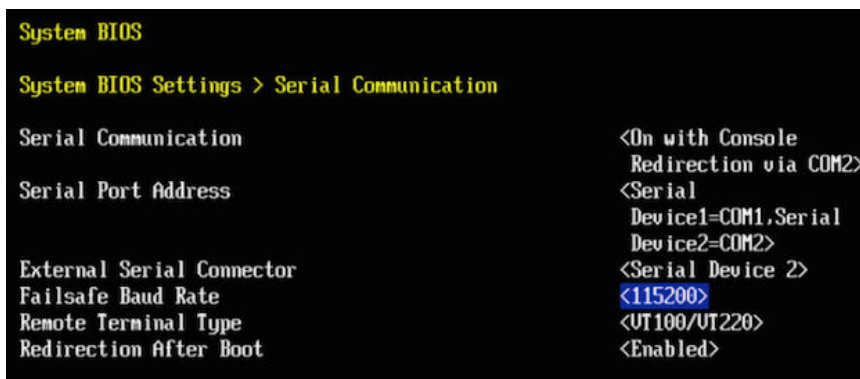
3. Press **Enter** to return to the **Serial Communication** screen.
- e. Select **External Serial Connector**. A pop-up window displays the available options.
- f. Select **Remote Access Device** in the **External Serial Connector** pop-up window, then press **Enter** to return to the previous screen.

Figure 22. 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.

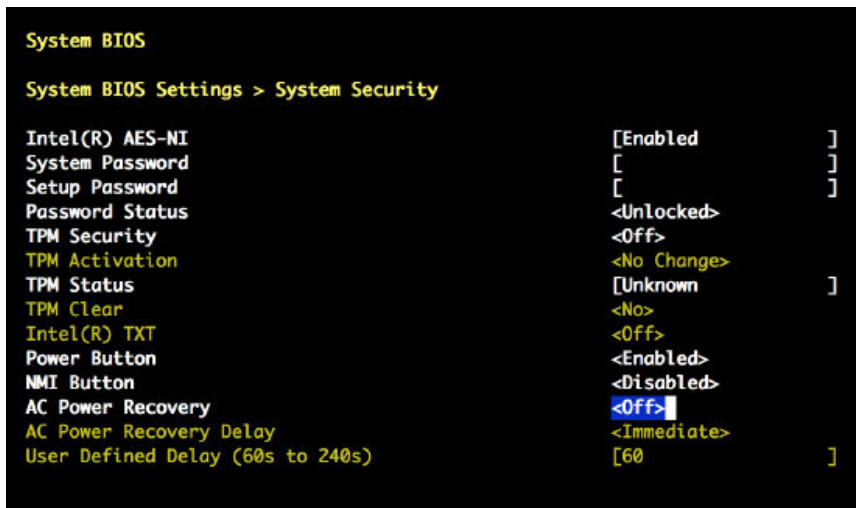
Figure 23. Serial Communication: Select 115200 Failsafe Baud Rate



- 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 24. System Security: AC Power Recovery Off

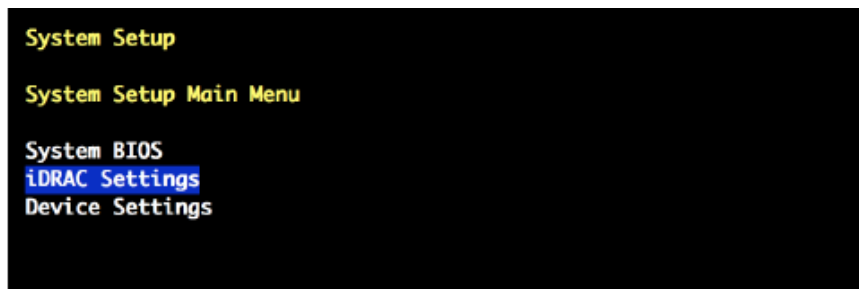


- b. Press the **Escape** key to exit the **System Security** screen.

————— CHANGE iDRAC SETTINGS —————

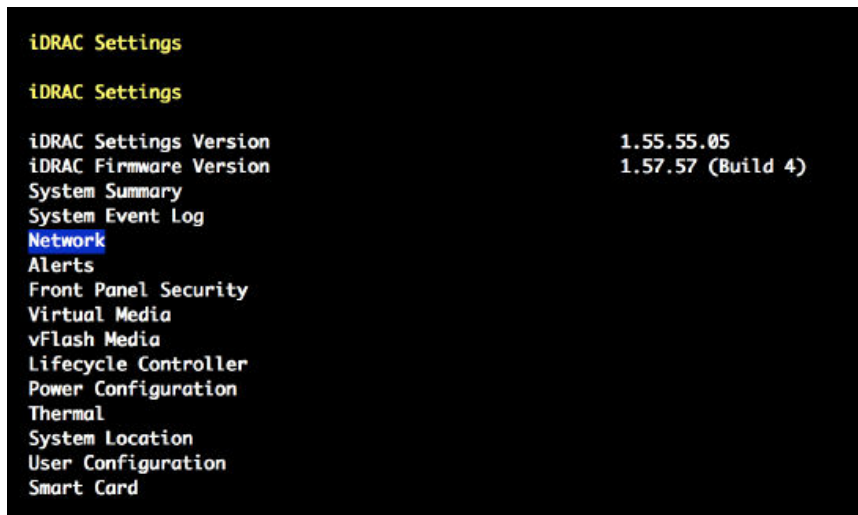
6. On the **System Setup Main Menu** screen, select **iDRAC Settings**, then press **Enter**.

Figure 25. System Setup Main Menu: iDRAC Settings



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

Figure 26. iDRAC Settings: Network



A long list of network settings is displayed.

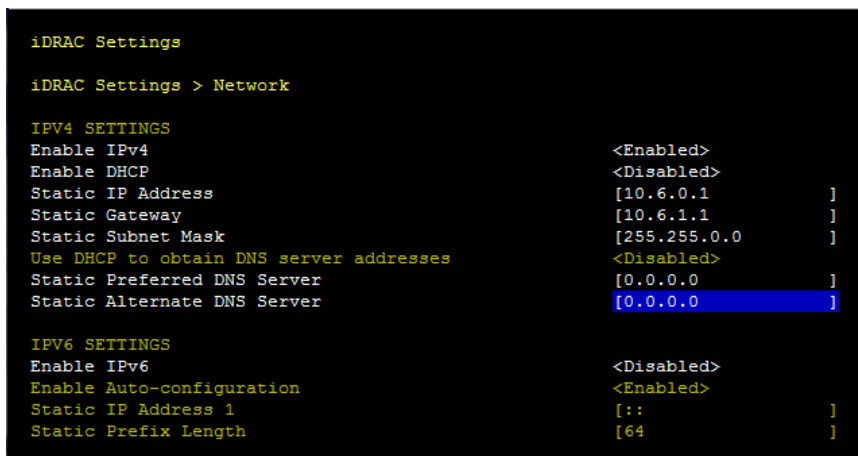
8. Change the iDRAC IP address.

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 27. Network IPv4 SETTINGS



- 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.

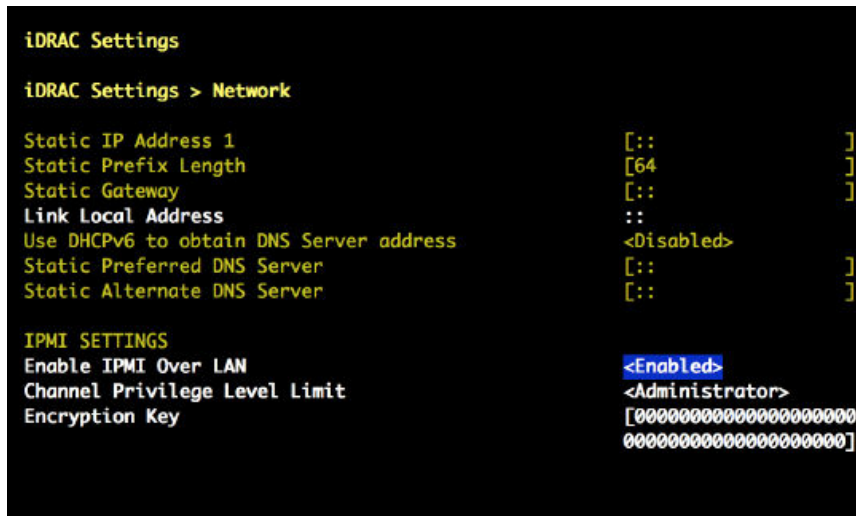
This must match the IP address of the SMW eth6 interface on the external-ipmi-net network.

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:

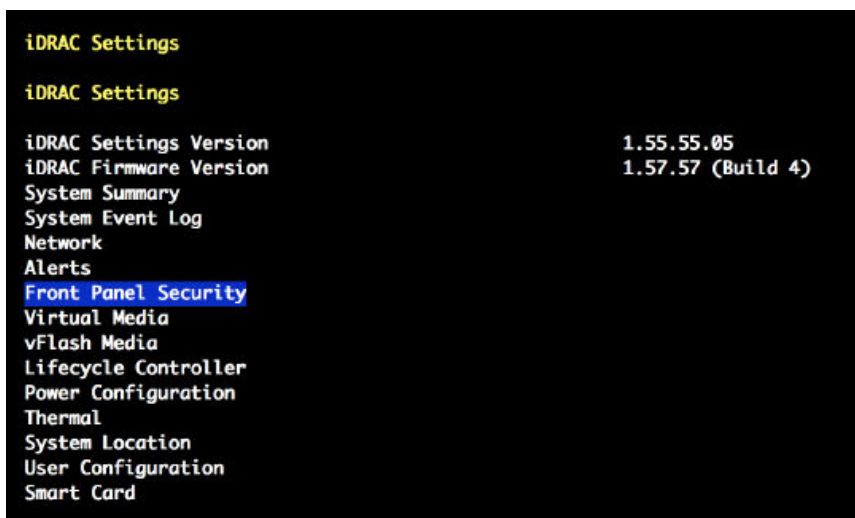
1. Select **Enable IPMI over LAN**, then press **Enter**.
2. Select **Enabled** in the pop-up window.

Figure 28. Network IPMI SETTINGS: Enable IPMI over LAN



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 29. 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 30. 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 31. System Setup Main Menu: Device Settings

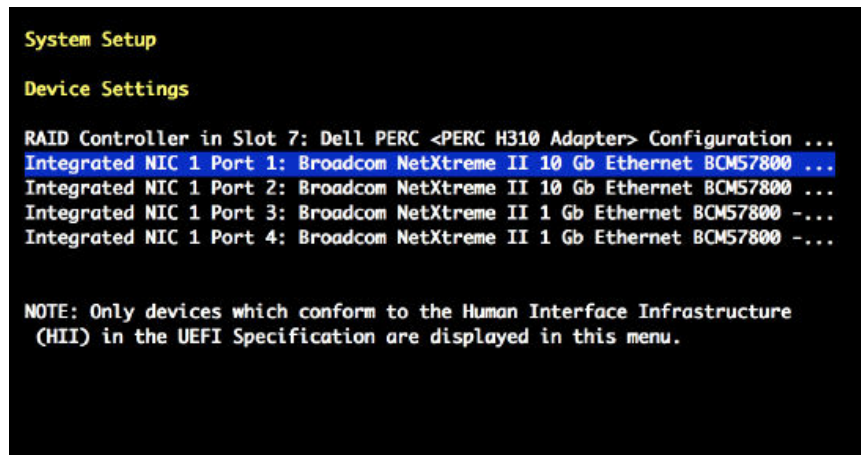


- b. Select **Integrated NIC 1 Port N ...** on the **Device Settings** screen, then press **Enter**. The **Main Configuration Page** opens.

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 ...**

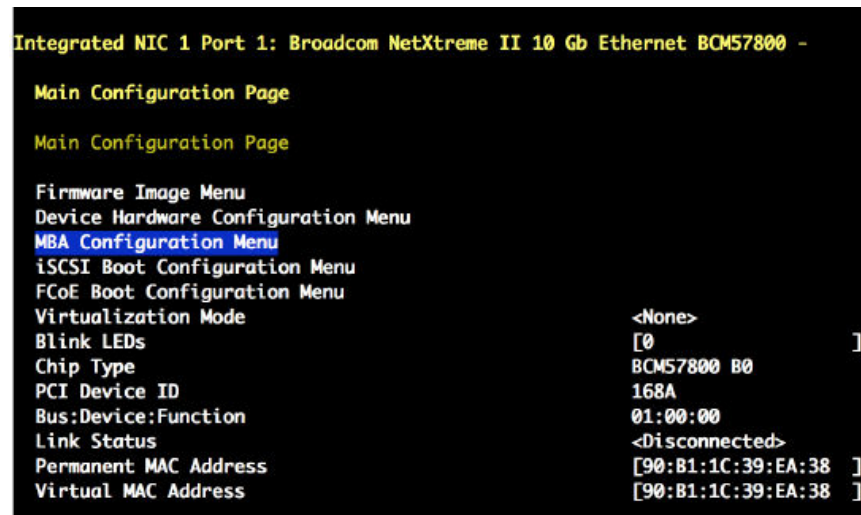
Figure 32. Main Configuration Page: Select Integrated NIC 1 Port #



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

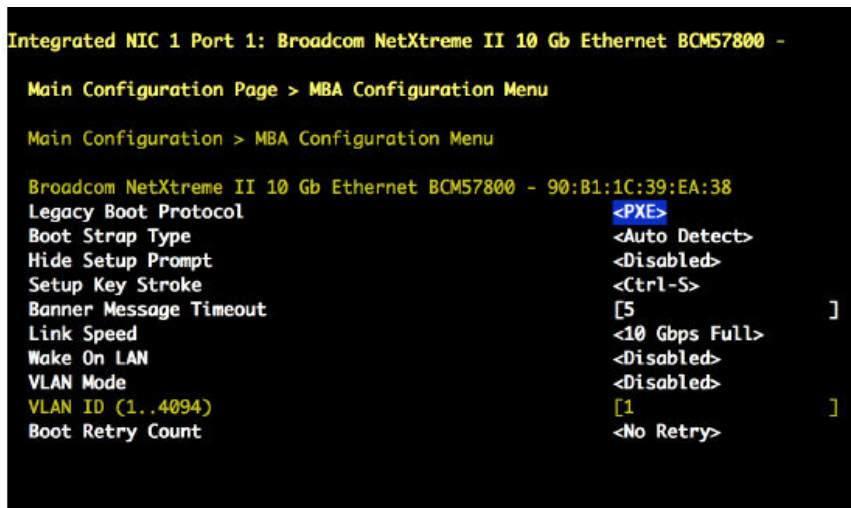
- Select **MBA Configuration Menu** on the **Main Configuration Page** screen, then press **Enter**.

Figure 33. Main Configuration Page: MBA Configuration Menu



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

Figure 34. MBA Configuration Menu: Legacy Boot Protocol - PXE



- 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
```

12.2 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 35. 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

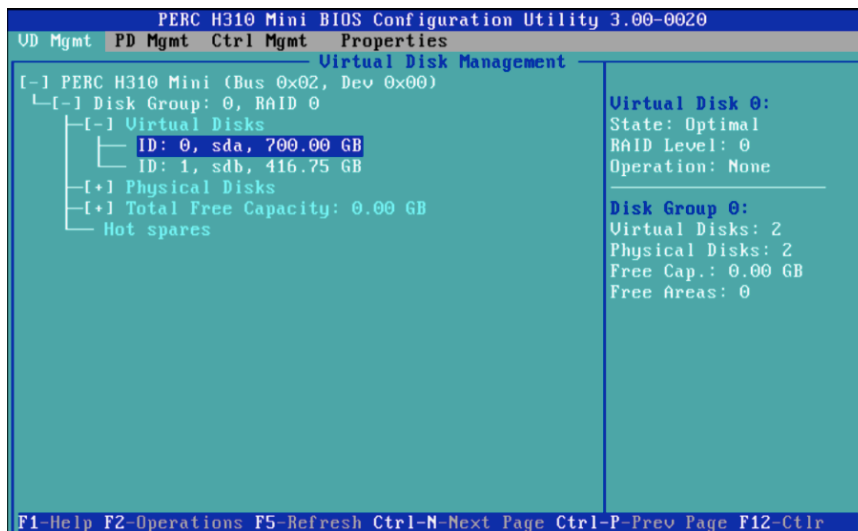
Initializing Serial ATA devices...
Port J: HL-DT-ST DVD-ROM DU90N

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

```

The RAID configuration screen opens.

Figure 36. RAID Configuration Screen: eLogin

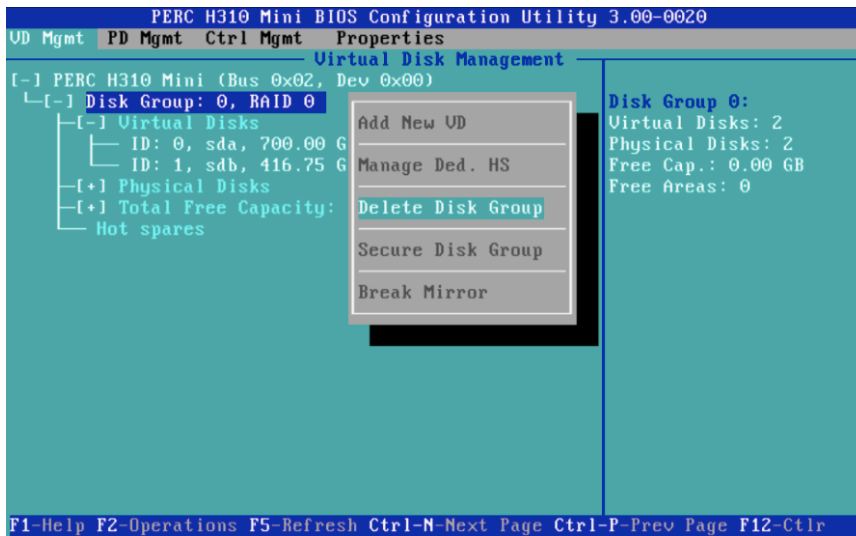


3. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration. Otherwise, skip this step.

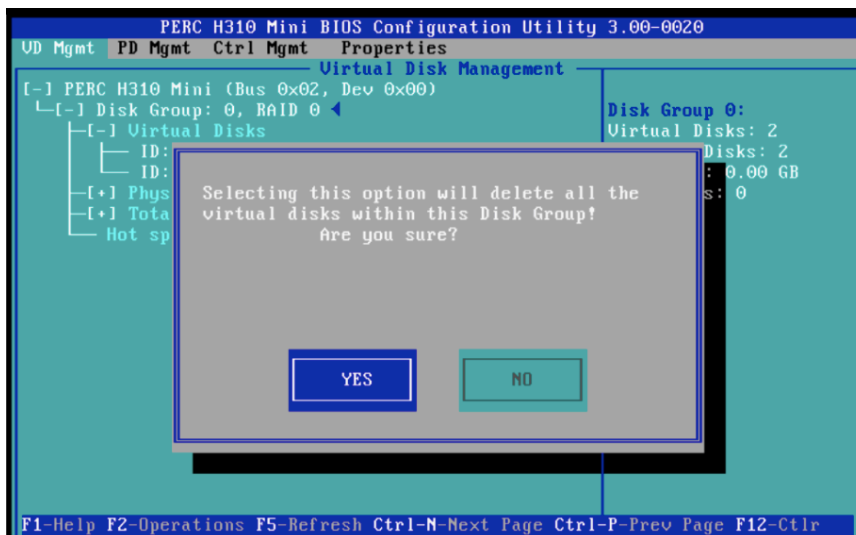
Occasionally disks are not viewable by the OS after RAID reconfiguration. This may be caused by residual metadata on the disk from the previous RAID configuration. To clear the metadata, remove the disks from any RAID configuration, and then initialize the disks. After initialization completes, reconfigure 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 37. Delete Disk Group: eLogin BIOS RAID Setup

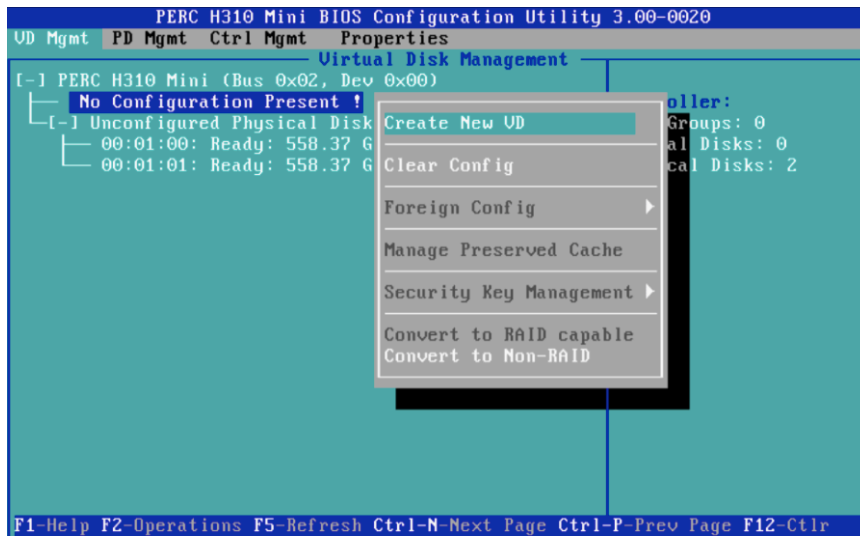


- d. Confirm the selection **Yes**, and press return.



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 38. Create Virtual Disk A: eLogin BIOS RAID



The **Create New VD** window opens.

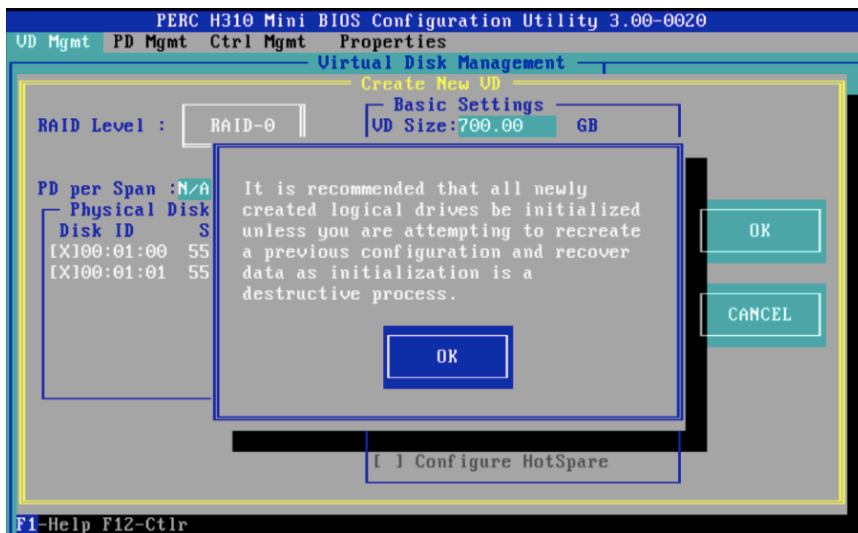
5. Move the cursor to select the disk ID in the **Create New VD** window, and then press spacebar on the keyboard to add disk to RAID.
6. Set the RAID Level to **RAID 0**.
7. Set **VD Size** and **VD Name** for virtual disk A.
 - a. Set the **VD Size** for virtual disk A to **700 GB** of disk space.

IMPORTANT: 700 GB is sufficient to accommodate the partition sizes specified in the default storage profile for eLogin nodes, `ellogin_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`.

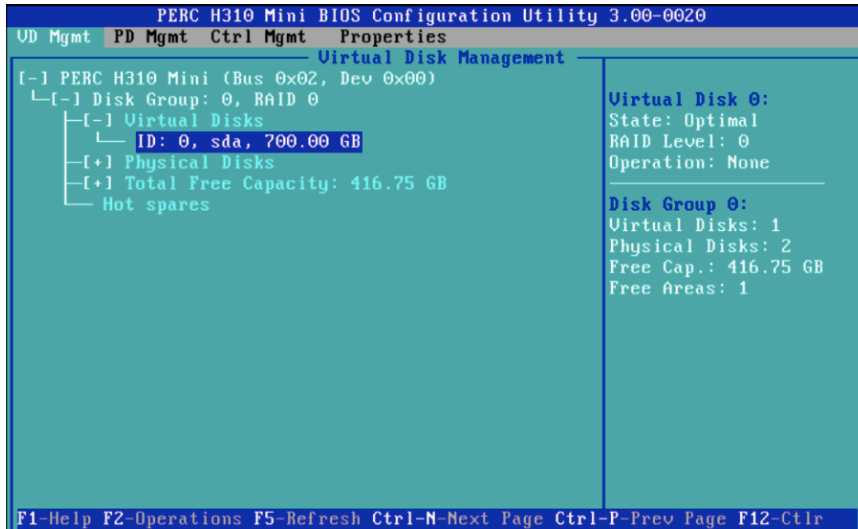
Figure 39. Disk Size and Name Setting for Virtual Disk A: eLogin



- 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**.

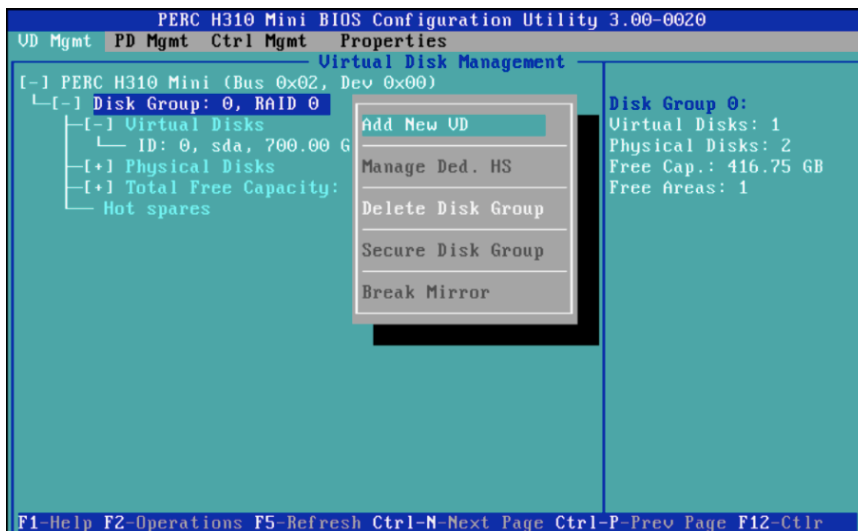
- Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- In the **Initialization** submenu, select **Fast Initialization**.

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**.

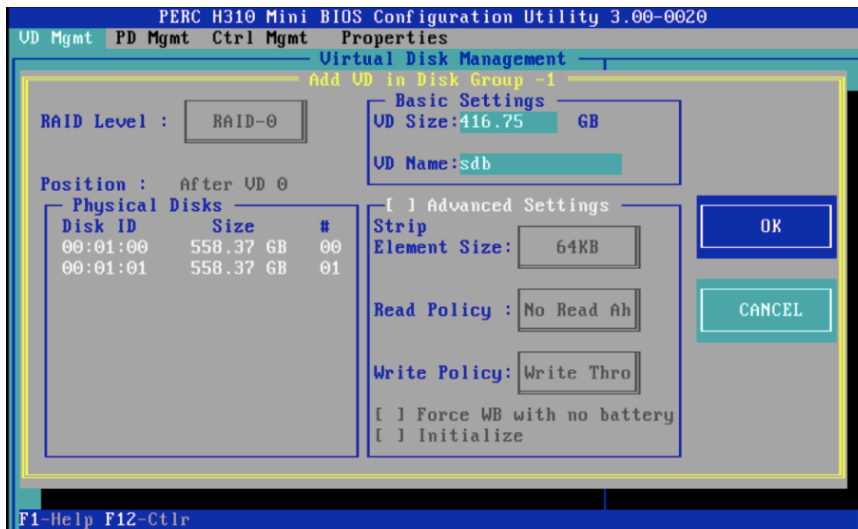
Figure 40. Create New Virtual Disk B: eLogin BIOS RAID



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 41. Disk Size and Name Setting for Virtual Disk B: eLogin

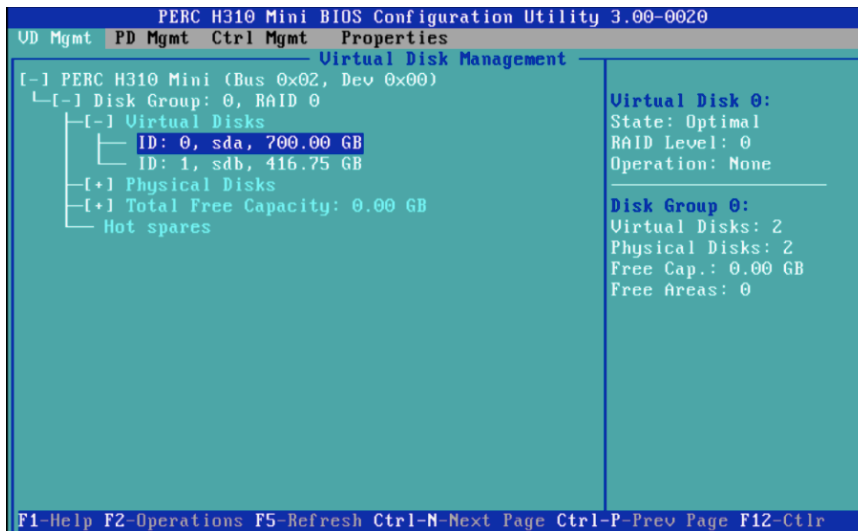


- e. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Two virtual disks are now available.

Figure 42. Two Virtual Disks Available: eLogin BIOS RAID



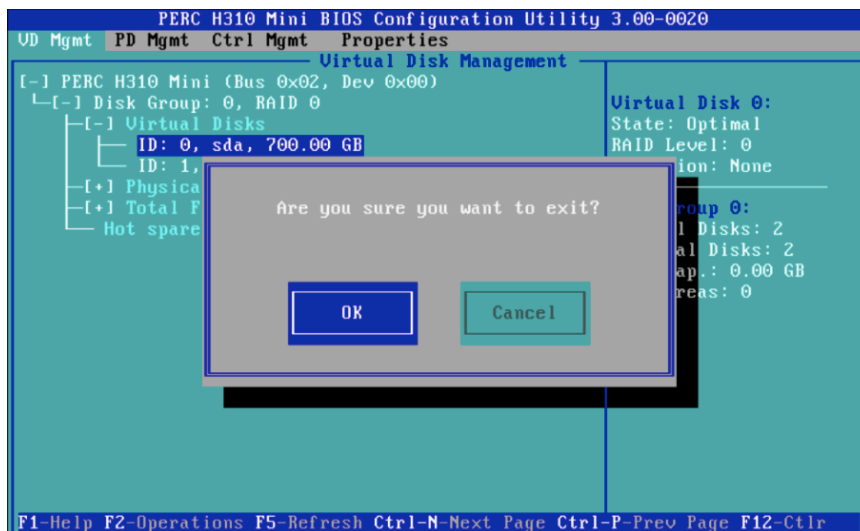
10. Initialize virtual disk B using Fast Initialization.

- 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 43. 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.

13 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 103.

- 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.

13.1 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 `eloin_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_eloin` in this example).

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

cray_storage.settings.storage_profiles.data.new_eloin: null
cray_storage.settings.storage_profiles.data.new_eloin.enabled: true

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

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

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

cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partitions.label.CRASH: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partitions.CRASH.type: ext4
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partitions.CRASH.size: 10GiB
...
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partitions.label.PERSISTENT: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partitions.PERSISTENT.type: xfs
cray_storage.settings.storage_profiles.data.new_eloin.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_eloin.layouts.device./dev/sdb: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sdb.partition_type: gpt
cray_storage.settings.storage_profiles.data.new_eloin.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
- To enable the eLogin node to boot, the `partition_flags` list for the GRUB partition must be set to a list containing `bios_grub` instead of the empty list (the default value for that field).

- 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 103.

For more information about binary values, see [Prefixes for Binary and Decimal Multiples](#) on page 151.

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 ellogin1
```

9. Reboot the eLogin node.

This example reboots an eLogin node named `ellogin1`.

```
smw# enode reboot --pxe ellogin1
```

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 ellogin1
```

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
```

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.

13.2 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
- To enable the eLogin node to boot, the `partition_flags` list for the GRUB partition must be set to a list containing `bios_grub` instead of the empty list (the default value for that field).
- 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 103.

For more information about binary values, see [Prefixes for Binary and Decimal Multiples](#) on page 151.

4. If creating a new storage profile, copy `eloin_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_eloin` in this example).

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

```
cray_storage.settings.storage_profiles.data.new_eloin: null
cray_storage.settings.storage_profiles.data.new_eloin.enabled: true

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

cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.label.GRUB: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.GRUB.type: ext3
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.GRUB.size: 1MiB
...
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.label.BOOT: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.BOOT.type: ext3
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.BOOT.size: 2GiB
...
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.label.WRITELAYER: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.WRITELAYER.type: ext4
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.WRITELAYER.size: 20GiB
...
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.label.TMP: null
cray_storage.settings.storage_profiles.data.new_eloin.layouts./dev/sda.partitions.TMP.type: xfs
cray_storage.settings.storage_profiles.data.new_eloin.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 `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
```

13.3 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 `ellogin1`:

```
smw# conman -j ellogin1
```

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 `ellogin1`.

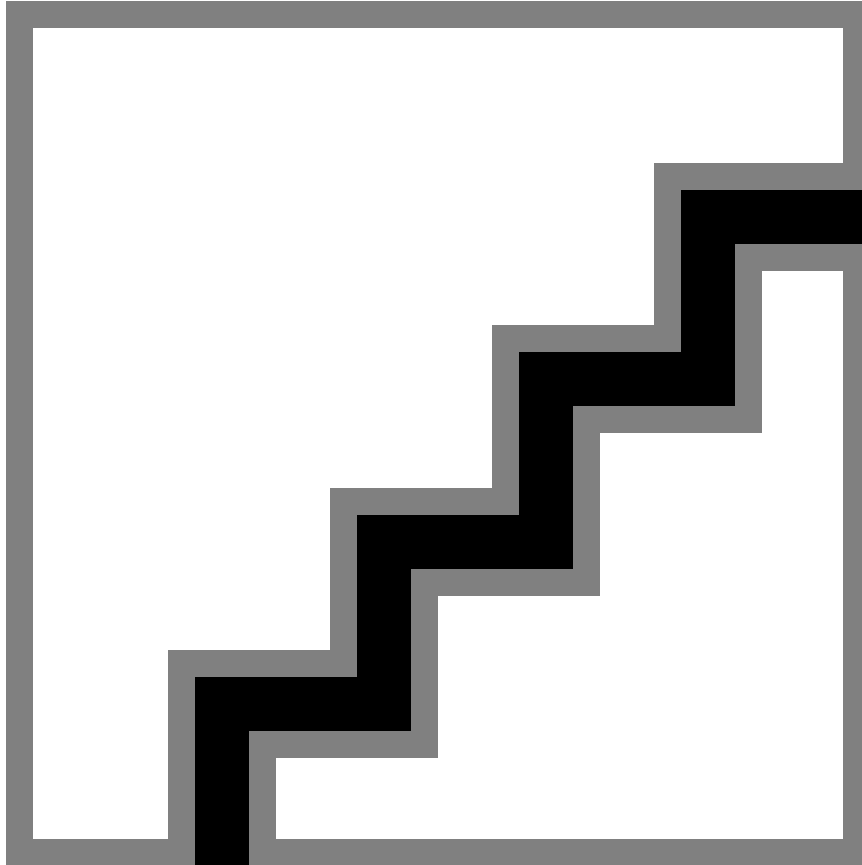
```
smw# enode reboot --bios ellogin1
```

3. Open the BIOS RAID configuration screen.

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

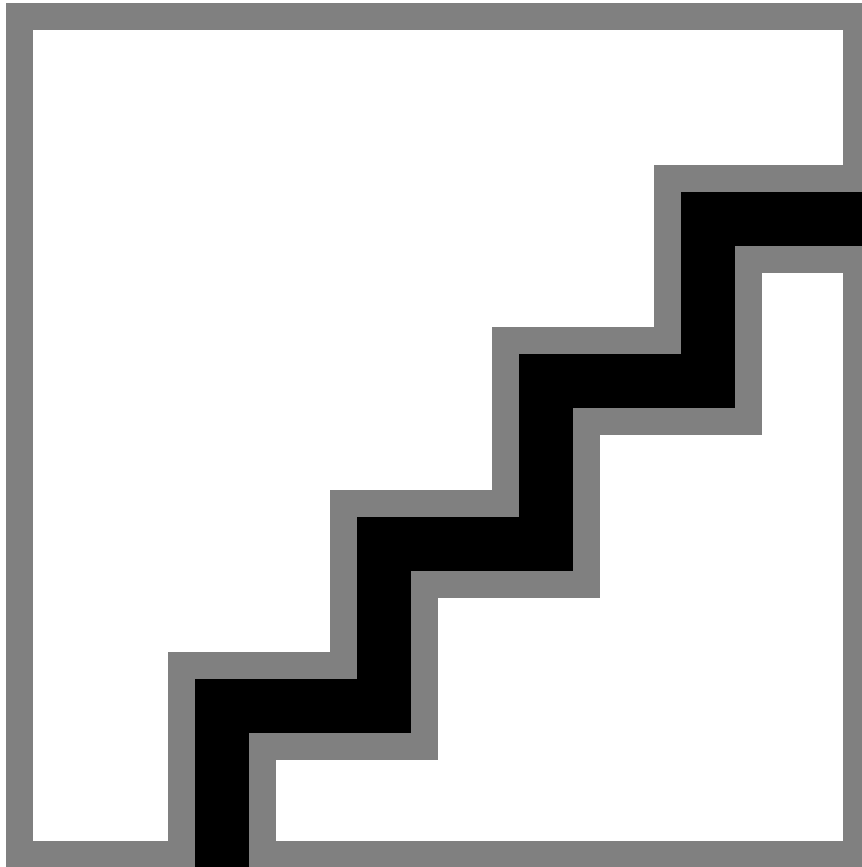
NOTE: The BIOS RAID configuration screens may appear different if accessing from `conman` versus the physical console, but the functionality is the same.

Figure 44. Initial Boot Menu for BIOS RAID Configuration: eLogin



The RAID configuration screen opens.

Figure 45. RAID Configuration Screen: eLogin

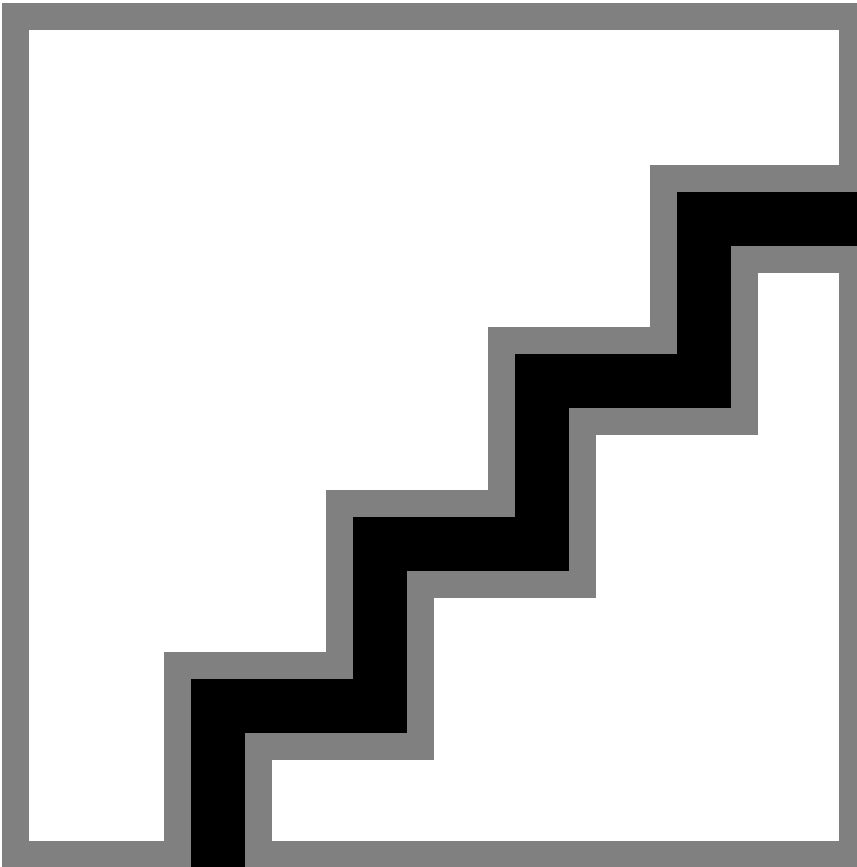


4. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration, as defined in `cray_storage` in the CLE config set. If there are none to delete, skip this step.

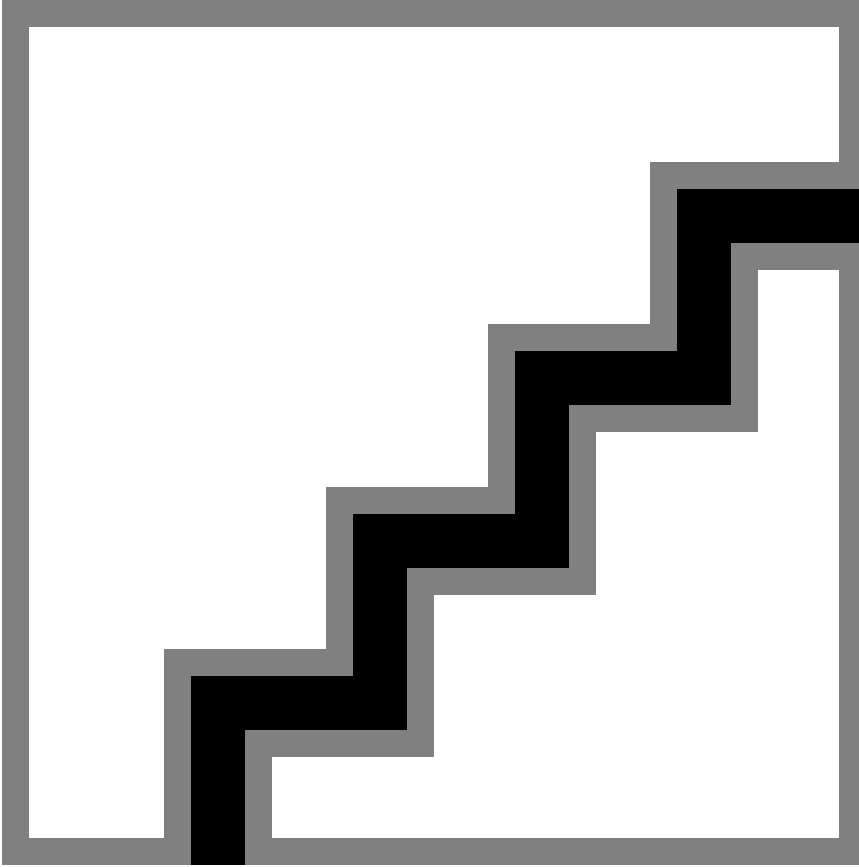
Occasionally disks are not viewable by the OS after RAID reconfiguration. This may be caused by residual metadata on the disk from the previous RAID configuration. To clear the metadata, remove the disks from any RAID configuration, and then initialize the disks. After initialization completes, reconfigure 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 46. Delete Disk Group: eLogin BIOS RAID Setup

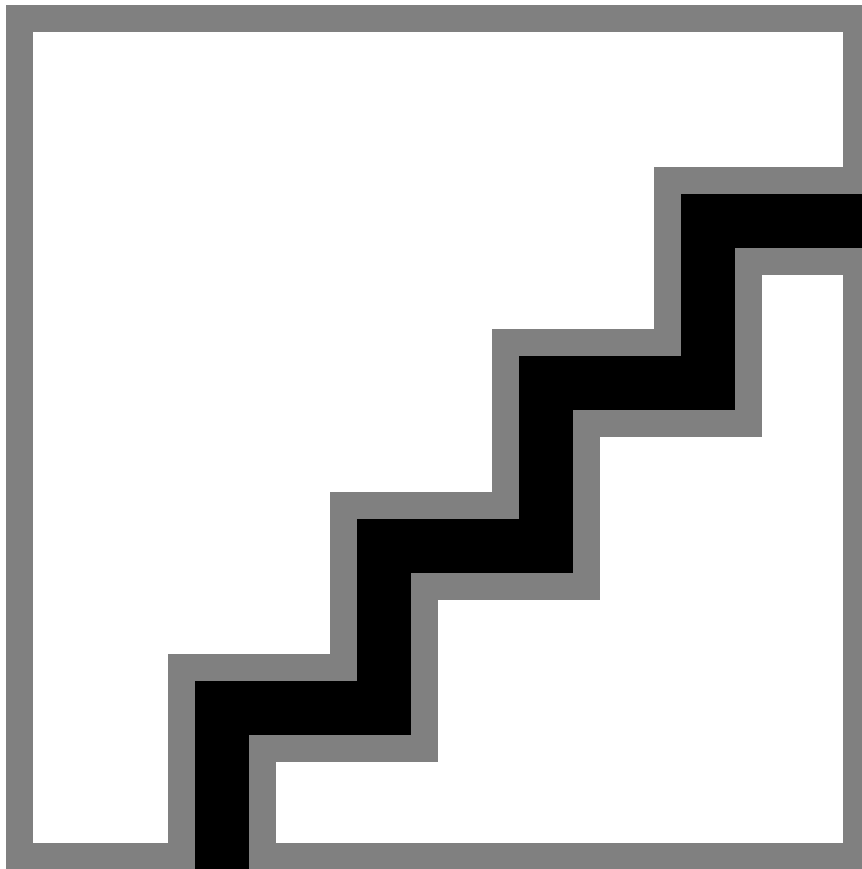


- d. Confirm the selection **Yes**, and press return.



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 47. Create Virtual Disk A: eLogin BIOS RAID

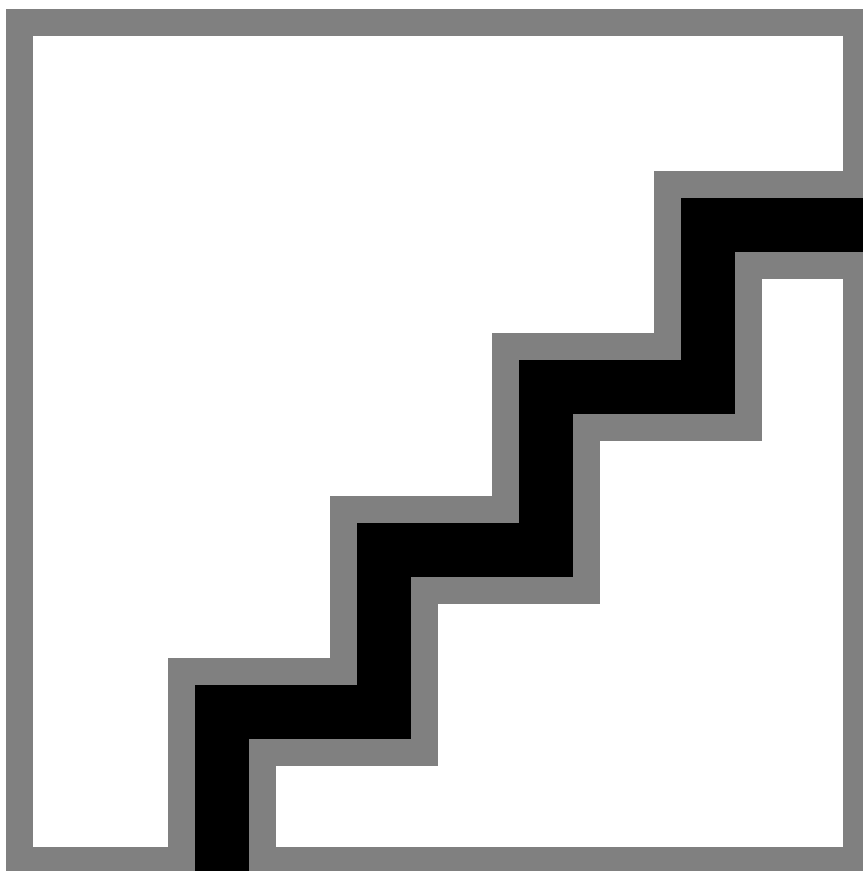


The **Create New VD** window opens.

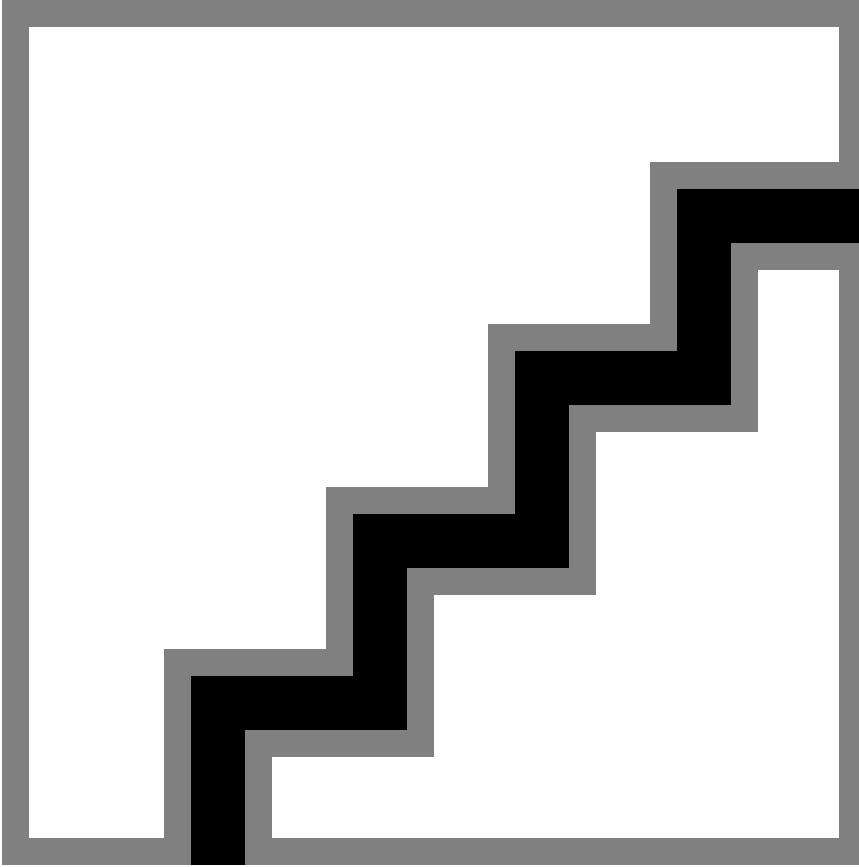
6. Move the cursor to select the disk ID in the **Create New VD** window, and then press spacebar on keyboard to add disk to RAID.
7. Set the RAID Level to **RAID 0**.
8. Set **VD Size** and **VD Name** for virtual disk A.
 - a. Set the **VD Size** for virtual disk A to **700 GB** of disk space.

IMPORTANT: 700 GB is sufficient to accommodate the partition sizes specified in the default storage profile for eLogin nodes, `ellogin_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`.

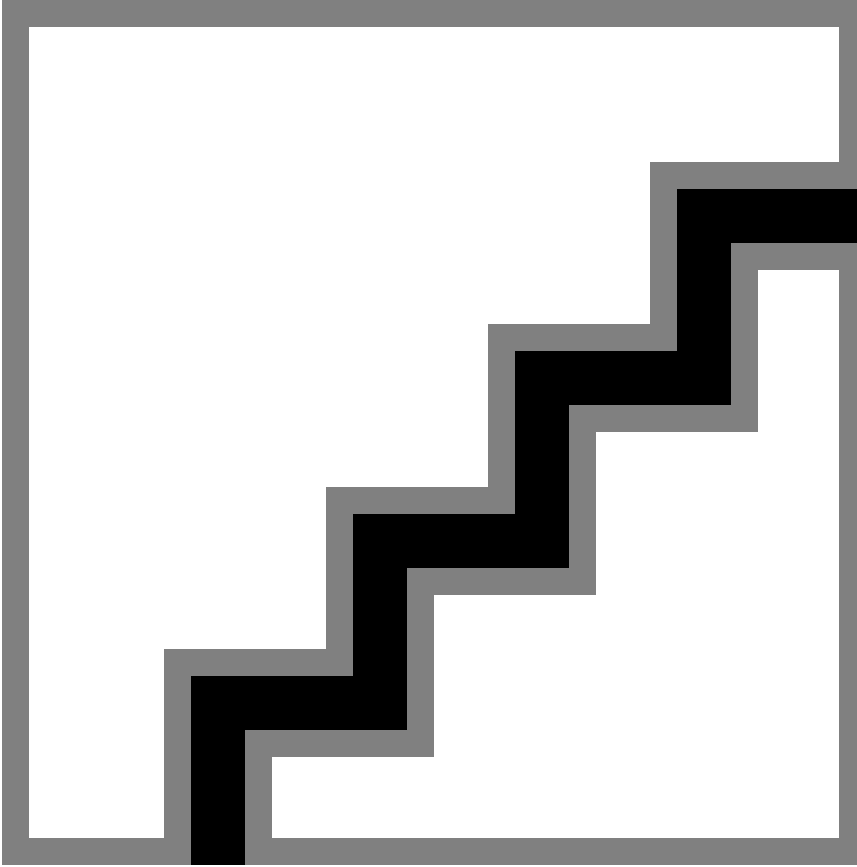
Figure 48. Disk Size and Name Setting for Virtual Disk A: eLogin



- 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.**

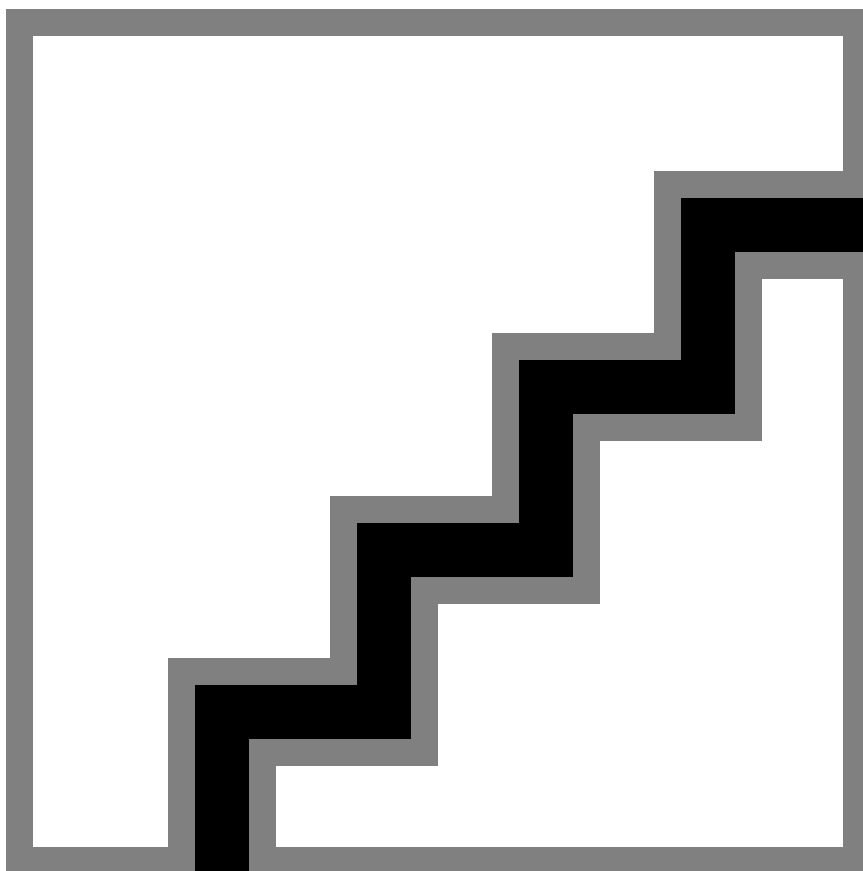
- a. Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
- b. Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
- c. In the **Initialization** submenu, select **Fast Initialization**.

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

10. Create a new virtual disk B.

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

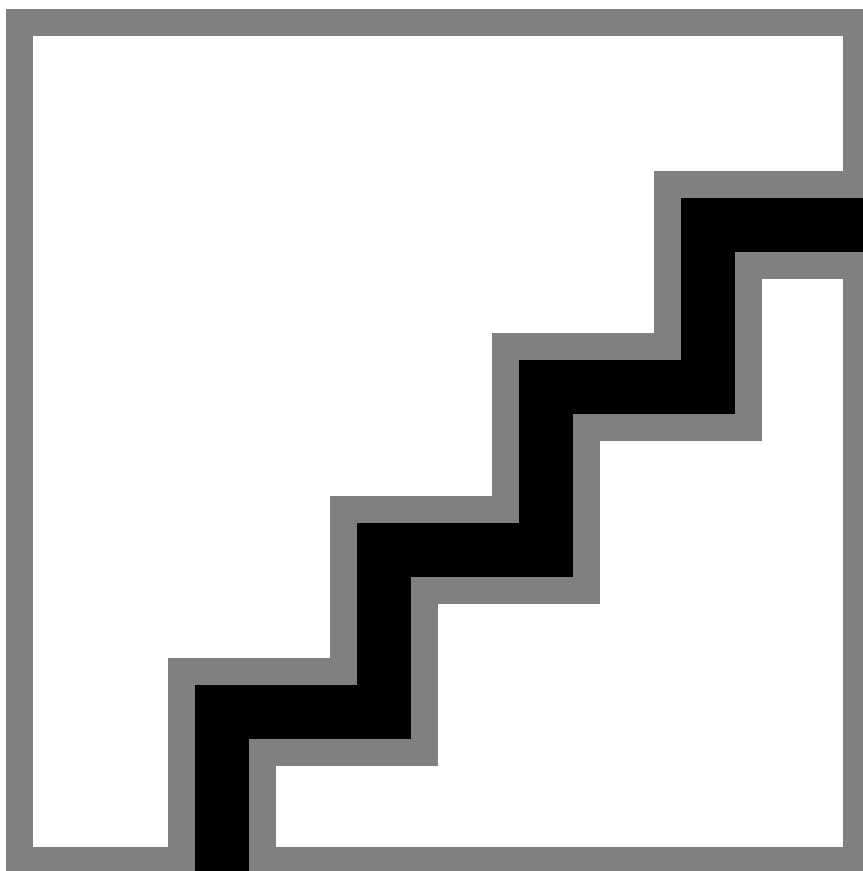
Figure 49. Create New Virtual Disk B: eLogin BIOS RAID



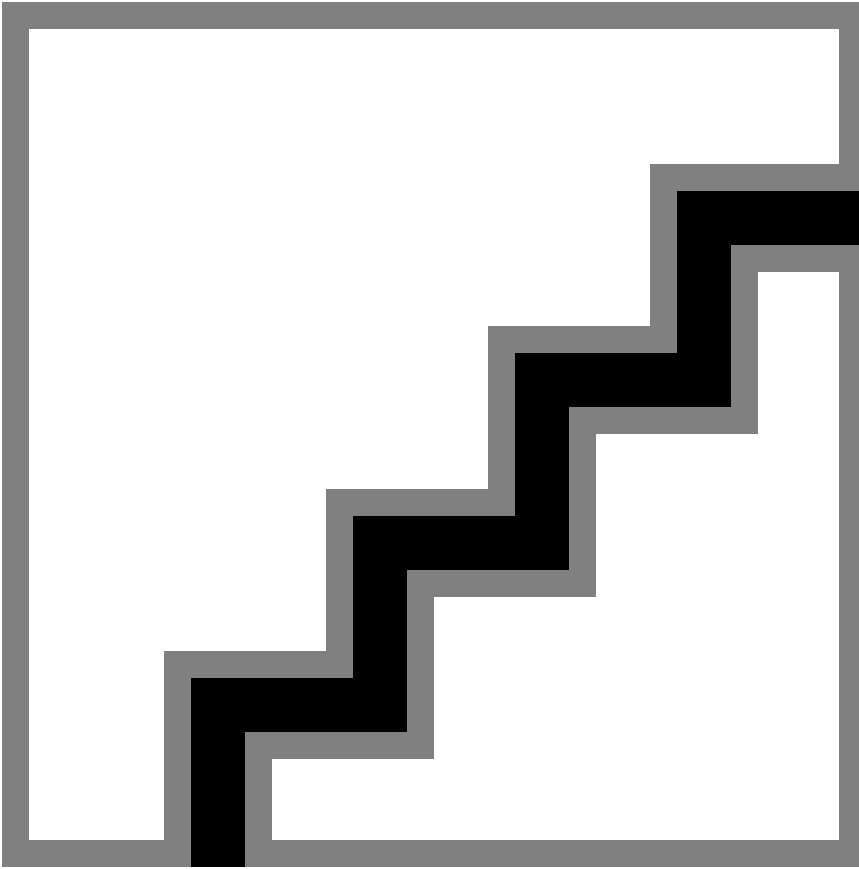
The **Add VD in Disk Group 0** window opens.

- d. In the window, set the **VD Name** to **sdb**, and verify that the **VD Size** is set to the remaining disk space.

Figure 50. Disk Size and Name Setting for Virtual Disk B: eLogin

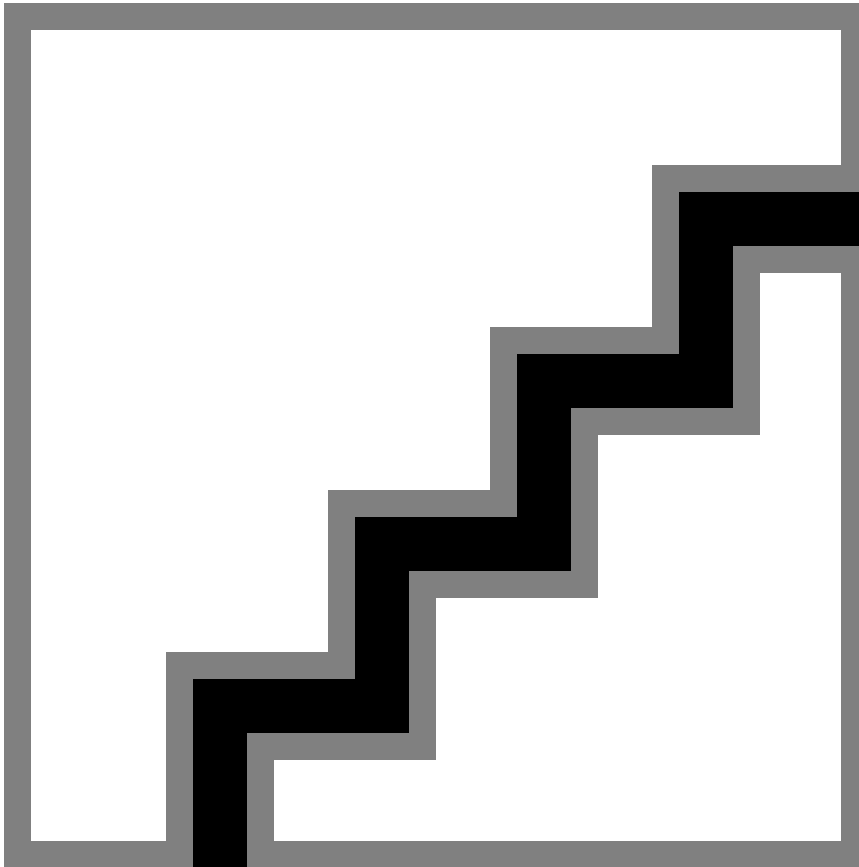


- e. Select **OK** in the window, and then in the initialization message pop-up window, select **OK**.



Two virtual disks are now available.

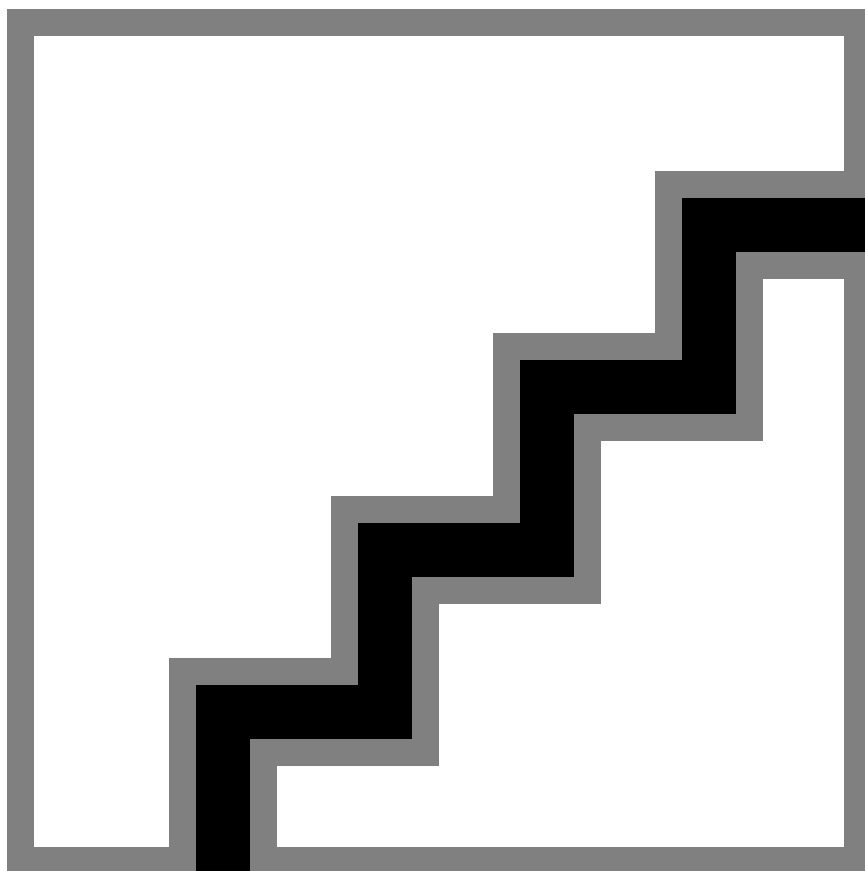
Figure 51. Two Virtual Disks Available: eLogin BIOS RAID



11. Initialize virtual disk B (sdb) using **Fast Initialization**.
 - a. Select **Virtual Disk #** and press **F2** to display the menu of available actions on the **Virtual Disk Management** screen.
 - b. Select **Initialization** and press the right-arrow key to display the **Initialization** submenu options.
 - c. 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 52. Exit BIOS Configuration: eLogin



The BIOS configuration utility screen is now closed.

- 13.** Press **Ctrl+Alt+Delete** from the keyboard to reboot the node.

14 File System Configuration

14.1 AutoFS

AutoFS uses template files for configuration which are located in `/etc/autofs`. The main template is called `auto.master`, which points to one or more other templates for specific media types.

An administrator can manually add the appropriate configuration changes to the active config set to support AutoFS. Configuration of AutoFS via Cray supported Ansible plays is not planned.

14.2 Connect eLogin Nodes to a Lustre File System

Prerequisites

Initial installation and configuration of eLogin is complete.

About this task

A Lustre file system may be mounted on the eLogin nodes to provide access to a shared file system.

Procedure

1. Ensure that the first InfiniBand interface is physically connected to the Lustre server.
2. Verify that the following settings in the `cray_elogin_lnet` config set are correct for the site.

```
smw# cfgset search -s cray_elogin_lnet config_set

# 1 match for '.' from cray_elogin_lnet_config.yaml
#-----
cray_elogin_lnet.settings.local_lnets.data.o2ib.ip_wildcard: 10.149.*.*

smw# cfgset search -s cray_net cfgset
...
cray_net.settings.networks.data.lnet.description: Infiniband network to external Lustre
cray_net.settings.networks.data.lnet.ipv4_network: 10.149.0.0
cray_net.settings.networks.data.lnet.ipv4_netmask: 255.255.0.0
cray_net.settings.networks.data.lnet.ipv4_gateway: # (empty)
cray_net.settings.networks.data.lnet.dns_servers: # (empty)
cray_net.settings.networks.data.lnet.dns_search: # (empty)
cray_net.settings.networks.data.lnet.ntp_servers: # (empty)
...
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: # (empty)
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.ipv4_address: 10.149.0.123
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.mtu: 65520
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.extra_attributes:
IPOIB_MODE='connected'
```

```
smw# cfgset search -s cray_lustre_client -l advanced cfgset
```

```
# 9 matches for '.' from cray_lustre_client_config.yaml
#-----
cray_lustre_client.settings.module_params.data.libcfs_panic_on_lbug: True
cray_lustre_client.settings.module_params.data.ptlrpc_at_min: 40
cray_lustre_client.settings.module_params.data.ptlrpc_at_max: 400
cray_lustre_client.settings.module_params.data.ptlrpc_ldlm_enqueue_min: 260
cray_lustre_client.settings.client_mounts.data.rindl.lustre_fs_name: rindl
cray_lustre_client.settings.client_mounts.data.rindl.mount_point: /lus/rindl
cray_lustre_client.settings.client_mounts.data.rindl.mgs_lnet_nids: 10.149.0.1@o2ib
cray_lustre_client.settings.client_mounts.data.rindl.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.rindl.mount_at_boot: True
cray_lustre_client.settings.client_mounts.data.rindl.mount_locations: login, compute,
elogin
```

- Ensure that the LNet IP wildcard in `cray_elogin_lnet` matches the LNet IPv4 network and netmask in `cray_net`.
- Verify that the IPv4 address for the eLogin `ib0` interface is unique within the LNet.
- Check that all Lustre mounts include `elogin` in the mount locations list.
- Update missing or incorrect settings.

```
smw# cfgset update -l advanced -s service -m interactive config_set
```

- Proceed to step 4 on page 118 if no updates are needed.

- Push the config set to the eLogin node and reboot if any configuration settings were changed.

```
smw# cfgset push -d elogin1 global
smw# cfgset push -d elogin1 config_set
smw # enode reboot elogin1
```

- Verify that Lustre functions correctly.

```
elogin# ip addr show ib0
4: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast state UP
group default qlen 256
    link/ether 80:00:00:00:00:00:02:c9:03:00:a4:5d:f1 brd 00:ff:ff:ff:ff:12:40:1b:ff:ff:
    00:00:00:00:00:00:ff:ff:ff:ff
        inet 10.149.0.123/16 brd 10.149.255.255 scope global ib0
            valid_lft forever preferred_lft forever
        inet6 fe80::202:c903:a4:5df1/64 scope link
            valid_lft forever preferred_lft forever

elogin# lctl list_nids
10.149.0.123@o2ib

elogin# mount | grep lustre
10.149.0.1@o2ib:/rindl on /lus/rindl type lustre (rw,flock,lazystatfs)
```

15 Other Configuration Options

15.1 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 12.

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 123.

————— 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
```

```
eloin# 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
```

15.1.1 Ensure that NFS Port is Open in Firewall

Prerequisites

SMW/CLE software is installed and configured.

About this task

If the `SuSEfirewall2` 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 SuSEfirewall2
```

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.

15.2 PBS License Server Configuration

PBS requires FlexLM licensing, which is handled automatically by the PBS install script that runs during the image recipe build. The process puts the license string in a file in the image. When the image is booted, PBS looks for the file and adds the license string to the server settings. No intervention is required.

15.3 User Authentication

The `root` password for the eLogin server is set to the value configured in the active config set template; this is the same `root` password used elsewhere in a Cray installation.

LDAP Authentication

Configure an eLogin node to authenticate users against an Lightweight Directory Access Protocol (LDAP) server. Because eLogin shares configuration data with the SMW, a Cray XC series system configured for LDAP authentication automatically configures eLogin nodes for LDAP authentication against the same source.

NIS Authentication

Deferred implementation: support for authenticating against a Network Information Service (NIS) server is currently not supported.

15.4 Configure Passwordless SSH

About this task

Users running eLogin wrapped commands benefit from configuring passwordless Secure Shell (SSH). Without passwordless SSH, a user must enter a password to run each command.

Procedure

1. Generate an SSH key pair.

```
cmc# ssh-keygen
```

2. Add the key pair to the `.ssh/authorized_keys` file on the eLogin node.

```
eloin# ssh-copy-id eloin_name
```

15.5 Configure eProxy to Wrap Reduced Set of Commands

Prerequisites

- The eLogin installation is complete
- eProxy is configured to connect to the XC system

About this task

Users for specific sites may desire to modify the available eProxy-wrapped command set. This section instructs how to make a minimal set of Slurm commands available by not including the full command set in the `ESWRAP_CMDS` list of wrapped commands. The list of wrapped commands may also be modified for other purposes, such as: disabling ALPS, or removing specific DataWarp commands.

DataWarp is an intermediate layer of high bandwidth, file-based storage that uses SSDs to provide high-performance storage to compute and eLogin nodes in a variety of ways. Refer to *XC™ Series DataWarp™ Installation and Administration Guide (S-2564)* for more information.

Procedure

1. Connect to the SMW.

```
# ssh root@smw
```

2. Configure the eLogin image with minimal set of Slurm commands.

- a. Change directory to the eLogin image.

```
smw# cd /var/opt/cray/imps/image_roots/eloin-image-name
```

- b. Change directory to the image's eProxy directory.

```
smw# cd opt/cray/eloin/eproxy/default/bin
```

- c. Edit the `eproxy_config.py` file.

```
smw# vi eproxy_config.py
```

- d. Delete all non desired Slurm commands listed in the `ESWRAP_CMDS` section. Do not delete the commands you want enabled, example, `salloc` and `srun`.

```

ESWRAP_CMD_FIELDS = ['category', 'command', 'preamble', 'x11']

ESWRAP_CMDS = [
    ['eproxy', 'eproxy', None, False],
    ['xt', 'cnselect', 'module load sdb', False],
    ['xt', 'xtprocadmin', 'module load sdb', False],
    ['xt', 'xtnodestat', 'module load nodestat', False],
    ['aprun', 'aprun', 'module load alps', False],
    ['alps', 'apcount', 'module load alps', False],
    ['alps', 'apmgr', 'module load alps', False],
    ['alps', 'apkill', 'module load alps', False],
    ['alps', 'apstat', 'module load alps', False],
    ['ccm', 'ccmrun', 'module load ccm', False],
    ['ccm', 'ccmlogin', 'module load ccm', False],
    ['debug', 'ls', None, False],
    ['debug', 'csh', None, False],
    ['debug', 'xterm', None, True],
    ['slurm', 'salloc', 'module load slurm', False], <=== KEEP THIS ONE
    ['slurm', 'srun', 'module load slurm', False], <=== KEEP THIS ONE
    ['datawarp', 'dwstat', 'module load dws', False],
    ['datawarp', 'dwcli', 'module load dws', False],
    ['datawarp', 'dwgateway', 'module load dws', False],
    ['datawarp', 'dw_wlm_cli', 'module load dw_wlm', False]
]

```

3. Update the config set's eProxy section to include your system's internal login node and eLogin node host names.

```
smw# cfgset update -S all --service cray_eproxy p0
```

4. Set or verify that eProxy is enabled.

```
cray_eproxy.enabled
[<cr>=keep 'true', <new value>, ?=help, @=less] $
```

5. Add the internal login node to the eProxy map.

```
cray_eproxy.settings.eproxy_map
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
```

Use the default login, or specify the internal login host required.

```
cray_eproxy.settings.eproxy_map.data.eproxy_host
[<cr>=set 'login', <new value>, ?=help, @=less] $ intlogin-p0
```

6. Add one or more elogin nodes associated with the internal login specified in the previous step.

```
cray_eproxy.settings.eproxy_map.data.intlogin-p0.elogin_hosts
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
```

```
Add elogin_hosts (Ctrl-d to exit) $ cray-elogin99
Add elogin_hosts (Ctrl-d to exit) $ ^d
```

7. Click **Enter** at the next prompt to display the new configuration.

```
Configured Values:
1) 'intlogin-p0'
```

```
a) eloin_hosts:
    cray-eloin99
```

15.6 Use eProxy Utility

Prerequisites

The node must be in the `node_up` state.

About this task

The `eProxy` utility is a wrapper that lets users access a subset of Cray Linux Environment (CLE) and Programming Environment (PE) commands from an eLogin node. `eProxy` uses `ssh` to launch the wrapped command on the Cray system, and then displays the output on the eLogin node so that it appears to the user that the wrapped command is running on a Cray internal login node.

Procedure

1. List the available wrapped commands.

```
eLogin$ eProxy
eProxy version 2.0.3
Will connect to host 'eLogin1'
Usage: eProxy [--install] | [--check]
Environment variables:
    ESWRAP_LOGIN:    Forces eProxy to ssh to named host.
    ESWRAP_DEBUG:    Turns on internal debug output.
    ESWRAP_KEYFILE:  Optional ini file.
                    Default /opt/cray/eloin/eProxy/etc/eProxy.ini
    ESWRAP_ENVFILE:  Environment variable configuration file.
                    Default /opt/cray/eloin/eProxy/etc/eProxy.env
    ESWRAP_ROOT:     Allows root to execute command.
    ESWRAP_USER:     Login node user name.
    ESWRAP_CWD:      Login node working directory.
    ESWRAP_PREFIX:   Command to execute before wrapped commands.

Valid commands:
    eProxy
    cnselect
    xtprocadmin
    xtnodestat
    aprun
    apcount
    apmgr
    apkill
    apstat
    dwstat
    dwcli
    dwgateway
    dw_wlm_cli
```

2. Run a wrapped command (example, `xtprocadmin`):

```
ellogin$ xtprocadmin
```

NID	(HEX)	NODENAME	TYPE	STATUS	MODE
1	0x1	c0-0c0s0n1	service	up	interactive
2	0x2	c0-0c0s0n2	service	up	interactive
5	0x5	c0-0c0s1n1	service	up	interactive
6	0x6	c0-0c0s1n2	service	up	interactive

16 Diagnostics

16.1 Access the eLogin Console

Prerequisites

The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 33).

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 147).

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.

16.2 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.

```
eloin# 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 eloin 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 eloin kernel: Initializing cgroup subsys cpuset
Nov 13 17:45:11 eloin kernel: Initializing cgroup subsys cpu
Nov 13 17:45:11 eloin kernel: Initializing cgroup subsys cpuacct
Nov 13 17:45:11 eloin 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 eloin kernel: Command line: initrd=initrd imagename=htg.test.
20171113.sqsh cfg_set=p0 storage_profile=eloin_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.

```
eloin# journalctl -f /usr/sbin/ntpd
-- Logs begin at Mon 2017-11-13 17:45:11 CST. --
Nov 13 17:49:08 eloin 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

```

16.3 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 `dhcpcd` (Dynamic Host Configuration Protocol Server) daemon will log messages to this file. The `dhcpcd` daemon will log startup messages which may indicate problems with the DHCP configuration. As each node begins a PXE boot process, `dhcpcd` 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 `dhcpcd` 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 `dhcpcd`, 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 (`vmlinuz`), 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.

16.3.1 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`.

16.4 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.

This example shows CLE 6.0.UP06 image names. Actual output will show image names for this system and the current CLE release.

```
smw# enode list
NAME          CONFIGSET      STORAGE_PROFILE  ESD_GROUP
IMAGE
MGMT_MAC      PARAMETERS  STATE
  elogin1  p0          elogin_default  elogin      elogin-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
  elogin2  p0          elogin_default  elogin      elogin-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
  elogin3  p0          elogin_default  elogin      elogin-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
  elogin4  p0-elogin4-5  elogin_default  elogin      elogin-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
  elogin5  p0-elogin4-5  elogin_default  elogin      elogin-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 `kdumptool 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:

```
SIZE_LOW = (Recommendation * RAM_in_TB) + (Adjustment)M
```

```
SIZE_LOW = (72 * 3) + (40)M
```

```
SIZE_LOW = 256M
```

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:

```
SIZE_HIGH = (Recommendation * RAM_in_TB) + (LUNs/2)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:

```
SIZE_HIGH = (Recommendation * RAM_in_TB) + (LUNs/2)M
```

```
SIZE_HIGH = (116 * 3) + (6 / 2)M
```

```
SIZE_HIGH = 351M
```

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 eloin2
Shutting down the following node(s):
eloin2
['eloin2']: All node(s) started shutdown process.
smw# enode status eloin2
NODE          PING  POWER          STATE
eloin2  Down  Chassis Power is off  node_off
smw# enode boot --pxe eloin2
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.


```
eloin2# 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.

16.5 Analyze KDUMP vmcore Files

Prerequisites

- SMW must have the crash utility installed and ample disk space available
 - If the SMW does not meet these requirements, another SUSE linux system can be used instead.

This procedure assumes knowledge of the Linux crash utility. See `crash(8)` for more information.

About this task

After a kernel panic has occurred and the KDUMP procedure is complete, analyze the vmcore file to debug the issue that caused the panic.

Procedure

1. Locate the KDUMP directory with the date/time stamp of the panic of interest in `/var/crash/` and verify the files produced by KDUMP reside in that directory.

```
eloin2# cd /var/crash
eloin2# ls
2018-01-23-22:32  2018-01-23-23:46  2018-01-29-20:32  lost+found
eloin2# ls 2018-01-29-20:32
dmesg.txt  README.txt  System.map-4.4.92-6.18-default  vmcore
vmlinuz-4.4.92-6.18-default.gz
```

In this example, the kernel version is `4.4.92-6.18` and the date stamp of the panic is `2018-01-29-20:32`. Substitue accurate values for the site's system.

Table 10. KDUMP Files

File	Description
dmesg.txt	kernel messages log ring buffer
README.txt	detailed information about the KDUMP files provided

File	Description
System.map-4.4.92-6.18-default	reference table to correlate symbol names to their addresses in memory
vmcore	memory image used to debug and determine the cause of the crash
vmlinux-4.4.92-6.18-default.gz	statically linked executable file that contains the Linux kernel

2. Copy the KDUMP directory to the SMW.

If the SMW does not have enough disk space available, use another SUSE linux system with the crash utility installed.

```
smw# cd /directory/used/for/KDUMP/debug
smw# scp -r root@eloin:/var/crash/2018-01-29-20:32 .
smw# ls
2018-01-29-20:32
smw# cd 2018-01-29-20:32
smw# ls
dmesg.txt  README.txt  System.map-4.4.92-6.18-default  vmcore
vmlinux-4.4.92-6.18-default.gz
```

3. Copy the kernel-default-debuginfo RPM (kernel-default-debuginfo-4.4.92-6.18x86_64.rpm) to the KDUMP debug directory used in the previous step.

The kernel-default-debuginfo RPM must have the same version number as the System.map and vmlinux versions. This file can be downloaded from Cray archive servers at

/data/cf/replicated/mirrors/repos/suse/SUSE/Updates/SLE-SERVER/<VER>/x86_64/update_debug/x, where <VER> is the kernel version that is running on the system. If there are any problems accessing the archived kernel-default-debuginfo RPMs, contact Cray Technical Services.

```
smw# scp root@example-machine:/directory/to/archived/rpms/kernel-default-
debuginfo-4.4.92-6.18.1.x86_64.rpm .
smw# ls
dmesg.txt  kernel-default-debuginfo-4.4.92-6.18.1.x86_64.rpm  README.txt
System.map-4.4.92-6.18-default  vmcore  vmlinux-4.4.92-6.18-default
```

4. Find the name of the vmlinux debug file.

```
smw# rpm -qlp kernel-default-debuginfo-4.4.92-6.18.1.x86_64.rpm | grep vmlinux
/usr/lib/debug/boot/vmlinux-4.4.92-6.18-default.debug
```

5. Extract the vmlinux-4.4.92-6.18-default.debug file from the kernel-default-debuginfo-4.4.92-6.18.1.x86_64.rpm file.

```
smw# rpm2cpio ./kernel-default-debuginfo-4.4.92-6.18.1.x86_64.rpm | \
cpio -iv --to-stdout ./usr/lib/debug/boot/vmlinux-4.4.92-6.18-default.debug > ./
vmlinux-4.4.92-6.18-default.debug
./usr/lib/debug/boot/vmlinux-4.4.92-6.18-default.debug
3724189 blocks
smw# ls
dmesg.txt  kernel-default-debuginfo-4.4.92-6.18.1.x86_64.rpm  README.txt
System.map-4.4.92-6.18-default  vmcore  vmlinux-4.4.92-6.18-default
vmlinux-4.4.92-6.18-default.debug
```

6. Execute the crash utility using the `vmlinux-4.4.92-6.18-default`, `vmlinux-4.4.92-6.18-default.debug`, and `vmcore` files.

```
smw# crash ./vmlinux-4.4.92-6.18-default ./vmlinux-4.4.92-6.18-default.debug ./
vmcore
    ARCH: x86_64
    Using: /usr/bin/crash_x86_64

    crash_x86_64 7.2.0
    Copyright (C) 2012-2013 Cray Inc.
    Copyright (C) 2002-2017 Red Hat, Inc.
    Copyright (C) 2004, 2005, 2006, 2010 IBM Corporation
    Copyright (C) 1999-2006 Hewlett-Packard Co
    Copyright (C) 2005, 2006, 2011, 2012 Fujitsu Limited
    Copyright (C) 2006, 2007 VA Linux Systems Japan K.K.
    Copyright (C) 2005, 2011 NEC Corporation
    Copyright (C) 1999, 2002, 2007 Silicon Graphics, Inc.
    Copyright (C) 1999, 2000, 2001, 2002 Mission Critical Linux, Inc.
    This program is free software, covered by the GNU General Public License,
    and you are welcome to change it and/or distribute copies of it under
    certain conditions. Enter "help copying" to see the conditions.
    This program has absolutely no warranty. Enter "help warranty" for
details.

    GNU gdb (GDB) 7.6
    Copyright (C) 2013 Free Software Foundation, Inc.
    License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/
gpl.html>
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law. Type "show copying"
    and "show warranty" for details.
    This GDB was configured as "x86_64-unknown-linux-gnu"...

    crash_x86_64: SECTION_SIZE_BITS = 27
        KERNEL: ./vmlinux-4.4.92-6.18-default
    DEBUG KERNEL: ./vmlinux-4.4.92-6.18-default.debug
    DUMPFILE: ./vmcore [PARTIAL DUMP]
        CPUS: 16
        DATE: Mon Jan 29 14:31:57 2018
        UPTIME: 00:05:57
    LOAD AVERAGE: 0.20, 0.34, 0.18
        TASKS: 376
    NODENAME: elogin2
    RELEASE: 4.4.92-6.18-default
    VERSION: #1 SMP Fri Oct 20 18:58:48 UTC 2017 (a69df70)
    MACHINE: x86_64 (2600 Mhz)
        MEMORY: 64 GB
        PANIC: "BUG: unable to handle kernel NULL pointer dereference
at (null)"
        PID: 18502
    COMMAND: "bash"
        TASK: ffff8807fc444780 [THREAD_INFO: ffff88080cf60000]
        CPU: 3
        STATE: TASK_RUNNING (PANIC)

    Setting scroll off while initializing PyKdump
    /usr/lib64/crash/extensions-python2/mpykdump64-py2.so: shared object loaded

    crash_x86_64>
```

The dump can now be analyzed using the crash utility. For example, to get a backtrace:

```
crash_x86_64> bt
PID: 18502 TASK: ffff8807fc444780 CPU: 3 COMMAND: "bash"
#0 [ffff88080cf63b00] machine_kexec at ffffffff81057a4c
#1 [ffff88080cf63b50] __crash_kexec at ffffffff81113cda
#2 [ffff88080cf63c10] crash_kexec at ffffffff81113dac
#3 [ffff88080cf63c20] oops_end at ffffffff8101a564
#4 [ffff88080cf63c40] no_context at ffffffff81064be7
#5 [ffff88080cf63c90] __bad_area at ffffffff81079dd7
#6 [ffff88080cf63cc8] __do_page_fault at ffffffff81065a57
#7 [ffff88080cf63d38] do_page_fault at ffffffff81065b2b
#8 [ffff88080cf63d60] page_fault at ffffffff8160c628
[exception RIP: sysrq_handle_crash+18]
RIP: ffffffff814156a2 RSP: ffff88080cf63e18 RFLAGS: 00010286
RAX: 0000000000000016 RBX: ffffffff81ef8600 RCX: 0000000000000000
RDX: 0000000000000001 RSI: 0000000000000246 RDI: 0000000000000063
RBP: 0000000000000063 R8: ffffffff8222e9c0 R9: ffff88102ff68771
R10: 0000000000006460 R11: 0000000000000000 R12: 0000000000000000
R13: 0000000000000007 R14: 0000000000000002 R15: 0000000001453be0
ORIG_RAX: ffffffff814156a2 CS: 0010 SS: 0018
#9 [ffff88080cf63e18] __handle_sysrq at ffffffff81415dbc
#10 [ffff88080cf63e40] write_sysrq_trigger at ffffffff814161f4
#11 [ffff88080cf63e50] proc_reg_write at ffffffff8126c9b9
#12 [ffff88080cf63e68] __vfs_write at ffffffff812059e3
#13 [ffff88080cf63ee0] vfs_write at ffffffff812066bd
#14 [ffff88080cf63f18] sys_write at ffffffff81207732
#15 [ffff88080cf63f50] entry_SYSCALL_64_fastpath at ffffffff8160a26e
RIP: 00007f26656032d0 RSP: 00007ffd471089d8 RFLAGS: 00000246
RAX: ffffffff814156a2 RBX: 0000000000000001 RCX: 00007f26656032d0
RDX: 0000000000000002 RSI: 00007f2666169000 RDI: 0000000000000001
RBP: 00007f26658c4280 R8: 000000000000000a R9: 00007f2666153700
R10: 000000000dabaa997 R11: 0000000000000246 R12: 000000000139d310
R13: 0000000000000001 R14: 0000000000000000 R15: 00007ffd47108988
ORIG_RAX: 0000000000000001 CS: 0033 SS: 002b
crash_x86_64>
```

16.6 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 kumps on eLogin nodes, perform a workaround if there is no boot session ID, and trigger a `kdump` if an eLogin node has no kumps 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 `xtumpsys` 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 `xtumpsys` 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# xtumpsys -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, but the kdump utility is enabled, go to step 5 on page 144.
- If the node has no kdump, and the kdump utility has NOT been enabled, do steps 2–5 in [Enable and Start kdump](#) on page 133, and then return to this procedure and go to step 5 on page 144.

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] eloin2
[582K] eloin_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] kdump
[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.eloin2.swp
[341K] console.eloin
[ 25M] console.eloin1
[ 17M] console.eloin2
[6.6M] console.eloin3
[9.3M] console.eloin4
[9.0M] console.eloin5
[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.

- xtdumpsys has been run on an eLogin node.

- There are no kdump 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 133, 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 kdump 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: kdump 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 kdump. 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.

6. 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 kdump 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
```



```
eloin2 '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@orioneloin4:/
var/crash/2017-12-21-18:18.tar.gz' /var/opt/cray/dump/
p0-20171221t080455-1712211853/edumpsys/eloin2/kdumps'
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
eloin2 '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:
#####
```

17 Troubleshooting

17.1 Boot the eLogin Node with the DEBUG Shell

Prerequisites

- Node is in the `node_off` state.
- The node registry must have the required fields for each node (see [Register eLogin Nodes](#) on page 33).
- 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 multiple nodes.

Procedure

PXE Boot

1. Set the DEBUG shell and boot the node.
 - To PXE boot, go to step 2 on page 146
 - To boot from disk, go to step 3 on page 147
2. Set the DEBUG shell and PXE boot the node.
 - a. Update the node to enable the DEBUG shell.

```
smw# enode update --set-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 --set-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 147).

5. Disable DEBUG shell for next boot.

```
smw# enode update --unset-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
```

17.2 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>.

17.3 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`.

17.4 Disable the Intel TOC Watchdog Timer on eLogin Nodes

Prerequisites

- KDUMP utility is configured and enabled on the eLogin node
- The system is currently experiencing unexplained hard reboots and all debug information is lost with the reboot

About this task

The Intel TCO (Total Cost of Ownership) Watchdog Timer is a hardware watchdog whose purpose is to reboot the computer when the system hangs. If the watchdog does not receive a ping at a regular interval it will cause a hardware reset. An Intel TCO watchdog timer triggered reboot may occur prior to the KDUMP utility being triggered to create a vmcore memory dump. Once the watchdog triggered hard reboot has begun, the vmcore necessary for debugging the watchdog timeout has been lost.

The resolution to this problem is to verify that the Intel TOC Watchdog Timer is loaded and active on the eLogin node and to then remove the module(s) from the system. With the Intel hardware watchdog removed, a KDUMP can be triggered on the hung system to capture a vmcore dump file.

Procedure

1. Verify the Intel TOC Watchdog Timer is loaded and active on the eLogin node.

Search the system messages on the eLogin node for "iTOC":

```
ellogin# dmesg | grep iTCO
[ 164.077562] iTCO_vendor_support: vendor-support=0
[ 164.121436] iTCO_wdt: Intel TCO WatchDog Timer Driver v1.11
```

```
[ 164.121465] iTCO_wdt: Found a Patsburg TCO device (Version=2,
TCOBASE=0x0860)
[ 164.130806] iTCO_wdt: initialized. heartbeat=30 sec (nowayout=0)
```

The messages listed above indicate the hardware watchdog is loaded and active.

2. Identify the names of the Intel TOC Watchdog Timer modules.

```
eLogin2# lsmod | grep iTCO
iTCO_wdt                16384  0
iTCO_vendor_support     16384  1 iTCO_wdt
```

In this example, the modules are named `iTCO_wdt` and `iTCO_vendor_support`.

3. Remove the Intel TOC Watchdog Timer modules from the currently running system.

```
eLogin# rmmod iTCO_wdt iTCO_vendor_support
[70561.902904] iTCO_wdt: Watchdog Module Unloaded
[70561.928297] iTCO_vendor_support: Module Unloaded
```

The Intel TOC Watchdog Timer modules are now removed. This ensures if a system hang occurs on the eLogin node, the KDUMP vmcore dump utility can be triggered. After rebooting the eLogin, the Intel TOC Watchdog Timer will be loaded and active once again.

For persistent disabling of the iTCO watchdog timer driver, edit the `/etc/modprobe.d/blacklist-watchdog` file and add `blacklist iTCO_wdt`.

For more information, see *Using the Intel ICH Family Watchdog Timer (WDT)* (<http://application-notes.digchip.com/027/27-45785.pdf>).

18 Supplemental Information

18.1 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)
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>.

18.2 Glossary

Term	Definition
ACL	access control list

Term	Definition
	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.
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.

Term	Definition
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.
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

Term	Definition
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.