



XC™ Series eLogin Installation Guide (CLE 6.0.UP01)

Contents

1 About the XC Series eLogin Installation Guide.....	4
2 Source ISOs.....	6
3 eLogin Architecture.....	7
3.1 eLogin Network Architecture.....	8
3.2 The Provisioning Process.....	12
4 The eLogin Installation Process.....	13
4.1 CSMS Configuration Worksheet.....	13
4.2 Configure CMC Hardware BIOS.....	15
4.3 Install CentOS 7.1 on Cray Management Controller (CMC).....	21
4.4 Install Cray System Management Software on CMC.....	28
4.5 Install eLogin Software on CMC.....	30
4.6 Configure CSMS On Management Controller.....	30
4.7 Install SMW and Configure the CMC Connection.....	36
4.8 eLogin Node Installation.....	38
4.8.1 eLogin Hardware and BIOS RAID Setup.....	38
4.8.2 Create a Minimum eLogin Config Set.....	45
4.8.3 Ironic Node Enrollment.....	53
4.8.4 Configure OpenStack Fuel.....	55
4.9 Configure and Manage an eLogin Image.....	56
5 Enable LiveUpdates Support for eLogin Nodes.....	62
6 Diagnostics.....	64
6.1 eLogin Console Access.....	64
6.2 The journalctl Command.....	64
6.3 The /var/log Directory.....	65
6.4 Ansible Install Logs.....	65
6.5 The cray_dumpsys Command.....	66
6.6 Configure eLogin Cray_dumpsys Plugin.....	66
6.7 OpenStack Log File Locations.....	67
6.8 OpenStack Diagnostics.....	67
6.8.1 Heat Diagnostic Commands.....	68
6.8.2 Nova Diagnostic Commands.....	69
6.8.3 Ironic Diagnostic Commands.....	69
6.9 Common Issues.....	70
6.9.1 Disk Space On CMC and eLogin Node.....	70
6.9.2 Recovering from Broken CSMS Installation.....	71

6.9.3 Repeated Cycle Rebooting CentOS Deploy Image.....	72
---	----

1 About the XC Series eLogin Installation Guide

The XC™ Series eLogin Installation Guide provides installation procedures for Cray eLogin nodes running externally to Cray XC series systems.

Audience and Scope

This publication is intended for system installers, administrators, and anyone who installs and configures software on a Cray XC Series system. It assumes competence with standard Linux and open source tools.

Record of Revision

Revision	Date	Content Information
XC Series eLogin Installation Guide CLE6.0 UP01 (this publication)	June 15, 2016	<ul style="list-style-type: none">• Content aligned with Alpaca 1.1.1• Upgrade processes not included
CLE6.0 UP01 draft v.0.9.4	June 7, 2016	<ul style="list-style-type: none">• Content aligned with Alpaca 1.1.1• CMC BIOS updates• CentOS-7 install updates• BIOS RAID setup updates• Upgrade processes not included
CLE6.0 UP01 v.1	May 27, 2016	<ul style="list-style-type: none">• Content aligned with Alpaca 1.1.1• Upgrade processes not included

This publication was previously titled eLogin Installation Guide CLE6.0 UP01 v.1. The new title XC Series eLogin Installation Guide CLE6.0 UP01 complies with the standard titling convention adopted and implemented for all publications within the technical publications department as of June 10, 2016. Previous versions of this publication will not be retitled.

Typographic Conventions

Monospace

A **Monospace** font indicates program code, reserved words or library functions, screen output, file names, path names, and other software constructs

Monospaced Bold

A **bold monospace** font indicates commands that must be entered on a command line.

Oblique or Italics

An *oblique* or *italics* font indicates user-supplied values for options in the syntax definitions

Proportional Bold

A **proportional bold** font indicates a user interface control, window name, or graphical user interface button or control.

Alt-Ctrl-f

Monospaced hyphenated text typically indicates a keyboard combination

Feedback

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

2 Source ISOs

Installing the eLogin software requires three ISO files. These ISOs constitute the software for installing the Cray System Management Software (CSMS) and eLogin onto the Cray Management Controller (CMC) hardware:

- `Cray-CentOSbase7.1-201605121026.iso`
- `csms_centos71-1.1.1-201605231632.iso`
- `ellogin-6.0.96-201605131428.iso`

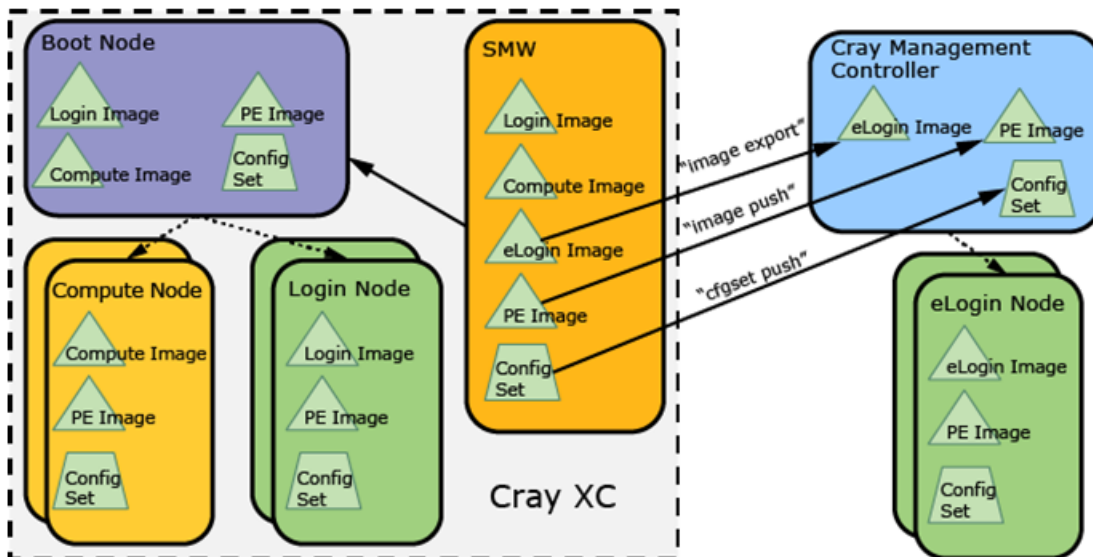
3 eLogin Architecture

A Cray eLogin node expands the role of the internal login node, by providing an external login (eLogin) and software development environment, with access to the Cray Lustre file system (CLFS) or Cray Sonexion, for Cray XC series systems.

The external system uses the Cray System Management Software (CSMS) installed on the Cray Management Controller (CMC) to manage the deployment of eLogin images to the Cray Development and Login (CDL) nodes. The CMC connects to the Cray Software Management Workstation (SMW). The SMW provides shared image and configuration services.

The diagram below shows the general architecture of the nodes used by eLogin. Configuration data is shared between Cray internal nodes and the eLogin nodes. The Cray Programming Environment (PE) is shared between the internal Cray nodes and the eLogin nodes. The canonical data for all nodes is always stored on the SMW node.

Figure 1. General Architecture for eLogin Nodes



Each node type in the system has a specified hardware platform and a software release package that provides an operating system and custom Cray software to support its role.

HARDWARE

The CMC is deployed to Dell R730 rack servers. For eLogin node deployment, either the Dell R730 or R630 is specified for use depending on the customer requirement.

SOFTWARE

CentOS 7 Operating System	CentOS 7 is the base operating system for CSMS and is installed using the CentOS 7 release media.
SLES12 Operating System	eLogin nodes run the SUSE Linux Enterprise Server (SLES™) operating system. The eLogin installation process installs SLES 12 during the image creation step on the SMW. The repositories are installed on the SMW during the SMW installation process.
Cray System Management Software (CSMS)	CSMS is Cray's supported implementation of the OpenStack framework; it contains the base OpenStack installation as well as eLogin specific customizations. The eLogin installation process installs CSMS on top of the base CentOS 7 installation from the CSMS installation disk, and then adds eLogin customizations via the eLogin installation ISO.
eLogin Node Software	<p>In addition to SLES 12, eLogin nodes require eLogin software configuration to the Cray Linux Environment (CLE) and Programming Environment (PE). CMC system software controls the eLogin software, which is distributed in repositories installed as a part of SMW installation.</p> <p>The eLogin image recipe on the SMW determines the specific software installed on the eLogin node. Cray provides a default recipe that can be cloned and modified to reflect site specific customizations.</p>

3.1 eLogin Network Architecture

The Cray System Management Software (CSMS) installation requires that the Cray Management Controller (CMC) and eLogin nodes are already attached to the appropriate networks in order to function. The CMC should have its first network device connected to a site administrative network. This may be the network connected to the SMW, thus making the CMC a peer of the SMW or a private network behind the SMW.

The SMW and CMC must be connected. Each site determines if the CMC and SMW are peers on the site administrative network, or if the CMC is behind the SMW.



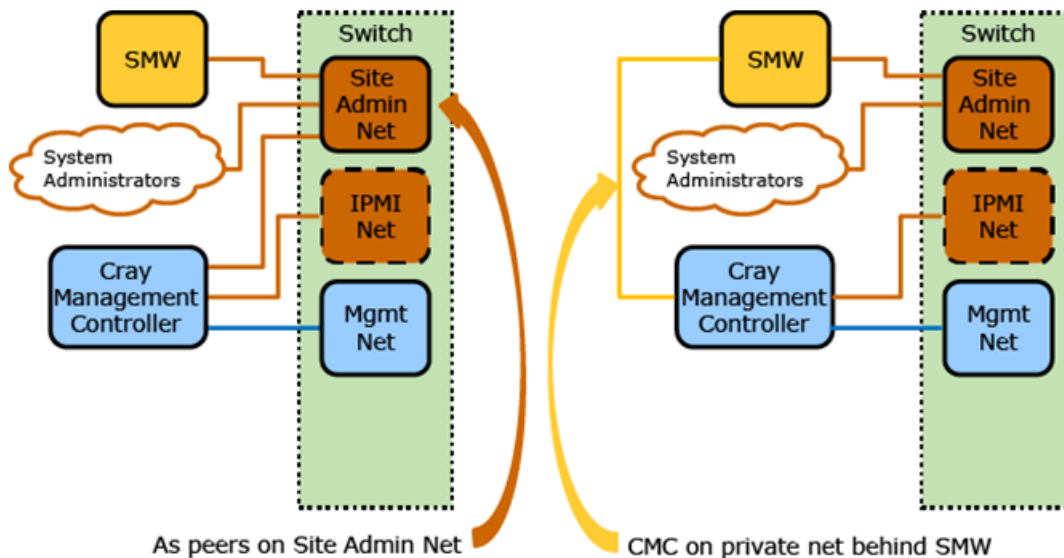
WARNING: For security reasons, configure the CMC behind the SMW on a private network.

The CMC should have its second network device connected to the management network, which is used for image provisioning, and its third network device connected to the IPMI network, which is used for remote console and power control.

The following diagram shows the two methods of connecting the SMW to the CMC:

- As peers on Site Admin Net
- CMC on private net behind SMW

Figure 2. Connecting SMW to Management Controller: eLogin System



eLogin Networks

eLogin software uses internal and external designations to classify networks. For example, the *Maint-Network* is classified as an internal network that is accessible only to the CMC. External networks such as the *Site-User-Network* and *Site-Admin-Network* enable users from outside the system to gain access.

The diagrams below show an overview of the hardware components and networks used in an eLogin system. There may be additional network connections as needed by a site. The following list describes the networks that are used in an eLogin system.

- Mgmt-Network** An internal management network that connects the CMC to the eLogin nodes, switches, RAID controllers, and IPMI devices. This network allows CSMS to manage and provision the eLogin systems.
- IPMI-Network** An internal management network that connects the CMC to the eLogin IPMI devices. This network allows CSMS to provide remote console access and power control.
- Site-Admin-Network** An external administration network that enables site administrators to log into the CMC and SMW. The IP address of this network can be customized during CSMS installation. Cray recommends that the IPMI interface of the CMC also be connected to this network to provide remote console and power management for the CMC.
- Site-User-Network** External user (site) network used by eLogin nodes. This network provides user access and may provide authentication services like LDAP. The name and IP addresses on this network are provided in the configuration set. Connections to additional site-specific networks are optional.
- IB-Network** Internal Infiniband® network used for high-speed Lustre LNet traffic.

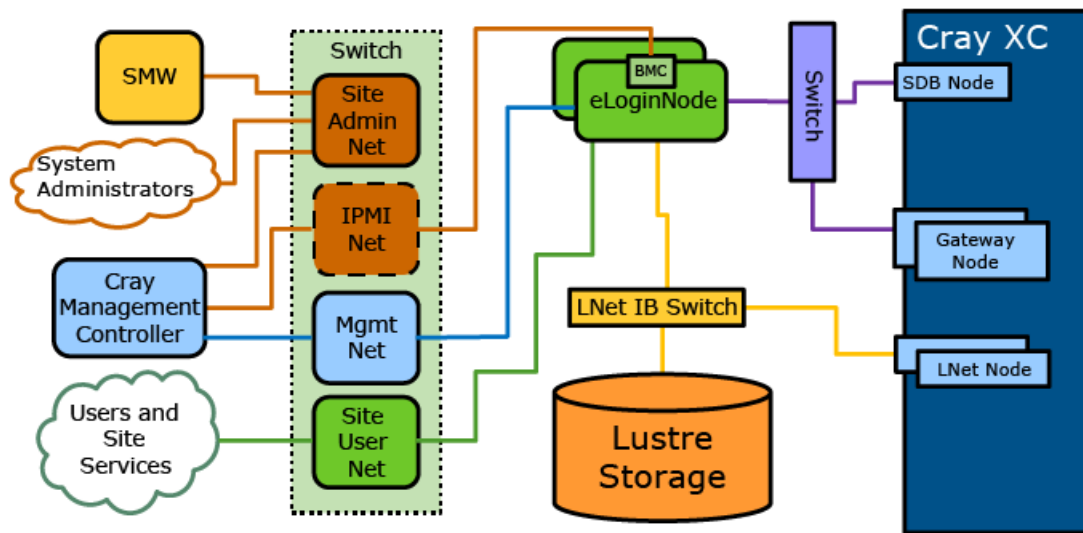
eLogin to Cray Network Attachment

There are four distinct configurations for attaching eLogin nodes to a Cray XC system. The first 1GbE device of each eLogin node must be connected to the management network. Depending on the eLogin hardware configuration, this may be the first Ethernet device in the case of the 4x-1GbE LOM, or the third Ethernet device in the case of the 2x-10GbE / 2x-1GbE LOM option. The dedicated IPMI device port must be connected to the IPMI network. An Ethernet device must also be connected to the site user network to allow users to log onto the eLogin node. This site user network may be 1GbE or 10GbE depending on site infrastructure.

eLogin Nodes Direct Connection to SDB Node

This configuration connects the service database (SDB) node directly to the eLogin nodes via a switch. Access to the Cray XC is via the eLogin node.

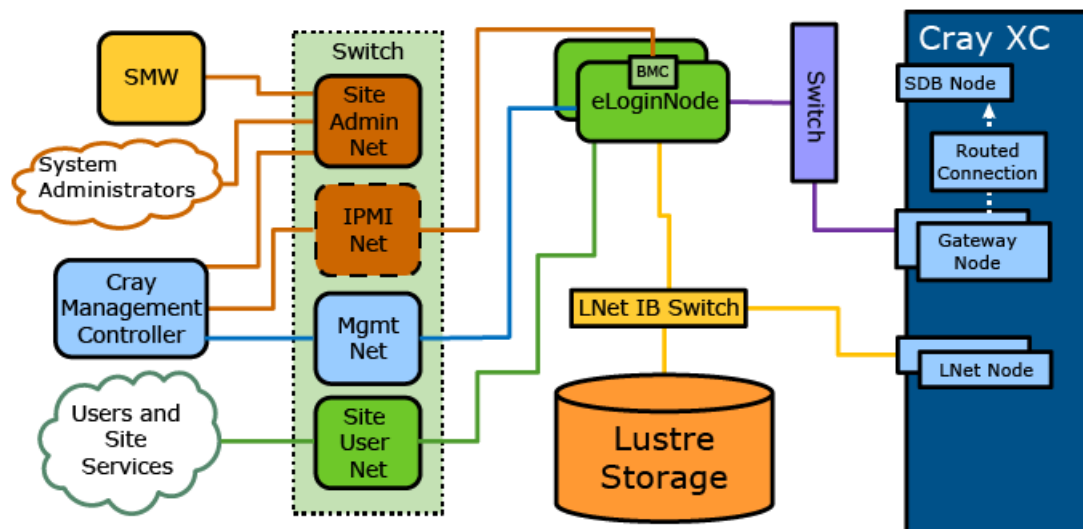
Figure 3. eLogin Nodes Direct Connection to SDB Node: eLogin System Topology



eLogin Nodes Routed Via Gateway to SDB Node

This configuration connects the SDB node to the eLogin nodes routed through the Gateway node. Job submission routes from the eLogin node through the Gateway node.

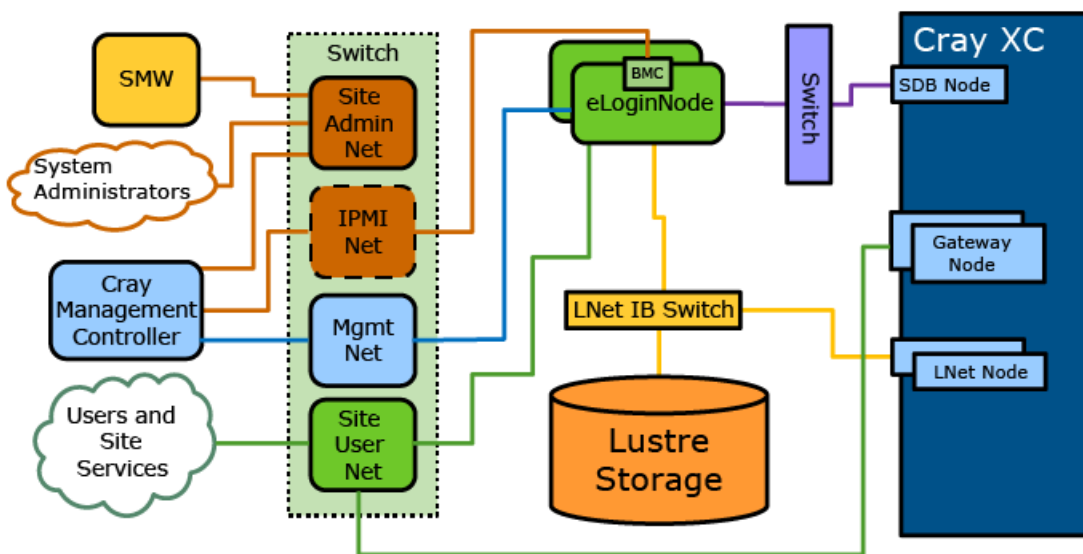
Figure 4. eLogin Nodes Routed Via Gateway to SDB Node: eLogin System Topology



eLogin Nodes Direct Connection to SDB Node with Site User Accessible Gateway

This configuration places Gateway nodes on the site user network (allowing site users to connect to the Gateway nodes directly) and connects the eLogin nodes directly to the SDB node via a switch.

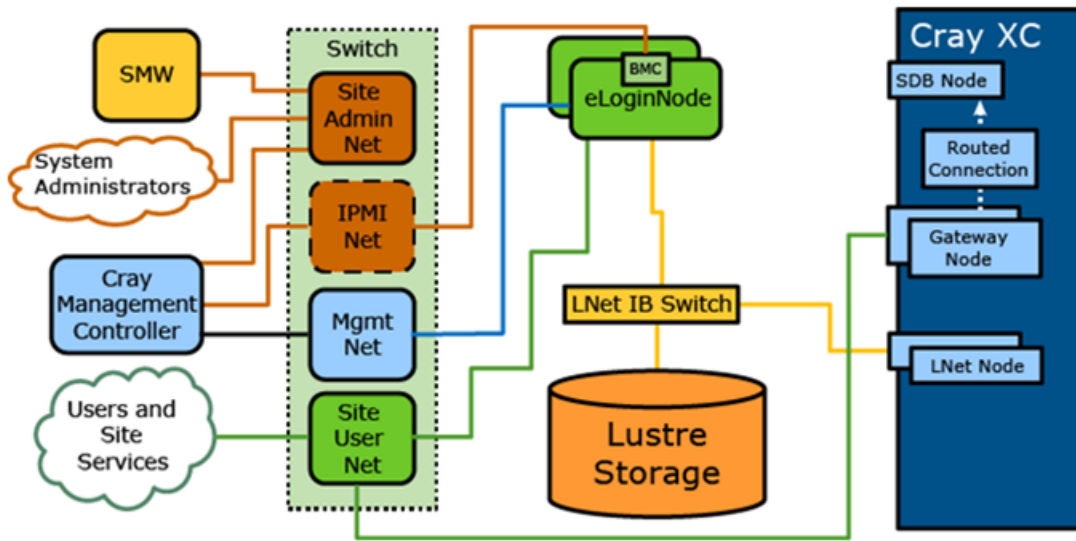
Figure 5. eLogin Nodes Direct Connection to SDB Node with Site User Gateway: eLogin System Topology



eLogin Nodes Routed to SDB Node with User Accessible Gateway

This configuration places the Gateway nodes on the site user network (allowing site users to connect to the gateway nodes directly) and connects the SDB node via the Gateway node. Job submission routes from the eLogin node through the Gateway node.

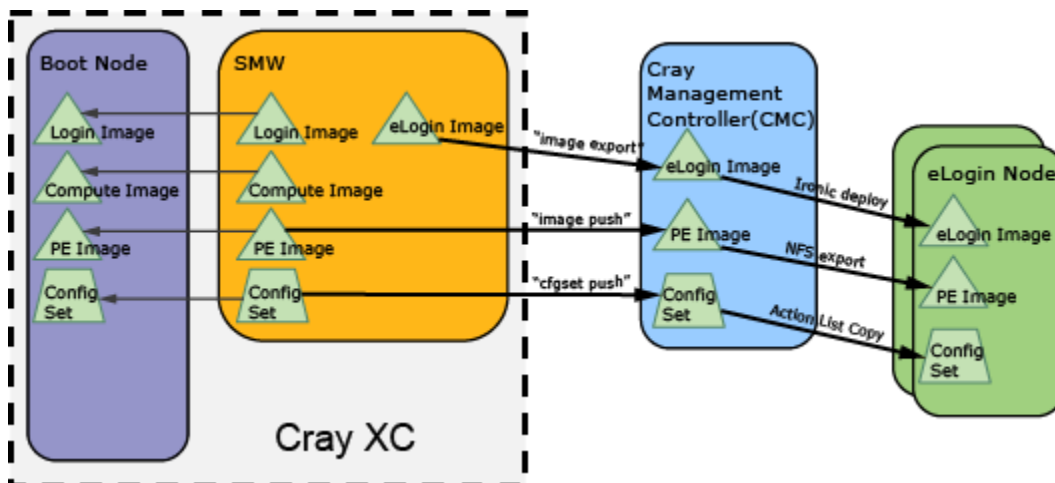
Figure 6. eLogin Nodes Routed to SDB Node via User Accessible Gateway: eLogin System Topology



3.2 The Provisioning Process

Provisioning an eLogin node starts on the Cray System Management Workstation (SMW) with building the eLogin and Cray Programming Environment (PE) images and preparing the config set. From the SMW, an administrator exports the eLogin image to Glance on the Cray Management Controller (CMC) and pushes the PE image and config set. From the CMC, Ironic deploys the eLogin image to a Cray Development and Login (CDL) node. During the boot process, the PE image is exported to the eLogin node with `nfs_export`.

Figure 7. Provisioning Process for eLogin Node



4 The eLogin Installation Process

Perform the following procedure sections for the initial installation of an eLogin node:

1. Configure CMC Hardware BIOS
2. Install CentOS and Cray System Management Software (CSMS) on the Cray Management Controller (CMC).
3. Install CSMS on CMC.
4. Install eLogin software on CMC.
5. Install SMW and configure the CMC connection.
6. eLogin node installation.

4.1 Cray System Management Software (CSMS) Configuration Worksheet

The following table lists the configuration items for which site-specific values must be known during the CSMS installation process. Gather this information prior to installation.

Item	Configuration Variable	Value
Hostname		
Hardware Platform	platform	
Site Network Interface (e.g. eth0)		
Default Gateway	default_gateway	
Site (external) IP address for site system administration	site_ip	
Site Subnet	site_subnet	
Site Routing Prefix	site_prefix_length	
Site gateway	site_gateway, defaults to default_gateway	
Management interface (e.g., eth1)	management_network_device	
Management Network IP Address for tenant node management and image deployment.	management_ip	

Item	Configuration Variable	Value
Management Network Subnet	management_subnet	
Management Network Prefix	management_prefix_length	
Management Network Gateway	management_gateway	
Management Allocation Pool Start	management_allocation_pool_start	
Management Allocation Pool End	management_allocation_pool_end	
DNS servers	dns1_server_ip dns2_server_ip	
DNS Domain	domain	
External NTP host	ntp_servers (a list)	
OpenStack Admin Password	admin_password	
Keystone Password	keystone_mysql_password	

Each CDL node also needs the following information:

Item	Value
BMC IP address	
Boot interface MAC address	

Common Configuration Options

The following tables lists common Ansible configuration values. The default values are typically declared in `/etc/opt/cray/openstack/ansible/group_vars/all/all`.



WARNING: Do not directly change the default values in the file: `/etc/opt/cray/openstack/ansible/group_vars/all/all`. Instead, use site override files to modify the values.

Table 1. Common Configuration Options

Configuration Option	Definition	Purpose	Type	Acceptable Values
<code>platform</code>	Target hardware platform of managed nodes, and not the management controller itself.	Specifies the target platform to trigger hardware specific options, such as console support.	Fixed string	CS300, Dell, Newisys, HSSCapmc, Libvirt, Virtualbox
<code>base_dir</code>	Path to the system installed ansible configuration.	Specifies the base directory	Directory path	Valid file system path

The default value for the `platform` option is `CS300`, whereas that for the `base_dir` option is `/etc/opt/cray/openstack/ansible`.

4.2 Configure CMC Hardware BIOS

About this task

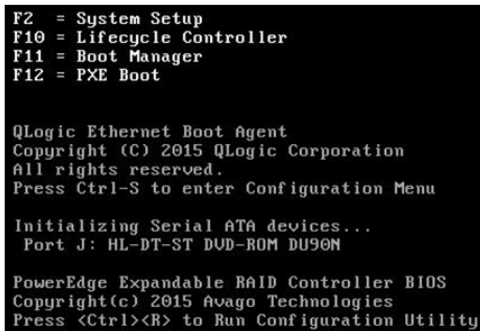
The Cray Management Controller (CMC) hardware has six 1TB disks available (RAID-5 with hot spare). The CMC must be configured to have two virtual disks visible to the operating system. The first virtual disk is used for the base operating system (O/S), and must be of sufficient size to hold the O/S, logs, and Cray's Programming Environment (PE).

Perform the following procedure to setup disks for the CMC and configure the hardware BIOS:

Procedure

1. On startup of the CMC node, press **Ctrl-R** when prompted to enter RAID setup.

Figure 8. Enter RAID Setup: CMC BIOS



```
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
```

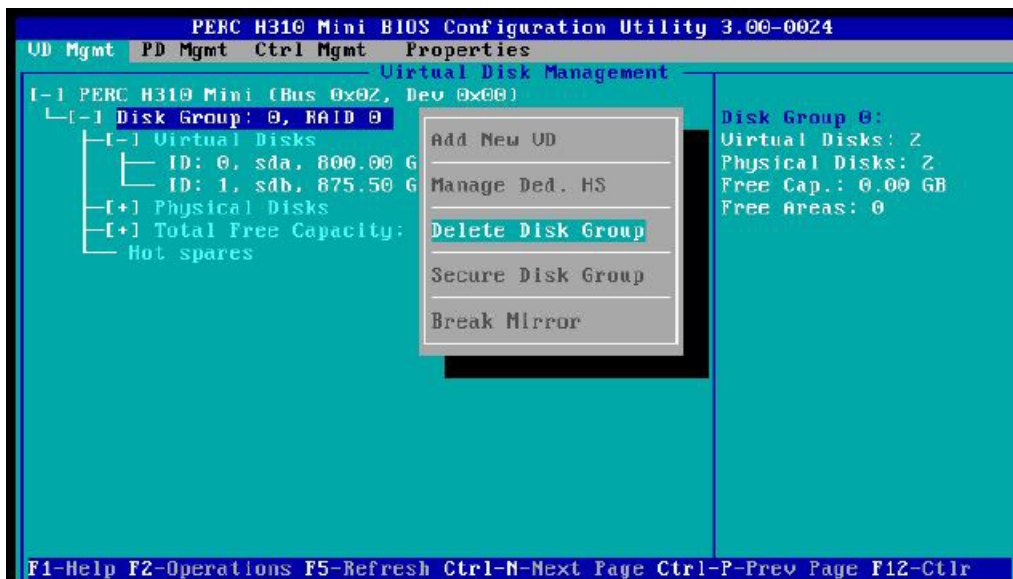
The RAID configuration screen opens.

Figure 9. RAID Configuration Screen: CMC BIOS

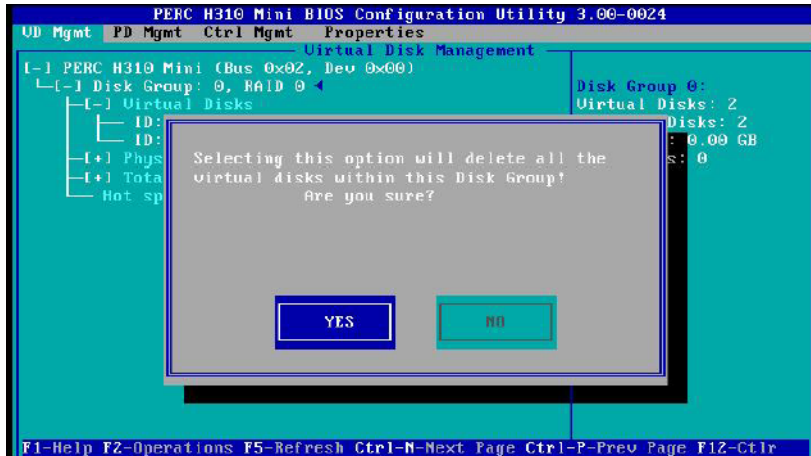


2. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration. Otherwise, skip this step.
 - a. Select the disk.
 - b. Press **F2** key to get a list of operations.
 - c. Select **Delete Disk Group** and press **Enter**.

Figure 10. Delete Disk Group: CMC BIOS

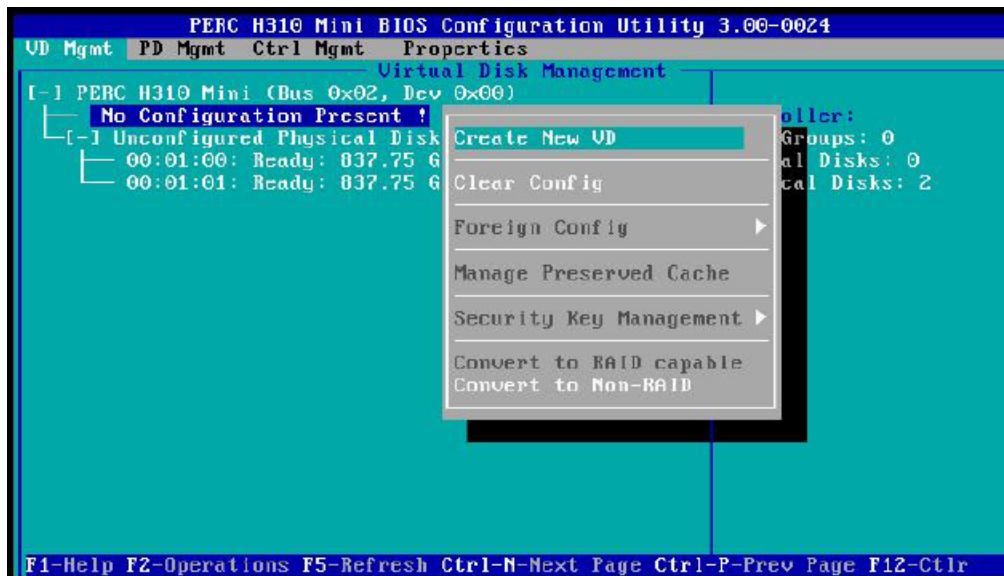


- d. Confirm the selection **Yes**, and press **Enter**.



3. Create a new Virtual Disk A.
 - a. In the virtual disk management window, 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 11. Create Virtual Disk: CMC BIOS



4. Move cursor to select the disk ID, and then press spacebar on keyboard to add disk to RAID.
5. Set the RAID Level to **RAID 5**, with hot spare.

Figure 12. Set RAID Level 5: CMC BIOS

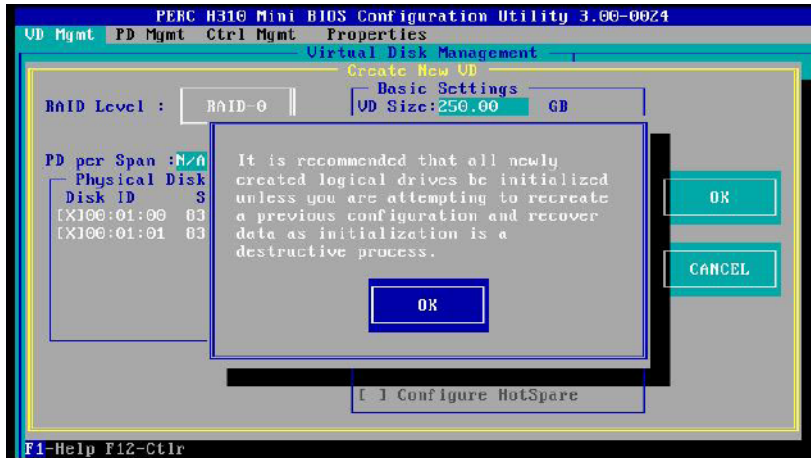


6. Set **VD Size** and **VD Name** for virtual disk A.
 - a. Set the **VD Size** of the first disk to two-thirds (2/3) of total available disk space.
 - b. Set the **VD Name** to **SDA**.

Figure 13. Disk Size and Name Setting for Virtual Disk A: CMC BIOS



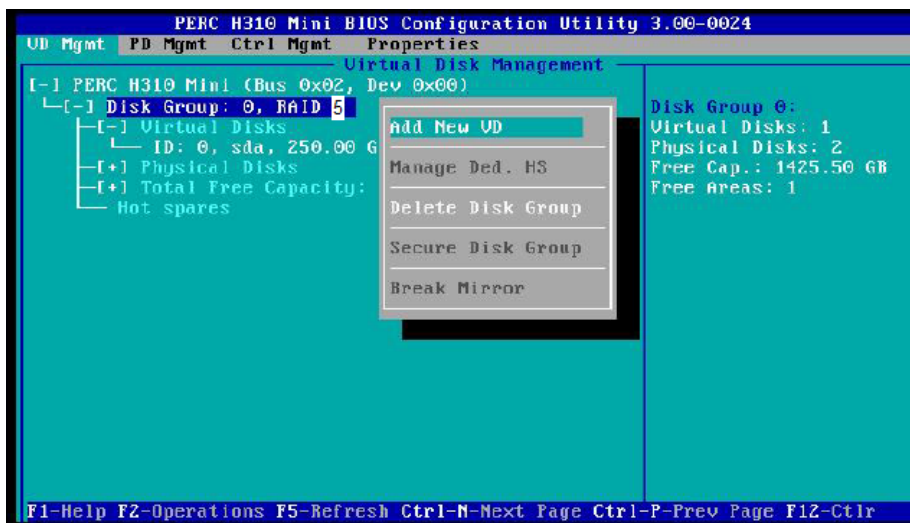
- c. Select **Ok** in the window, and then in the initialization message pop-up window, select **Ok**.



7. Create a new Virtual Disk B.

- In the virtual disk management window, navigate to **Disk Group: 0 RAID-5** using the keyboard up/down arrows.
- Press **F2** to access the disk creation menu.
- Select **Add New VD**.

Figure 14. Create New Virtual Disk B: CMC BIOS



- Set the **VD Name** to **SDB**.

The VD size should be set to the remaining disk space and remain in that state.

Figure 15. Disk Size and Name Setting for Virtual Disk B: CMC BIOS



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



The CMC now has two disks available to install on.

Figure 16. Two Virtual Disks Available: CMC BIOS



8. Press **Esc** on the keyboard to exit the virtual disk BIOS configuration, and then select **Ok** to confirm in the window.

Figure 17. Exit BIOS Configuration: CMC BIOS



The BIOS configuration utility screen is now closed.

9. Press **Ctrl+Alt+Delete** from the keyboard to reboot.

4.3 Install CentOS 7.1 on Cray Management Controller (CMC)

Prerequisites

- Current ISO image for the CentOS
- Assumes a first time install of Cray System Management Software (CSMS)

The following information, available from the [Cray System Management Software \(CSMS\) Configuration Worksheet](#), is required to complete this procedure:

- Management controller hostname
- Management controller IP address
- Netmask
- Gateway IP address

About this task

CentOS is the base operating system for the CSMS. CentOS must be installed on the Cray Management Controller (CMC) before installing the CSMS.

IMPORTANT: Configuration settings set during initial install are for bootstrapping purposes only. To persist, continue to capture these settings during the configuration process.

Procedure

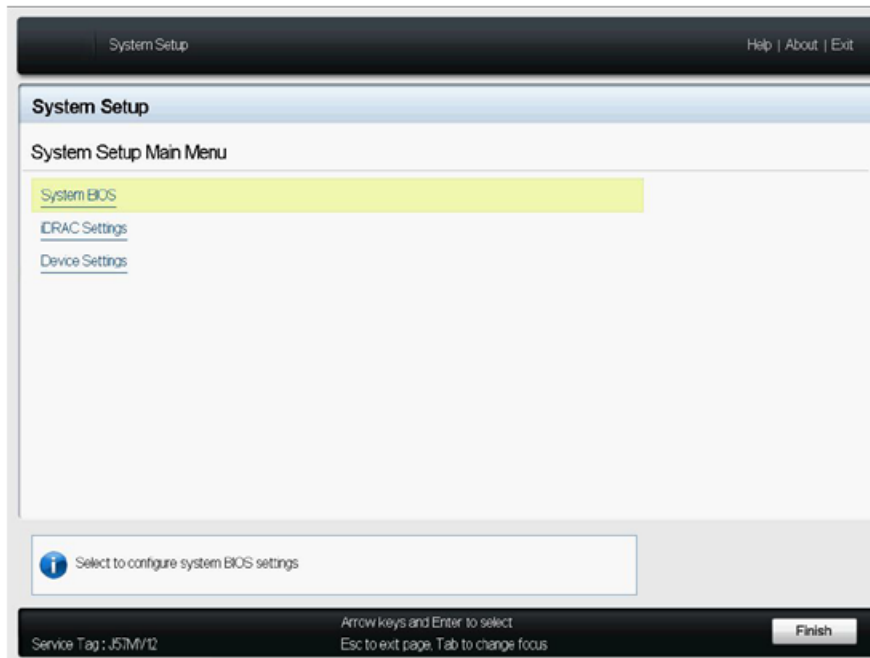
1. Gain access to the console of the machine being installed by either physically connecting a keyboard, mouse, and monitor to the system; or via the BMC. A serial connection over LAN is not sufficient.
2. Insert the Cray Bootable CentOS installation disk into the DVD drive.
3. Power up the management controller. If the machine is already powered on, reboot it.
The BIOS power-on self-test (POST) process begins.
4. On reboot, press **F2** on the keyboard to enter BIOS settings.

A screenshot of a BIOS System Setup screen. The text is displayed in a monospaced font on a black background. It shows system information including processor details, memory, and BIOS version. Navigation options F10, F11, and F12 are listed at the top right.

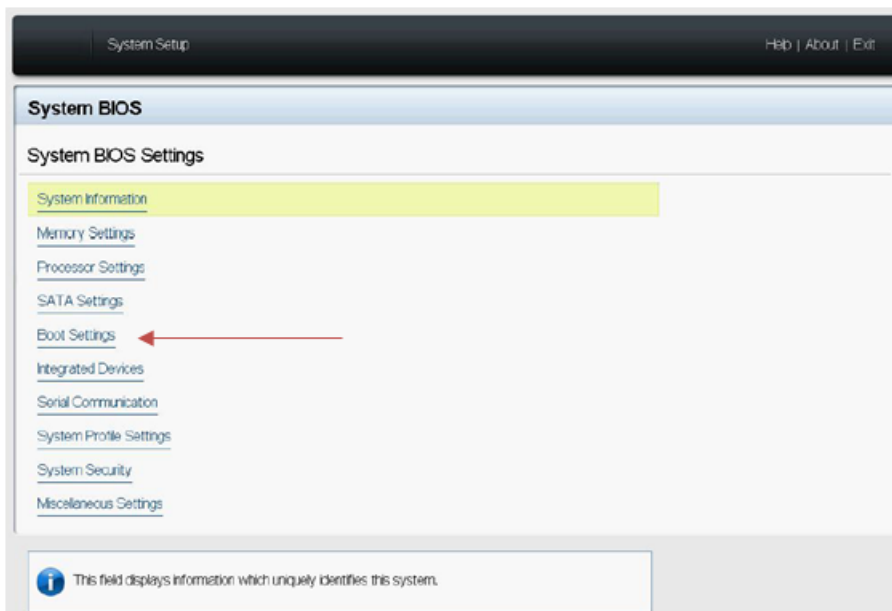
```
                Entering System Setup
                F10 = Lifecycle Controller
                F11 = BIOS Boot Manager
                F12 = PXE Boot
Two 2.60 GHz Eight-core Processors, Bus Speed:8.00 GT/s, L2/L3 Cache:2 MB/20 MB
System running at 2.60 GHz
System Memory Size: 64.0 GB, System Memory Speed: 1600 MHz, Voltage: 1.35V
Dell Serial ATA AHCI BIOS Version 1.0.2
Copyright (c) 1988-2014 Dell Inc.
```

The **System Setup** UI window should open. In some cases, a text only version of the BIOS utility based on console settings, may be implemented. The options for each method are the same.

5. Select **System BIOS** from the **System Setup Main Menu**.

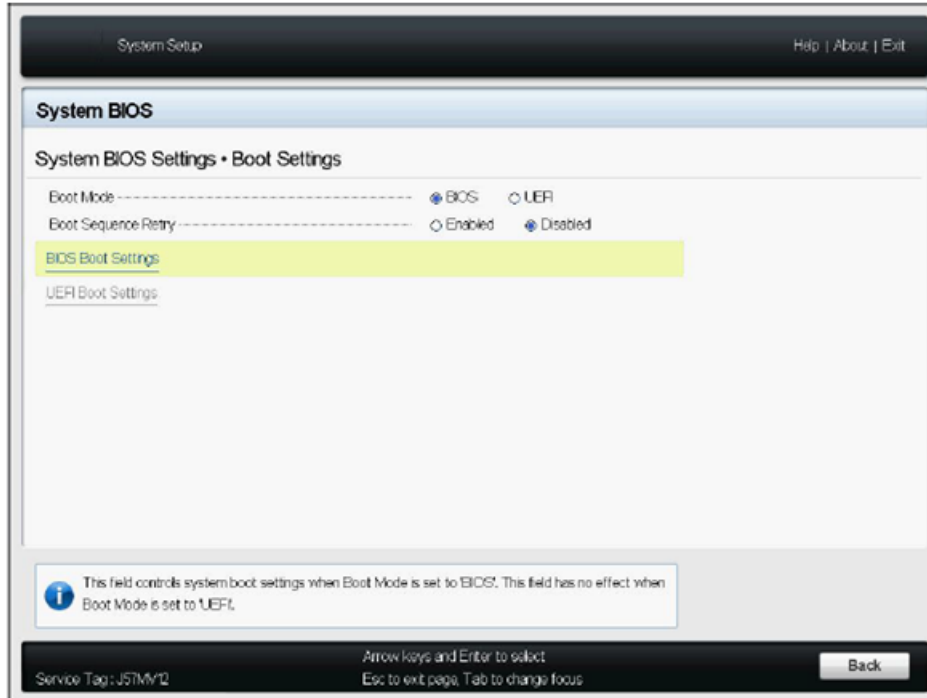
Figure 18. System Setup Main Menu: Install CentOS 7.1

6. Change the boot settings to allow the node to boot from the CentOS DVD.
 - a. Select **Boot Settings** from the **System BIOS** menu.

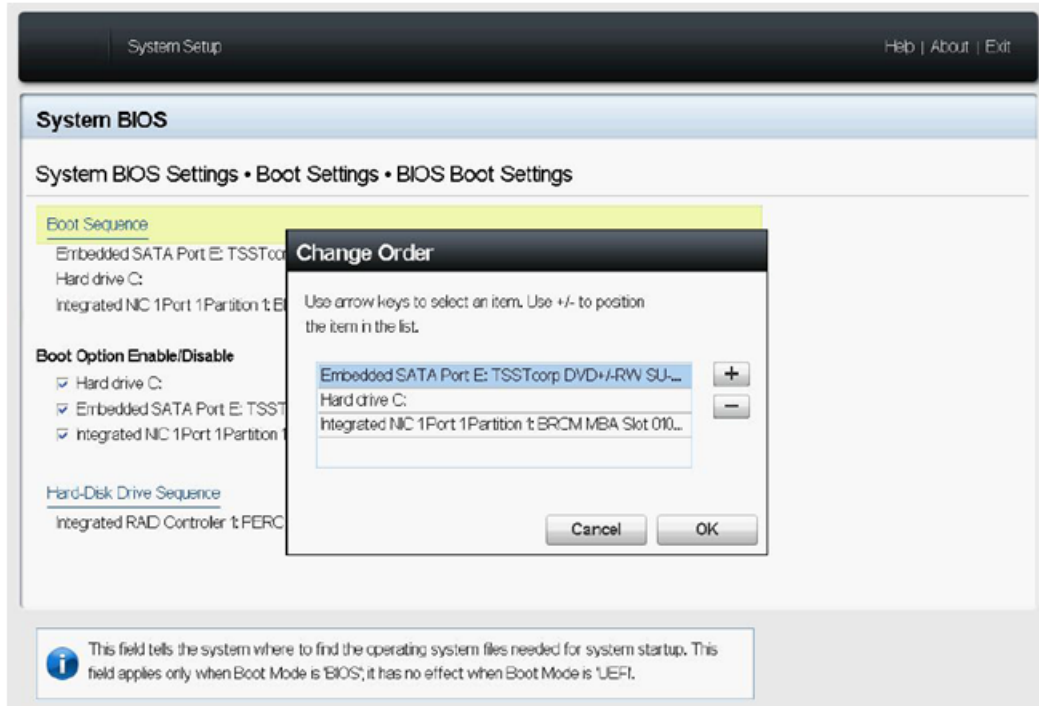
Figure 19. System BIOS Menu: Install CentOS 7.1

- b. In the **Boot Settings** menu, verify that **Boot Mode** is set to **BIOS**, and **Boot Sequence Retry** is **Disabled**.

Figure 20. System BIOS Boot Settings: Install CentOS 7.1



- c. Select **BIOS Boot Settings** from the menu.
- d. In the **BIOS Boot Settings** window, select **Boot Sequence**.
- e. In the **Change Order** pop-up window, ensure that the boot-order sequence list (top to bottom) is: **DVD** drive, followed by **Hard drive C:**, and then optionally **Integrated NIC**.

Figure 21. System BIOS Boot Sequence Order: Install CentOS 7.1

f. Click **Ok** in the **Change Order** window.

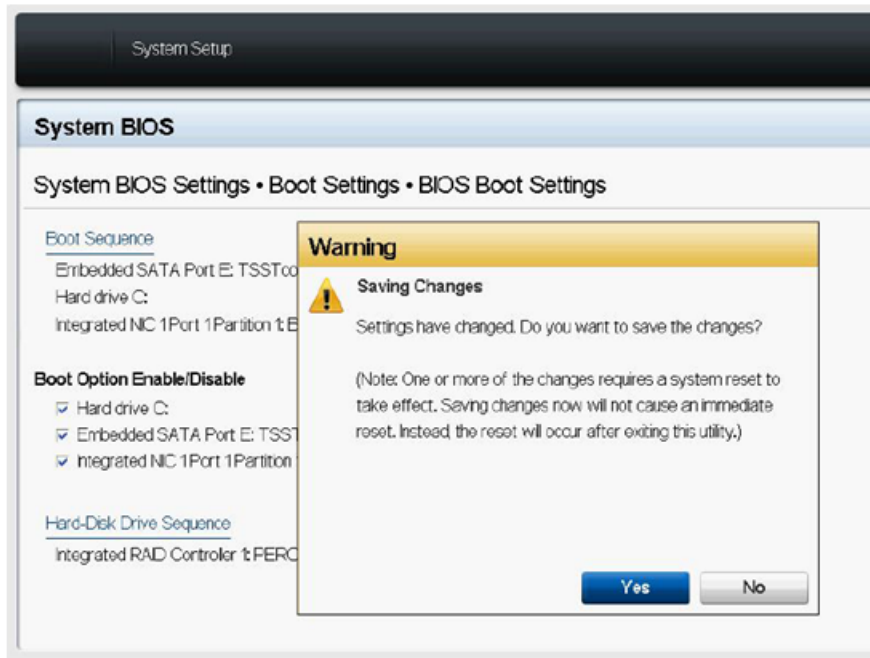
At this point, you may choose to further configure the front panel text and the BMC.

7. Insert the CentOS install DVD in the DVD drive for the node.

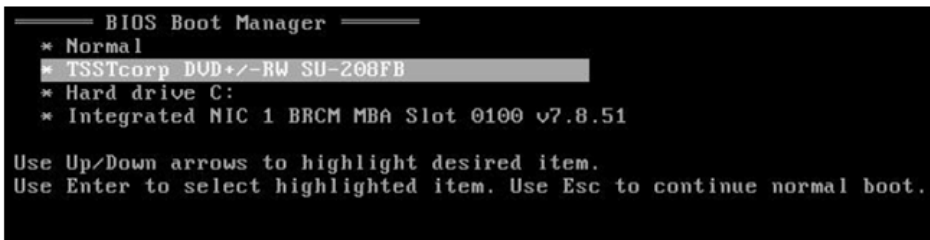
8. Restart the CMC node.

- a. Press the **Esc** key on the keyboard to open the **Saving Changes** pop-up window.
- b. Click **Yes** to save changes.

Figure 22. System BIOS Boot Settings, Save Changes: Install CentOS 7.1



- c. If the CMC displays a boot manager menu, select the DVD as the boot source.



The CMC should now boot from the DVD, and display a **CentOS 7** boot menu.

9. Select **Install CentOS 7 for Cray CMC** option, from the **CentOS 7** boot menu window.

Figure 23. CentOS-7 Boot Menu



The installer should start up and enter the main install screen.

10. Configure networking

- a. Select **NETWORK & HOSTNAME**.
- b. Select the interface that is connected to the site network from the list of interfaces on the left side of the screen, and then click on the **Configure** button.
This opens an editing window.
- c. Select the **General** tab and do the following:
 1. Ensure that **Automatically connect to this network when it is available** is selected.
 2. Ensure that **All users may connect to this network** is selected.
- d. Select the **IPv4 Settings** tab and do the following:
 1. Set **Method** to **Manual** using the pull down menu.
 2. Click the **add** button to add addresses.
 3. Enter the site network address and press **Enter**.
 4. Enter the site netmask and press enter **Enter**.
 5. Enter the site gateway and press enter **Enter**.
 6. Click within the **DNS servers** field and enter a comma-separated list of DNS servers.
 7. Click within the **Search Domains** field and enter the domain.
 8. Click **Save**.
- e. Click within the **Hostname** field and enter the hostname.

- f. Switch the **On/Off** toggle to **On** in the upper right corner to enable the network. Make sure that the network is enabled. If desired, ping the machine from another host to verify that networking is correctly configured.
- g. Click **Done**.

11. Select DATE & TIME.

- a. Click the appropriate map location to set the time zone, and verify that it is correct.
- b. Switch the **Network Time** toggle to **On** to enable NTP.
- c. Click **Done**.

12. Select Begin Installation.

The system finishes the install and reboots.

4.4 Install Cray System Management Software (CSMS)

Prerequisites

- This procedure requires a successful installation of CentOS
- This procedure assumes that the user has network access to the management server via the site network.
- This procedure assumes a first time install of Cray System Management Software (CSMS). If performing an update, please see the upgrade section.

About this task

After the OS has been installed, the CSMS can be installed on the management controller, as described in this procedure.

Procedure

1. SSH to the management server as the root user, entering `initial0` as the password. For this procedure it is assumed that the management server's IP address is `192.168.1.10`. From the CSMS [Cray System Management Software \(CSMS\) Configuration Worksheet](#), substitute the site (external) IP address in place of the example IP.

```
# ssh root@192.168.1.10
Are you sure you want to continue connecting (yes/no)? yes
```

2. Create a directory titled `isos` under the `/root` directory

```
root@csms# mkdir -p /root/isos
```

3. Log out of the management server

```
root@csms# exit
```

ATTENTION: If installing CSMS for CentOS 7.2, skip to Step 5, otherwise perform step 4 if installing CSMS for CentOS 7.1

4. Copy the CSMS and Cray Bootable 7.1 CentOS ISOs into the `/root/isos/` directory from the management server or another network accessible client containing the CSMS and Cray Bootable CentOS ISOs. Enter `yes` when the system displays the message: Are you sure you want to continue connecting (yes/no)? and enter `initial0` when prompted for a password.

```
root@smw# scp csms_centos71-1.1.2-201606240115.iso root@192.168.1.10:/root/isos
Are you sure you want to continue connecting (yes/no)? yes
password:
root@smw# scp CentOSbase7.1-201509231647.iso root@192.168.1.10:/root/isos
password:
```

- a. SSH back to the management node as root
- b. Mount and invoke the CSMS installer.

```
root@csms# mount /root/isos/csms_centos71-1.1.2-201606240115.iso /mnt
root@csms# cd /mnt
root@csms# ./install.py
[installer runs/completes]
```

- c. Unmount the ISO.

```
root@csms# cd /root
root@csms# umount /mnt
```

- d. Proceed to [CSMS Installation and Configuration Process Using Ansible](#).

5. Copy the CSMS and Cray Bootable 7.2 CentOS ISOs into the `/root/isos/` directory from the management server or another network accessible client containing the CSMS and Cray Bootable CentOS ISOs. Enter `yes` when the system displays the message: Are you sure you want to continue connecting (yes/no)? and enter `initial0` when prompted for a password.

```
root@smw# scp csms_centos72-1.1.2-201606240115.iso root@192.168.1.10:/root/isos
Are you sure you want to continue connecting (yes/no)? yes
password:
root@smw# scp Cray-CentOSbase7-1511-201605031030.iso root@192.168.1.10:/root/
isos
password:
```

- a. SSH back to the management node as root.



WARNING: For this release of CSMS, the ISO installer expects a CentOS ISO named `Cray-CentOSbase7-1511-201604201604.iso`. This will cause errors when using the bootable CentOS installer `csms_centos72-1.1.2-201606240115.iso` or newer. To work around execute the following:

```
root@csms# cd /root/isos
root@csms# ln -s Cray-CentOSbase7-1511-201605031030.iso \
Cray-CentOSbase7-1511-201604201604.iso
```

- b. Mount and invoke the CSMS installer.

```
root@csms# mount /root/isos/csms_centos72-1.1.2-201606240115.iso /mnt
root@csms# cd /mnt
```

```
root@csms# ./install.py  
[installer runs/completes]
```

- c. Unmount the ISO.

```
root@csms# cd /root  
root@csms# umount /mnt
```

4.5 Install eLogin Software on CMC

Prerequisites

Installation of the Cray System Management Software (CSMS) on the Cray Management Controller (CMC).

Procedure

1. Copy the eLogin ISO onto the management controller `/root/isos/` directory.

```
cmc# scp <location of elogin iso>elogin-6.0.96-201605131428.iso /root/isos/
```

2. Change directory to the ISO root and run the eLogin installation script. (*Estimated time: 1 minute*)

```
cmc# mount /root/isos/elogin-6.0.96-201605131428.iso /mnt  
cmc# cd /mnt  
cmc# ./install.py  
[installer runs]
```

3. Unmount the ISO root.

```
cmc# cd /root  
cmc# umount /mnt
```

4.6 Configure CSMS On Management Controller

Prerequisites

Successful installations of:

- Cray System Management Software (CSMS) on the Cray Management Controller (CMC) for eLogin
- Cray eLogin customizations

About this task

Procedure

1. Replace the hostname in `/etc/opt/cray/openstack/ansible/hosts/hosts`. The top of this file contains two occurrences of `localhost`, which must be replaced with the actual hostname of the

management controller. Either edit the file, or replace `example-csms` with the actual CMC hostname in the following example:

```
cmc# perl -p -i -e 's/^localhost/example-csms/g' \
/etc/opt/cray/openstack/ansible/hosts/hosts
```

- a. Confirm that the `/etc/opt/cray/openstack/ansible/hosts/hosts` file contains the proper CMC hostname.

After replacing the CMC hostname, you are ready to configure the CSMS.

2. Configure the CSMS by modifying the eLogin site overrides file, located at: `/etc/opt/cray/openstack/ansible/config/site/elogin-site-overrides.yaml`.

There is also a product overrides file

in: `/etc/opt/cray/openstack/ansible/config/product/elogin-overrides.yaml`. This file describes the system defaults. The product overrides file is controlled by the eLogin release and subject to change per revision. The eLogin overrides file is provided as a reference only to the default configuration. All changes must be made to the `elogin-site-overrides` file. The site overrides file takes priority over the product file. In the case of any duplicated items, the value set in the site file prevails.

- a. Edit the file `/etc/opt/cray/openstack/ansible/config/site/elogin-site-overrides.yaml`, with the site-appropriate values. If the default behavior is desired instead of the site customization, delete the line or comment it out.

The key values to set (edit) for the eLogin site overrides file, are:

domain

(Required) This sets the search domain for `resolv.conf`.

management_ip

Override the IP address of the management controller on the Openstack management network. (For most sites, revert to the default.)

management_subnet

Override the subnet of the management controller on the Openstack management network. The management IP must be contained within the management subnet. (For most sites, revert to the default.)

ipmi_ip

Override the IP address of the management controller on the IPMI network. (Current status: *Comment Out* all IPMI values.)

ipmi_subnet

Override the subnet of the management controller on the IPMI network. The management IP must be contained within the IPMI subnet. (Current status: *Comment Out* all IPMI values.)

site_ip

(Required) The address of the CSMS on the site administrative network.

site_subnet

(Required) The network for the site administrative network. The IP address of the CSMS must be contained within the site subnet. Typically this is set to the `site-ip`, with the final octet set to 0.

default_gateway

(Required) The IP address of the gateway for the site-administrative network. If connected directly to the SMW, set this to the IP address of the SMW on the interface connected to the CMC.

dns1_server_ip | dns2_server_ip

Set both of these to the DNS servers for the site management network. If the site administrative network does not have a DNS server, use the default.

physical_networks

Physical networks represent distinct networks in the system not directly connected. Cray recommends using a dictionary data structure (`dict`), with keys being the name of the physical networks to use for convenience. (For most sites, revert to the default.) The values may contain the following fields:

- **inventory_regex:** (Required) A regular expression used to match against MAC address column names in the inventory file.
- **physical_network:** (Required) The name of the physical network used in Neutron as the provider. The `physical_network` attribute of networks on this physical network, are as follows:

```
physical_networks:
  site:
    inventory_regex: site
    physical_network: site_network
  management:
    inventory_regex: management
    physical_network: mgmt_network
  ipmi:
    inventory_regex: ipmi
    physical_network: ipmi_network
```

subnets

(Required) This sets the subnet mask and gateway for your site network, as well as when the management network and IPMI networks are customized (example, changing the management network from the recommended default of 10.142.0.0/16). Ensure your site subnet prefix (example, bitmask) is correct for your network. IP addresses and ranges must exist within the defined subnet for that network.

```
subnets:
- name: site
  physical_network: "{{ physical_networks.site }}"
  address: "{{ site_subnet }}"
  prefix: 24
  gateway: 111.222.333.1

- name: management
  physical_network: "{{ physical_networks.management }}"
  address: "{{ management_subnet }}"
  prefix: 16
  # Ensure that this is different to management_ip as a neutron
  router on
  # this network will take this IP.
  gateway: 10.142.0.2
  allocation_pool_start: 10.142.0.99
  allocation_pool_end: 10.142.0.200

- name: ipmi
  physical_network: "{{ physical_networks.ipmi }}"
```



```

address: "{{ ipmi_subnet }}"
prefix: 16
# Ensure that this is different to management_ip as a neutron
router on
# this network will take this IP.
gateway: 10.148.0.2
allocation_pool_start: 10.148.0.99
allocation_pool_end: 10.148.0.200

```

networks

(Required) This section maps the subnet settings (above) to the physical network devices on the CMC node. Your network devices may be different than: em1, em2, and em3. Verify the network connections for each device before setting.

```

networks:
  # Site network
  - device: em1
    boot_protocol: none
    ip_address: "{{ site_ip }}"
    subnet: "{{ subnets[0] }}"

  # Management network
  - device: em2
    boot_protocol: none
    ip_address: "{{ management_ip }}"
    subnet: "{{ subnets[1] }}"

  # IPMI network
  - device: em3
    boot_protocol: none
    ip_address: "{{ ipmi_ip }}"
    subnet: "{{ subnets[2] }}"

```

3. Clear the Swift disk of any formatting. Use the `dd` command to write over the first part of the device.



CAUTION: The `dd` command is destructive. Ensure that you have the right device before clearing.

```
cmc# dd if=/dev/zero of=/dev/sdb bs=1M count=100
```

4. Setup the vault password.

Cray highly recommends to set the vault password interactively (secure), or if you prefer, have it read from a file (unsecure). Perform either step A or B to set the vault password.

- a. Set the vault password interactively by running the `csms_gen_creds.py` script with no arguments:

```

cmc# cd /etc/opt/cray/openstack/ansible
cmc# cp vars/credentials.yaml .
cmc# ./csms_gen_creds.py

```

Or

- b. (Option): Store the password in a cleartext file for convenience. If you prefer to keep the password in a file (unsecure), run the `csms_gen_creds.py` script with arguments:

```

cmc# cd /etc/opt/cray/openstack/ansible
cmc# cp vars/credentials.yaml .
cmc# echo initial0 > /etc/opt/cray/openstack/ansible/vault-password.txt

```

```
cmc# ./csms_gen_creds.py --write-vault-password --vault-password-file \
/etc/opt/cray/openstack/ansible/vault-password.txt
```

NOTE: Most sites only want to store the cleartext password to file in a test environment, if at all. Be aware of your site security requirements when choosing to store the password.

5. Edit the `eloin-overrides` file to enable the `haproxy` and `ssl` settings.

```
cmc# vi /etc/opt/cray/openstack/ansible/config/product/eloin-overrides.yaml
```

Change these two settings from `true` to `false`:

- `disable_haproxy: false`
- `disable_ssl: false`

6. Edit the firewall Ansible play to change the number of networks (because eLogin requires three networks instead of the default two).

```
cmc# vi /etc/opt/cray/openstack/ansible/firewall.yaml
```

In the Ansible play, change these two settings from 2 to 3:

- `Configure default IPv4 OpenStack firewall rules for high speed network\`
`(networks|length > 3`
- `Disable default IPv4 OpenStack firewall rules for high speed network\`
`(networks|length > 3`

7. Apply the CSMS configuration. (*Estimated time ~20 minutes*)

An administrator password is required as part of the CSMS configuration. To avoid supplying the password to the shell's command history, `csms_install.sh` uses the `getpass.sh` utility, provided in `/etc/opt/cray/openstack/ansible`.



WARNING: Run `csms_install.sh` from the console command prompt, not the graphical terminal. If the network changes substantially, an interrupt to the SSH connection may occur that would abort the output indicating progress, and may also terminate the session running the script.

From the console command prompt, do the following:

- a. Install and configure Cray OpenStack

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_install.sh
Enter admin password:
[...]
Vault password:
```

The password is created when the OpenStack `admin` user is created; the password is set when you run `csms_install.sh`.

- b. Pull specific OpenStack environment variables into your shell for future commands and playbooks to function correctly. If requested, use the administration password that was used when applying the configuration in (6a).

```
cmc# . /root/admin.openrc
Enter OpenStack Password:
cmc#
```

Failure messages indicated in the logs may be ignored unless ansible stops with a failure. After completion, ansible gives an count of plays. A return of `failed=0` indicates that ansible was successful. A log of the installation is saved in `/var/log/ansible.log`.

8. Apply Openstack Horizon branding changes.

Run the Ansible playbook commands to apply Cray branding changes to the Openstack Horizon web portal. Select one of two password options:

- a. To be asked for password, run the following:

```
cmc# ansible-playbook -i /etc/opt/cray/openstack/ansible/hosts\
--ask-vault-pass/etc/opt/cray/openstack/ansible/horizon-branding.yaml
```

Or

- b. To read the password from file (if the vault password was stored in a cleartext file), run the following:

```
cmc# ansible-playbook -i /etc/opt/cray/openstack/ansible/hosts/\
--vault-password-file=/etc/opt/cray/openstack/ansible/vault-password.txt\
/etc/opt/cray/openstack/ansible/horizon-branding.yaml
```

9. Generate root `ssh` keys.

Run `ssh-keygen` at the command line:

```
cmc# ssh-keygen
```

Cray recommends as a default, to generate an `ssh` key with no passphrase. Your security requirements may differ. When generating keys, enter <CR> for the defaults.

10. Configure the Kibana Index

After the system configuration has been applied, manually configure the index pattern in the Kibana user interface (UI).

NOTE: The index name in an ElasticSearch database is conceptually similar to an SQL database name.

- a. Open a web browser and navigate to the management controller using the hostname or IP address, such as `http://example-csms`.
- b. Log into the Horizon Dashboard with the username `admin`, and the Vault password created previously in the Setup Vault Password step.
- c. Select **Monitoring** and then **Overview** using the panel on the left-hand side.
- d. Select the **Log Management** button in the **Monitoring** pane.
- e. Change the provided default of `logstash-*` to `csms-logs_*`, and leave other fields at their default values.
- f. Click the **Create** button (when presented by Kibana UI) to save the updated index pattern. The UI then presents a web page, listing the fields in the `csms-logs_*` index.
- g. Select **Discover** in the UI to access a real-time summary and chart of streaming `csms-logs_*` data.

4.7 Install SMW and Configure the CMC Connection

Prerequisites

- Install of CentOS 7.1 on CMC.
- Install CSMS for eLogin
- Configuration of CSMS on CMC
- Install of eLogin ISO

About this task

This procedure installs the System Management Workstation (SMW) and configures the connection to the Cray Management Controller (CMC) for eLogin.

Procedure

Deployment Setup

1. Run the deployment setup playbook.

This playbook sets up the OpenStack environment to allow instance creation by registering required dependencies. These include:

- Removing quotas to avoid hitting resource limits
- Creating a network and subnet in Neutron for the management network
- Adding a SSH pair to Nova

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_deploy_setup.sh
```

2. Run any eLogin specific Ansible playbooks.

These plays configure log rotation and the NFS exports required for PE synchronization.

```
cmc# cd /etc/ansible
cmc# ansible-playbook elogin*.yaml
```

SMW Installation and Setup

3. Follow the usual procedures for an SMW and CLE installation if they are not already installed.

The default eLogin image recipe, `elogin_cle_6.0up01_sles_12_x86-64_ari`, is installed as part of CLE. If the image has not been generated, do so now.

4. Log onto the SMW and add the site administration network address of the management controller as an entry to `/etc/hosts`. The IP is the address of the CMC as seen from the SMW. (This should be the site IP address given for the CMC on the site administrative network.)

```
cmc# ssh example-smw
smw# echo "IP_address example-smw" >> /etc/hosts
```

IMPORTANT: The SMW and management controller must be able to communicate. If the ping fails, address the issue before continuing.

5. Test the connectivity between the SMW and management controller.

```
smw# ping -c10 <cmc>
```

IMPORTANT: The SMW and management controller must be able to communicate. If the ping fails, address the issue before continuing.

6. Add the management controller SSH key to the `~/.ssh/known_hosts` file on the SMW.

```
smw# ssh-keyscan -H <cmc> >> ~/.ssh/known_hosts
```

7. Set up SSH keys between the SMW and management controller.

- a. Generate a SSH key pair if one does not already exist.

```
smw# ssh-keygen
```

- b. Add the key pair to the `.ssh/authorized_keys` file on the management controller.

```
smw# ssh-copy-id <cmc>
```

Setup IMPS for Keystone and Glance Integration

8. Copy the `admin.openrc` file from the CMC to the SMW. This file contains the required connection settings that allow secure communication between the SMW and the CSMS.

```
smw# scp <cmc>:admin.openrc /root/
```

9. If SSL connections to Openstack are enabled, **copy** any `OS_CACERT` certificate file (`/etc/ssl/public_api.cert`), and reconfigure the `admin.openrc` file on the SMW. SSL is enabled by default.

- a. Copy the `OS_CACERT` file to the SMW.

```
smw# scp <cmc>:/etc/ssl/public_api.cert /root/
```

- b. Modify the SMW `admin.openrc` file to reflect the location of the newly copied `OS_CACERT` file.

The `export OS_CACERT=/etc/ssl/public_api.cert` field should have a new path of `/root/public_api.cert` if following the examples verbatim.

10. Source the environment variables before creating the Keystone connection information. This allows the interface message processors (IMPS) to read and query information regarding Glance service endpoints.

```
smw# . /root/admin.openrc
```

11. Create the Keystone and Glance endpoint connection.

Creation of the Keystone and Glance service endpoints simultaneously in IMPS requires that the Keystone server is operational and has the associated Glance service registered. If the Glance service is not operational, create the Keystone and Glance service associations using the `glancecon` and `keystonecon` command-line interfaces separately. This may allow for a more asynchronous setup.

```
smw# keystonecon create <cmc> --env
```

12. Verify the connection registration.

```
cmc# keystonecon list
NAME          URL                                     OS_USERNAME
example-cmc   https://172.30.12.47:5000/v2.0      admin
smw# /var/opt/cray/imps/config/sets/global/config # glancecon list
NAME          URL                                     KEYSTONE_CONN
example-cmc   https://172.30.12.47:9292          example-cmc
```

4.8 eLogin Node Installation

Installation of an eLogin node requires that both the SMW and management controller are successfully installed. Additionally, these eLogin hardware details must be known:

- Amount of memory
- Size of hard drive

The procedures for the eLogin node installation process are:

- eLogin hardware and BIOS RAID setup
- Create a minimum eLogin config set
- Ironic node enrollment
- Configure Fuel for eLogin

For installing eLogin software on an eLogin node, refer to [Configure and Manage an eLogin Image](#).

4.8.1 eLogin Hardware and BIOS RAID Setup

Prerequisites

Successful install and configuration of the SMW and Cray Management Controller (CMC).

About this task

The eLogin hardware must be configured properly before the eLogin image is deployed to it.

The eLogin software requires two drives:

- SDA: The operating system and swap
- SDB: Persistent storage

Procedure

1. Boot the eLogin hardware.
2. Press **Ctrl-R** on the keyboard to enter the BIOS RAID configuration boot menu.

Figure 24. 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 25. RAID Configuration Screen: eLogin

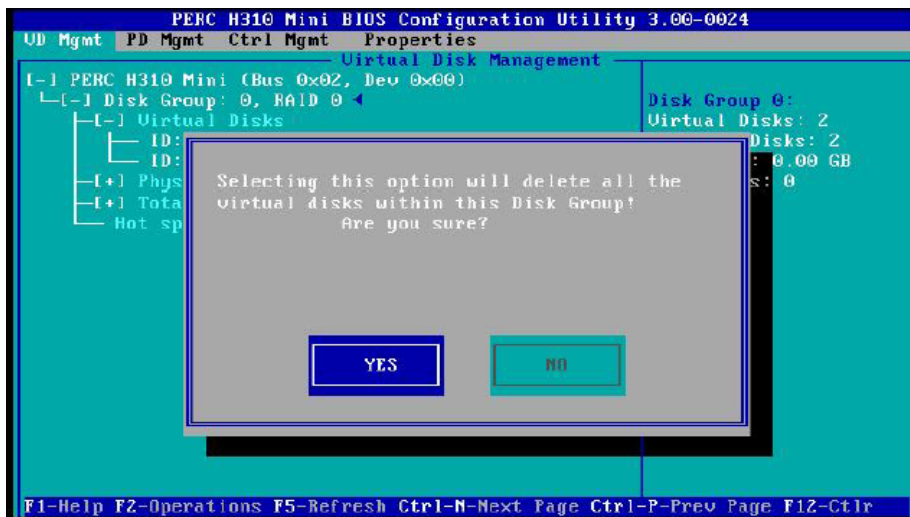


3. (Conditional): Delete any virtual disks (if present) that do not meet the required disk configuration. Otherwise, skip this step.
 - a. Select the disk.
 - b. Press **F2** key to get a list of operations.
 - c. Select **Delete Disk Group** and press **Enter**.

Figure 26. Delete Disk Group: eLogin BIOS RAID Setup

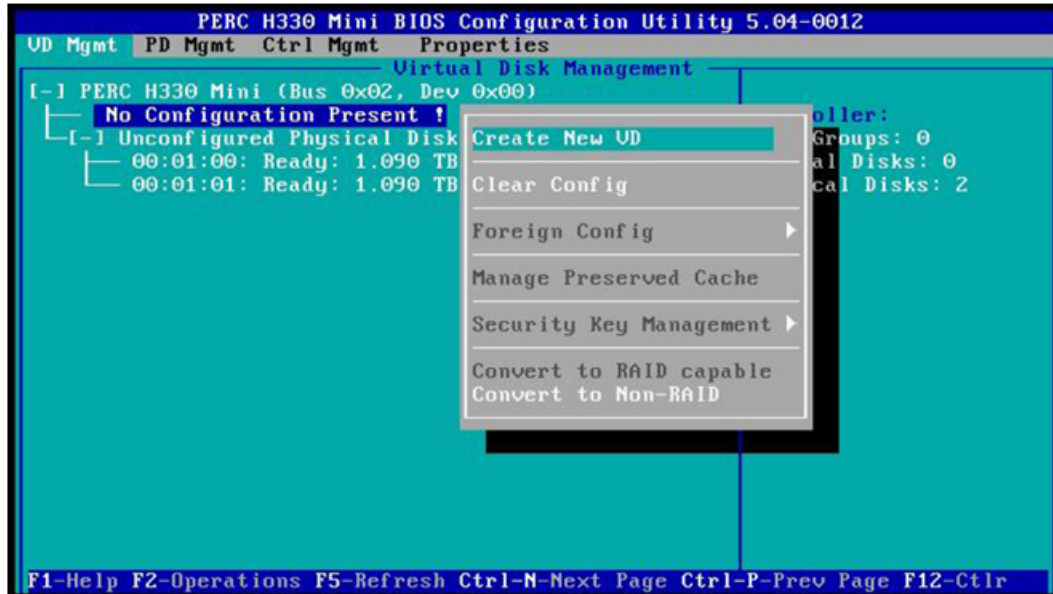


- d. Confirm the selection **Yes**, and press return.



4. Create a new Virtual Disk A.
 - a. In the virtual disk management window, 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 27. Create Virtual Disk A: eLogin BIOS RAID



The Create New Virtual Disk (VD) UI window opens.

5. In the create new VD window, move the cursor to select the disk ID, and then press spacebar on keyboard to add disk to RAID.
6. Set the RAID Level to **RAID 0**.

Figure 28. Add Disk to RAID Level 0: eLogin



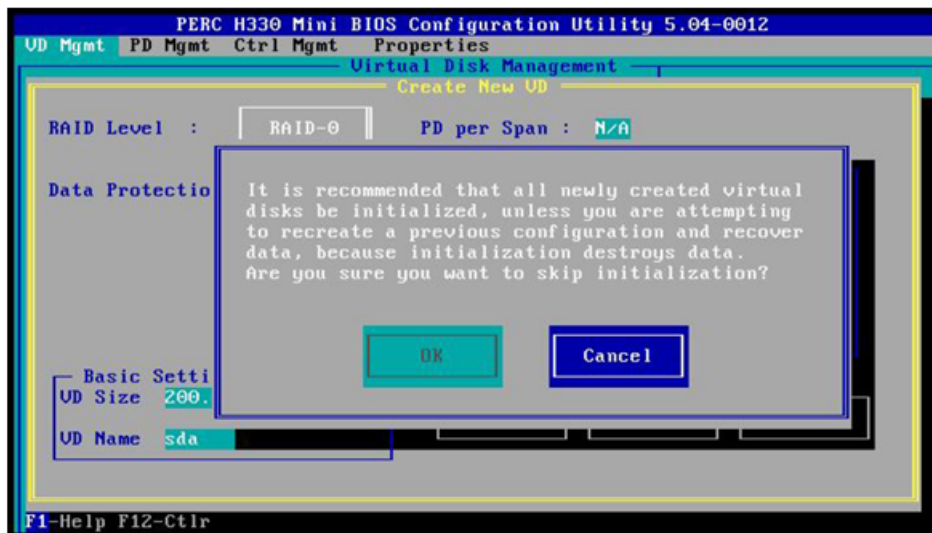
7. Set **VD Size** and **VD Name** for virtual disk A.

- a. Set the **VD Size** for virtual disk A to **200 GB** of disk space.
- b. Set the **VD Name** to **SDA**.

Figure 29. 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**.



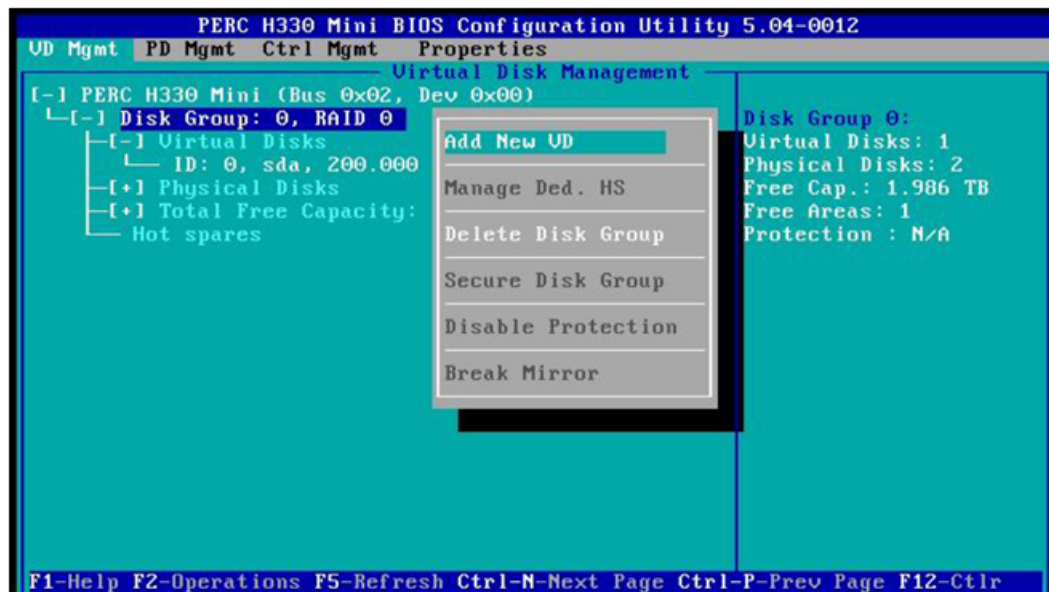
The SDA is now created.



8. 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**.

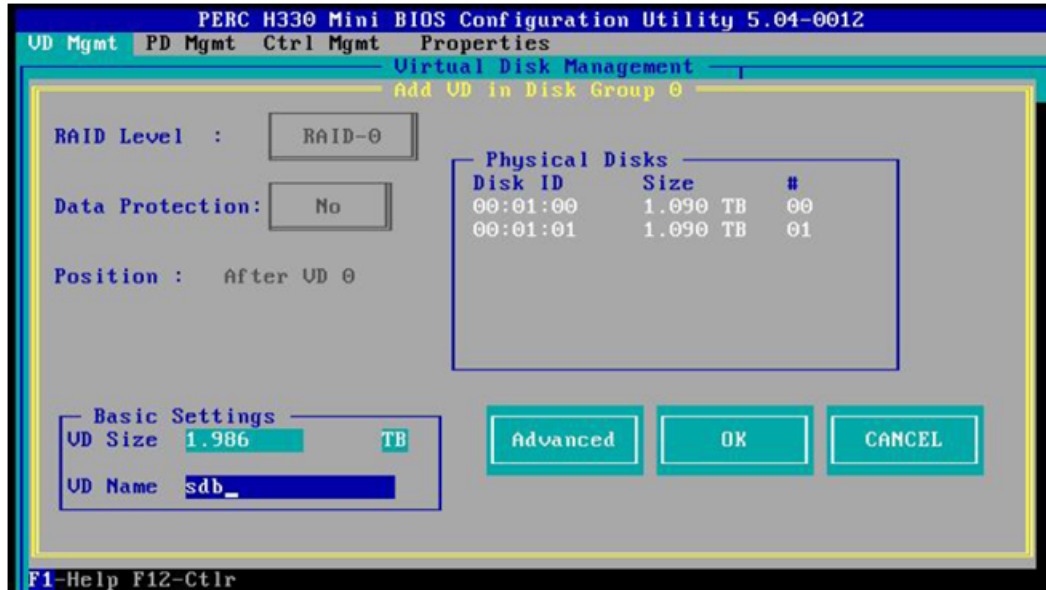
Figure 30. Create New Virtual Disk B: eLogin BIOS RAID



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

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

Figure 31. Disk Size and Name Setting for Virtual Disk B: eLogin

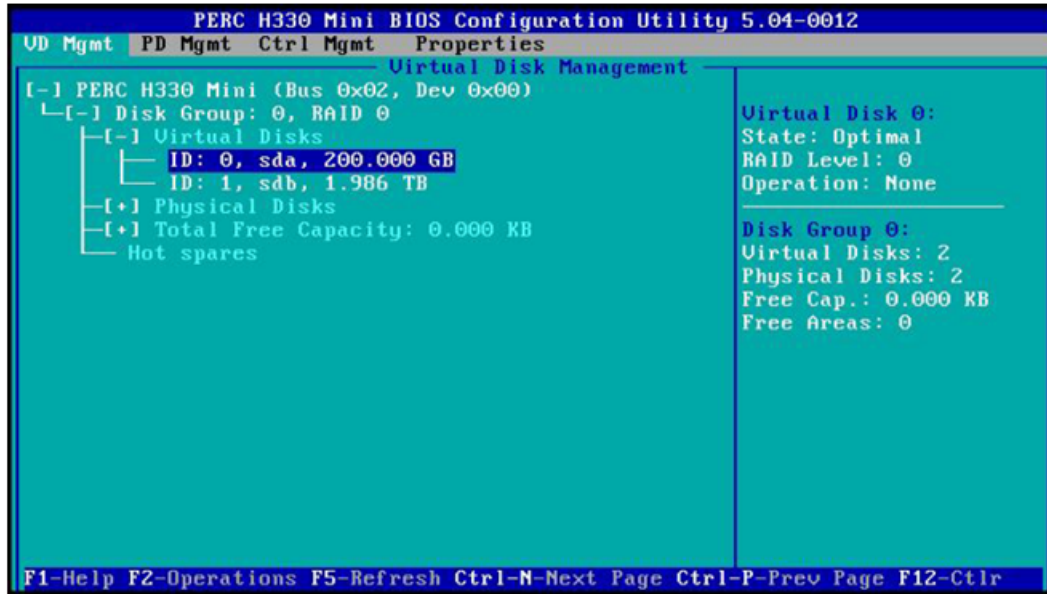


- e. Select **OK** in the window, and then in the initialization message pop-up window, select **Ok**.



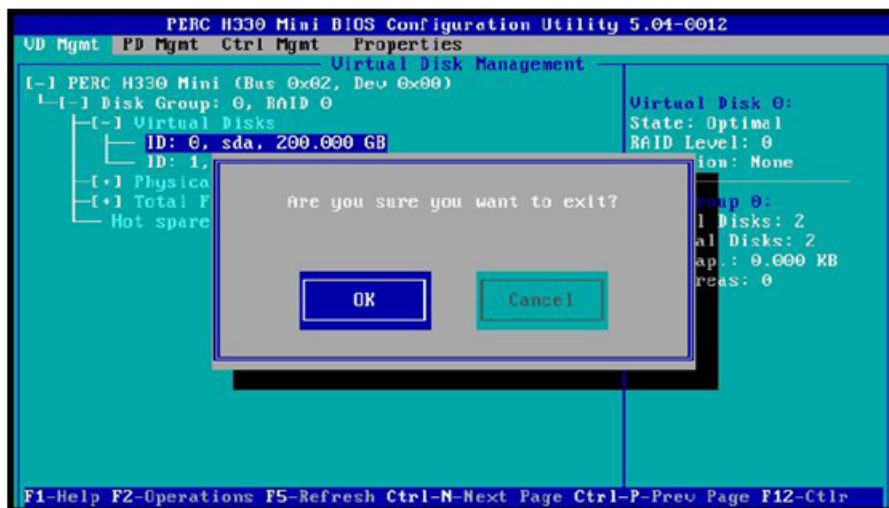
Two virtual disks are now available.

Figure 32. Two Virtual Disks Available: eLogin BIOS RAID



9. Press **Esc** on the keyboard to exit the BIOS configuration, and then select **Ok** to confirm in the window.

Figure 33. Exit BIOS Configuration: eLogin



The BIOS configuration utility screen is now closed.

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

4.8.2 Create a Minimum eLogin Config Set

Prerequisites

Successful install and configuration of the SMW and Cray Management Controller (CMC).

About this task

The boot process uses standard interface message processors (IMPS) configuration templates to convey configuration information to the eLogin image. The system administrator should, therefore, expect to answer eLogin specific questions when running config set operations. The eLogin attempts, where possible, to reuse existing configuration values to avoid duplicate questions.

eLogin references the following Cray XC config set templates:

- `cray_local_users`
- `cray_time`
- `cray_user_settings`
- `cray_auth`
- `cray_ssh`
- `cray_lustre_client`
- `cray_net`
- `cray_image_layering`
- `cray_simple_sync`

Procedure

1. Configure `cray_net` within the config set. Cray networking sets up network connections for eLogin nodes to the management controller, the site network, and LNet.

```
smw# cfgset update -s cray_net -S all -m interactive config_set

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

cray_net.settings.networks
[<cr>=set 6 entries, +=add an entry, ?=help, @=less] $ +

cray_net.settings.networks.data.name
[<cr>=set '', <new value>, ?=help, @=less] $ openstack_mgmt

cray_net.settings.networks.data.openstack_mgmt.description
[<cr>=set '', <new value>, ?=help, @=less] $ OpenStack Management Network

cray_net.settings.networks.data.openstack_mgmt.ipv4_network
[<cr>=set '', <new value>, ?=help, @=less] $ OpenStack_IP_network

cray_net.settings.networks.data.openstack_mgmt.ipv4_netmask
[<cr>=set '', <new value>, ?=help, @=less] $ OpenStack_IP_netmask

cray_net.settings.networks.data.openstack_mgmt.ipv4_gateway
[<cr>=set '', <new value>, ?=help, @=less] $ ellogin_IP_gateway
```



```

cray_net.settings.networks.data.openstack_mgmt.dns_servers
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks.data.openstack_mgmt.dns_search
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks.data.openstack_mgmt.ntp_servers
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks
[<cr>=set 7 entries, +=add an entry, ?=help, @=less] $ +

cray_net.settings.networks.data.name
[<cr>=set '', <new value>, ?=help, @=less] $ site

cray_net.settings.networks.data.site.description
[<cr>=keep 'eLogin site network', <new value>, ?=help, @=less] $ eLogin site network

cray_net.settings.networks.data.site.ipv4_network
[<cr>=keep '123.45.67.0', <new value>, ?=help, @=less] $ ellogin_IP_network

cray_net.settings.networks.data.site.ipv4_netmask
[<cr>=keep '234.567.890.0', <new value>, ?=help, @=less] $ ellogin_IP_netmask

cray_net.settings.networks.data.site.ipv4_gateway
[<cr>=keep '345.67.89.1', <new value>, ?=help, @=less] $ ellogin_IP_gateway

cray_net.settings.networks.data.site.dns_servers
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add dns_servers (Ctrl-d to exit) $ dns_IP_address
Add dns_servers (Ctrl-d to exit) $ dns_IP_address
Add dns_servers (Ctrl-d to exit) $ <Ctrl-d>

cray_net.settings.networks.data.site.dns_servers
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks.data.site.dns_search
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add dns_search (Ctrl-d to exit) $ dns_server_name
Add dns_search (Ctrl-d to exit) $ dns_server_name
Add dns_search (Ctrl-d to exit) $ dns_server_name
Add dns_search (Ctrl-d to exit) $ dns_server_name
Add dns_search (Ctrl-d to exit) $ <Ctrl-d>

cray_net.settings.networks.data.site.dns_search
[<cr>=set 4 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks.data.site.ntp_servers
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add ntp_servers (Ctrl-d to exit) $ ntp_server_name
Add ntp_servers (Ctrl-d to exit) $ ntp_server_name
Add ntp_servers (Ctrl-d to exit) $ <CR>

cray_net.settings.networks.data.site.ntp_servers
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.networks
[<cr>=set 8 entries, +=add an entry, ?=help, @=less] $ <CR>

```

```

cray_net.settings.hosts
[<cr>=set 6 entries, +=add an entry, ?=help, @=less] $ +

cray_net.settings.hosts.data.common_name
[<cr>=set '', <new value>, ?=help, @=less] $ ellogin_name

cray_net.settings.hosts.data.example_eLogin.description
[<cr>=set '', <new value>, ?=help, @=less] $ eLogin Node

cray_net.settings.hosts.data.example_eLogin.aliases
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.roles
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.hostid
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.host_type
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.hostname
[<cr>=set '', <new value>, ?=help, @=less] $ ellogin_hostname

cray_net.settings.hosts.data.example_eLogin.interfaces
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +

cray_net.settings.hosts.data.example_eLogin.interfaces.common_name
[<cr>=set '', <new value>, ?=help, @=less] $ eth0

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.name
[<cr>=set '', <new value>, ?=help, @=less] $ eth0

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.description
[<cr>=set '', <new value>, ?=help, @=less] $ OpenStack eth0

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.aliases
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.network
[<cr>=set '', <new value>, ?=help, @=less] $ openstack_mgmt

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.ipv4_address
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.bootproto
[<cr>=set 'static', <new value>, ?=help, @=less] $ dhcp

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.mtu
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.extra_attributes
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add extra_attributes (Ctrl-d to exit) $ DHCLIENT_SET_DEFAULT_ROUTE=no
Add extra_attributes (Ctrl-d to exit) $ <Ctrl-d>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.extra_attributes
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ +

```



```
cray_net.settings.hosts.data.example_eLogin.interfaces.common_name
[<cr>=set '', <new value>, ?=help, @=less] $ eth1

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.name
[<cr>=set '', <new value>, ?=help, @=less] $ eth1

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.description
[<cr>=set '', <new value>, ?=help, @=less] $ Site eth1

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.aliases
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.network
[<cr>=set '', <new value>, ?=help, @=less] $ site

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.ipv4_address
[<cr>=set '', <new value>, ?=help, @=less] $ eLogin_Site_IP_address

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.bootproto
[<cr>=set 'static', <new value>, ?=help, @=less] $ static

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.mtu
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.extra_attributes
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ +

cray_net.settings.hosts.data.example_eLogin.interfaces.common_name
[<cr>=set '', <new value>, ?=help, @=less] $ ib0

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.name
[<cr>=set '', <new value>, ?=help, @=less] $ ib0

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.description
[<cr>=set '', <new value>, ?=help, @=less] $ IB to External Lustre

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.aliases
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.network
[<cr>=set '', <new value>, ?=help, @=less] $ lnet

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.ipv4_address
[<cr>=set '', <new value>, ?=help, @=less] $ eLogin_LNet_address

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.bootproto
[<cr>=set 'static', <new value>, ?=help, @=less] $ static

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.mtu
[<cr>=set '', <new value>, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces.ib0.extra_attributes
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_net.settings.hosts.data.example_eLogin.interfaces
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $ <CR>
```

```
cray_net.settings.hosts
[<cr>=set 7 entries, +=add an entry, ?=help, @=less] $ <CR>
```

2. Verify the values entered in the previous steps.

```
smw# cfgset search -t openstack_mgmt -s cray_net <config_set>
cray_net.settings.networks.data.openstack_mgmt.description: OpenStack
Management Network
cray_net.settings.networks.data.openstack_mgmt.ipv4_network:
OpenStack_IP_network
cray_net.settings.networks.data.openstack_mgmt.ipv4_netmask:
OpenStack_IP_netmask
cray_net.settings.networks.data.openstack_mgmt.ipv4_gateway:
OpenStack_IP_gateway
cray_net.settings.networks.data.openstack_mgmt.dns_servers: dns_IP_address,
dns_IP_address
cray_net.settings.networks.data.openstack_mgmt.dns_search: dns_server_name
cray_net.settings.networks.data.openstack_mgmt.ntp_servers: ntp_server_name
```

```
smw# cfgset search -t site -s cray_net <config_set>
cray_net.settings.networks.data.site.description: eLogin site network
cray_net.settings.networks.data.site.ipv4_network: IP_address
cray_net.settings.networks.data.site.ipv4_netmask: IP_address
cray_net.settings.networks.data.site.ipv4_gateway: IP_address
cray_net.settings.networks.data.site.dns_servers: # (empty)
cray_net.settings.networks.data.site.dns_search: # (empty)
cray_net.settings.networks.data.site.ntp_servers: # (empty)
```

```
smw# cfgset search -t example_eLogin -s cray_net <config_set>
cray_net.settings.hosts.data.example_eLogin.description: eLogin Node
cray_net.settings.hosts.data.example_eLogin.aliases: # (empty)
cray_net.settings.hosts.data.example_eLogin.roles: # (empty)
cray_net.settings.hosts.data.example_eLogin.hostid: # (empty)
cray_net.settings.hosts.data.example_eLogin.host_type: # (empty)
cray_net.settings.hosts.data.example_eLogin.hostname: example-eLogin
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.description:
OpenStack eth0
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.aliases: # (empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.network:
openstack_mgmt
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.ipv4_address: #
(empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.bootproto: dhcp
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.mtu: # (empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth0.extra_attributes: #
(empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.name: eth1
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.description: Site
eth1
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.aliases: # (empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.network: site
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.ipv4_address:
IP_address
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.bootproto: static
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.mtu: # (empty)
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.extra_attributes: #
(empty)
```

3. Configure eLogin networking, which determines the postfix relay each eLogin node uses.

```
smw# cfgset update -s cray_elogin_networking -S all config_set

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

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

cray_elogin_networking.settings.elogin_networking.data.hostname
[<cr>=set '', <new value>, ?=help, @=less] $ elogin-hostname

cray_elogin_networking.settings.elogin_networking.data.<elogin
hostname>.postfix_relay_host
[<cr>=keep 'cims-mgmt', <new value>, ?=help, @=less] $ <CR>

cray_elogin_networking.settings.elogin_networking
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>
```

NOTE: The default `cims-mgmt` is a variable reference to the actual hostname of the management controller. Keep as is.

4. Verify that the information was entered correctly.

```
smw# -t postfix_relay_host -s cray_elogin_networking config_set
cray_elogin_networking.settings.elogin_networking.data.example-
elogin.postfix_relay_host:
cims-mgmt
```

5. Configure eLogin Lustre networking (LNet), which controls how eLogin nodes connect to the Lustre server.

```
smw# cfgset update -s cray_elogin_lnet -l basic -S all config_set

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

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

cray_elogin_lnet.settings.local_lnets.data.lnet_name
[<cr>=set '', <new value>, ?=help, @=less] $ o2ib

cray_elogin_lnet.settings.local_lnets.data.o2ib.ip_wildcard
[<cr>=set '', <new value>, ?=help, @=less] $ 10.149.*.*

cray_elogin_lnet.settings.local_lnets
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>
```

6. Configure the `eswrap` service. The `cray_eswrap` configuration template maps each eLogin node to an internal login node for running `eswrap` commands. It also defines which commands are wrapped.



WARNING: A mapping must exist for each eLogin node configured in the `cray_elogin_networking` config set; otherwise, the eLogin node will not boot.

```
smw# cfgset update -s cray_eswrap -S all config_set

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

```

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

cray_eswrap.settings.eswrap_map.data.eswrap_host
[<cr>=set 'login', <new value>, ?=help, @=less] $ login_node_hostname

cray_eswrap.settings.eswrap_map.data.<login node hostname>.ellogin_hosts
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add ellogin_hosts (Ctrl-d to exit) $ ellogin_hostname
Add ellogin_hosts (Ctrl-d to exit) $ <Ctrl-d>

cray_eswrap.settings.eswrap_map.data.<login node hostname>.ellogin_hosts
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>

cray_eswrap.settings.eswrap_map
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>

```

7. Optional: For systems without a workload manager, enable aprun wrapping.

```

smw# cfgset update -s cray_eswrap -S all -l advanced -m interactive config_set

# Select the aprun setting in the wrapped category by entering its number
Cray eswrap Service Menu [default: save & exit - Q] $ 5

Cray eswrap Service Menu [default: configure - C] $ <CR>

cray_eswrap.settings.wrapped.data.aprun
[<cr>=keep 'false', <new value>, ?=help, @=less] $ true

Cray eswrap Service Menu [default: save & exit - Q] $ <CR>

```

8. Configure any eLogin node groups, if desired. Node groups allow targeting of `simple_sync` actions to groups of nodes rather than individually.

```

smw# cfgset update -s cray_node_groups -S all config_set

# Enable cray_node_groups.
cray_node_groups.enabled
[<cr>=keep 'true', <new value>, ?=help, @=less] $ true

# Type '+' to add a new group.
cray_node_groups.settings.groups
[<cr>=set 18 entries, +=add an entry, ?=help, @=less] $ +

# Add a group for all eLogin nodes (named all_elogins in this example).
cray_node_groups.settings.groups.data.group_name
[<cr>=set '', <new value>, ?=help, @=less] $ all_elogins

# Add a description for the group.
cray_node_groups.settings.groups.data.all_elogins.description
[<cr>=set '', <new value>, ?=help, @=less] $ All eLogin nodes

# Add all the eLogin node hostnames to the list of members.
cray_node_groups.settings.groups.data.all_elogins.members
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add members (Ctrl-d to exit) $ ellogin1
Add members (Ctrl-d to exit) $ ellogin2
Add members (Ctrl-d to exit) $ ellogin3

```

```
Add members (Ctrl-d to exit) $ <Ctrl-d>

# Set the three entries by hitting <cr>.
cray_node_groups.settings.groups.data.all_elogins.members
[<cr>=set 3 entries, +=add an entry, ?=help, @=less] $

# Set all node group entries by hitting <cr>.
cray_node_groups.settings.groups
[<cr>=set 19 entries, +=add an entry, ?=help, @=less] $

# Verify the new 'all_elogins' settings.
cfgset search -s cray_node_groups --level basic <config_set> | grep all_elogins
cray_node_groups.settings.groups.data.all_elogins.description: All eLogin nodes
cray_node_groups.settings.groups.data.all_elogins.members: elogin1, elogin2,
elogin3
```

9. Confirm that the config set is valid.

```
smw# cfgset validate config_set
```

4.8.3 Ironic Node Enrollment

About this task

The `csms_ironic_enrollment.sh` script registers the bare metal nodes with Ironic, the bare metal flavor with Nova, and the deployment ramdisk and kernel with Glance. If the `inspection_enabled` variable is set in the `/etc/opt/cray/openstack/ansible/group_vars/all` configuration file, a hardware inspection process runs on the nodes in the inventory to populate their properties in Ironic.

Procedure

1. Log onto the management controller as `root`.
2. Define the bare metal nodes.

The bare metal nodes must be defined in an inventory file before enrollment.

- a. Create the file `/etc/opt/cray/openstack/ansible/inventory.csv`, and add the node(s) for registration with Ironic.

The following is an example. Each node should have a single line after the header.

IMPORTANT: The header line is required.

```
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUS, ARCH, RAM_MB, DISK_GB, NODE_DESC
example-elogin,bmc_ip,mac_addr,n_cpus,x86_64,ram_mb,disk_gb,example-elogin1
```

Where:

NODE_NAME Name of the node.

BMP_IP IP address assigned to the BMC interface. It is set in the BIOS for each eLogin node. This IP must be on the maintenance network.

MAC_ADDR	MAC address of the maintenance network interface of the CDL node. This is the device that is PXE booted. In the BIOS of the CDL, this interface must be set to start with this MAC address.
N_CPUs	Number of CPUs that Ironic should report for the node.
ARCH	The architecture that Ironic should report for the node.
RAM_MB	Amount of RAM that Ironic should report for the node.
DISK_GB	Size of the disk that Ironic should report for the node.
NODE_DESC	Description of node.

For full details of the possible options, see the *eLogin Administration Guide*.

3. Change to the `ansible` directory, and register the Ironic node.

Prior to running the script `./csms_ironic_enrollment.sh`, ensure the BMC/DRAC for each node is powered up and configured with the correct IP address, and the DRAC successfully pings the management node. When these details are confirmed, run the following at the command line to register the Ironic node:

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_ironic_enrollment.sh
```

If a failure results during the execution of the script `./csms_ironic_enrollment.sh`, the node must be deleted from Ironic prior to re-running the command.

To delete the node, run this command line:

```
cmc# ironic node-delete nodename
```

IMPORTANT: Any changes to the CDL node inventory, requires a corresponding change to `inventory.csv`, and a rerun of the node registration script.

Recovery for Password Not Accepted

During the execution of the script `./csms_ironic_enrollment.sh`, the vault password must be entered.

(Conditional) If the vault password is not accepted, perform these steps:

- a. Run this command line (enter password and confirm):

```
cmc# ansible-vault rekey ./group_vars/all/service_passwords
```

Ansible-vault will prompt for the vault password, then for the new vault password, and again to confirm the new vault password.

- b. Run this command line (enter **initial0** for current password, then new password and confirm):

```
cmc# ansible-vault rekey ./vars/credentials.yaml
```

Ansible-vault will prompt for the vault password as before, but this time, enter **initial0** instead of the current vault password setting. Then enter the new vault password and confirm.

4. Create the eLogin Nova flavor.

This describes the hardware and partitioning used when deploying the hardware. Example values are presented when executing the `nova` command. Note that these are recommended values for the end hardware. The site-specific hardware may differ. The values in the flavor specify the minimum hardware on which eLogin nodes will be deployed. For this reason, Cray recommends keeping all hardware specification

below what the actual hardware is. If the CDL node hardware is below any of the minimum specification, the node deployment will fail.

```
cmc# nova flavor-create flavor_name id ram disk vcpus --swap swap
```

Where:

flavor_name	Name of the flavor (example, <code>eloginflavor</code>)
id	Use <code>auto</code> for automatic generation
ram	Minimum amount of RAM present on the system
disk	Size of the root file system partition (<code>/dev/sda2</code>)
vcpus	Set to 16
swap	Desired swap space (<code>/dev/sda1</code>) in MB

Run the following command line using `nova flavor-create` to set the minimum eLogin hardware requirements:

```
cmc# nova flavor-create eloginflavor auto 2048 100 16 --swap 16384
```

4.8.4 Configure OpenStack Fuel

Prerequisites

Successful install and configuration of CSMS on the CMC.

About this task

The Ironic Fuel driver requires that both a cloud default configuration JSON file and a deploy configuration JSON file exist in the resource storage (Glance for eLogin).

For detailed information regarding Fuel configuration, see the *eLogin Administration Guide*.

Procedure

1. Optional: Override the `fuel_enabled` Ansible variable in the `/etc/opt/cray/openstack/ansible/group_vars/ironic` file in a site override file to make Fuel the default Ironic driver if needed.

```
# Whether the Ironic Fuel agent is enabled.
fuel_enabled: true
```

2. Create the `cloud_default_deploy_config` Glance image.

```
cmc# glance image-create --is-public True --disk-format raw \
--container-format bare --name cloud_default_deploy_config --file \
/etc/opt/cray/openstack/fuel/deploy_config/cloud_default_deploy_config.json
```

3. Create the `deploy_config_elogin` Glance image.

```
cmc# glance image-create --is-public True --disk-format raw \
--container-format bare --name deploy_config_elogin --file \
/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin.json
```

4.9 Configure and Manage an eLogin Image

Prerequisites

A complete successful CSMS/eLogin installation.

About this task

Image and config set management is the core of eLogin node management. All image management is done via IMPS on the SMW.

Append images with ‘_YYYYMMDD’. For example, if generating `elogin-large_cle_6.0up01_sles_12_x86-64_ari` on June 1st, 2015, the image should be named `elogin-large_cle_6.0up01_sles_12_x86-64_ari_20150601`. These image names match the naming scheme of the internal login image, with eLogin prepended.

SMW Image Creation and Export

Procedure

1. Connect to the SMW.

```
# ssh smw
```

2. Select an eLogin image type.

There are two types of images: regular eLogin image and eLogin large image. This mirrors the internal login structure. The eLogin large image contains an expanded set of tools. This documentation uses the eLogin large image for all examples.

Use the regular eLogin image only if there are specific size constraints for the eLogin, or if the image is only to be used for test. (In which case, the smaller image allows for shorter boot times.)

3. Optional: Create a custom eLogin image recipe.

Perform this step if either one of these conditions apply:

- Additional packages are required (example, for workload managers)
- The OpenStack network interface is not `eth0`

Create a new eLogin image recipe by cloning

`elogin-large_cle_6.0up01_sles_12_x86-64_ari_20150601`. Prepend the function of the customization to the original user name of a custom image (example, `username-function`).

```
smw# recipe create custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari
smw# recipe update -r elogin-large_cle_6.0up01_sles_12_x86-64_ari \
custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari
```


4. Build the eLogin image.

```
smw# image create -r custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari \
custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari-YYYYMMDD
```

5. Source the `admin.openrc` file to set up the authentication to Glance and eliminate multiple password prompts.

```
smw# source ~/admin.openrc
```

6. Push the eLogin image from the SMW to Glance running on the cmc.

Move the eLogin image to the CMC machine. This includes both an image format conversion to `qcow2`, and the transfer of the image to the Glance database. For a large image, the estimated time to complete is half an hour.



WARNING: Glance allows multiple images with the same name to be stored on the cmc, but it can only deploy an image with a unique name. If duplicate image names are used, Glance will not deploy to the eLogin node. To recover from this situation, remove the image from Glance using the universally unique identifier (UUID), not the name.

Ensure that the image being pushed is unique. Remove any images with used names from the CMC before pushing a new image from the SMW.

```
smw# image export --format qcow2 -d glance:csms-hostname\
:custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari_YYYYMMDD \
custom-elogin-large_cle_6.0up01_sles_12_x86-64_ari_YYYYMMDD
```

Repeat this image deploy step each time the image is modified on the SMW.

7. Push the config set to the CMC.

The config set was generated during CLE installation and modified in [Create a Minimum eLogin Config Set](#) on page 45.

```
smw# cfgset push -d csms-name global
smw# cfgset push -d csms-name config_set_name
```

The config set is cached on the CMC. This makes it possible to reprovision eLogin nodes if the SMW is not available for any reason.

Whenever the config set changes, push it to the CMC to allow the eLogin node to access the changes.

8. Push the CLE Programming Environment (PE) to the CMC.

The PE is shared between the Cray XC system and the eLogin node. The PE is built during the SMW installation and is also cached on the CMC for accessibility in the circumstance where the SMW is not available.

```
smw# image push -d csms-name pe_compute_image
```

The estimated time to complete this process is ~10 to 30 minutes, depending on: the size of the PE and the speed of the networking link between the SMW and the CMC.

Whenever the PE is modified, the built image must be pushed to the CMC in order for the updated PE to be available to the eLogin node. Only changes are pushed; subsequent pushes are likely to be faster barring large change sets.

CSMS Image Deployment

9. Connect to the CMC node.

```
# ssh cmc
```

10. Source the `admin.openrc` file. This sets up the authentication to Glance and eliminates multiple password prompts.

```
cmc# source ~/admin.openrc
```

11. Upload the config set to Swift using the `add_configset` utility.

The config set must be loaded into Swift to allow placement on the eLogin node during the deployment. This must be done for each config set (though not global). The `add_configset` utility scrubs the config set of data not required or desired on the eLogin node for security or operational reasons. The list of files and directories to scrub are contained in an exclude list file.

An exclude list file is provided for use as a basis for a site specific list. This file is located at `/etc/opt/cray/ellogin/exclude_lists/ellogin_cfgset_excludelist` and should be modified as required by the site.

The contents of the exclude list are set by default to ensure security over functionality. Typically, the required components of the config set are disabled by default. It is often necessary to enable `munge` and `ssh` keys. These filters are enacted at a file-by-file level. Review all changes with the relevant site security team.



WARNING: If `munge` is enabled on the SMW, the `munge` line must be commented out of the file. Failing to do so will result in the CDL node booting to an inaccessible, unconfigured state.

The contents of the `ellogin_cfgset_excludelist` are as follows. The files or directories to exclude are rooted at the config set directory: `/var/opt/cray/imps/config/sets/<config_set_name>`

```
worksheets
config/cray_sdb_config.yaml          # sdb configuration
files/roles/common/etc/ssh           # ssh keys
files/roles/common/root               # ssh and nodehealth
files/roles/munge                     # munge
files/roles/common/etc/opt/cray/xtremoted-agent
files/roles/merge_account_files      # site provided user account info
```

- a. Run the following command to scrub and upload the config set into Swift.

```
cmc# add_configset -c config_set_name -e /etc/opt/cray/ellogin/\
exclude_lists/ellogin_cfgset_excludelist
```

IMPORTANT: Whenever the config set changes and then pushed to the CMC, the config set must be loaded to Swift to allow the eLogin node to access the changes.

If the Heat stack was previously deployed, the stack must be deleted and redeployed.

- b. Run `heat stack-list` at the command line to check the status of the Heat stack deployment.

```
cmc# heat stack-list
```

Run steps (c, d, and e) only in the circumstance where the Heat stack is deployed.

- c. (Conditional): Delete the Heat stack to shut down the node.

```
cmc# heat stack-delete stack_name
```

- d. (Conditional): Verify that the Heat stack was deleted before re-deploying.

```
cmc# heat stack-list
```

- e. (Conditional): Re-deploy the Heat stack to the node.

```
cmc# /etc/opt/cray/openstack/heat/templates/deploy_elogin_name.sh
```

12. Create the config set action list:

- a. Move to the Heat stack template directory.

```
cmc# cd /etc/opt/cray/openstack/heat/templates
```

- b. Copy the `copy_p0.template` to `copy_config_set_name`, where `config_set_name` is the name of the config set to be used by the image.

```
cmc# cp copy_p0.template copy_config_set_name
```

- c. Edit the `copy_config_set_name`, so that instances of `p0` are replaced with the name of the config set. Replace all instances of `p0` with the config set name. If the config set is named `p0`, no changes are required.

To replace all instances of `p0` with the config set name, change the following:

```
"args": "-pxzvf /tmp/configset_name_configset.tar.gz -C /mnt/",
"url": "swift:configset_name_configset/vconfigset_name_configset.tar.gz",
"target": "/tmp/configset_name_configset.tar.gz"
```

- d. Add the action list to Glance.

```
cmc# glance image-create --is-public True \
--disk-format raw --container-format bare --name copy_config_set_name\
--file copy_config_set_name
```

Perform this step only once for each config set. Repeat this step for each config set name change.

13. Configure the deployment of images and deploy.

OpenStack nodes are deployed by creating a Heat stack using a template. A set of key-value parameters containing configuration information is supplied by an environment file.

- a. Log on to the CMC, and change directory to: `/etc/opt/cray/openstack/heat/templates`.

```
cmc# cd /etc/opt/cray/openstack/heat/templates
```

- b. Copy the appropriate eLogin environment file to: `elogin_name-env.yaml`

- If dynamic management IP addresses are desired, use: `elogin-env.yaml.template`
- If static management IP addresses are desired, use: `elogin-env-fixed-ip.yaml.template`.

```
cmc# cp chosen-template elogin_name-env.yaml
```

- c. Edit the copied file with site-appropriate settings for the node:

```
cmc# vi elogin_name-env.yaml
```

```
parameters:
  image_id: elogin_name.qcow2
  host_name: elogin_name
  fixed_ip: IP_address
  instance_flavor: eloginflavor
  cray_config_set: p0
  cims_host_name: example-cims
  ironic_id: elogin_node_uuid
  actions_list: copy_p0
```

image_id	Name of the image pushed from the SMW and appended with <code>.qcow2</code> . To display the image name, use <code>glance image-list</code> .
host_name	The host name of the node to be deployed.
fixed_ip	The static IP address of the management interface on this eLogin node. This must be an IP address in the management network that is unique to the node. The <code>fixed_ip</code> address is only available in the <code>elogin-env-fixed-ip.yaml.template</code> .
instance_flavor	Nova flavor of the CDL being booted. In most cases, use <code>eloginflavor</code> .
cray_config_set	Name of config set to use.
cims_host_name	Host name of the management controller (not an alias).
ironic_id	UUID of the node being booted by this stack. To determine the UUID, use the <code>ironic node-list</code> command. This is used to target specific hardware.
actions_list	A list of additional actions to take. This list must have the value of the config set action list uploaded above for the appropriate config set.

d. Create a Heat template.

Copy `deploy_elogin.sh.template` to `deploy_<elogin_name>.sh`.

```
cmc# cp deploy_elogin.sh.template deploy_<elogin_name>.sh
```

Edit the `deploy_<elogin_name>.sh` file with site-appropriate settings:

```
TEMPLATE_FILE=/etc/opt/cray/openstack/heat/templates/elogin_template.yaml
ENV_FILE=/etc/opt/cray/openstack/heat/templates/elogin-env.yaml
STACK_NAME=elogin
```

TEMPLATE_FILE Full path to the Heat template file. Use the same template file used in step 13B:

- `elogin_template.yaml`: If `elogin-env.yaml.template` was used.
- or
- `elogin_template_fixed_ip.yaml`: If `elogin-env-fixed-ip.yaml.template` was used.

ENV_FILE Full path to the `<elogin_name>-env.yaml` file from the previous step.

STACK_NAME The stack name to use in Heat, usually the name of the eLogin node.

e. Create the Heat stack.

This step requests that Openstack deploy the image to the eLogin node.

```
cmc# ./deploy_elogin_name.sh
```

At this point, the node boots. To monitor the boot, observe the console. Use `ironic_conman` to connect to the console.

To access the console of an eLogin node:

1. Find the Ironic name of the eLogin node.

```
cmc# ironic node-list
```

2. Attach to the console with `ironic_conman` using the Ironic name of the eLogin node.

```
cmc# ironic_conman ironic_name
```

For more details, refer to [eLogin Console Access](#).

Conman takes over the session, transferring into a serial-over-Lan console session with the node. All keystrokes are forwarded to the node.

The process pauses for ~5 to 10 minutes on nullwaiting for notification of completion. At this time, the base image is converted and copied (via Linux `dd`) to the disk of the node, and then the node restarts. The node boots to a root log-in state. The process may take an hour or more for the PE to synchronize before user access is enabled.

At boot time, the PE is copied to the node. The estimated time for this process is one hour or more on the first boot.

To monitor progress, log into the console as root, and watch the synchronization log.

```
cmc# ssh example-elogin
elogin# tail -f /var/opt/cray/persistent/pe_sync.log
```

`ironic_conman` logs the console output to: `/var/log/conman/ironic-UUID.log`.

To escape or disconnect the console, the command-line characters are:

- Escape: Type "&."
- Disconnect: Type "@."

14. Repeat the previous step for each eLogin node.

5 Enable LiveUpdates Support for eLogin Nodes

About this task

LiveUpdates is a repository content distribution mechanism designed to route IMPS repository content information from a central location to client nodes. It provides client nodes the ability to dynamically install update packages or install new software using the distribution native package manager (`zypper`).

LiveUpdates is disabled by default for eLogin nodes due to security concerns. With LiveUpdates, eLogin nodes dynamically pull content from the management controller. Although the use of RPM and package managers is limited to users with root access, the distribution of RPM content is over HTTP. This effectively allows any user to pull package content, but not install it locally without elevated privileges.

IMPORTANT: Software Environment Congruence

Keep the eLogin software stack as close as possible with the XC login node software. Divergence between these two software stacks can cause programs that rely on agreed APIs and protocols to function in unexpected ways.

If LiveUpdates is used to maintain and update XC nodes, eLogin environments must also be kept up-to-date. If LiveUpdates is configured for eLogin use, then the same package manager update commands are used to accomplish this. If not, a new eLogin image must be created on the SMW, sent to Glance, and then deployed and booted on a CDL node, as described in [Configure and Manage an eLogin Image](#).

To enable LiveUpdates support for eLogin nodes, follow this procedure to configure the necessary IPtable and firewall settings on both the SMW and management controller.

On eLogin nodes, the booted config set determines if the LiveUpdates service is turned on. If LiveUpdates is turned on, it configures local repositories to reference the upstream repositories of origin on the SMW. The package manager invocations will not function if the SMW and management controller configuration procedure steps are not completed.

Procedure

SMW Configuration

1. Register the LiveUpdates service with the SMW firewall.
 - a. (For `SuSEFirewall12`) Append the service `liveupdates` to the `/etc/sysconfig/SuSEFirewall12` configuration file and then apply the updated firewall setting to the SMW.

```
smw# vi /etc/sysconfig/SuSEFirewall12
smw# /sbin/SuSEFirewall12 start
```

- b. (For other firewalls) Ensure that port traffic is routable over port 8242 through `eth0`.

```
smw# iptables -I INPUT -i eth0 -p tcp --dport 8242 -m state \
--state NEW,ESTABLISHED -j ACCEPT
```

CSMS Configuration

2. Configure the management controller firewall to allow traffic over 8242 from the eLogin nodes.

Manually configure the IPtables by specifying the Ethernet interface responsible for providing connectivity to the eLogin nodes (this is dependent on the cabling of the management controller).

```
cmc# ping -I br-em2 eloin_name
PING eloin2.us.cray.com (172.30.50.174) from 172.30.50.129 br-em2: 56(84) bytes of data.
64 bytes from eloin2.us.cray.com (172.30.50.174): icmp_seq=1 ttl=64 time=0.408 ms
64 bytes from eloin2.us.cray.com (172.30.50.174): icmp_seq=2 ttl=64 time=0.349 ms
64 bytes from eloin2.us.cray.com (172.30.50.174): icmp_seq=3 ttl=64 time=0.375 ms

cmc# iptables -I INPUT -i br-em2 -p tcp --dport 8242 -m state \
--state NEW,ESTABLISHED -j ACCEPT
```

3. Enable and start the LiveUpdates service.

```
cmc# systemctl enable liveupdates
cmc# systemctl start liveupdates
```

6 Diagnostics

6.1 eLogin Console Access

About this task

Use `ironic_conman` to connect to an eLogin console.

To access the console of an eLogin node, connect to it using `ironic_conman`.

Procedure

1. Find the Ironic name of the eLogin node.

```
cmc# ironic node-list
```

2. Attach to the console with `ironic_conman` using the Ironic name of the eLogin node.

```
cmc# ironic_conman ironic_name
```

Conman takes over, putting you into a serial-over-LAN console session with the node. All keystrokes are forwarded to the node.

Conman logs the console output to: `/var/log/conman/ironic-UUID.log`.

For debugging purposes, follow by running `tail` on the file.

```
cmc# tail /var/log/conman/ironic-UUID.log
```

To escape or disconnect the console, the command-line characters are:

- Escape: Type "&."
- Disconnect: Type "@."

6.2 The journalctl Command

`systemd` (on both the management controller 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 the database is through the `journalctl` tool.

The command, `journalctl -a`, displays all kernel messages and other available information.

```
eLogin# journalctl -a
-- Logs begin at Mon 2015-06-08 19:28:53 UTC, end at Thu 2015-06-11 22:15:01 UTC. --
Jun 08 19:28:53 example-eLogin2 systemd-journal[1681]: Runtime journal is using 8.0M\
(max allowed 4.0G, trying to leave 4.0G free of 252.4G available → current limit 4.0G).
Jun 08 19:28:53 example-eLogin2 systemd-journal[1681]: Runtime journal is using 8.0M \
(max allowed 4.0G, trying to leave 4.0G free of 252.4G available → current limit 4.0G).
Jun 08 19:28:53 example-eLogin2 kernel: Initializing cgroup subsys cpuset
Jun 08 19:28:53 example-eLogin2 kernel: Initializing cgroup subsys cpu
Jun 08 19:28:53 example-eLogin2 kernel: Initializing cgroup subsys cpuacct
Jun 08 19:28:53 example-eLogin2 kernel: Linux version 3.12.28-4-default \
(geeko@buildhost) (gcc version 4.8.3 20140627 [gcc-4_8-branch revision 212064] \
(SUSE Linux) ) #1 SMP Thu Sep 25 17:02:34 UTC 2014 (9879bd4)
Jun 08 19:28:53 example-eLogin2 kernel: Command line: \
initrd=/var/lib/tftpboot/e79e85cd-57f5-4fcd-ba43-14ccea0375e7/ramdisk \
root=UUID=f09a21f4-1bb1-4b1e-8a12-c5329e4b9073 ro text nofb nomodeset vga=normal \
BOOT_IMAGE=/var/lib/tftpboot/e79e85cd-57f5-4fcd-ba43-14ccea0375e7/kernel BOOTIF=01-90-
b1-1c-39-ea-3c
```

The command `journalctl -f` behaves like `tail -f`, displaying updates as they happen. For example, `journalctl -f /usr/sbin/ntpd` monitors `ntpd`-related messages. Any system daemons that produce output visible to `journalctl` can be filtered similarly.

```
eLogin# journalctl -f /usr/sbin/ntpd
-- Logs begin at Mon 2015-06-08 19:28:53 UTC, end at Thu 2015-06-11 22:15:01 UTC. --
Jun 08 19:30:00 example-eLogin2 ntpd[3436]: ntpd 4.2.6p5@1.2349-o Wed Oct 8 14:41:40 UTC
2014 (1)
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: proto: precision = 0.103 usec
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: ntp_io: estimated max descriptors: 1024, initial
socket boundary: 16
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen and drop on 1 v6wildcard :: UDP 123
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen normally on 3 eth2 10.142.0.111 UDP 123
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen normally on 4 lo ::1 UDP 123
Jun 08 19:30:00 example-eLogin2 ntpd[3437]: Listen normally on 5 eth2
fe80::92b1:1cff:fe39:ea3c UDP 123
```

6.3 The /var/log Directory

System log message files are located in `/var/log/messages` directory. The message files contain helpful information about the state of the system. Other system services log to their standard locations in `/var/log`. Most log files are only visible for the user `root`.

6.4 Ansible Install Logs

There are two log files on the eLogin node that track installation and configuration of the system:

/var/opt/cray/log/ansible/sitelog-init	Initial configuration of the system before <code>systemd</code> startup.
/var/opt/cray/log/ansible/sitelog-booted	Configuration of the system during <code>systemd</code> startup.

6.5 The `cray_dumpsys` Command

The `cray_dumpsys` script gathers data needed to debug the CSMS. It dumps the state of the OpenStack services, configuration and log files, and background information about the system. The files are compressed and the results are stored in the `/var/tmp/` directory. By default, only recent logs are dumped.

`cray_dumpsys` includes the `--all-logs` option to dump all rotated logs. Additionally, the `--days` option dumps logs up to a certain number of days. For example:

```
cmc# cray_dumpsys --days 4
/root/admin.openrc sourced!

sosreport (version 3.2)

This command will collect diagnostic and configuration information from
[...]
Setting up archive ...
Setting up plugins ...
Running plugins. Please wait ...
Running 1/12: memory...
Running 2/12: mysql...
Running 3/12: networking...
[...]
Running 12/12: newtplugin...

Creating compressed archive...

Your sosreport has been generated and saved in:
/var/tmp/sosreport-newt-20150923124808.tar.xz

The checksum is: bb87d9323f88813e07659e53aebb16b6

Please send this file to your support representative.
```

6.6 Configure eLogin `cray_dumpsys` Plugin

To include information from eLogin nodes in the `cray_dumpsys` report, an eLogin plug-in must be created. To configure this plug-in, edit the `/etc/cray_tools/cray_tools.conf` file, adding eLogin to the list of enabled plugins, and a space-separated list of eLogin node names in the `eloin.nodes` option.

For example:

```
[cray_dumpsys]
# List of enabled Cray_dumpsys plugins.
plugins =
    process
    networking
```

```

openvswitch
mysql
openstack_cinder
openstack_horizon
openstack_keystone
openstack_neutron
openstack_nova
openstack_swift
newtplugin
eloin
eloin

# List of Cray_dumpsys plugin options.
options =
  openstack_cinder.log=off
  openstack_horizon.log=off
  openstack_keystone.log=off
  openstack_nova.cmds=on
  openstack_nova.log=off
  openstack_swift.log=off
  eloin.nodes="eloin1 eloin2"

```

To override the configured node list, use the `cray_dumpsys` option: `--extra-option eloin.nodes="eloin1 eloin2"`.

6.7 OpenStack Log File Locations

Log files of each OpenStack service are stored in the `/var/log/service` directory on the management server/controller.

Table 2. OpenStack Services Log File Locations

OpenStack Service	Log File Location
Cinder	/var/log/cinder
Glance	/var/log/glance
Heat	/var/log/heat
Ironi	/var/log/ironi
KeyStone	/var/log/keystone
Neutron	/var/log/neutron
Nova	/var/log/nova
Swift	/var/log/swift

More detailed information about logging and monitoring in OpenStack is available at: <http://www.openstack.org>. Specific information about logs of each service can also be found in the documentation of the service under consideration.

6.8 OpenStack Diagnostics

The management controller uses standard OpenStack commands to manage most components. The OpenStack diagnostic commands for Heat, Nova, and Ironi are described in the following sections. For each command, usage and a short description are listed, including an example of a successful output.

For additional information on these commands and a full list of available commands, the OpenStack documentation is available at: <http://docs.openstack.org/cli-reference/content/> or by typing: `OpenStack_component help command`. (Example, `heat help stack-list`.)

Note the OpenSource documentation may reflect a different version of OpenStack than is installed on the management controller.

6.8.1 Heat Diagnostic Commands

The csms uses standard OpenStack commands to manage most components. For additional information on these commands and a full list of available commands, the OpenStack documentation is available at: <http://docs.openstack.org/cli-reference/content/> or type `OpenStack_component help component_command`.

- `heat stack-list`

```
cmc# heat stack-list
```

id	stack_name	stack_status	creation_time
4452df3e-46f1-4345-8b61-c489bbbc863f	eLogin1	CREATE_COMPLETE	2015-06-11T20:52:39Z

- `heat stack-show stack_name_or_id`

This command describes the stack.

```
cmc# heat stack-show eLogin1
```

Property	Value
capabilities	[]
creation_time	2015-06-11T20:52:39Z
description	Simple deploy template with parameters
disable_rollback	True
id	4452df3e-46f1-4345-8b61-c489bbbc863f
links	http://172.30.50.129:8004/v1/acc067874bfd45dcbce9f44d1516910a/ \ stacks/eLogin1/4452df3e-46f1-4345-8b61-c489bbbc863f (self)
notification_topics	[]
outputs	[{ "output_value": { "management": ["10.149.0.157"] }, "description": "IP assigned to the instance", "output_key": "instance_ip" }]
parameters	{ "network_id": "management", "OS::stack_id": "4452df3e-46f1-4345-8b61-c489bbbc863f", "OS::stack_name": "eLogin1", "cray_config_set": "sta_p2", "key_name": "default", "instance_flavor": "eLoginflavor", "cray_cims_ip": "10.149.0.1", "image_id": "whale_eLogin2.qcow2", "host_name": "eLogin1" }
parent	None

stack_name	eLogin1
stack_owner	admin
stack_status	CREATE_COMPLETE
stack_status_reason	Stack CREATE completed successfully
template_description	Simple deploy template with parameters
timeout_mins	None
updated_time	None

6.8.2 Nova Diagnostic Commands

The CSMS uses standard OpenStack commands to manage most components. For additional information on these commands and a full list of available commands, the OpenStack documentation is available at: <http://docs.openstack.org/cli-reference/content/> or type: `OpenStack_component help component_command`.

- `nova list`

This command lists active servers.

```
cmc# nova list
```

ID	Name	Status	Task State	Power State	Networks
ac6384e2-...-4c9e1885	eLogin1	ACTIVE	-	Running	management=10.142.0.156

- `nova show server_name_or_id`

This command displays details about the given server.

```
cmc# nova show example-elogin
```

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	nova
OS-EXT-SRV-ATTR:host	cims
OS-EXT-SRV-ATTR:hypervisor_hostname	e63ffc33-029f-44ac-8808-c55909f85f2f
OS-EXT-SRV-ATTR:instance_name	instance-00000050
OS-EXT-STS:power_state	1
OS-EXT-STS:task_state	-
OS-EXT-STS:vm_state	active
OS-SRV-USG:launched_at	2015-06-11T21:01:16.000000
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
config_drive	
created	2015-06-11T20:52:40Z
flavor	eloginflavor (012435a2-54f7-458b-8734-6cdefe58b52e)
hostId	9e184dc6993ac9954652611f13f3faaaa797b5ff1625869be0edeb80
id	ac6384e2-4ca0-421f-9e6e-4c9e138f8785
image	eLogin2_new.qcow2 (1cc535c0-9f71-446a-8f4e-66aacc2617fe)
key_name	default
management network	10.149.0.156
metadata	{"cray_config_set": "sta_p2", "cray_cims_ip": "10.147.0.1", "cray_cims_rsync_password": "fab9--679b47aca4", "cray_cims_rsync_username": "eLogin"}
name	eLogin1
os-extended-volumes:volumes_attached	[]
progress	0
security_groups	default
status	ACTIVE
tenant_id	acc067874bfd45dcbce9f44d1516910a
updated	2015-06-11T21:01:16Z
user_id	762d33ecbe64356a933e27bce688579

6.8.3 Ironic Diagnostic Commands

The CSMS uses standard OpenStack commands to manage most components. For additional information on these commands and a full list of available commands, the OpenStack documentation is available at: <http://docs.openstack.org/cli-reference/content/> or type: `OpenStack_component help component_command`. For example: `heat help stack-list`.

- `ironic node-list`

Lists nodes that are registered with the Ironic service.

```
cmc# ironic node-list
```

UUID	Instance UUID	Power State	Provisioning State	Maintenance
e6300c33-...-c55952f9f	ac6384e2-...-4c9e138f85	power on	active	False

- `ironic node-show identifier`

Displays detailed information for a node, where *identifier* is an ID, UUID, or instance ID.

```
cmc# ironic node-show e63ffc33-029f-44ac-8808-c55909f85f2f
```

Property	Value
instance_uuid	ac6384e2-...-4c9e138f85
target_power_state	None
properties	{u'memory_mb': 131072, u'cpu_arch': u'x86_64', u'local_gb': 550, u'cpus': 32}
maintenance	False
driver_info	{u'pxe_deploy_ramdisk': u'd867d80f-8847-40f0-a499-50de8a31a997', u'pxe_deploy_kernel': u'a46d386c-3307-45fe-a395-8fc1d5c2285c', u'ipmi_address': u'10.142.0.5', u'ipmi_username': u'root', u'ipmi_password': u'initial0'}
extra	{u'description': u'elogin1'}
last_error	None
created_at	2015-05-18T21:25:54+00:00
target_provision_state	None
driver	pxe_ipmitool_cm
updated_at	2015-06-12T14:54:29+00:00
maintenance_reason	
instance_info	{u'ramdisk': u'960017eb-ef0b-439a-9dd1-331c9ed449ac', u'kernel': u'9a51857a-008c-47cb-9768-92622fad7314', u'root_gb': u'100', u'image_source': u'1cc535c0-9f71-446a-8f4e-66aacc2617fe', u'deploy_key': u'J0047487N9F4CLNMFHSS9QLHHBTNLPF', u'swap_mb': u'10240'}
chassis_uuid	None
provision_state	active
reservation	None
power_state	power on
console_enabled	False
uuid	e6300c33-...-c55952f0f

6.9 Common Issues

6.9.1 Disk Space On CMC and eLogin Node

Disk Space Issues On CMC

There are multiple places on the Cray Management Controller (CMC) where pressure potentially builds up on the file system:

- Images fill up space in `/var/lib/glance`.
Solution: Remove using Glance commands only.
- Images fill up space in `/var/lib/tftpboot`.
Solution: These are removed automatically following a successful deployment. If they remain, remove manually.
- PE, config sets, and repositories fill up space in subdirectories of `/var/opt/cray`.
Solution: Remove manually.

Disk Space on eLogin Node

The eLogin node is partitioned into two virtual disks:

- `sda` contains the OS, and other data that can be rewritten. If an image is re-deployed, all data on `sda` will be overwritten. There should be no space concerns.
- `sdb` is configured as persistent storage for the node. Config sets, PE, and some job submission details for workload managers are stored here. If the partition is destroyed, all data specified by the config set is re-synchronized upon reboot. Administrators can safely delete any data here.

6.9.2 Recovering from Broken CSMS Installation

Prerequisites

- Configuration of CMC hardware BIOS.
- Successful install of CentOS on CMC.
- CSMS install on CMC failed.

About this task

If errors occur when configuring `group_vars/all` or `group_vars/cims`, the CSMS installation fails. To recover from this type of error, rerun the main Ansible playbook to reconfigure the SQL used by OpenStack.

Procedure

1. Stop all OpenStack services.

```
cmc# ansible-playbook stop_openstack_services.yaml
```

2. Drop schemas directly related to OpenStack.

```
cmc# mysql
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 34562
Server version: 5.5.41-MariaDB MariaDB Server
+-----+
14 rows in set (0.00 sec)
MariaDB [(none)]> drop database cinder;
Query OK, 21 rows affected (0.08 sec)

MariaDB [(none)]> drop database glance;
Query OK, 13 rows affected (0.06 sec)
```

```

MariaDB [(none)]> drop database heat;
Query OK, 13 rows affected (0.04 sec)

MariaDB [(none)]> drop database ironic;
Query OK, 5 rows affected (0.01 sec)

MariaDB [(none)]> drop database keystone;
Query OK, 18 rows affected (0.67 sec)

MariaDB [(none)]> drop database neutron;
Query OK, 142 rows affected (0.88 sec)

MariaDB [(none)]> drop database neutron_ovs;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> drop database swift;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> drop database nova;
Query OK, 108 rows affected (1.04 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| hssds |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.00 sec)

```

3. Correct values in the configuration files, and then rerun the main Ansible play.

```

cmc# cd /etc/opt/cray/openstack/ansible
cmc# ansible-playbook -i hosts stop_openstack_services.yaml
cmc# ansible-playbook -i hosts main.yaml

```

6.9.3 Repeated Cycle of Rebooting CentOS Deploy Image

Prerequisites

- Successful install of CentOS on CMC
- Successful install and configuration of CSMS
- Deployment of the eLogin image fails (causing reboot cycle)

About this task

The failure signature of this issue is a repeated cycle of rebooting the deploy image until the Heat stack timeout on boot attempts is reached. The console logs show the following output.

```

[FAILED] Failed to start LSB: Bring up/down networking.
See 'systemctl status network.service' for details.

```



```
[ OK ] Reached target Network is Online.
Starting Ironiic Callback...
[ 11.749171] bnx2x 0000:01:00.2: msix capability found
[ 11.755303] bnx2x 0000:01:00.2: part number 394D4342-30383735-30305430-473030
[FAILED] Failed to start Ironiic Callback.
```

The cause of the reboot cycle is that the deploy image cannot bring up the management network interface in order to continue the deployment of the eLogin image. This has been seen on eLogin nodes with the 2 x 10 GbE / 2 x 1 GbE LOM configuration. The root cause is out-of-date LOM firmware. The firmware on the LOM must be FFV7.2.20 or later. The Family Firmware Version (FFV) is available by connecting to the iDRAC of the eLogin node using a browser. The following procedure describes how to verify the FFV.

Procedure

1. Find the iDRAC (BMC) IP address for the eLogin server.

```
csms# source ~/admin.openrc
csms# ironic node-show percival-eLogin3 | grep ipmi_address \
| driver_info | {u'ipmi_password': u'*****', u'ipmi_address': u'10.142.0.7'}
```

2. Open a browser on the CMC, and enter the `ipmi_address` value for the URL. (10.142.0.7, in this example.)
3. Enter the credentials for the iDRAC (`root/initial0`).
4. Locate **Hardware** directory from left-hand side of window, and click to expand.
5. Click on **Network Devices** under **Hardware**.
6. Click on **Integrated NIC1** in the main window.
7. Click on **+** to expand the information for **Port 3**.
8. Browse to **Port Properties** and verify the **Family Firmware Version** is 7.2.20 or later.

If the FFV is earlier than 7.2.20, please contact Cray Support to obtain the latest firmware for your eLogin node(s).