XC™ Series eLogin Installation Guide (CLE 6.0.UP03) S-2566 Rev C

Contents

1 About the XC Series eLogin Installation Guide

The XC™ Series eLogin Installation Guide CLE6.0 UP03 provides installation procedures for Cray eLogin nodes running externally to Cray XC series CLE6.0 UP03 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. Use of the term *user* throughout refers to the intended audience, not to end users of the system. This publication assumes users have competence with standard Linux and open source tools.

Record of Revision

Revision	Date	Content Information
XC Series eLogin Installation Guide CLE 6.0 UP03 S-2566 Rev C	6/21/2017	UP03 Rev C Final (this publication)
XC Series eLogin Installation Guide CLE 6.0 UP03 S-2566 Rev B	3/21/2017	UP03 Rev B Final release
XC Series eLogin Installation Guide CLE 6.0 UP03 S-2566 Rev A	3/6/2017	UP03 Rev. A Final release
XC Series eLogin Installation Guide CLE 6.0 UP03 S-2566	2/14/2017	UP03 Final release ● CSMS 1.1.3, CentOS 7.2 ● CSMS 1.1.3 patch sets (PS01-PS04)
XC Series eLogin Installation Guide CLE6.0 UP02 S-2566 Rev A	11/14/2016	UP02 Rev. A Final release
XC Series eLogin Installation Guide CLE6.0 UP02 S-2566	10/31/2016	UP02 Final release ● CSMS 1.1.3, CentOS 7.2 ● Upgrade processes included
XC Series eLogin Installation Guide CLE6.0 UP01 S-2566	06/15/2016	UP01 Final release ● CSMS 1.1.1, CentOS 7.1 ● Upgrade processes not included

This publication was previously titled eLogin Installation Guide CLE6.0 UP01. The new title XC Series eLogin Installation Guide CLE6.0 UP03 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.

Reference Documents

Specific sections in this document reference the following Cray publication:

- *XC Series eLogin Administration Guide CLE6.0 UP03 (S-2570) Rev C*

Command Prompt Conventions

Host name and account in command prompts

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

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

`smw#`	Run the command on the SMW as `root`.
`cmc#`	Run the command on the CMC as `root`.
elogin#	Run the command on the eLogin node as `root`.
`crayadm@boot>`	Run the command on the boot node as the `crayadm` user.
`hostname``#`	Run the command on the specified system as `root`.
`user@`*`hostname`*`>`	Run the command on the specified system as any non-`root` user.

Directory path in command prompt

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

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

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

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

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

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
Proporational 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

Trademarks

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

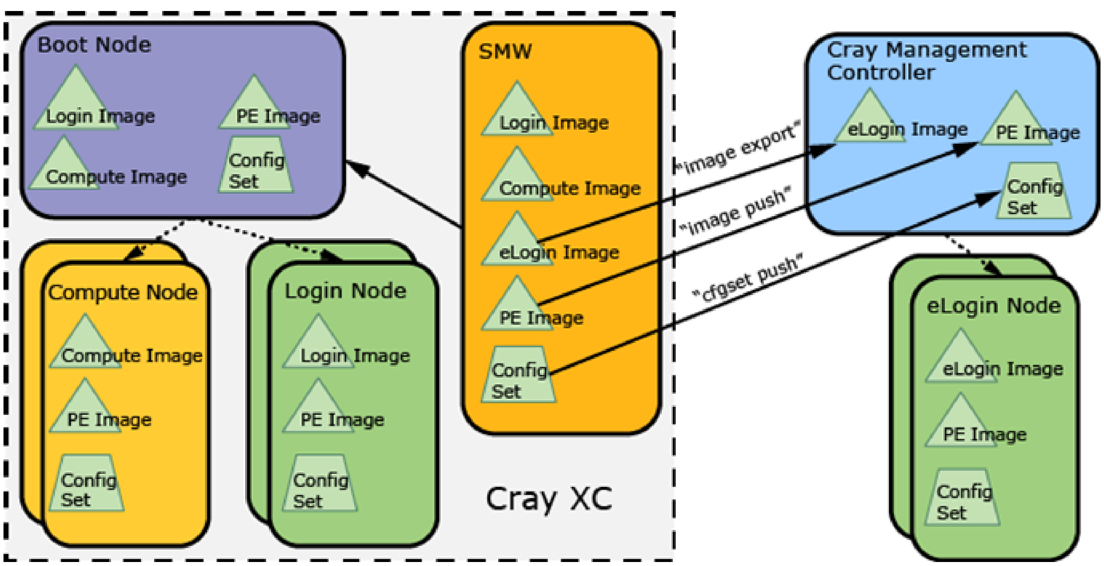
2 eLogin Architecture

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 System 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. Gereral 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 SP2 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	In addition to SLES 12 SP2, 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. 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.

2.1 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.

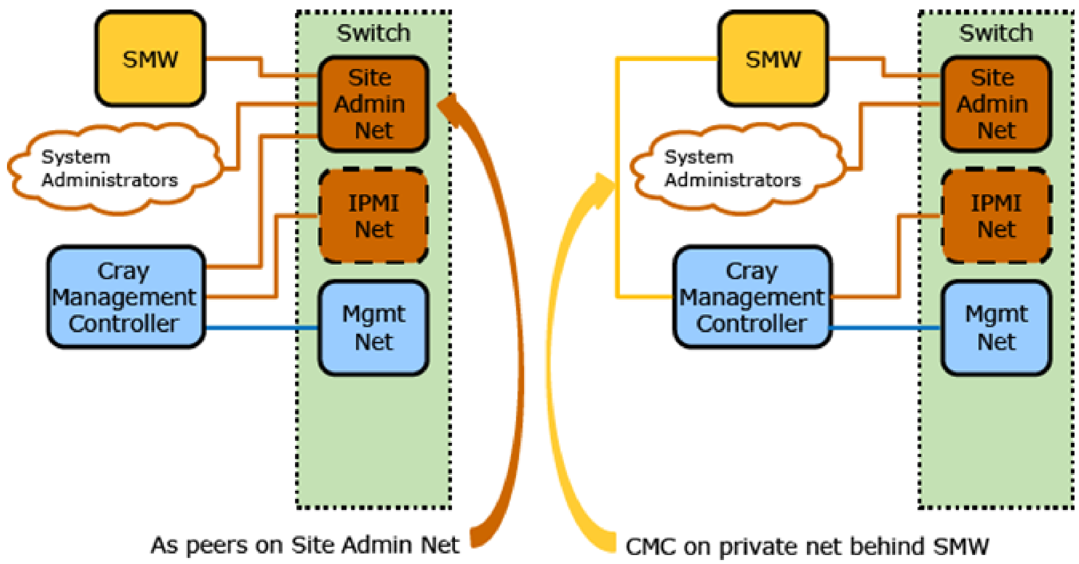
WARNING: For security reasons, configure the CMC behind the SMW on a private network.

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.

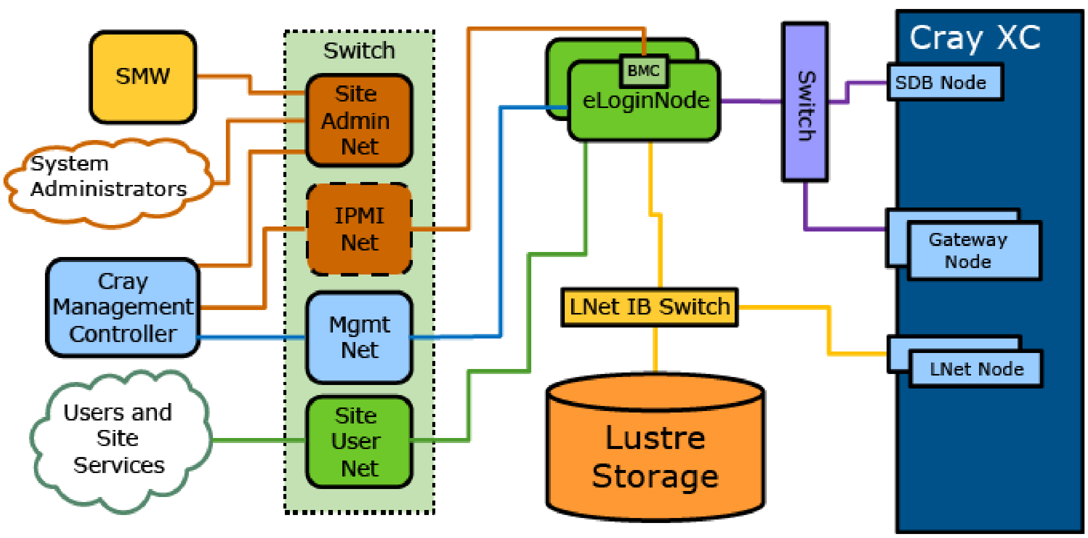
eLogin to Cray Network Attachment

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 / 10GbE / 40GbE 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.

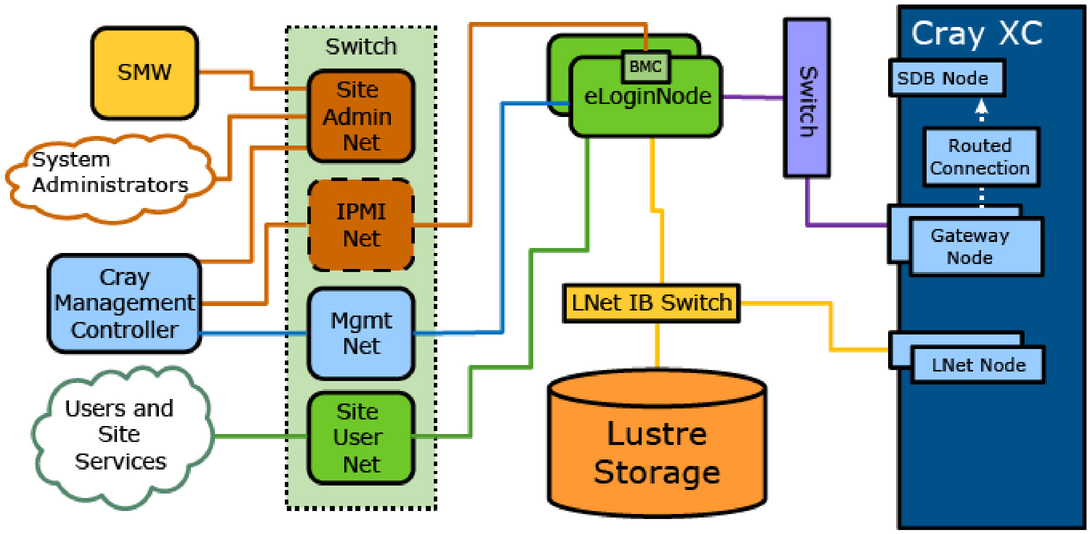
Figure 3. eLogin Nodes Direct Connection to SDB Node: eLogin System Topology



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.

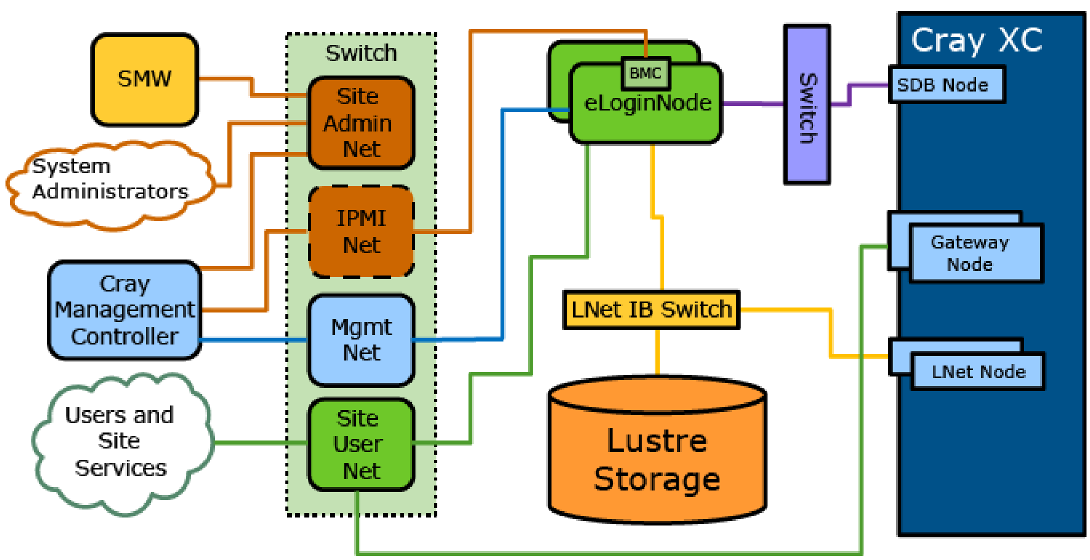
Figure 4. eLogin Nodes Routed Via Gateway to SDB Node: eLogin System Topology



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.

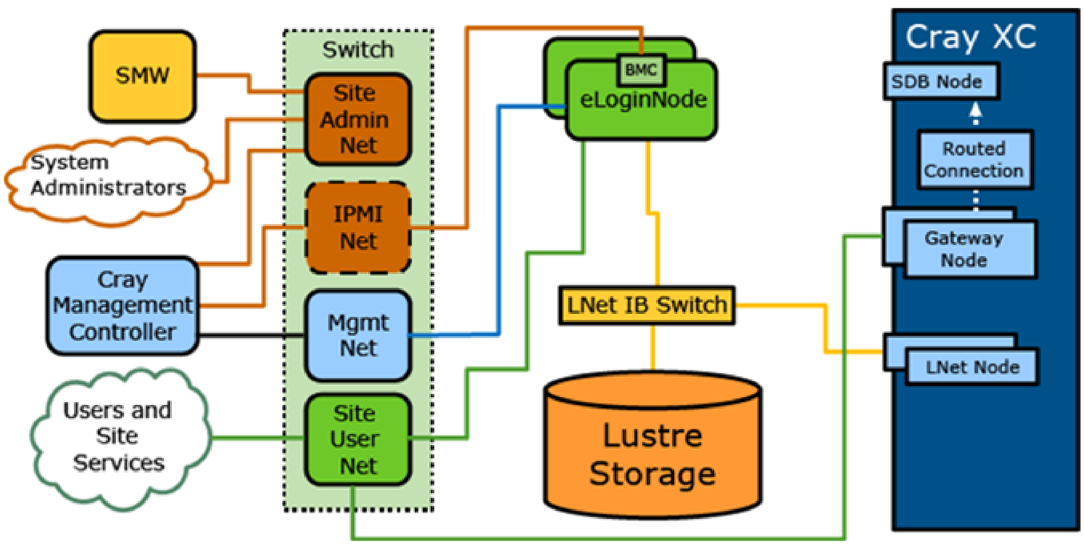
Figure 5. eLogin Nodes Direct Connection to SDB Node with Site User Gateway: eLogin System Topology



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.

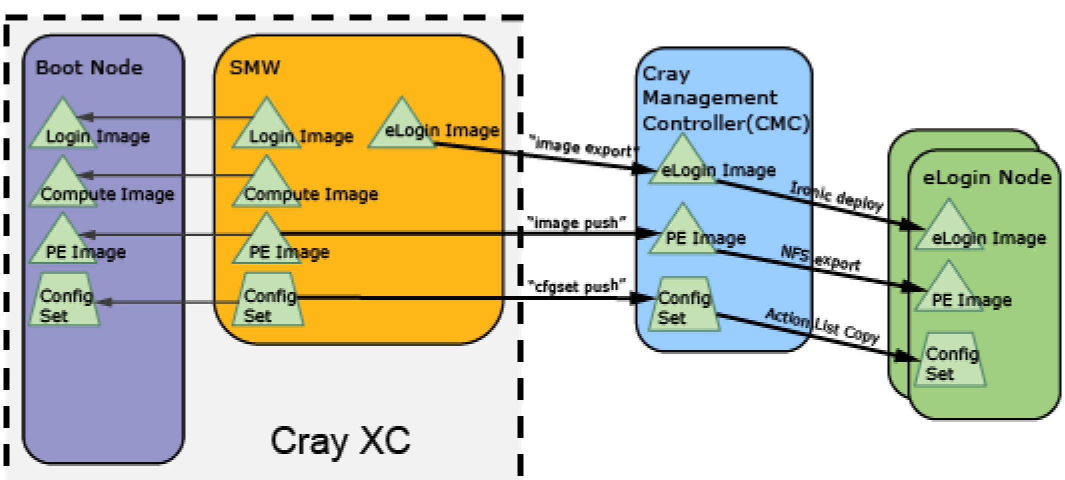
Figure 6. eLogin Nodes Routed to SDB Node via User Accessible Gateway: eLogin System Topology



2.2 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 initial provisioning process, the config set is copied to disk using a fuel action list. The config set is then copied to the final location during the first boot after initial deployment. On each boot, the CDL attempts to mount a read only NFS share that has PE on it. PE is then re-synchronized (`rsync`) from the NFS mount to the final location on the persistent disk.

Figure 7. Provisioning Process for eLogin Node



3 Source ISO Images for eLogin

The eLogin (CLE 6.0 UP03) software release supports the install of the following CentOS and Cray Software Management System (CSMS) versions:

- CentOS 7.2
- CSMS 1.1.3
- CSMS 1.1.3 patch sets:
  - CSMS 1.1.3 PS01
  - CSMS 1.1.3 PS02
  - CSMS 1.1.3 PS03
  - CSMS 1.1.3 PS04

Installing the eLogin UP03 software requires the following ISO image files. These ISOs constitute the software for installing the CentOS, CSMS, and eLogin onto the Cray Management Controller (CMC) hardware.

Table 1. eLogin (CLE 6.0 UP03) Software Release ISO Image Files

ISO Name	ISO Image File Name
CentOS 7.2 (bootable)	`Cray-CentOSbase7-1511-201605031030.iso`
CSMS 1.1.3 for CentOS 7.2	`csms_centos72-1.1.3-201608190116.iso`
CSMS 1.1.3 PS01 patch set	`CSMS_1.1.3.PS01.iso`
CSMS 1.1.3 PS02 patch set	`CSMS_1.1.3.PS02.iso`
CSMS 1.1.3 PS03 patch set	`CSMS_1.1.3.PS03.iso`
CSMS 1.1.3 PS04 patch set	`CSMS_1.1.3.PS04.iso`
eLogin	`elogin-6.0.3055-201701182038.iso`

Download ISOs on CrayPort

All ISOs are available on CrayPort. To download eLogin ISOs on CrayPort, do the following:

1. Open a browser to *CrayPort*.

   The CrayPort Account Access interface opens.

Figure 8. CrayPort Account Access Interface

2. Access your account. Type-in your CrayPort account credentials (username/password), and then click the **Sign In** button.

   NOTE: To register for a CrayPort account, click the **Register for an Account** button.

   After a successful log-in to your account, the CrayPort homepage opens.

3. Click **Support** on the toolbar.

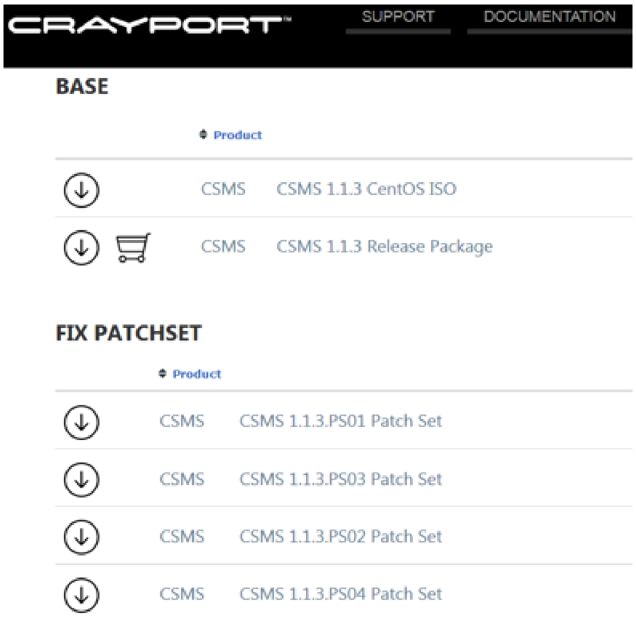
   Figure 9. CrayPort Homepage Support

4. Click **Download Software** from the **Support** drop-down menu.
5. Search for keyword: **`CLE6.0UP03`** for the eLogin ISO.
6. Locate the `CLE 6.0.UP03 Release Package` listed on the page.
7. Click the download arrow button on left-side of page to download/list the files in the Release Package.
8. Locate and select the eLogin ISO from those listed on page to download, and then follow the remaining download instructions.
9. Return to the Home page, and click **Support** on the toolbar.
10. Click **Download Software** from the **Support** drop-down menu.
11. Search for keyword: **`CSMS 1.1.3`** for CSMS base ISOs and CSMS 1.1.3 patch set ISOs.

    See the `CSMS 1.1.3 Release Package` and four CSMS patch sets (`CSMS 1.1.3.PS<01-04>`) listed on the page.

Figure 10. CrayPort CSMS 1.1.3 Release Package and CSMS 1.1.3 Patch Sets



12. Select the `CSMS 1.1.3 Release Package` to download the CSMS 1.1.3 ISO, or `CSMS 1.1.3.PS<01-04>` to download the CSMS 1.1.3 patch set (PS01-PS04) ISOs individually.

13. Click the download arrow button on left-side of page to download/list the files.

14. Select the ISO for download, and then follow the remaining download instructions.

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. Configure CSMS on CMC.
6. Configure connection beween SMW and CMC.
7. 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`: • `0.pool.ntp.org` • `1.pool.ntp.org`	
OpenStack Admin Password	admin_password	
Keystone Password	keystone_mysql_password	

Common Configuration Options

Each CDL node also needs the following information:

Item	Value
BMC IP address	
Boot interface*	
MAC address*	

* Refer to *Determine Boot Interface and MAC Address* on page 121 section of the *XC Series eLogin Installation Guide CLE 6.0 UP03 Rev C*.

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 RAID device must be configured to have two virtual disks (VDs) visible to the operating system. The first VD is configured on 1,920 GB of disk space for the base operating system (O/S), logs, and Cray's Programming Environment (PE). The first VD is configured as VD name '**sda**'. The second VD is configured on the remaining disk space, as VD name '**sdb**'. The VD '**sdb**' is also referred to as 'Swift disk' in eLogin installation documentation.

During the CMC installation and configuration procedure, the VD '**sdb**' (or Swift disk) can be used for Ironic Fuel driver deployment. Once the CMC RAID device is configured correctly, the devices `/dev/sda` and `/dev/sdb` are reserved for Cray Operating System software.

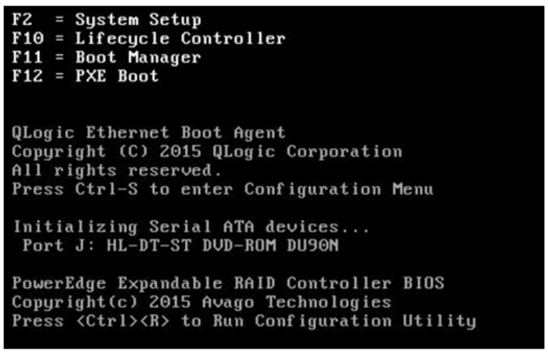
> **ATTENTION:** Stop! Do not perform this procedure if your current system requires a storage device change to an SSD defined as anything other than **sdc**. In this case, refer to *Configure SSDs on CDL Nodes*.

Perform this procedure to configure the hardware BIOS as an initial disk setup for the CMC:

Procedure

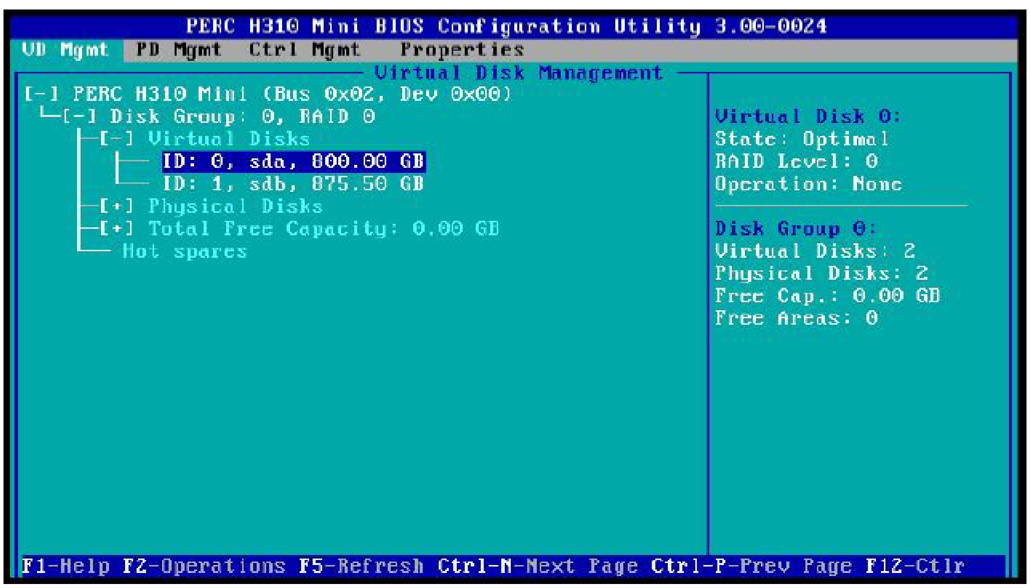
1. On startup of the CMC node, press **Ctrl-R** when prompted to enter RAID setup.

   Figure 11. Enter RAID Setup: CMC BIOS

   The RAID configuration screen opens.

   Figure 12. 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.

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

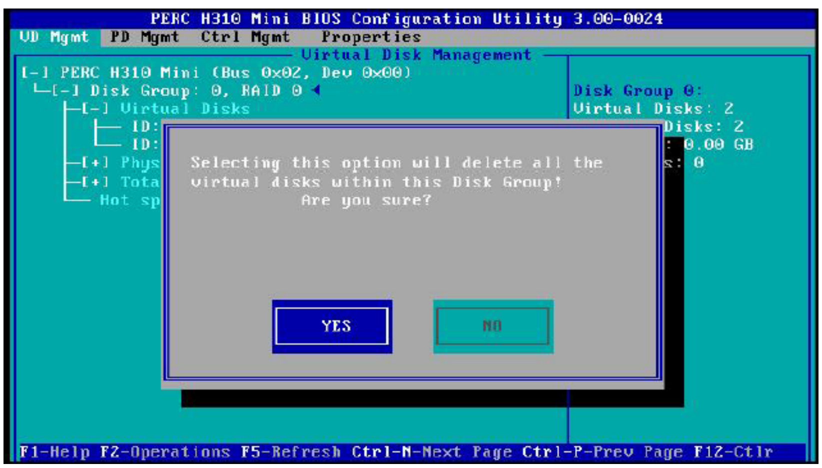
a. Select the disk.

b. Press **F2** key to get a list of operations.

c. Select **Delete Disk Group** and press **Enter**.

Figure 13. Delete Disk Group: CMC BIOS

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



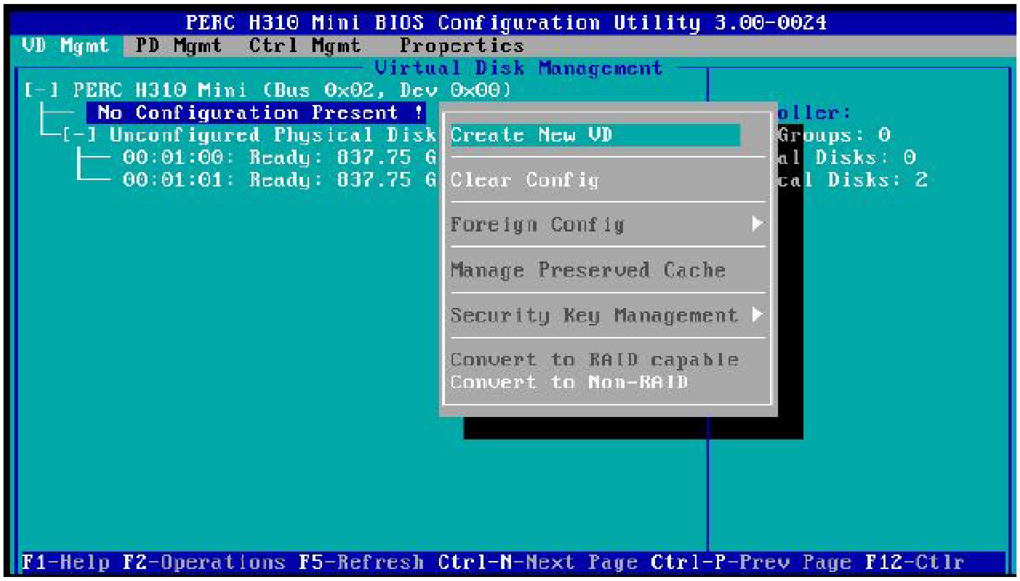
3. Create a new virtual disk (VD) A.

a. In the VD 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.

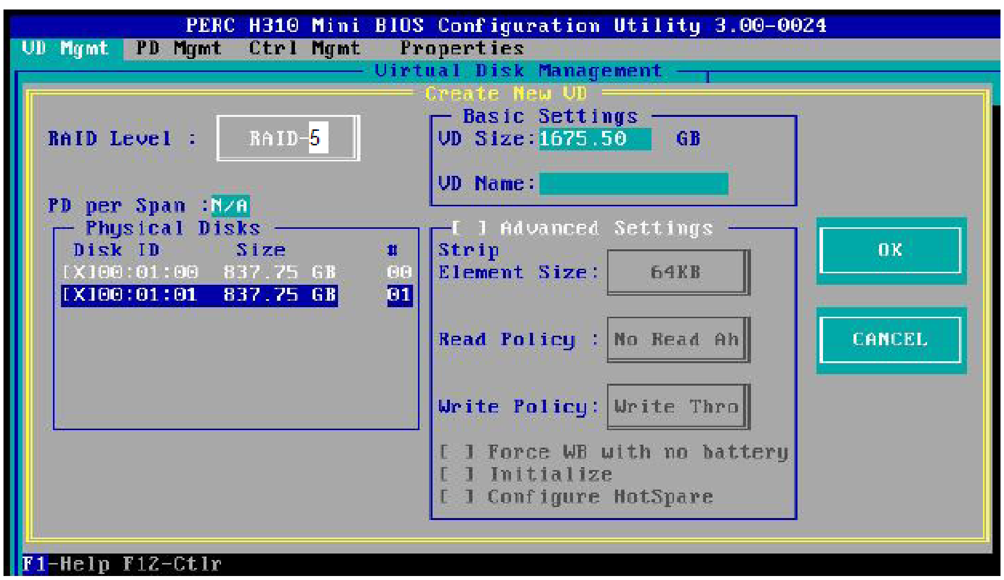
Figure 14. Create Virtual Disk: CMC BIOS



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 15. Set RAID Level 5: CMC BIOS

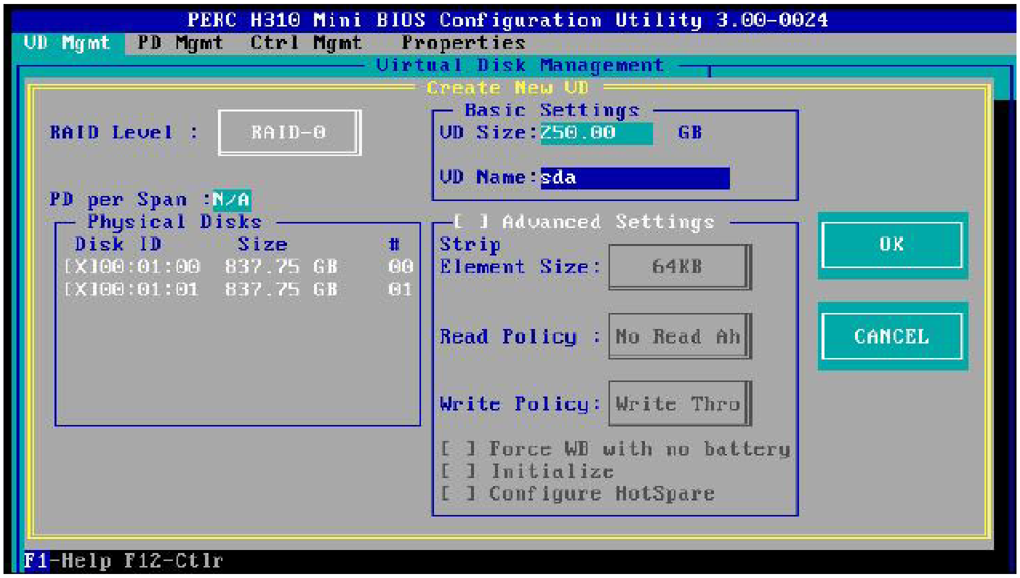


6. Set **VD Size** and **VD Name** for virtual disk A.

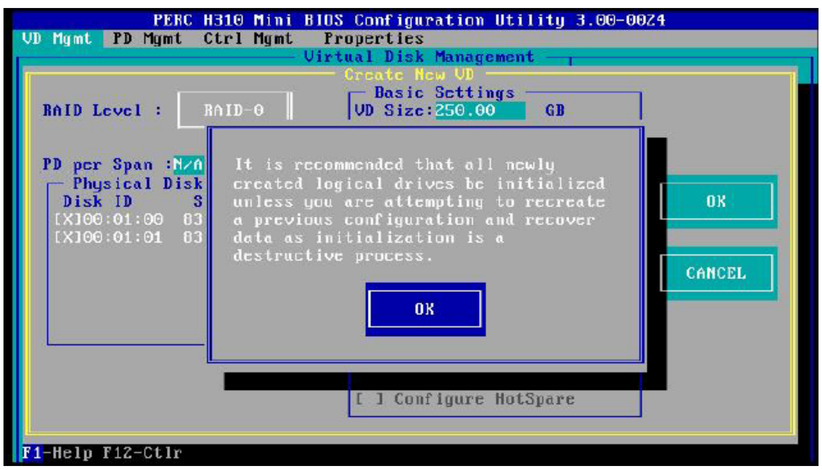
a. Set the **VD Size** of the first disk to **1,920 GB** disk space.

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

Figure 16. 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**.



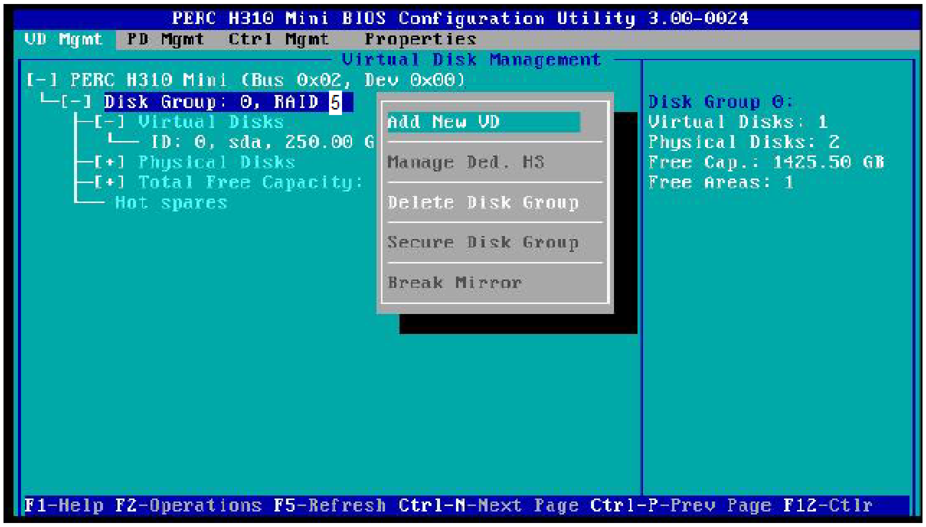
7. Create a new virtual disk (VD) B.

a. In the VD management window, navigate to **Disk Group: 0 RAID-5** using the keyboard up/down arrows.

b. Press **F2** to access the disk creation menu.

c. Select **Add New VD**.

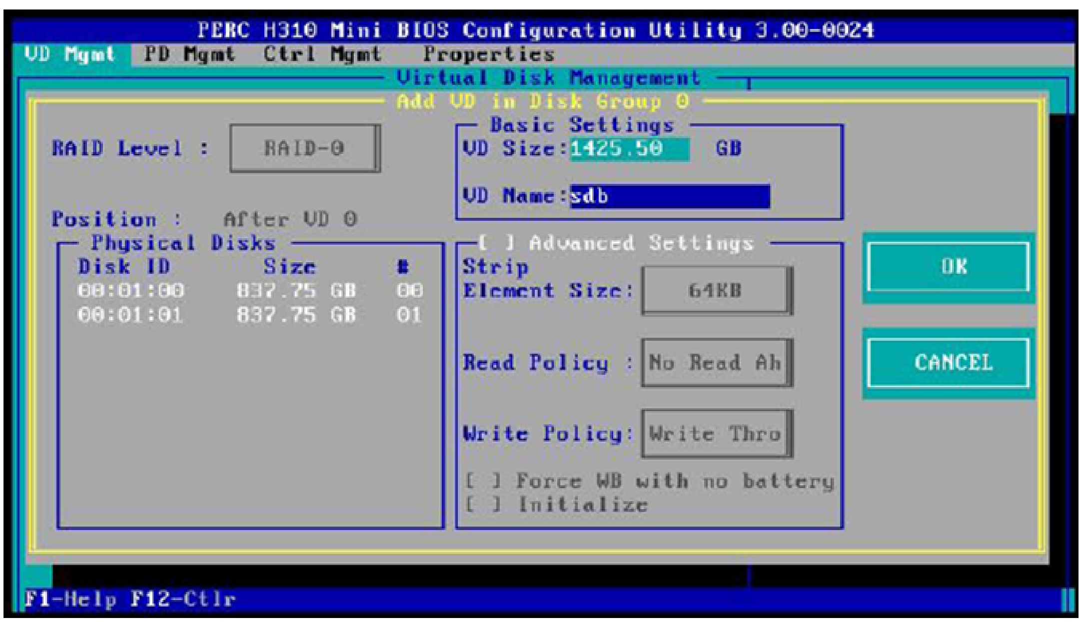
Figure 17. Create New Virtual Disk B: CMC BIOS



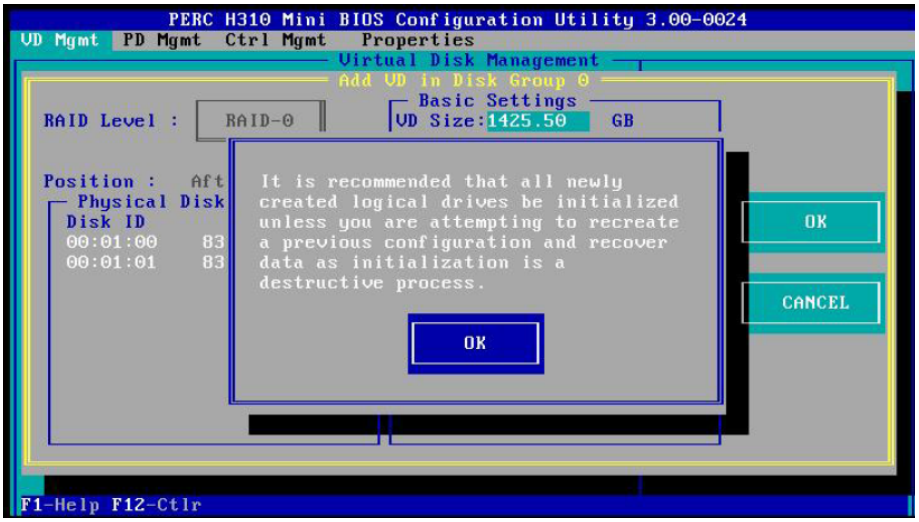
d. Set the **VD Name** to **sdb**.

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

Figure 18. 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**.

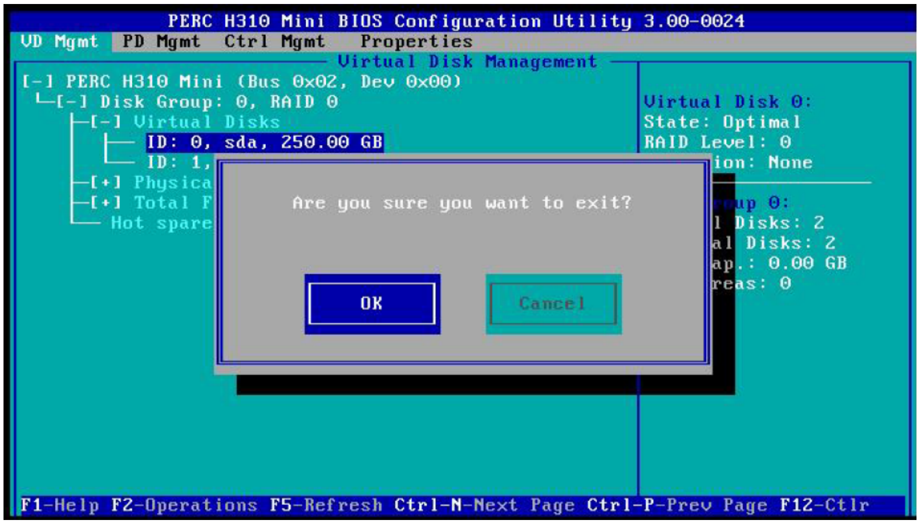


The CMC now has two disks available to install on.

Figure 19. 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 20. 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 on the CMC

Prerequisites

- The CentOS ISO image is required for this procedure.
- This procedure assumes a first time installation 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 CSMS and must be installed on the Cray Management Controller (CMC) before installing 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 for performing this procedure.

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.

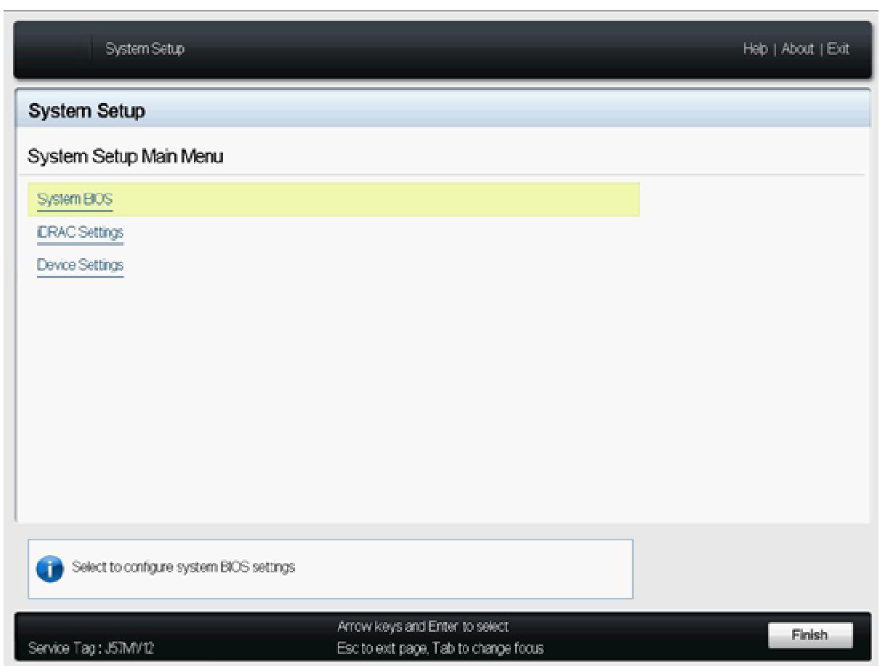
   The BIOS power-on self-test (POST) process begins.

4. Press **F2** on the keyboard during reboot to enter BIOS settings.

   The **System Setup** UI window opens. 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. Click **System BIOS** from the **System Setup Main Menu**.

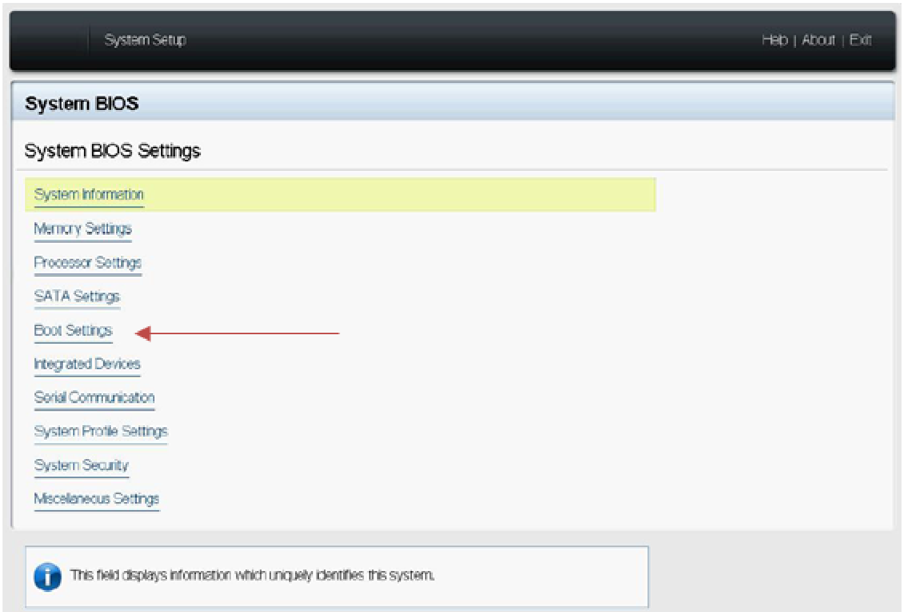
   Figure 21. System Setup Main Menu: Install CentOS



6. Change the boot settings to allow the node to boot from the CentOS DVD.

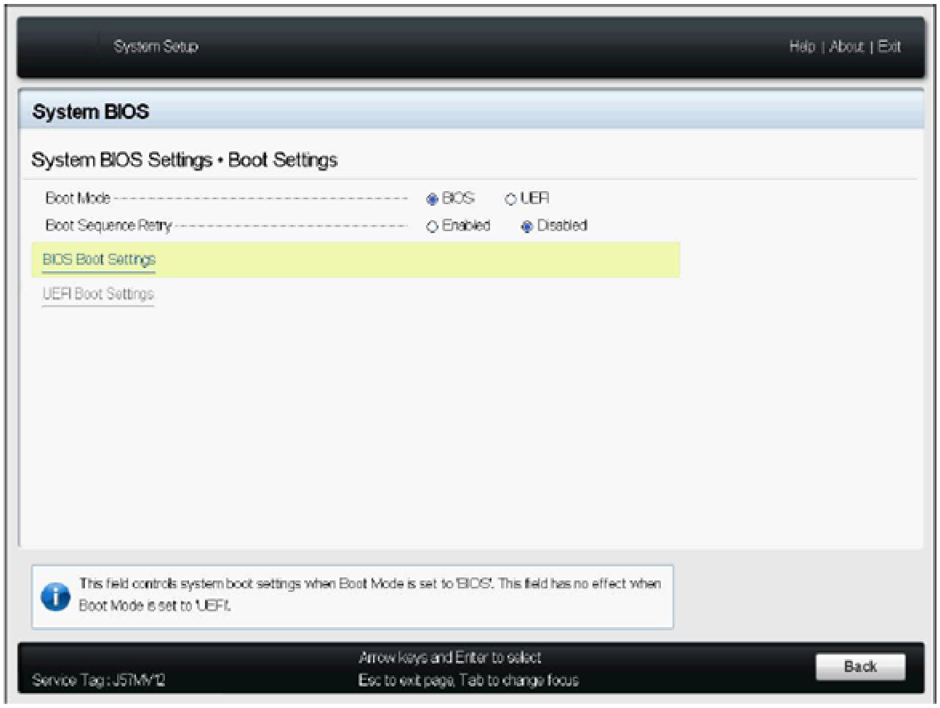
a. Click **Boot Settings** from the **System BIOS** menu.

Figure 22. System BIOS Menu: Install CentOS



b. Verify that **Boot Mode** is set to **BIOS**, and **Boot Sequence Retry** is **Disabled**, in the **Boot Settings** window.

Figure 23. System BIOS Boot Settings: Install CentOS



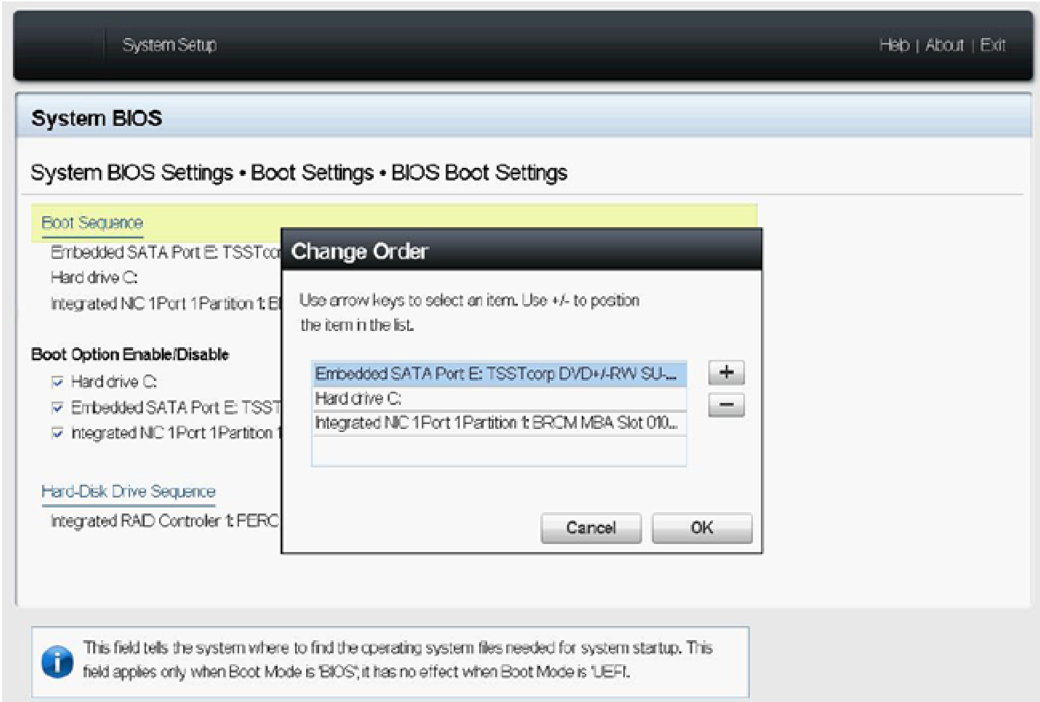
c. Click **BIOS Boot Settings** in the window.

d. Click **Boot Sequence** in the **BIOS Boot Settings** window.

   The **Change Order** pop-up window opens.

e. Modify the boot sequence of the server. 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 24. System BIOS Boot Sequence Order: Install CentOS



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.

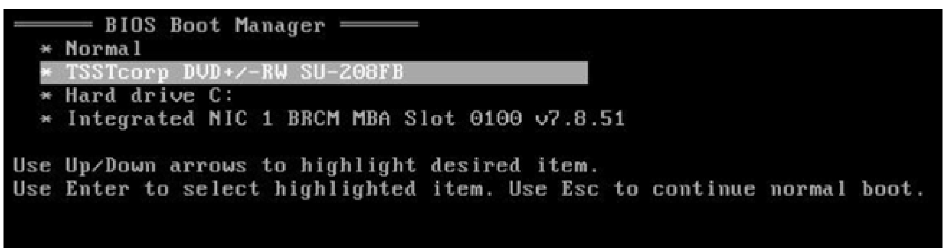
8. Restart the CMC node.

a. Press the **Esc** key on the keyboard to open the **Saving Changes** pop-up window.

b. Click **Yes** to save changes.

Figure 25. System BIOS Boot Settings, Save Changes: Install CentOS

c. (Conditional) Select the DVD as the boot source, if the CMC displays a boot manager menu.



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

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

Figure 26. CentOS-7 Boot Menu

The installer starts and the main install screen opens.

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. On the CMC, this is generally the `em1` interface or `eth0` interface.

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**.

    NOTE: If the **Save** button remains gray'ed out, look at the Network, Netmask, and Gateway site address entries and ensure the correct format.

e. Click within the **Hostname** field in the lower-left corner of the window, and enter the hostname for the CMC.

f. Switch the **On/Off** toggle to **On** in the upper-right corner of the window to enable the network. If desired, ping the machine from another host to verify that networking is correctly configured.

g. Click **Done** in the upper-left corner of the window to complete the site-network configuration.

11. (Conditional) Configure storage.

SKIP THIS STEP if your site is using default configuration. Perform this step only if your site needs non-default configuration.

If your site requires storage configuration that is not default, do the following:

a. Select **Installation Destination**.

b. Select the drives on which it is required to install CentOS and CSMS software.

c. Select **I will configure partitioning** if required to configure a non-default storage allocation.

   If configuring a non-default storage allocation the **Manual Partitioning** interface will be displayed.

d. Use the **Manual Partitioning** interface to configure the partitioning scheme designed during installation planning

12. Configure the Network Time Protocol (NTP).

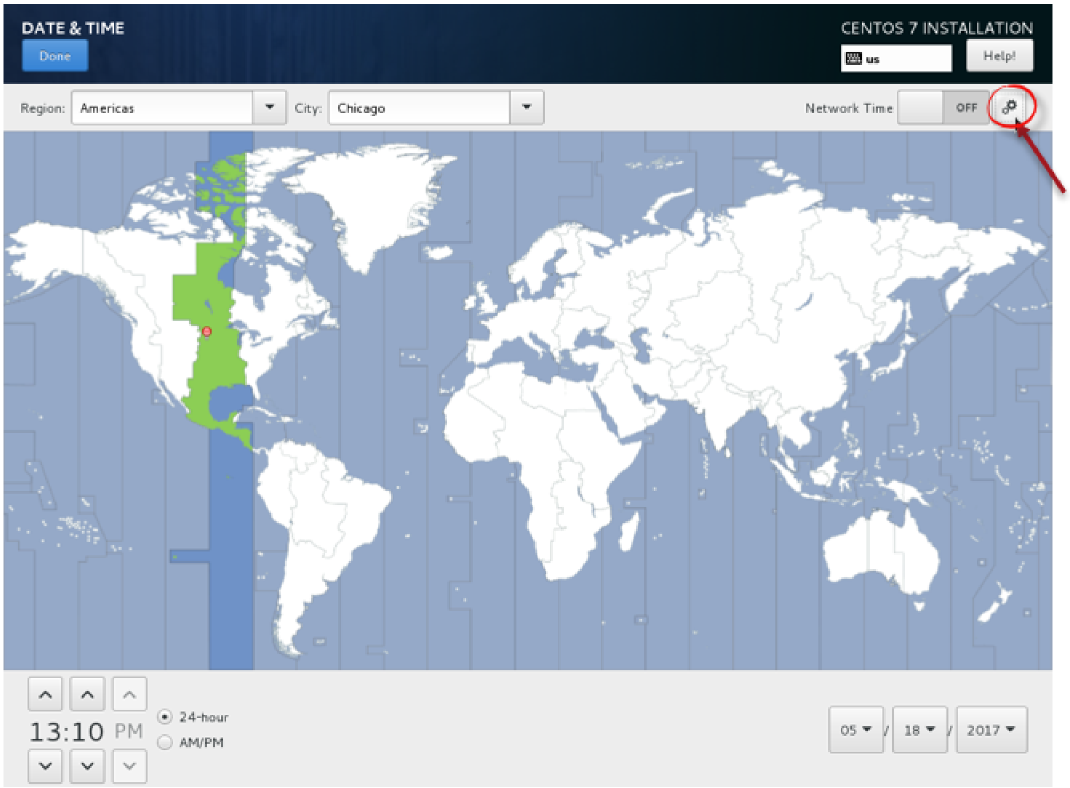
a. Select **DATE & TIME**.

b. Click the appropriate map location to set the time zone, and verify that it is correct.

c. Switch the **Network Time** toggle to **On** to enable NTP.

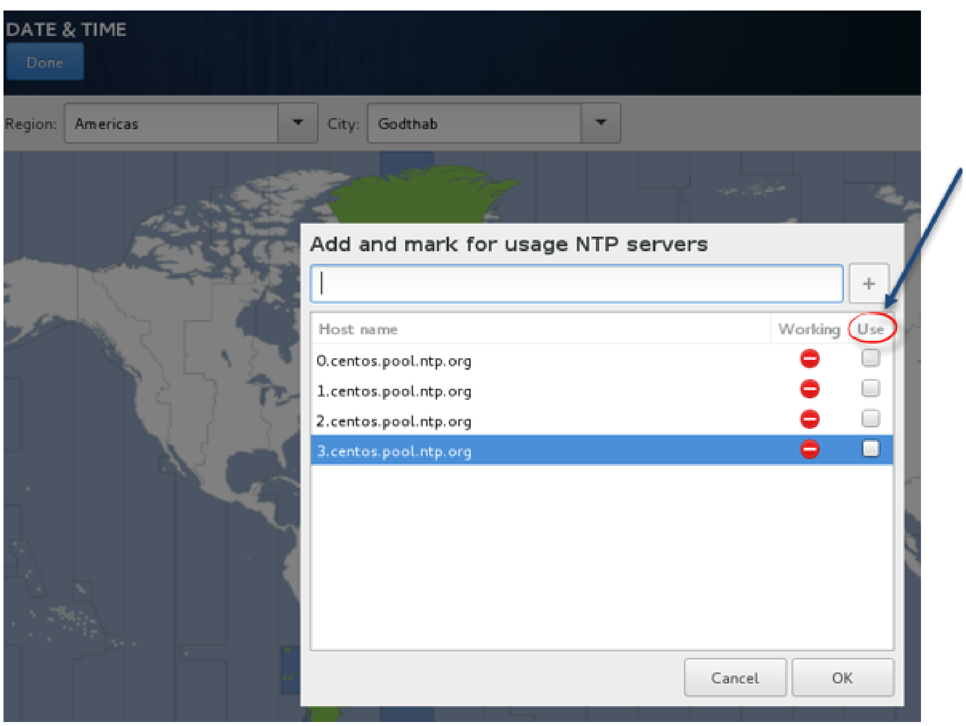
d. Click the gear icon to enter the NTP settings dialog.

Figure 27. Enter the NTP Settings Dialog by Clicking Gear Icon



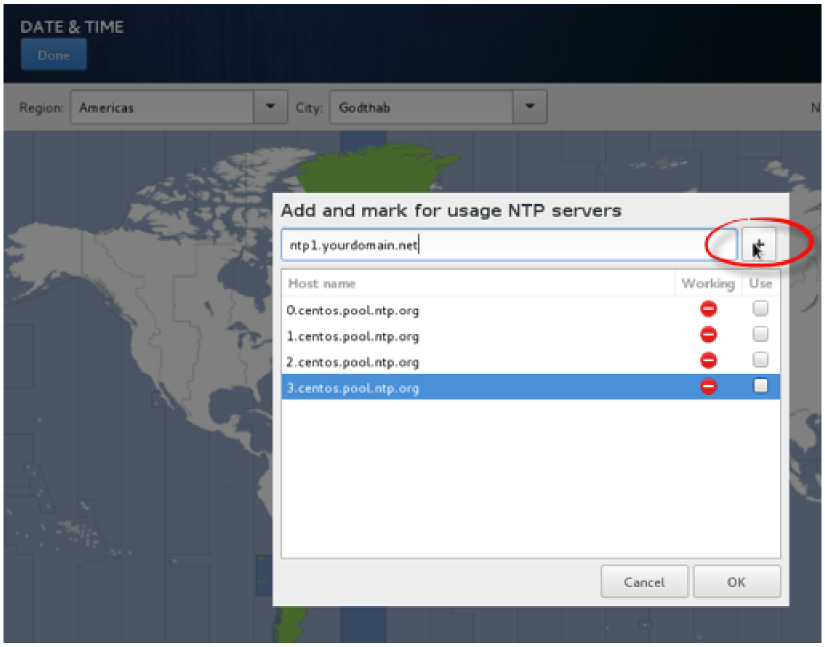
e. Uncheck the "**Use**" boxes to deselect all the default `ntp.org` servers.

Figure 28. Deselect Default ntp.org Servers by Unchecking "Use" Boxes



f. Click the "**+**" button to add your site's NTP server, and repeat for any additional servers.

Figure 29. Add Site's NTP Server and Any Additional Servers



g. Ensure the "**Use**" boxes are checked for your site's NTP servers added in previous substep.

h. Click **Ok** to exit the NTP settings dialog.

i. Click **Done** at the top left of the **DATE & TIME** screen.

13. Select **Begin Installation**.

The system finishes the install and reboots.

4.4 Install Cray System Management Software on CMC

Prerequisites

- CentOS 7 successfully installed on the Cray Management Controller (CMC).
- User has network access to the CMC via the site network.
- Assumes a first time install of the Cray System Management Software (CSMS). If performing an update,refer to the upgrade section.

About this task

This procedure describes how to install the CSMS on the CMC for eLogin nodes.

Replace all instances of localhost to the hostname of the site's machine.

Procedure

1. SSH to the CMC's IP address as the root user, entering `initial0` as the password.

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

2. (Optional) Start a typescript. (This step is optional, but recommended to assist in debugging and traceability of the installation.)

```
cmc# export TODAY=`date +%Y%m%d`
cmc# script -af ${TODAY}.insert_todays_topic.script_index
cmc# PS1="\u@\h:\w \t# "
```

3. Create a directory titled `isos` under the CMC `/root` directory.

```
cmc# mkdir -p /root/isos
```

4. Copy the CSMS and the Cray bootable CentOS ISOs to the CMC `/root/isos/` directory.

```
cmc# scp location of csms iso/csms_centos72-1.1.3-201608190116.iso /root/isos/
cmc# scp location of Centos iso/Cray-CentOSbase7-1511-201605031030.iso /root/isos/
```

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

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

5. Mount and invoke the CSMS installer. (Estimated process time is ~5 minutes.)

```
cmc# mount csms_centos72-1.1.3-201608190116.iso /mnt
cmc# cd /mnt
cmc# ./install.py
[installer runs]
```

6. Unmount the ISO root.

```
cmc# cd /root
cmc# 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 CMC `/root/isos/` directory.

```
root@cmc# scp location of elogin iso/elogin-6.0.3055-201701182038.iso /root/isos/
```

2. Change directory to the ISO root.

```
cmc# cd /root/isos
```

3. Mount and invoke the eLogin installation script. (*Estimated time: 1 minute*)

```
cmc# mount /root/isos/elogin-6.0.3055-201701182038.iso /mnt
cmc# cd /mnt
cmc# ./install.py
[installer runs]
```

4. Change directory and unmount the ISO root.

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

4.6 Configure CSMS On Management Controller

Prerequisites

Successful installation of:

- CentOS and Cray System Management Software (CSMS) on the Cray Management Controller (CMC)
- Cray eLogin software on the CMC

About this task

The procedure configures the CSMS on the CMC for eLogin.

Replace all instances of localhost to the hostname of the site's machine.

Procedure

1. Edit the file `/etc/opt/cray/openstack/ansible/hosts/hosts`, and replace all instances of localhost to the hostname of the site's machine. The top of this file contains two ocurrences of `localhost`, and the string `example-cmc` that must be replaced with the hostname of site CMC.

```
cmc# sed -i 's/^localhost/example-cmc/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 in: `/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 CMC on the Openstack management network. (For most sites, revert to the default.)

management_subnet

Override the subnet of the CMC 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 CMC on the IPMI network. (Current status: *Comment Out* all IPMI values.)

ipmi_subnet

Override the subnet of the CMC 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.

Set the vault password (secure) interactively by running the `csms_gen_creds.py` script with no arguments:

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_gen_creds.py
```

The `csms_gen_creds.py` script generates keys and credentials for multiple services. If your site's security requires to periodically regenerate keys (for example, Fuel), refer to *Change CSMS Passwords* in the *XC Series eLogin Administration Guide CLE 6.0 UP03 Rev C*, for more details.

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

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

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 step 4.

```
cmc# . /root/admin.openrc
Enter OpenStack password:
root@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`.

6. Apply Openstack Horizon branding changes.

Run the Ansible playbook commands to apply Cray branding changes to the Openstack Horizon web portal.

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_common.py -a horizon-branding.yaml
```

This command line invokes a secure password prompt.

7. 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.

8. 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 Kabana 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 CSMS 1.1.3 Patch Sets for Fresh Install

Prerequisites

- For system in process of a fresh install of CSMS 1.1.3
- Initial install of CSMS 1.1.3 is complete
- *Configure CSMS On Management Controller* is complete
- Root privileges
- Access to the CSMS 1.1.3 patch sets (PS01-PS04); refer to *Source ISO Images for eLogin*.

About this task

This procedure installs CSMS 1.1.3 patch sets (PS01-04) for systems in process of a fresh initial install of CSMS 1.1.3. Perform this procedure immediately after completion of *Configure CSMS On Management Controller*.

The CSMS 1.1.3 patch sets contain critical software fixes available for sites to install on a Cray Management Controller (CMC).

CSMS 1.1.3 requires four patch sets: PS01, PS02, PS03, and PS04. Each patch set is made up of three files:

- `CSMS_1.1.3.PS##.readme`: Contains information regarding issues addressed by the patch set and install preparation instructions.
- `CSMS_1.1.3.PS##.install`: Contains instructions for installing fixes in the CSMS software.
- `CSMS_1.1.3.PS##.iso`: Contains the rpm's and source files for the CSMS fixes and updates.

The CSMS 1.1.3 patch sets are identified by a combination of the CSMS release number and patch set ID (for example, `1.1.3.PS01`). The patch set files are released in a directory of that name.

> **IMPORTANT:** Do not use the instructions listed in either the `.install` or `.readme` files to install the CSMS patch sets. These instructions are replaced with shell scripts to simplify the install process.

For detailed information regarding fixes included in a specific patch set, refer to the `CSMS_1.1.3.PS##.readme` file.

Procedure

1. Create a directory on the CMC, and copy the four patch set directories and prep/install shell scripts to the same directory.

```
cmc# mkdir -p /root/CSMS113PS

# scp -r 1.1.3.PS01 1.1.3.PS02 1.1.3.PS03 1.1.3.PS04 \
root@cmc:/root/CSMS113PS/

cmc# cd /root/patch_scripts

cmc# cp CSMS-psprep.sh CSMS113_PS01_03_install.sh CSMS113_PS04_install.sh \
/root/CSMS113PS/
```

2. Change directory to where the patch sets were copied.

```
cmc# cd /root/CSMS113PS
```

3. Prepare the four patch sets for installation.

```
cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS01
cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS02
cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS03
cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS04
```

4. Install patch sets PS01 through PS03 using the `CSMS113_PS01_03_install.sh` script.

```
cmc# ./CSMS113_PS01_03_install.sh
```

Patch sets PS01-PS03 are now installed.

5. Install patch set PS04 using the `CSMS113_PS04_install.sh` script, including the command-line parameter of the *`IP address of the host, to which logs are to be forwarded`*.

```
cmc# ./CSMS113_PS04_install.sh 111.222.333.444
```

Patch set PS04 is installed. The install for all CSMS 1.1.3 patch sets (PS01-PS04) is now complete.

4.8 Configure Connection Between SMW and CMC

Prerequisites

- Install of CentOS on CMC.
- Install CSMS on CMC
- Configuration of CSMS on CMC
- Install of SMW/CLE software on SMW
- Install of eLogin ISO (default eLogin image recipe)

About this task

This procedure configures the connection between the System Management Workstation (SMW) and 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
Vault password:
```

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 and CMC Connection

3. Create an environment variable for the CMC name (`$CMCNAME`).

```
smw# CMCNAME=cmc-name
```

4. Log on to the SMW, and add the site-admin network address of the CMC as an entry to `/etc/hosts`. The IP is the address of the CMC as seen from the SMW. (This is the site IP address given for the CMC on the site-admin network.)

```
cmc# ssh smw

smw# echo "IP_address $CMCNAME" >> /etc/hosts
```

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

5. Test the connectivity between the SMW and CMC.

```
smw# ping -c10 $CMCNAME
```

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

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

```
smw# ssh-keyscan -H $CMCNAME >> ~/.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 $CMCNAME
```

Setup IMPS for Keystone and Glance Integration

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

```
smw# scp -p $CMCNAME:admin.openrc /root/
```

9. Copy the `OS_CACERT` certificate file (``` ``/etc/ssl/public_api.cert`` ```), and reconfigure the `admin.openrc` file on the SMW. SSL is enabled on the CMC by default.

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

```
smw# scp -p $CMCNAME:/etc/ssl/public_api.cert /root/
```

b. Edit 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 Image Management and Provisioning System (IMPS) to read and query information regarding Glance service endpoints.

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

11. Create the Keystone and Glance endpoint connections.

The `keystonecon create` command creates both the Keystone and Glance endpoints.

```
smw# keystonecon create $CMCNAME --env
```

12. Verify the connection registration for the Keystone and Glance endpoints.

```
smw# keystonecon list
NAME          URL                               OS_USERNAME
example-cmc#  https://172.30.50.129:5000/v2.0   admin

smw# glancecon list
NAME         URL                          KEYSTONE_CONN
example-cmc  https://172.30.50.129:9292   example-cmc

smw#
```

4.9 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*.

4.9.1 Configure eLogin Hardware for Deployment

Prerequisites

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

About this task

The eLogin hardware must be configured before deploying the eLogin image.

The eLogin software requires two drives:

- sda: The operating system and swap
- sdb: Persistent storage

Procedure

1. Boot the eLogin hardware.
2. Open the BIOS RAID configuration boot menu, by using one of the following methods:
   - Press **Ctrl-R** on the keyboard.

   Or

   - Use `ipmitool` from the CMC, if access to BIOS RAID setup from the eLogin hardware's physical console is not available, or not practical.
     1. Open an SSH session to connect to the CMC.

        ```
        # ssh root@cmc
        cmc#
        ```
     2. Activate the console.

        ```
        cmc# ipmitool -I lanplus -U root -P initial0 -H drac ip sol activate
        ```
     3. Open a separate window (SSH session) to connect to the CMC.

        ```
        # ssh root@cmc
        cmc#
        ```
     4. Run the `ipmitool` power reset command to restart the node to access the disk configuration menu in the second window.

        ```
        cmc# ipmitool -I lanplus -U root -P initial0 -H drac ip power reset
        ```

        NOTE: The appearance of the BIOS RAID configuration screens may vary when accessing via `ipmitool` vs. the physical console, but the functionality is the same.

Figure 30. Initial Boot Menu for BIOS RAID Configuration: eLogin

The RAID configuration screen opens.

Figure 31. RAID Configuration Screen: eLogin

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

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

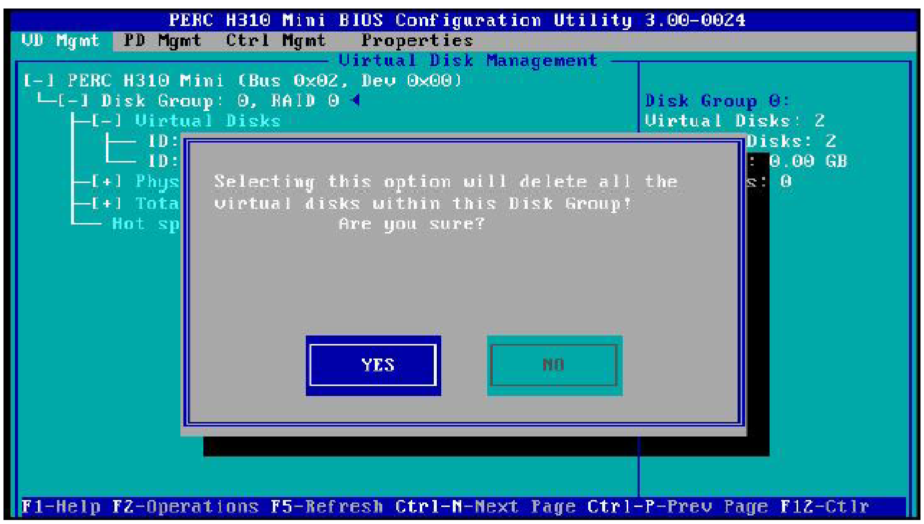
a. Select the disk.

b. Press **F2** key to get a list of operations.

c. Select **Delete Disk Group** and press **Enter**.

Figure 32. Delete Disk Group: eLogin BIOS RAID Setup

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



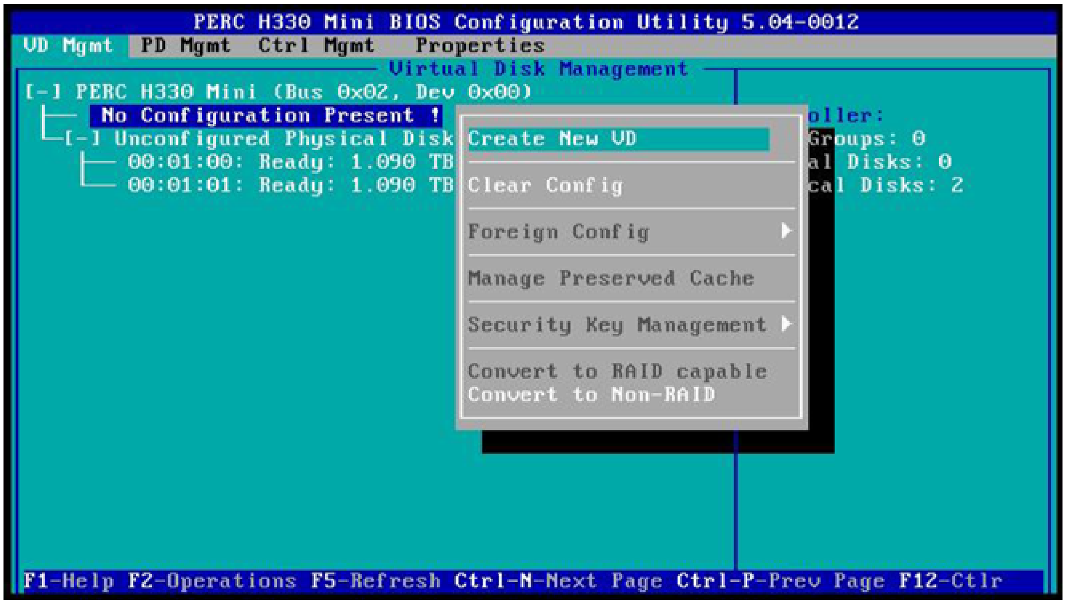
4. Create a new virtual disk A.

a. In the virtual disk (VD) 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 33. Create Virtual Disk A: eLogin BIOS RAID



The Create New VD window opens.

5. Move the cursor to select the disk ID in the Create New VD window, and then add disk to RAID by pressing spacebar on keyboard.

6. Set the RAID Level to **RAID 0**.

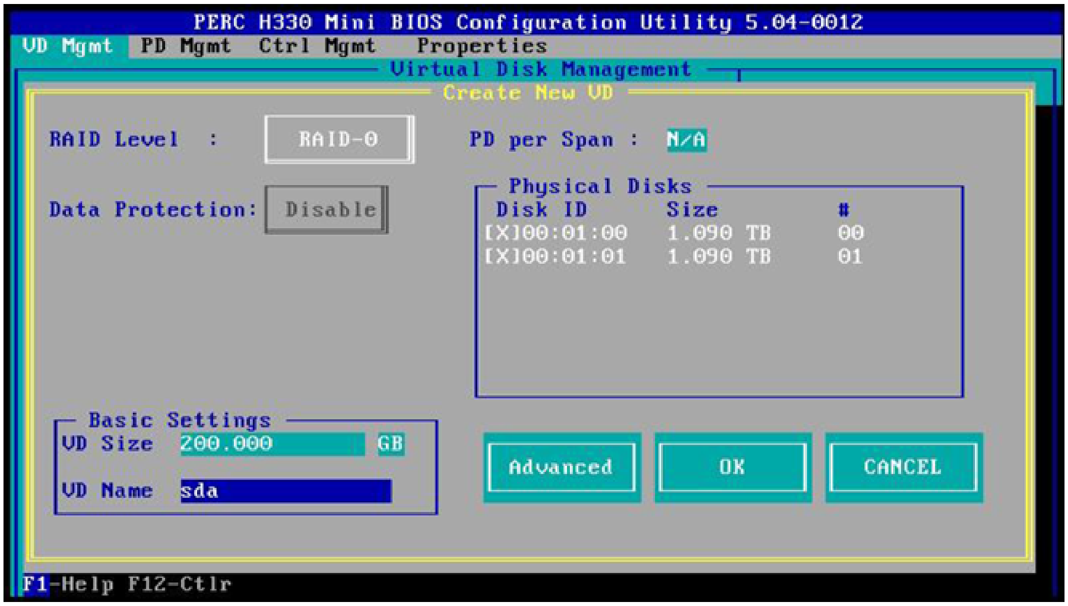
Figure 34. Add Disk to RAID Level 0: eLogin

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

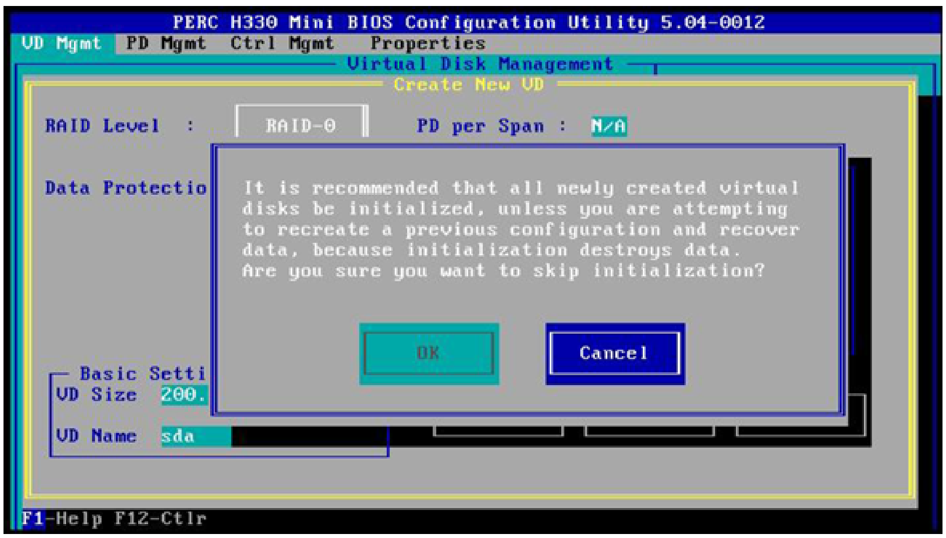
a. Set the **VD Size** for virtual disk A to **200 GB** of disk space.

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

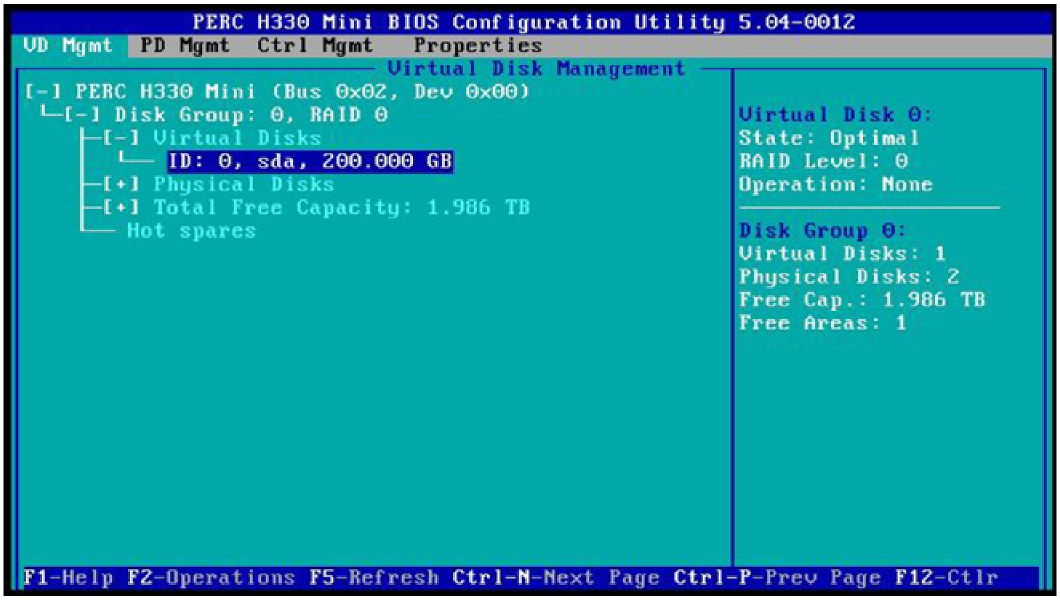
Figure 35. Disk Size and Name Setting for Virtual Disk A: eLogin



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



The sda is now created.



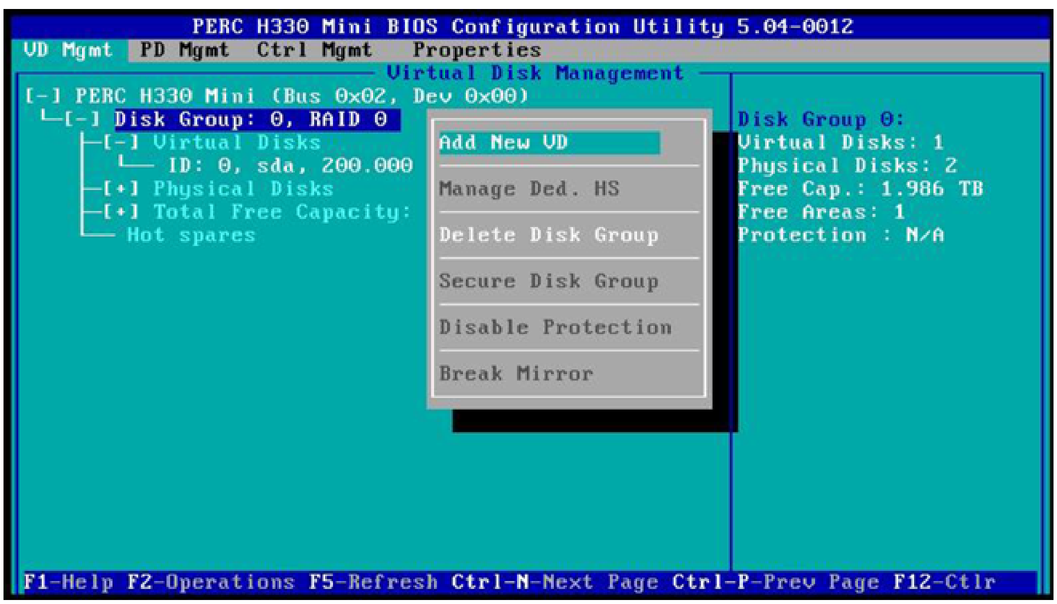
8. Create a new virtual disk B.

a. In the Virtual Disk Management window, navigate to **Disk Group: 0 RAID-0** using the keyboard up/down arrows.

b. Press **F2** to access the disk creation menu.

c. Select **Add New VD**.

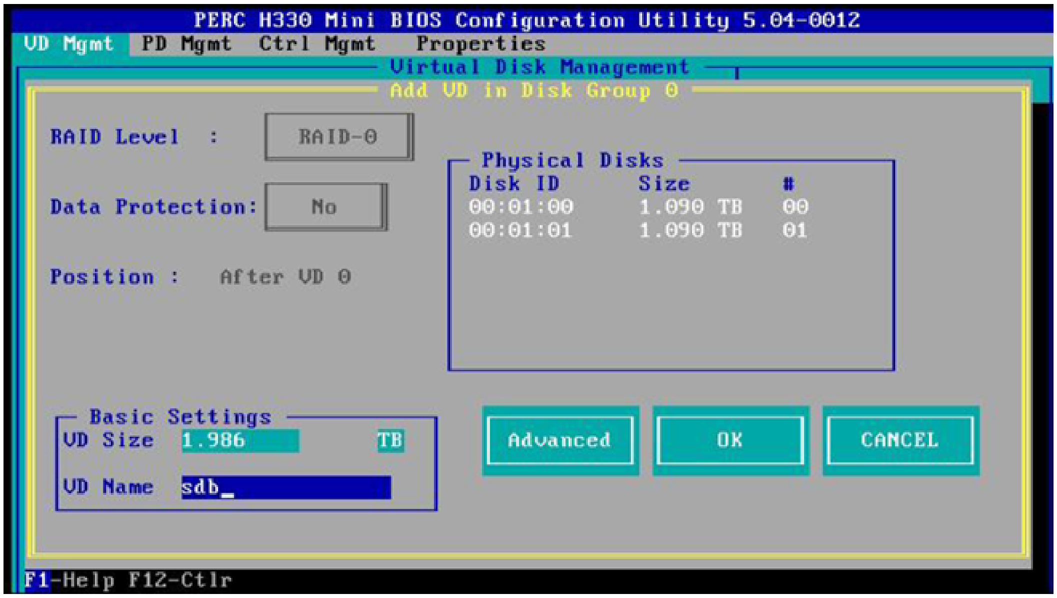
Figure 36. Create New Virtual Disk B: eLogin BIOS RAID



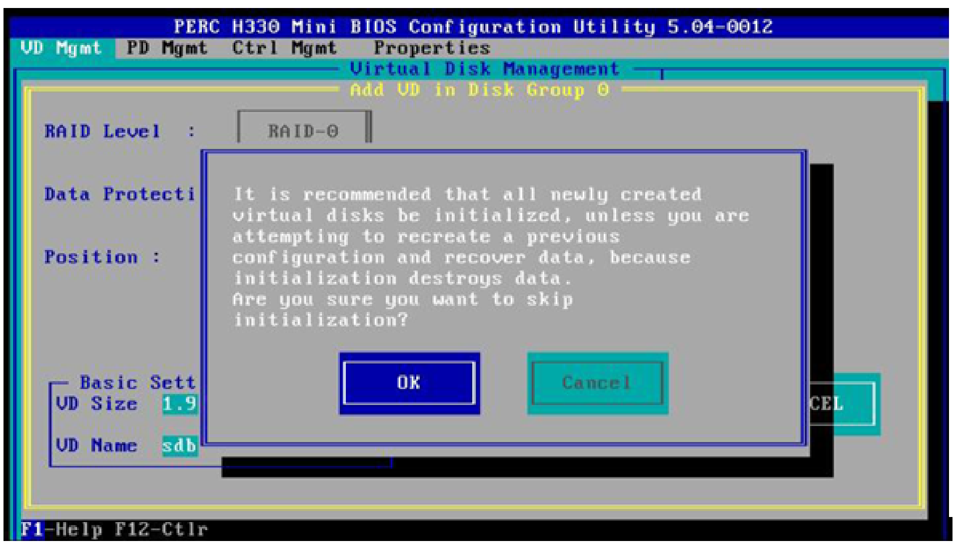
The **Add VD in Disk Group 0** window opens.

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

Figure 37. 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**.

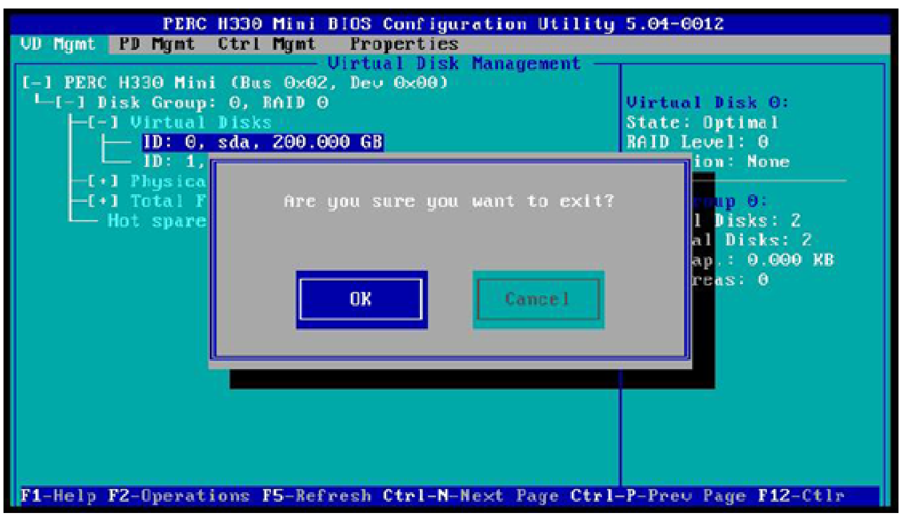


Two virtual disks are now available.

Figure 38. 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 39. Exit BIOS Configuration: eLogin



The BIOS configuration utility screen is now closed.

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

4.9.2 Configure Minimum Services Required for eLogin

Prerequisites

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

About this task

The boot process uses standard configuration templates to convey configuration information to the eLogin image. The system administrator is expected 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 Linux Environment (CLE) 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. Verify that the configuration interface values match your site hardware interface type, as one of the following:

   4x-1GbE LOM

   eLogin OpenStack management interface is on (`eth0`); the site network interface is on (`eth1`).

   2x-10GbE / 2x-1GbE LOM

   eLogin OpenStack management interface is on (`eth2`); the site network interface is on (`eth3`)

2. Configure `cray_net` within the CLE config set. Cray networking sets up network connections for eLogin nodes to the CMC, the site network, and LNet.

   The command used in this step is `cfgset update` (auto mode). To add a new entry for networks, hosts, and interfaces, use carriage returns <`cr`> to navigate (skip current entries) to the line. For example:

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

   When configuring `cray_net`, ensure the following are correct:

   - For these multi-value settings, the key must exactly match the value shown in the example.
   - In the configuration settings for the common name (`cray_net.settings.hosts.data.common_name`): If your eLogin hostname contains a hyphen,

such as *example-elogin*, you must use an underscore rather than a hyphen for the common name key. Only use the underscore for the hostname (*example_elogin* for the purposes of this key.

Run the following command line to configure cray_net within the CLE config set. (The example shown is for management interface (eth0), and site network interface (eth1.)

If your site hardware interface is (2x-10GbE / 2x-1GbE LOM), configure the management interface to (eth2), and site network interface to (eth3) respectively.

```
smw# cfgset update -s cray_net -S all -m auto 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] $ 10.142.0.0

cray_net.settings.networks.data.openstack_mgmt.ipv4_netmask
[<cr>=set '', <new value>, ?=help, @=less] $ 255.255.0.0

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

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] $ elogin_IP_network

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

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

**HOSTS**

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] $ elogin_name

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

cray_net.settings.hosts.data.eLogin_name.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>
```

3. Verify that the network connections for eLogin node values were configured correctly.

```
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 eLogin_name -s cray_net config_set
cray_net.settings.hosts.data.eLogin_name.description: eLogin Node
cray_net.settings.hosts.data.eLogin_name.aliases: # (empty)
cray_net.settings.hosts.data.eLogin_name.roles: # (empty)
cray_net.settings.hosts.data.eLogin_name.hostid: # (empty)
cray_net.settings.hosts.data.eLogin_name.host_type: # (empty)
cray_net.settings.hosts.data.eLogin_name.hostname: example-elogin
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.description: OpenStack
eth0
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.aliases: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.network: openstack_mgmt
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.ipv4_address: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.bootproto: dhcp
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.mtu: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth0.extra_attributes: #
(empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.name: eth1
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.description: Site eth1
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.aliases: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.network: site
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.ipv4_address:
IP_address
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.bootproto: static
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.mtu: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.eth1.extra_attributes: #
(empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.description: IB to
External Lustre
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.aliases: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.ipv4_address:
eLogin_LNet_address
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.mtu: # (empty)
cray_net.settings.hosts.data.eLogin_name.interfaces.ib0.extra_attributes: #
(empty)
```

4. 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.

5. Verify that the eLogin networking postfix relay information was configured correctly.

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

6. 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>
```

7. Verify that the eLogin LNet information was configured correctly.

```
smw# cfgset search -t cray_elogin_lnet -l basic -S config_set \
cray_elogin_networking.settings.elogin_networking.data.example-elogin.cray_elogin_lnet:cims-mgmt
```

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

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_eproxy -S all config_set

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

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

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

cray_eproxy.settings.eproxy_map.data.login_node_hostname.elogin_hosts
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add elogin_hosts (Ctrl-d to exit) $ elogin_hostname
Add elogin_hosts (Ctrl-d to exit) $ <Ctrl-d>

cray_eproxy.settings.eproxy_map.data.login_node_hostname.elogin_hosts
[<cr>=set 1 entries, +=add an entry, ?=help, @=less] $ <CR>
```

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

9. Verify that the `eproxy` service information is configured correctly.

```
smw# cfgset search -t cray_eproxy -S config_set \
cray_elogin_networking.settings.elogin_networking.data.example-elogin.cray_eproxy:cims-mgmt
```

10. (Conditional) Enable `aprun` wrapping, for systems without a workload manager.

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

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

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

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

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

11. Update entire config set.

```
smw# cfgset update config_set
```

This update runs all pre- and post-config scripts, and refreshes the squash config set used in config set caching. Updating the config set is recommended as good practice when any config set services are changed.

12. Optional: Configure 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) $ elogin1
Add members (Ctrl-d to exit) $ elogin2
Add members (Ctrl-d to exit) $ elogin3
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
```

13. Configure `cray_elogin_motd` within the config set.

Cray MOTD provides the ability to enable or disable the automatic creation of the `/etc/motd` file on each eLogin node. The default is to generate the `/etc/motd` file. This example enables automatic generation of `/etc/motd` for `elogin1` and disables it for `elogin2`.

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

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

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

cray_elogin_motd.settings.elogin_motd.data.hostname
[<cr>=set '', <new value>, ?=help, @=less] $ elogin1

cray_elogin_motd.settings.elogin_motd.data.elogin1.cray_managed
[<cr>=set 'true', <new value>, ?=help, @=less] $ true

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

cray_elogin_motd.settings.elogin_motd.data.hostname
[<cr>=set '', <new value>, ?=help, @=less] $ elogin2

cray_elogin_motd.settings.elogin_motd.data.elogin2.cray_managed
[<cr>=set 'true', <new value>, ?=help, @=less] $ false

cray_elogin_motd.settings.elogin_motd
[<cr>=set 2 entries, +=add an entry, ?=help, @=less] $ <cr>
```

a. Verify the new `cray_elogin_motd` settings.

```
smw# cfgset search -s cray_elogin_motd --level basic config_set

INFO - Checking services for valid YAML syntax
INFO - Checking services for schema compliance
# 2 matches for '.' from cray_elogin_motd_config.yaml
#--------------------------------------------------------------------------------
----
cray_elogin_motd.settings.elogin_motd.data.elogin1.cray_managed: true
cray_elogin_motd.settings.elogin_motd.data.elogin2.cray_managed: false
```

14. Confirm that the config set is valid.

```
smw# cfgset validate config_set
```

4.9.3 Enroll an Ironic Node

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 on to the CMC as root.

2. Define the bare metal nodes.

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

   Only one of the file formats should be used. Choose either the CSV or YAML format inventory file and copy it to `/etc/opt/cray/openstack/ansible/` as either `inventory.csv` or `ironic_inventory.yaml`, respectively. Instructions for defining bare metal nodes in the inventory file are included in the comments at the top of the example inventory file being used. These files also provide information on the available fields. The inventory file will vary depending on the Ironic driver used, use cases and system configuration. The inventory file will also vary depending on whether hardware inspection is enabled:

   - If enabled, only a minimal inventory is required, as many hardware node attributes are discovered automatically.
   - If disabled, the inventory needs to contain all information required by Ironic.

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

   ```
   cmc# vi /etc/opt/cray/openstack/ansible/inventory.csv
   ```

   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
   elogin1,bmc_ip,mac_addr,n_cpus,x86_64,ram_mb,disk_gb,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.

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

The following command line is an example only:

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

Where:

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

Run the following command line using `nova flavor-create` to set the minimum eLogin hardware requirements. Use the same values entered for the inventory file in step 2a.

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

4.9.4 Configure OpenStack Fuel

Prerequisites

- Perform as root on the Cray Management Controller (CMC).
- A successful installation and configuration of CSMS on the CMC.

Procedure

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

```
cmc# . /root/admin.openrc
```

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
```

4.9.5 Change the Apache Kafka Data Retention Policy

Prerequisites

This procedure requires root privileges.

About this task

Kafka's data retention policy can be configured after installation. This requires setting values for some or all of the following variables in an override file, and then running the relevant Ansible play:

- `log_retention_hours`
- `log_retention_bytes`
- `log_segment_bytes`

Procedure

1. Log on to the management controller as root.
2. Add or change the required variables in the site override file, which is located under `/etc/opt/cray/openstack/ansible/config/site`.

```
cmc# echo "log_retention_hours: 12" >> /etc/opt/cray/openstack/ansible/config/site/local-kafka-settings.yaml
```

3. Run the `kafka.yaml` playbook to apply any changes and to restart Kafka.

```
cmc# cd /etc/opt/cray/openstack/ansible
cmc# ./csms_common.py -a kafka.yaml
```

4. Verfiy that Kafka has restarted.

```
cmc# systemctl status kafka
```

5. (Conditional) If Kafka has not started, try to restart Kafka again using the following command:

```
cmc# systemctl restart kafka
```

6. Optional: View the Kafka settings file's content to verify that it contains the expected values for the `log.retention.hours`, `log.retention.bytes` and `log.segment.bytes` parameters.

```
cmc# cat /etc/kafka/server.properties
```

5 eLogin Remote Access Controller Configuration

Prerequisites

Access to the console for each CDL being configured.

About this task

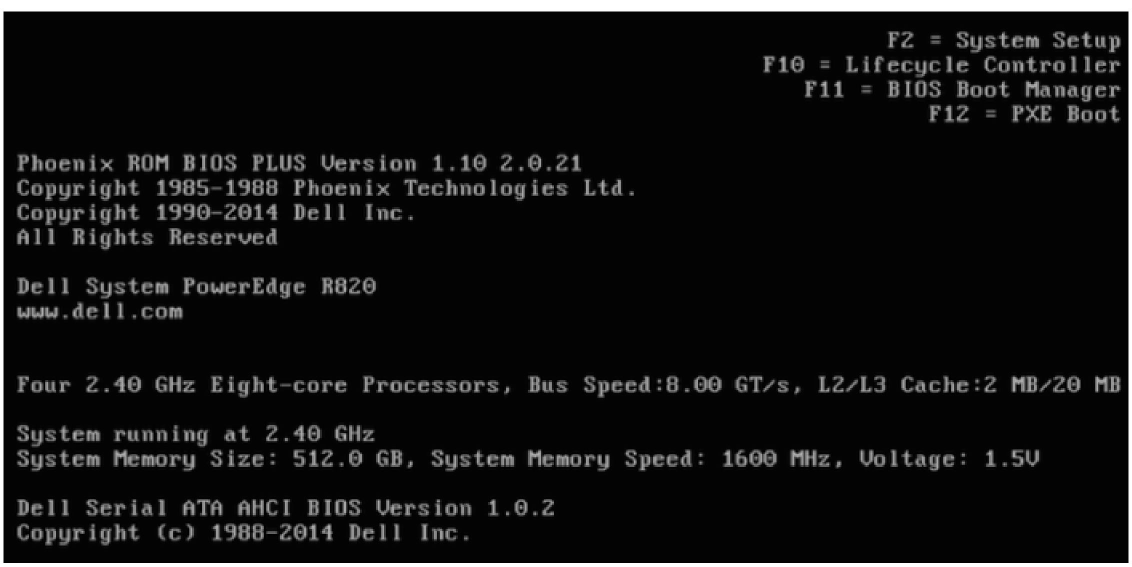
Before configuring a node, the BIOS and remote access controller (iDRAC) settings must be changed. Several of the following BIOS settings are required for `ironic_conman` (a console manager), to function properly.

Procedure

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

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

   Figure 40. Dell R820 BIOS Power-On Self-Test Menu Screen

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

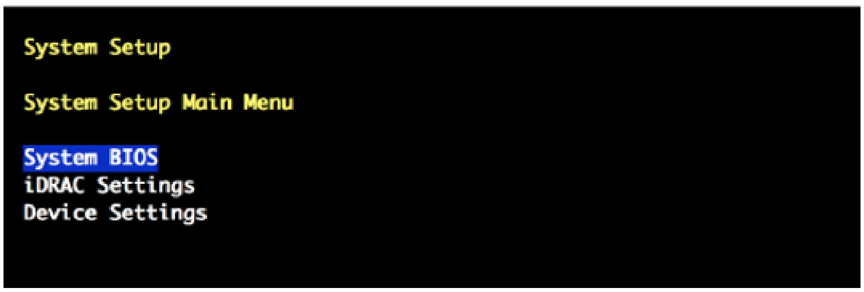
After the POST process completes and all disk and network controllers are initialized, the **Dell System Setup Main Menu** screen appears with the following sub-menus:

- System BIOS
- iDRAC Settings
- Device Settings

2. Change the system BIOS settings.

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

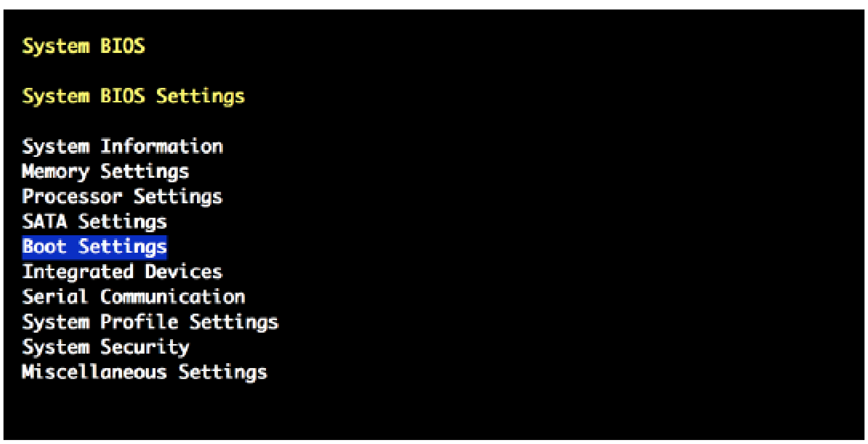
Figure 41. System Setup Main Menu: Select System BIOS



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

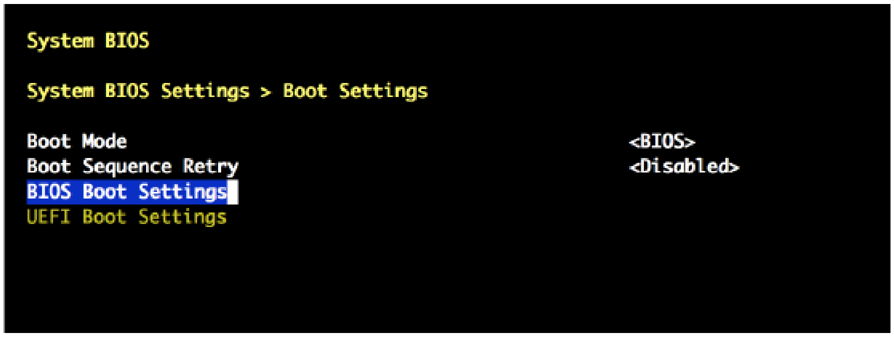
b. Select **Boot Settings** from the **System BIOS Settings** screen, then press **Enter**.

Figure 42. System BIOS Settings: Boot Settings



c. Select **BIOS Boot Settings**, then press **Enter**.

Figure 43. Boot Settings: BIOS Boot Settings



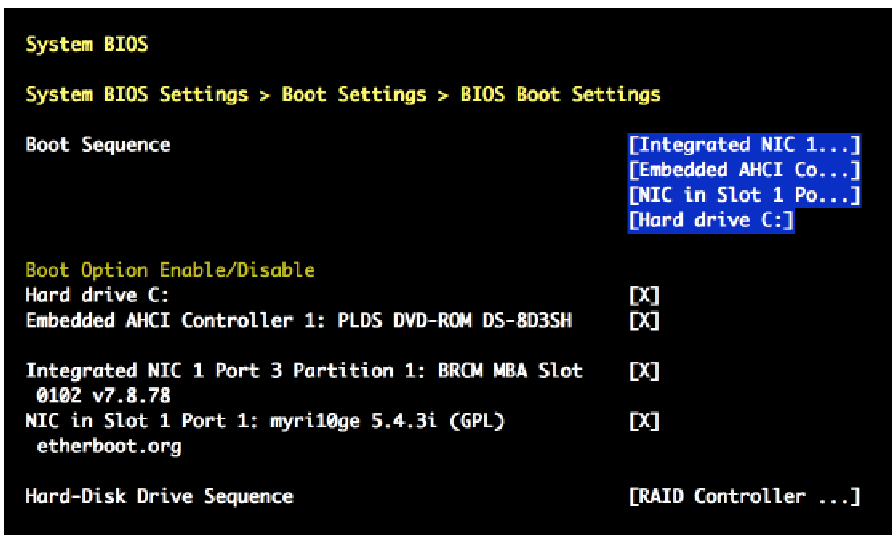
d. Select **Boot Sequence**, then press **Enter** to view the boot settings.

e. Set the boot sequence.

1. Change the boot order so that the **Integrated NIC** appears first, before the optical (DVD) drive. The hard drive must be last on the list.

2. Ensure that the **Integrated NIC Port** is enabled so the node can PXE boot from the CMC (embedded NIC) before attempting to boot from the local disk.

Figure 44. BIOS Boot Settings: Set Boot Sequence



f. Press **Enter** to return to the **BIOS Boot Settings** screen.

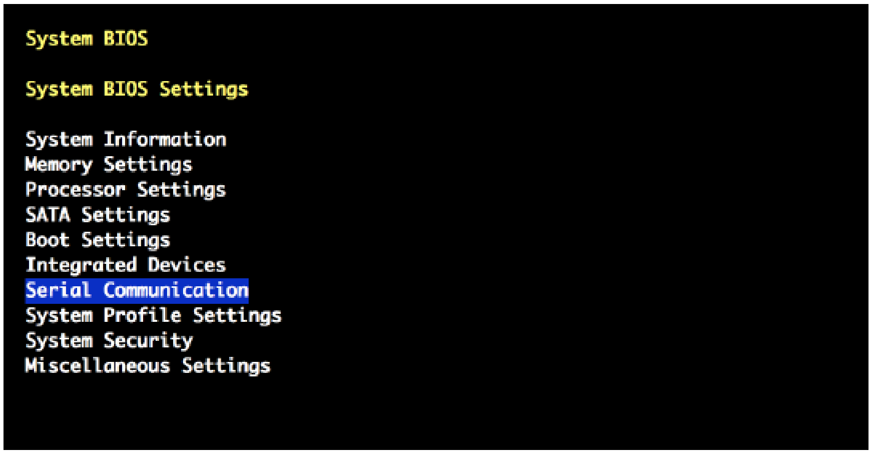
g. Press **Escape** to exit **BIOS Boot Settings**.

h. Press **Escape** to exit **Boot Settings** and return to the **System BIOS Settings** screen.

3. Change the serial communication settings.

a. On the **System BIOS Settings** screen, select **Serial Communication**.

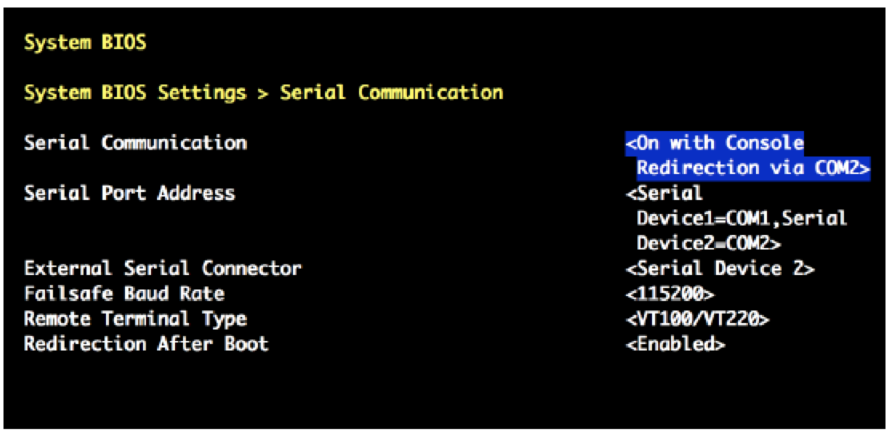
Figure 45. System BIOS Settings: Select Serial Communication



b. On the **Serial Communication** screen, select **Serial Communication**. A pop-up window displays the available options.

c. Select **On with Console Redirection via COM2**, then press **Enter**.

Figure 46. Serial Communication: Select Console Redirection via COM2



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

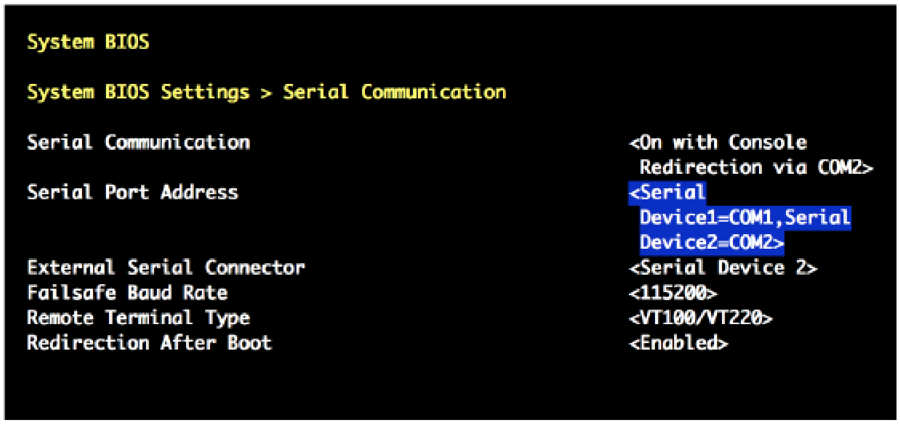
> **NOTE:** This setting enables the remote console. If this setting is incorrect, you cannot use a remote console to access the node.

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

1. Press **Enter** to display the available **Serial Port Address** options.

2. Change the setting to: `Serial Device1=COM1`, `Serial Device2=COM2`.

Figure 47. Serial Communication: Serial Port Address

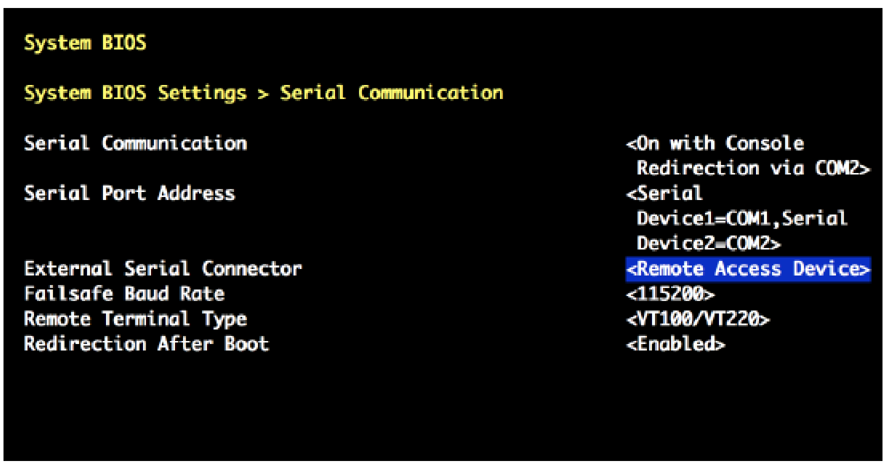


3. Press **Enter** to return to the **Serial Communication** screen.

e. Select **External Serial Connector**. A pop-up window displays the available options.

f. In the **External Serial Connector** pop-up window, select **Remote Access Device**, then press **Enter** to return to the previous screen.

Figure 48. External Serial Connector: Select Remote Access Device



g. Select **Failsafe Baud Rate**. A pop-up window displays the available options.

h. Select `115200` for the **Failsafe Baud Rate** in the pop up window, and then press **Enter** to return to the previous screen.

Figure 49. Serial Communication: Select 115200 Failsafe Baud Rate

i. Press the **Escape** key to exit the **Serial Communication** screen.

j. Press the **Escape** key to exit the **System BIOS Settings** screen.

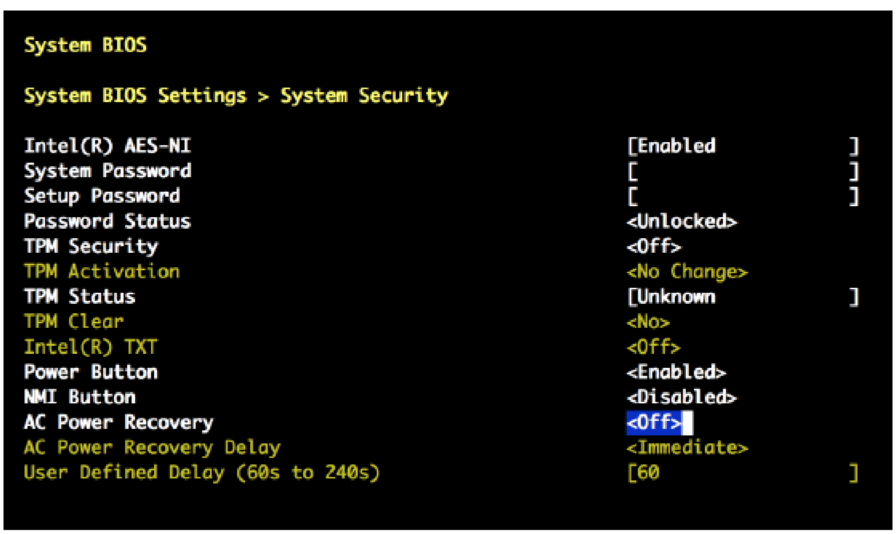
k. Press the **Escape** key to exit the **BIOS Settings** screen.

l. When the "`Settings have changed`" message appears, select **Yes** to save your changes.

m. When the "`Settings saved successfully`" message appears, select **Ok**.

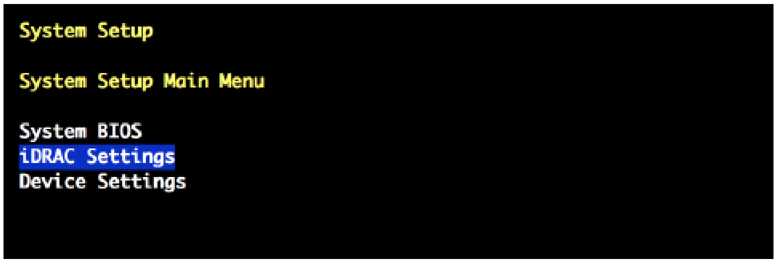
4. Open the **System BIOS Settings** menu, select **System Security**, and then **AC Power Recovery** to set the node to remain powered `Off` after a system power failure. Cray recommends that the CMC node power up and become operational before all client nodes.

Figure 50. System Security: AC Power Recovery Off



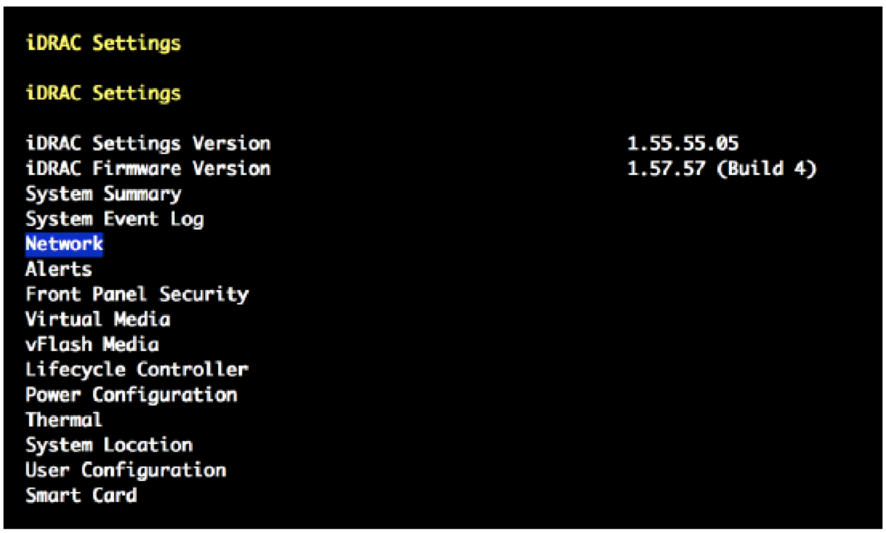
5. Select **iDRAC Settings** on the **System Setup Main Menu** screen, then press **Enter**.

Figure 51. System Setup Main Menu: iDRAC Settings



6. Select **Network** from **iDRAC Settings** screen, then press **Enter**. A long list of network settings is displayed.

   Figure 52. iDRAC Settings: Network



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

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

   b. Ensure that **IPMI over LAN** (or **Enable IPMI over LAN**) is `Enabled`.

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

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

Figure 53. Network IPMI SETTINGS: Enable IPMI over LAN

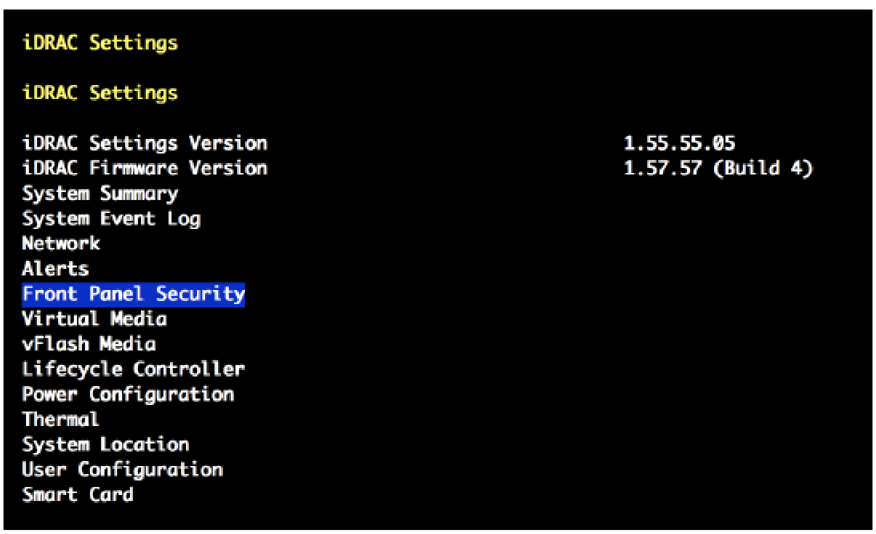
3. Press **Enter** to return to the previous screen.

c. Press the **Escape** key to exit the **Network** screen, and return to the **iDRAC Settings** menu.

8. Change the LCD configuration to show the hostname in the LCD display.

a. On the **iDRAC Settings** screen, scroll down using the `down-arrow` key to **LCD** (or **Front Panel Security**), and then press **Enter**.

Figure 54. iDRAC Settings: Front Panel Security



b. Select **Set LCD message**. A pop-up window opens.

c. Select **User-Defined String** in the pop-up window, and then press **Enter**.

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

Figure 55. ***Front Panel Security: User Defined String***

e. Enter the hostname (such as, `elogin1`) in the text pop-up window.

f. Press the **Escape** key to exit the **Set LCD message** screen.

g. Press the **Escape** key to exit the **Network** screen.

h. Press the **Escape** key to exit the **iDRAC Settings** screen.

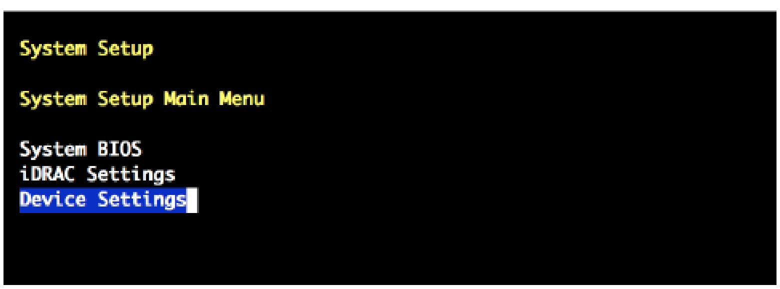
i. When the "`Settings have changed`" message appears, select **Yes** to save your changes.

j. When the "`Settings saved successfully`" message appears, select **Ok**, and then **Enter**.

9. Change the device settings so that the node can PXE boot from the CMC administration network (`maint-net`).

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

Figure 56. System Setup Main Menu: Device Settings

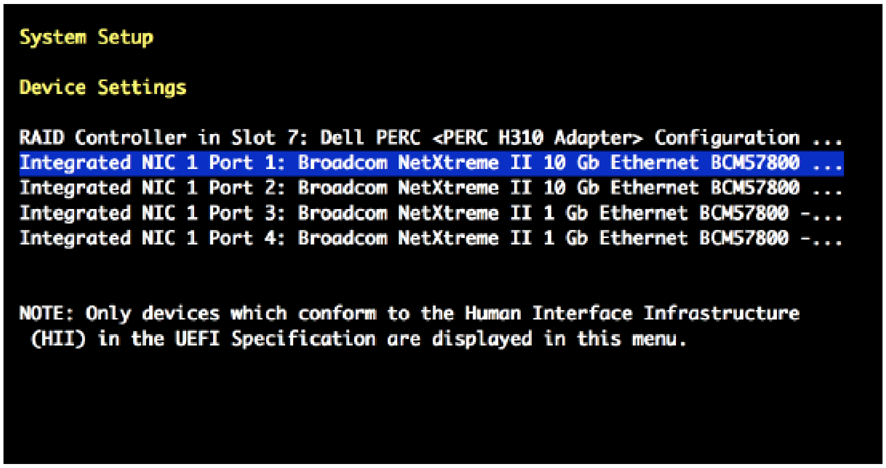


b. Select **Integrated NIC 1 Port N ...** on the **Device Settings** screen, then press **Enter**. The **Main Configuration Page** opens.

Choose the NIC port number that corresponds to the Ethernet port for the `maint-net` network:

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

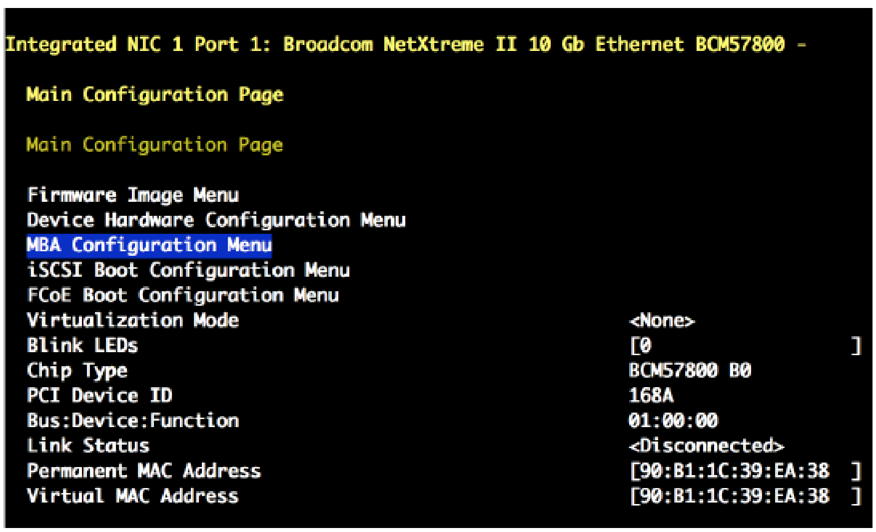
Figure 57. Main Configuration Page: Select Integrated NIC 1 Port #



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

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

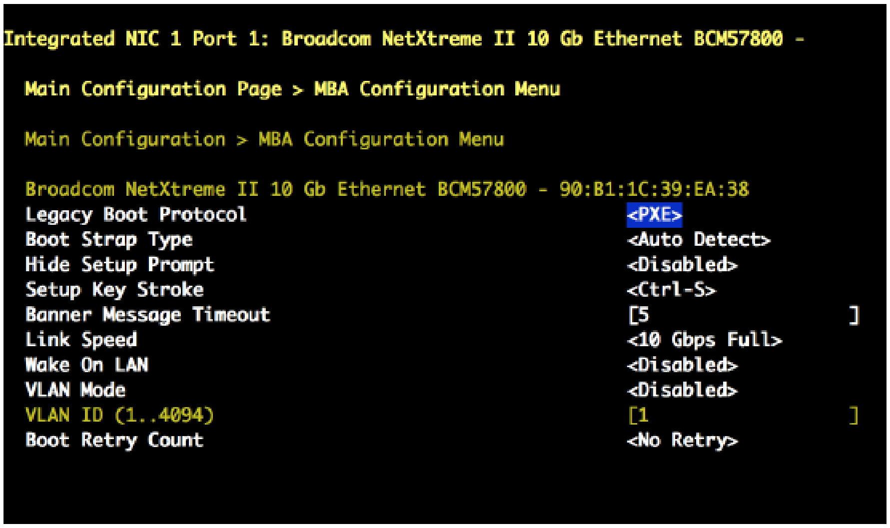
Figure 58. Main Configuration Page: MBA Configuration Menu



d. Select **Legacy Boot Protocol** on the **MBA Configuration Menu** screen, then press **Enter**. A pop-up window displays the available options.

e. In the pop-up window, use the `down-arrow` key to highlight **PXE**, then press **Enter**.

Figure 59. MBA Configuration Menu: Legacy Boot Protocol - PXE



f. Press the **Escape** key to exit the **MBA Configuration Menu** screen.

g. Verify that **Legacy Boot Protocol** is set to **None** for the other three Ethernet ports. If necessary, change the setting for these three ports by repeating substep **9b**.

h. Press the **Escape** key to exit the **Device Settings** screen.

i. When the `"Settings have changed"` message appears, select **Yes** to save your changes.

j. When the `"Settings saved successfully"` message appears, select **Ok**, and then **Enter**. The main screen (**System Setup Main Menu**) appears.

10. Save your changes and exit.

1. Press **Escape** to exit the **System Setup Main Menu**.

2. Select **Yes** when the utility displays the message `"Are you sure you want to exit and reboot?"`

The eLogin remote access configuration is now complete.

6 Configure and Manage an eLogin Image

Prerequisites

- A complete successful eLogin installation.
- Environment variable was set to `$CMCNAME` for *`CMC-name`* in Configure Connection Between SMW and CMC.

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.

Cray recommends appending images with '-*`YYYYMMDD`*'. For example, if generating `elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari` on Feb. 1st, 2017, the image should be named `elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari_20170201`. 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 root@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.

   NOTE: Use the regular eLogin image only if there are specific size constraints for the eLogin, or if the image is intended for test purposes. (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_6.0up04_sles_12sp2_x86-64_ari`. 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_6.0up04_sles_12sp2_x86-64_ari
smw# recipe update -i elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari \
custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari
```

4. Build the eLogin image.

```
smw# image create -r custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari \
custom-elogin-large_cle_6.0up04_sles_12sp2_12_x86-64_ari-YYYYMMDD
```

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

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

6. Create two environment variables: `$IMAGE` for the eLogin ISO image, and `$CONFIGSETNAME` for the *`config_set_name`*.

```
smw# IMAGE=custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari-YYYYMMDD

smw# CONFIGSETNAME=config_set_name
```

Note that the variable for the *`CMC-name`* (`$CMCNAME)` was created in previous section.

7. Push the eLogin image from the SMW to Glance running on the CMC. (Use the two environment variables - `$CMCNAME` and `$IMAGE`.)

Performing this step moves the eLogin image to the CMC machine that 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.

IMPORTANT: 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:$CMCNAME:$IMAGE $IMAGE
```

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

8. Push the config set to the CMC. (Use the two environment variables - `$CMCNAME` and `$CONFIGSETNAME`.)

The config set was generated during CLE installation and then modified in *Configure Minimum Services Required for eLogin* on page 51. Refer to the *XC Series eLogin Installation Guide CLE 6.0 UP03 Rev C*.

```
smw# cfgset push -d $CMCNAME global

smw# cfgset push -d $CMCNAME $CONFIGSETNAME
```

The config set is cached on the CMC. This allows a reprovision of 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.

9. Push the CLE Programming Environment (PE) to the CMC. (Use environment variable $CMCNAME.)

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 $CMCNAME 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

10. Connect to the CMC node.

```
# ssh root@cmc
```

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

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

12. 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/elogin/exclude_lists/elogin_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 elogin_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 $CONFIGSETNAME \
-e /etc/opt/cray/elogin/exclude_lists/elogin_cfgset_excludelist
```

IMPORTANT: In general, after config set changes are 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, delete the stack, and then redeploy.

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

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

Run steps (f., g., and h.) 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
```

13. 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_$CONFIGSETNAME
```

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/$CONFIGSETNAME_configset.tar.gz -C /mnt/",
"url": "swift:$CONFIGSETNAME_configset/$CONFIGSETNAME_configset.tar.gz",
"target": "/tmp/$CONFIGSETNAME_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_$CONFIGSETNAME \
--file copy_$CONFIGSETNAME
```

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

14. 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 approprate 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 `.qcow2`. To display the image name, use `glance image-list`.
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 `fixed_ip` address is only available in the `elogin-env-fixed-ip.yaml.template`.
instance_flavor	Nova flavor of the CDL being booted. In most cases, use `eloginflavor`.
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 `ironic node-list` 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. Select and use the same template file used in step 14B, as follows:

- `elogin_template.yaml`: If `elogin-env.yaml.template` was used.
- `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 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 elogin_name
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 `ironic_conman` console, the command-line characters are:

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

15. Repeat the previous step for each eLogin node.

7 Upgrade Process for CSMS and CentOS

This section includes procedures required for the CSMS and CentOS upgrade process:

- Prepare to Upgrade
- Upgrade CentOS 7.1 to 7.2 for eLogin
- Upgrade CSMS for eLogin
- Install elogin Software on CMC After Upgrade
- Install CSMS 1.1.3 Patch Sets After Upgrade

7.1 Prepare to Upgrade

Prerequisites

- System has CSMS and CentOS installed.
- This procedure requires root privileges.

About this task

Before upgrading CSMS, certain settings need to be put in place. These are described in this procedure.

Procedure

1. Ensure that no operations are performed on the management plane for the duration of the upgrade.

   This includes all communication with the CSMS service API endpoints. Existing deployed workloads will continue to function during the update.

2. SSH to the Cray Management Controller (CMC) as the root user, entering **`initial0`** as the password (default).

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

3. Disable all yum repositories that exist on the system other than those provided by Cray. This includes disabling the CentOS base, extras and update repositories.

   ```
   cmc# yum-config-manager --disable base extras updates
   ```

   The CSMS 2.0.* software is designed to run exclusively on CentOS 7.2 and does not support any newer releases of CentOS. By disabling the CentOS base, extras and updates yum repositories, you prevent the system from installing updated CentOS packages during the installation of this update or when performing a yum update operation.

4. Ensure all nodes are in a state where deployment images can be updated without any issues.

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

Any state, other than `available` or `active` could potentially lead to problems during the upgrade (such as when a node gets stuck during deployment). In such a case, move the nodes manually into a state where they can be updated:

```
cmc# ironic node-set-provision-state node  provision-state
```

Here *`node`* is the node name and *`provision-state`* should be `available` or `active`.

5. Backup all site files that were modified to meet specific site configuration requirements, in particular, from these two directories: `/etc/opt/cray/elogin`
and `/etc/opt/cray/openstack/ansible/config/site`.

Backup the entire contents of the previously mentioned directories with this command:

```
cmc# tar -cvf /root/siteconfigfiles.tar /etc/opt/cray/elogin \
/etc/opt/cray/openstack/ansible/config/site
```

All modified site files not in these two directories must also be backed up.

6. Verify that the system clock is set correctly.

It is critical that the system clock is set up correctly for the update to succeed.

a. Execute the timedatectl command.

```
cmc# timedatectl
```

b. Verify that in the output the Universal time correctly shows the current time (in UTC), accurate to within a second, and that the RTC time matches Universal time.

Following is an example of a valid output:

```
cmc# timedatectl set-time 11:15:00
Local time: Wed 2016-10-05 10:00:01 CDT
         Universal time: Wed 2016-10-05 14:00:01 UTC
               RTC time: Wed 2016-10-05 14:00:01
              Time zone: America/Chicago (CDT, -0500)
            NTP enabled: no
       NTP synchronized: yes
        RTC in local TZ: no
             DST active: yes
        Last DST change: DST began at
                         Sun 2016-03-13 01:59:59 CST
                         Sun 2016-03-13 03:00:00 CDT
        Next DST change: DST ends (the clock jumps one hour backwards) at
                         Sun 2016-11-06 01:59:59 CDT
                         Sun 2016-11-06 01:00:00 CST
```

If Universal time is incorrect or does not match RTC time, execute `timedatectl set-time` *HH:MM:SS* to manually set the time, specifying the time in the time zone listed in the output of the `timedatectl` command. For example:

```
cmc# timedatectl set-time 11:15:00
```

It is also possible to use an alternative method to set the time, such as via `ntpdate`, however it is important to ensure that any alternative mechanism also updates the RTC (hardware) clock.

7. Create a backup of certificates.

```
cmc# tar -cvf /root/certs.tar \
/etc/haproxy/public_api.pem /etc/httpd/ssl/public_api.cert /etc/httpd/ssl/public_api.key \
/etc/ssl/public_api.cert /etc/ssl/public_api.key
```

IMPORTANT: Due to a known issue, certificate backups are necessary. This issue affects the following:

- `/etc/haproxy/public_api.pem`
- `/etc/httpd/ssl/public_api.cert`
- `/etc/httpd/ssl/public_api.key`
- `/etc/ssl/public_api.cert`
- `/etc/ssl/public_api.key`

8. Exit the SSH session.

```
cmc# exit
```

Proceed to *Upgrade CentOS 7.1 to 7.2 for eLogin*.

7.2 Upgrade CentOS 7.1 to 7.2 for eLogin

Prerequisites

- This procedure requires a system with CentOS 7.1 and CSMS 1.1.1 or 1.1.2 installed.
- Root privileges.
- *Prepare to Upgrade* procedure is completed.
- Access to the CSMS CentOS 7.2 ISO. Refer to *Source ISO Images for eLogin*.
- Ensure that no operations are performed on the management plane for the duration of the update. This includes all communication with the CSMS service API endpoints. Existing deployed workloads will continue to function during the update.

About this task

Perform this procedure to upgrade Centos 7.1 to 7.2 on a system using CSMS 1.1.1 or 1.1.2.

Procedure

1. Copy the CSMS CentOS 7.2 ISO to the `/root/isos` directory from the SMW or another network accessible client containing the CSMS (1.1.1 or 1.1.2) 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.

```
smw# scp Cray-CentOSbase7-1511-201605031030.iso root@cmc:/root/isos
Are you sure you want to continue connecting (yes/no)? yes
password: initial0
```

Choose and run the command instructions that match the CSMS version installed on the system.

- For a system with CSMS 1.1.1, run the following:

  ```
  smw# scp csms_centos72-1.1.1-201605231635.iso root@cmc:/root/isos
  password: initial0
  ```

- For a system with CSMS 1.1.2, run the following:

  ```
  smw# scp csms_centos72-1.1.2-201606240115.iso root@cmc:/root/isos
  password: initial0
  ```

2. SSH to the Cray Management Controller (CMC) as the root user, entering **initial0** as the password (default).

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

3. Source the `admin.openrc` file.

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

4. Stop all the OpenStack services by executing the `csms_stop_all_services.sh` script from the `/etc/opt/cray/openstack/ansible/` directory

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

WARNING: For this release of CSMS, the ISO installer expects a CentOS ISO named `Cray-CentOSbase7-1511-201604201604.iso`. This causes errors when using the bootable CentOS installer `csms_centos72-1.1.1-201605231635.iso` or newer. To work around execute the following:

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

5. Create a directory named `csms` under the `/mnt` directory.

```
cmc# mkdir /mnt/csms
```

6. Mount the CSMS specific CentOS 7.2 ISO.

Choose and run the command instructions that match the CSMS version installed on the system.

- For a system with CSMS 1.1.1, run the following:

  ```
  cmc# mount /root/isos/csms_centos72-1.1.1-201605231635.iso /mnt/csms
  ```

- For a system with CSMS 1.1.2, run the following:

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

7. Switch to the `/mnt/csms` directory.

```
cmc# cd /mnt/csms
```

8. Run the installer.

   The Cray installer will check that only Cray provided repositories are enabled before proceeding with the installation of this update.

```
cmc# ./install.py
[installer runs/completes]
```

9. Unmount the CSMS specific CentOS 7.2 ISO

```
cmc# cd /root
cmc# umount /mnt/csms
cmc# rm -rf /mnt/csms
```

10. Locate any `*.rpmsave` and `*.rpmnew` files and merge any and all appropriate changes into the indicated configuration file(s).

    CAUTION: The location of various configuration files may have changed between releases. Cray recommends, where appropriate, that system configuration values be placed in override files rather than modifying configuration files directly. This is especially the case with files under the `/etc/opt/cray/openstack/group_vars/` directory. It is especially important to ensure that the `/etc/opt/cray/openstack/ansible/hosts/hosts` file is updated to include the correct hostname of the system.

```
cmc# find /etc -name '*rpmsave' -or -name '*rpmnew'
```

11. Execute the `csms_install.sh` script from the `/etc/opt/cray/openstack/ansible` directory to invoke the CSMS installer and to start all the OpenStack services.

    The `csms_install.sh` script checks that only Cray provided repositories are enabled before proceeding with the installation and configuration of CSMS software

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

12. Restore all certificates.

```
cmc# tar -xvf /root/certs.tar -C /
```

13. Remove the certificate backup file.

```
cmc# rm /root/certs.tar
```

14. Restart all OpenStack services.

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

15. Apply Cray branding changes to the OpenStack Horizon web portal by running the following commands to switch to the `/etc/opt/cray/openstack/ansible/` directory and apply the Cray branding changes.

NOTE: If this branding step was previously applied via an upgrade to CSMS 1.1.3 (this release) using CentOS 7.1, re-running this step as part of the CentOS 7.2 upgrade process may return a message indicating that it failed. In this scenario, this message can safely be ignored.

- If not using an Ansible Vault password file, execute:

```
cmc# cd /etc/opt/cray/openstack/ansible/
cmc# ./csms_common.py -a horizon-branding.yaml
```

- If using an Ansible Vault password file, execute:

```
cmc# cd /etc/opt/cray/openstack/ansible/
cmc# ./csms_common.py -a horizon-branding.yaml --password vault-password.txt
```

16. Verify that all the OpenStack services have been started by executing the `openstack-status` command.

```
cmc# openstack-status
```

17. Reboot the CMC.

```
cmc# reboot
```

18. Proceed to *Upgrade CSMS on CMC for eLogin*.

7.3 Upgrade CSMS on CMC for eLogin

Prerequisites

For Upgrade CSMS 1.1.x to CSMS 1.1.3 using CentOS 7.2, the following is required:

- The system must have CSMS 1.1.1 or 1.1.2 installed.
- CentOS 7.2 installed.
- *Prepare to Upgrade* procedure is completed.
- Access to the CSMS CentOS 7.2 ISO; refer to *Source ISO Images for eLogin*.
- This procedure requires root privileges.
- Ensure that no operations are performed on the management plane for the duration of the update. This includes all communication with the CSMS service API endpoints. Note that existing deployed workloads continue to function during the update.

About this task

Perform this procedure to upgrade the CSMS 1.1.x to CSMS 1.1.3, on a system running CentOS 7.2.

Procedure

1. Copy the CSMS (1.1.3) - CentOS (7.2) ISO to the `/root/isos` directory from the SMW, or another network accessible client containing the CSMS and Cray bootable CentOS ISOs.

```
smw# scp Cray-CentOSbase7-1511-201605031030.iso root@cmc:/root/isos
smw# scp csms_centos72-1.1.3-201608190116.iso root@cmc:/root/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.

```
Are you sure you want to continue connecting (yes/no)? yes
password: initial0
```

2. SSH to the Cray Management Controller (CMC) as the root user, entering **initial0** as the password (default).

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

3. Source the `admin.openrc` file.

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

4. Switch to the `/etc/opt/cray/openstack/ansible/` directory, and then stop all OpenStack services by executing the `csms_stop_all_services.sh` script.

```
cmc# cd /etc/opt/cray/openstack/ansible/

cmc# ./csms_stop_all_services.sh
```

5. Create a directory named `csms` under the `/mnt` directory.

```
cmc# mkdir /mnt/csms
```

6. Mount the CSMS (1.1.3) - CentOS (7.2) ISO on the `/mnt/csms/` directory.

```
cmc# mount /root/isos/csms_centos72-1.1.3-201608190116.iso /mnt/csms
```

7. Switch to the `/mnt/csms` directory.

```
cmc# cd /mnt/csms
```

8. Run the installer from the `/mnt/csms` directory.

   The Cray installer checks that only Cray provided repositories are enabled before proceeding with the installation of this update.

```
cmc# ./install.py
[installer runs/completes]
```

9. Unmount the CSMS specific CentOS ISO.

```
cmc# cd /root
cmc# umount /mnt/csms
cmc# rm -rf /mnt/csms
```

10. Locate any `*.rpmsave` and `*.rpmnew` files and merge any/all appropriate changes into the indicated configuration file(s).

CAUTION: The location of various configuration files may have changed between releases. Cray recommends, where appropriate, that system configuration values be placed in override files rather than modifying configuration files directly. This is especially the case with files under the `/etc/opt/cray/openstack/ansible/group_vars` directory. It is especially important to ensure that the `/etc/opt/cray/openstack/ansible/hosts/hosts` file is updated to include the correct hostname of the system.

```
cmc# find /etc -name '*rpmsave' -or -name '*rpmnew'
```

IMPORTANT: Leaving `*rpmsave` or `*rpmnew` files in place will result in ansible errors later in this procedure. All `*rpmsave` and `*rpmnew` files must be merged successfully. After merge, both `*rpmsave` and `*rpmnew` must be deleted.

11. Execute the `csms_install.sh` script from the `/etc/opt/cray/openstack/ansible` directory to invoke the CSMS installer and to start all the OpenStack services.

    The `csms_install.sh` script checks that only Cray provided repositories are enabled before proceeding with the installation and configuration of CSMS software

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

12. Restore all certificates.

    ```
    cmc# tar -xvf /root/certs.tar -C /
    ```

13. Remove the certificate backup file.

    ```
    cmc# rm /root/certs.tar
    ```

14. Restart all OpenStack services.

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

15. Apply Cray branding changes to the OpenStack Horizon web portal by running the following commands to switch to the `/etc/opt/cray/openstack/ansible/` directory and apply the Cray branding changes:

    ```
    cmc# cd /etc/opt/cray/openstack/ansible/
    cmc# ./csms_common.py -a horizon-branding.yaml
    ```

16. Verify that all the OpenStack services have been started by executing the `openstack-status` command.

    ```
    cmc# openstack-status
    ```

17. Proceed to *Install eLogin Software on CMC After Upgrade*.

7.4 Install eLogin Software on CMC After Upgrade

Prerequisites

- The *Upgrade CSMS on CMC for eLogin* is completed.
- Access to the eLogin ISO; refer to *Source ISO Images for eLogin*.

Procedure

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

```
cmc# scp location of elogin iso/elogin-6.0.3055-201701182038.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-6.0.3055-201701182038.iso /mnt
cmc# cd /mnt
cmc# ./install.py
[installer runs]
```

3. Unmount the ISO root.

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

4. Locate any `*.rpmsave` and `*.rpmnew` files and merge any and all appropriate changes into the indicated configuration file(s).

   CAUTION: The location of various configuration files may have changed between releases. Cray recommends, where appropriate, that system configuration values be placed in override files rather than modifying configuration files directly. This is especially the case with files under the `/etc/opt/cray/openstack/group_vars/` directory. It is especially important to ensure that the `/etc/opt/cray/openstack/ansible/hosts/hosts` file is updated to include the correct hostname of the system.

```
cmc# find /etc -name '*rpmsave' -or -name '*rpmnew'
```

   Proceed to *Install CSMS 1.1.3 Patch Sets After Upgrade*.

7.5 Install CSMS 1.1.3 Patch Sets After Upgrade

Prerequisites

- *Upgrade CSMS on CMC for eLogin* is complete.
- *Install eLogin Software on CMC After Upgrade* is complete.
- Access to the CSMS 1.1.3 patch sets (PS01-PS04); refer to *Source ISO Images for eLogin*.
- Root privileges

About this task

After completion of the *Upgrade CSMS on CMC for eLogin* procedure, perform this procedure to install all four CSMS 1.1.3 patch sets.

The CSMS 1.1.3 patch sets contain critical software fixes available for sites to install on a Cray Management Controller (CMC).

CSMS 1.1.3 requires four patch sets: PS01, PS02, PS03, and PS04. Each patch set is made up of three files:

- `CSMS_1.1.3.PS##.readme`: Contains information regarding issues addressed by the patch set and install preparation instructions.
- `CSMS_1.1.3.PS##.install`: Contains instructions for installing fixes in the CSMS software.
- `CSMS_1.1.3.PS##.iso`: Contains the rpm's and source files for the CSMS fixes and updates.

The CSMS 1.1.3 patch sets are identified by a combination of the CSMS release number and patch set ID (for example, `1.1.3.PS01`). The patch set files are released in a directory of that name.

> **IMPORTANT:** Do not use the instructions listed in the `.install` or `.readme` files to install the CSMS patch sets. These instructions are replaced with shell scripts to simplify the install process.

For detailed information regarding fixes included in a specific patch set, refer to the `CSMS_1.1.3.PS##.readme` file.

Procedure

1. Create a directory on the CMC, and copy the four patch set directories and prep/install shell scripts to the same directory.

```
cmc# mkdir -p /root/CSMS113PS

# scp -r 1.1.3.PS01 1.1.3.PS02 1.1.3.PS03 1.1.3.PS04 \
root@cmc:/root/CSMS113PS/

cmc# cd /root/patch_scripts

cmc# cp CSMS-psprep.sh CSMS113_PS01_03_install.sh CSMS113_PS04_install.sh \
/root/CSMS113PS/
```

2. Change directory to where the patch sets were copied.

```
cmc# cd /root/CSMS113PS
```

3. Prepare the four patch sets for installation.

```
cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS01

cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS02

cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS03

cmc# ./CSMS-psprep.sh /root/CSMS113PS/1.1.3.PS04
```

4. Install patch sets PS01 through PS03 using the `CSMS113_PS01_03_install.sh` script.

```
cmc# ./CSMS113_PS01_03_install.sh
```

Patch sets PS01-PS03 are now installed.

5. Install patch set PS04 using the `CSMS113_PS04_install.sh` script, including the command-line parameter of the *`IP address of the host, to which logs are to be forwarded`*.

```
cmc# ./CSMS113_PS04_install.sh 111.222.333.444
```

Patch set PS04 is installed. The install for all patch sets PS01-PS04 is now complete.

6. Proceed to *Update Fuel Deployment Images*.

8 Component Updates for eLogin

This section includes the following component updates for eLogin:

- Update Fuel Deployment Images
- Update eLogin Specific CSMS RPMs

8.1 Update Fuel Deployment Images

Prerequisites

Based on upgrade requirements, this procedure requires completion of either or both of the following procedures. If both procedures are required, they must be performed in the following order:

- Upgrade to CSMS 1.1.3 if the system uses CentOS 7.1
- Upgrade from CentOS 7.1 to CentOS 7.2

Procedure

1. Check if any Ironic nodes exist or if the Ironic node driver field contains the string: `fuel` by executing the following command, replacing *`node_name`* with the corresponding site-specific value. Existence of the string: `fuel` indicates that the Fuel driver is used.

```
cmc# ironic node-list
+--------------------------------------+---------------+---------------+-------------+---------------------+-------------+
| UUID                                 | Name          | Instance UUID | Power State | Provisioning State | Maintenance |
+--------------------------------------+---------------+---------------+-------------+---------------------+-------------+
| 97bc3fea-561c-46d4-b8f6-091c41272ebf | system        | None          | power off   | available           | False       |
+--------------------------------------+---------------+---------------+-------------+---------------------+-------------+

cmc# NODE=node_name

cmc# ironic node-show ${NODE} | grep "driver " | grep fuel
| driver | fuel_rsync_ipmi
```

   Ironic drivers using the Fuel deploy driver include `fuel_rsync_ipmi`, `fuel_swift_ipmi`, `fuel_rsync_hss`, `fuel_swift_hss`, and test drivers including `fuel_rsync_ssh`, `fuel_swift_ssh`, `fuel_rsync_vbox` and `fuel_swift_vbox`.

   If no Ironic nodes exist that use the Fuel driver, STOP here, and skip the remainder of this section.

   If Ironic nodes do exist that use the Fuel driver, continue on with the following steps.

2. Remove any existing `fuel-agent-deploy-reference-image` files from OpenStack Glance

```
cmc# glance image-delete fuel-agent-deploy-reference-image.initramfs \
fuel-agent-deploy-reference-image.vmlinuz
```

3. Create new Glance images for the updated Fuel deploy images

```
cmc# glance image-create --name fuel-agent-deploy-reference-image.vmlinuz \
--disk-format aki --container-format aki --file \
/var/opt/cray/openstack/images/deploy-images/fuel-agent-deploy-reference-image.vmlinuz \
--is-public True

cmc# glance image-create --name fuel-agent-deploy-reference-image.initramfs \
--disk-format ari --container-format ari --file \
/var/opt/cray/openstack/images/deploy-images/fuel-agent-deploy-reference-image.initramfs \
--is-public True
```

4. Perform the following steps for each Ironic node found via the previous step to update existing Ironic nodes to use the patched Fuel deploy images. Replace `KERNEL_ID` and `RAMDISK_ID` with the corresponding site-specific values.

```
cmc# glance image-show fuel-agent-deploy-reference-image.vmlinuz | grep " id "
| id |  6fa966c0-9267-4767-a787-a4b30d1a1444 |

cmc# KERNEL_ID=6fa966c0-9267-4767-a787-a4b30d1a1444

cmc# ironic node-update NODE replace driver_info/deploy_kernel=KERNEL_ID

cmc# glance image-show fuel-agent-deploy-reference-image.initramfs | grep " id "
| id |  3cc6ccb7-282e-4dcb-8eb7-5897519f872c

cmc# RAMDISK_ID=3cc6ccb7-282e-4dcb-8eb7-5897519f872c

cmc# ironic node-update NODE replace driver_info/deploy_ramdisk=RAMDISK_ID
```

8.2 Update eLogin-Specific CSMS RPMs

Prerequisites

A recent update of the eLogin ISO is required for this procedure.

About this task

The Red-hat Package Manager (RPM), is a program for install, uninstall, and management of software packages in Linux. RPMs are used to manage the Cray Software Management System (CSMS). This procedure is for updating the eLogin-specific CSMS RPMs.

Procedure

1. Log on to the Cray Management Controller (CMC) as root, and mount the updated eLogin ISO.

```
cmc# mount -o ro,loop /root/isos/elogin-image-xxx.iso /mnt
```

2. Change directory to the ISO root, and run the eLogin install.

```
cmc# cd /mnt
cmc# ./install.py
```

3. Unmount the ISO root.

```
cmc# cd /root
cmc# unmount /mnt
```

4. Run any eLogin-specific Ansible playbooks.

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

9 Component Upgrades for eLogin

This section includes the following component upgrades for eLogin:

- Upgrade PE for eLogin Node
- Upgrade Config Sets for eLogin Nodes
- Upgrade eLogin Node Image

9.1 Upgrade PE for eLogin Node

Prerequisites

Download the *Cray Programming Environment Installation Guide*.

Procedure

1. Update the PE software on the SMW (if not done). Refer to the *Update Programming Environment (PE) Software* section within the *Cray Programming Environment Installation Guide*.

2. Push the PE image to the CMC.

```
smw# image push -d cmc-name pe_image_root
```

3. Reboot the eLogin node to pull updated PE from the SMW.

```
cmc# source admin.openrc
cmc# nova reboot eLogin_name
```

9.2 Upgrade Config Sets for eLogin Nodes

Prerequisites

Config set updates on the SMW. Refer to *SMW Software Installation Guide*.

About this task

When the config set is updated on the SMW, the changes must to be pushed to the eLogin node, and the config set re-run. This procedure instructs how to upgrade the config sets to the eLogin nodes.

Procedure

1. Update the config set on the SMW (if not done). Refer to *SMW Software Installation Guide*.

2. Push the config set to the CMC.

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

3. Edit the exclude list file as required by site. The exclude list file is located at: `/etc/opt/cray/elogin/exclude_lists/elogin_cfgset_excludelist`.

   The exclude list file must be modified before the config set is loaded into swift to allow placement on the eLogin during deployment. This must be done for each config set (though not global). The `add_configset` utility is used to scrub 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 to use as a basis for a site specific list.

   > **IMPORTANT:** The contents of the exclude list are set by default to ensure security over functionality. Due to key components of the config set disabled by default, it is necessary to enable MUNGE and SSH keys. These filters are enacted via a file-by-file level. Ensure that any changes are reviewed with the relevant site security team.

   The contents of the `elogin_cfgset_excludelist` are as follows. The files or directories to exclude, are rooted at the `config/sets` 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
```

4. Push the config set changes to the CMC to allow the CDL to access the changes.

```
smw# cfgset push -d cmc-name config_set_name
```

5. Run the following command to scrub and upload the config set into Swift:

```
cmc# add_configset -c config_set_name \
-e /etc/opt/cray/elogin/exclude_lists/elogin_cfgset_excludelist
```

   > **IMPORTANT:** In general, to redeploy the eLogin node: first shut down the node by deleting the Heat stack, and then redeploy via the deploy shell script.

   If the Heat stack was previously deployed, delete the stack, and then redeploy.

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

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

   Run steps (b., c., and d.) only if the Heat stack is deployed. If no Heat stack is listed as deployed, move to step 6.

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

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

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

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

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

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

Perform step 6 only if no Heat stack was listed as deployed in step 5a.

6. 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 elogin_name
root@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 `ironic_conman` console, the command-line characters are:

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

7. Repeat the previous step for each eLogin node.

9.3 Upgrade eLogin Node Image

About this task

To upgrade the image on eLogin nodes, a Heat stack is updated and the nodes are deployed. This assumes that the administrator wants to preserve some aspects of the nodes and only change specific properties. If this is not the case, delete the stack and create a new one.

Perform this procedure to upgrade the eLogin node image:

Procedure

1. Generate a new eLogin image.

   Perform the procedure in: *Configure and Manage an eLogin Image*, except for the final deploy step.

2. Update the eLogin node environment file at: `/etc/opt/cray/openstack/ansible/eLogin_hostname-env.yaml`, on the CMC with the new `image_id`.

3. Update the Heat stack to deploy the new image.

   a. Copy the `/etc/opt/cray/openstack/heat/templates/update_elogin.sh.template` file to `/etc/opt/cray/openstack/heat/templates/update_eLogin_name.sh`.

   b. Edit the file (`update_eLogin_name.sh`) 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.

   ENV_FILE
   : Full path to the <eLogin_name>-env.yaml file.

   STACK_NAME
   : The stack name to update in Heat; usually the name of the eLogin node.

   ```
   cmc# ./update_eLogin_name.sh
   ```

   At this point, the node boots, reflecting the desired changes.

10 Update eLogin After SMW Upgrade

Prerequisites

- CSMS 1.1.3 with patch sets (PS01-PS04) installed on XC system
- Perform this procedure only after an upgrade to the SMW 8.0 / CLE 6.0 software to UP04

About this task

This procedure describes how to build and deploy the eLogin image after updating the SMW 8.0 / CLE 6.0 to UP04.

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

Cray recommends appending images with '-*YYYYMMDD*'. For example, if generating `elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari` on Feb. 1st, 2017, the image should be named `elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari-20170201`. 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 root@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.

   NOTE: Use the regular eLogin image only if there are specific size constraints for the eLogin, or if the image is intended for test purposes. (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_6.0up04_sles_12sp2_x86-64_ari`. 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_6.0up04_sles_12sp2_x86-64_ari

smw# recipe update -i elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari \
custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari
```

4. Build the eLogin image.

```
smw# image create -r custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari \
custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari-YYYYMMDD
```

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

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

6. Create three environment variables: CMC name (`$CMCNAME`), eLogin image (`$IMAGE`), and config set name (`$CONFIGSETNAME`).

```
smw# CMCNAME=name_of_cmc

smw# IMAGE=custom-elogin-large_cle_6.0up04_sles_12sp2_x86-64_ari-YYYYMMDD

smw# CONFIGSETNAME=config_set_name
```

7. Push the eLogin image from the SMW to Glance running on the CMC. (Use the two environment variables - `$CMCNAME` and `$IMAGE` - created in step 6.)

   Performing this step moves the eLogin image to the CMC machine that includes both an image format conversion to `qcow2`, and the transfer of the image to the Glance database. The estimated time to complete this process depends on the size of your image, and the speed of the networking link between the SMW and CMC.

   WARNING: Glance allows multiple images with the same name to be stored on the CMC, but only deploys 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.

   IMPORTANT: 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:$CMCNAME:$IMAGE $IMAGE
```

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

8. Push the config set to the CMC. (Use the two environment variables - `$CMCNAME` and `$IMAGE` - created in previous section.)

   The config set was generated during CLE installation and modified in *Configure Minimum Services Required for eLogin* on page 51. Refer to the *XC Series eLogin Installation Guide CLE 6.0 UP03 Rev C*.

```
smw# cfgset push -d $CMCNAME global

smw# cfgset push -d $CMCNAME $CONFIGSETNAME
```

   The config set is cached on the CMC. This allows a reprovision of 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.

9. Push the CLE Programming Environment (PE) to the CMC. (Use environment variable `$CMCNAME`.)

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

   ```
   smw# image push -d $CMCNAME pe_compute_image
   ```

   The estimated time to complete this process depends on the size of the PE, and the speed of the networking link between the SMW and CMC.

   Whenever the PE is modified, the built image must be pushed to the CMC 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

10. Connect to the CMC node.

    ```
    # ssh root@cmc
    ```

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

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

12. 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/elogin/exclude_lists/elogin_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 results in the CDL node booting to an inaccessible, unconfigured state.

    The contents of the `elogin_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 $CONFIGSETNAME \
-e /etc/opt/cray/elogin/exclude_lists/elogin_cfgset_excludelist
```

IMPORTANT: In general, after config set changes are pushed to the CMC, the config set must be loaded to Swift to allow the eLogin node to access the changes.

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

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

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

c. (Conditional): Delete the Heat stack to shut down the node (only when the Heat stack is deployed).

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

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. Edit the existing template files with site-appropriate settings for the new software `image_id`:

```
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 `.qcow2`. To display the image name, use `glance image-list`.
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 `fixed_ip` address is only available in the `elogin-env-fixed-ip.yaml.template`.
instance_flavor	Nova flavor of the CDL being booted. In most cases, use `eloginflavor`.
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 `ironic node-list` 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.

c. Create the Heat stack. This 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 elogin_name

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 `ironic_conman` console, the command-line characters are:

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

14. Repeat the previous step for each eLogin node.

11 Diagnostics for eLogin

This section includes the following eLogin diagnostic sections for an XC system:

- eLogin Console Access
- `journalctl` Command
- `/var/log` Directory
- Ansible Install Logs
- `cray_dumpsys` Command
- `kdump` and `crash`
- OpenStack Log File Locations
- OpenStack Diagnostics

11.1 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 "@."

For more information about using `ironic_conman`, refer to the man pages.

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

11.2 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 elogin 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 elogin 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 elogin kernel: Initializing cgroup subsys cpuset
Jun 08 19:28:53 elogin kernel: Initializing cgroup subsys cpu
Jun 08 19:28:53 elogin kernel: Initializing cgroup subsys cpuacct
Jun 08 19:28:53 elogin 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 elogin 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` function is similar to `tail -f`, displaying updates as they happen. For example, `journalctl -f /usr/sbin/ntpd` monitors `ntpd`-related messages. Any system daemons that produce output visible to `journalctl` can be filtered similarly.

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

11.3 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.

11.4 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 `systemd` startup.
/var/opt/cray/log/ansible/sitelog-booted	Configuration of the system during `systemd` startup.

11.5 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.
```

11.6 kdump and crash

The kdump utility is installed but NOT enabled by default. Kdump ensures that data is saved in the event of a server kernel panic.

To enable kdump on the eLogin node, refer to *Create Dump File Using kdump Service*, in the *XC Series eLogin Administration Guide CLE 6.0 UP03 Rev C*.

For details on customizing kdump, refer to the *Red Hat Kernel Crash Dump Guide*.

The crash command is also installed for use to examine crash dump data. Kernel crash dumps are stored in /var/crash/ by default. To use crash on the server, the kernel-debuginfo package must also be installed.

Users creating tenant images for deployment via OpenStack, are responsible for configuring the tenant images with kdump support if desired. Administrators are responsible for copying and/or removing kdump data from tenant nodes.

11.7 OpenStack Log File Locations

OpenStack log files for services managed by systemd are located under /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
Ironic	/var/log/ironic
Keystone	/var/log/keystone
Neutron	/var/log/neutron
Nova	/var/log/nova
Swift	/var/log/swift

Log files of OpenStack services running under Docker are stored in the kolla_logs Docker volume on the management server/controller. These files may be accessed via /var/lib/docker/volumes/kolla_logs/_data/*service*. For example, Keystone log files are located under /var/lib/docker/volumes/kolla_logs/_data/keystone.

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.

11.8 OpenStack Diagnostics

The management controller uses standard OpenStack commands to manage most components. The OpenStack diagnostic commands for Heat, Nova, and Ironic 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.

11.8.1 Heat Diagnostic Commands

The CMC 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": [                                                 |
|                      |         "10.142.0.156"                                                |
|                      |       ]                                                               |
|                      |     },                                                                |
|                      |     "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": "10.142.0.1",                                       |
```

```
|                       |   "image_id": "eLogin1.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                                                             |
+-----------------------+------------------------------------------------------------------+
```

11.8.2 Nova Diagnostic Commands

The CMC 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 elogin1
+--------------------------------------+------------------------------------------------------------+
| Property                             | Value                                                      |
+--------------------------------------+------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                     |
| OS-EXT-AZ:availability_zone          | nova                                                       |
| OS-EXT-SRV-ATTR:host                 | cmc                                                        |
| 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                                | eLogin1.qcow2 (1cc535c0-9f71-446a-8f4e-66aacc2617fe)       |
| key_name                             | default                                                    |
| management network                   | 10.142.0.156                                               |
| metadata                             | {"cray_config_set": "p2", "cray_cims_ip": "10.142.0.1",    |
|                                      | "cray_cims_rsync_password": "fab9--679b47aca4",            |
|                                      | "cray_cims_rsync_username": "eLogin1"}                     |
| 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                              | 762d33ecbeb64356a933e27bce688579                           |
+--------------------------------------+------------------------------------------------------------+
```

11.8.3 Ironic Diagnostic Commands

The CMC uses standard OpenStack commands to manage most components. For additional information on these commands, type: *`OpenStack_component`* `help` *`component_command`*. For example: `heat help stack-list`.

- `ironic node-list`

  List nodes that are registered with the Ironic service.

```
cmc# ironic node-list
+--------------------------+---------+--------------------------+-------------+--------------------+-------------+
| UUID                     | Name    | Instance UUID            | Power State | Provisioning State | Maintenance |
+--------------------------+---------+--------------------------+-------------+--------------------+-------------+
| 7baa31f0-...-ae5f147f78c3 | elogin1 | b1e37b80-...-39d3edfc8142 | power on    | active             | False       |
| 685fcdab-...-8f1c83b92f77 | elogin2 | fffb03cf-...-5c4157779237 | power on    | active             | False       |
| a2badf9e-...-cc0432e8b3fe | elogin3 | b37144df-...-aeeb6f41fc48 | power on    | active             | False       |
+--------------------------+---------+--------------------------+-------------+--------------------+-------------+
```

- `ironic node-show` *`identifier`*

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

```
cmc# ironic node-show elogin1
+------------------------+-------------------------------------------------------------------------+
| Property               | Value                                                                   |
+------------------------+-------------------------------------------------------------------------+
| target_power_state     | None                                                                    |
| extra                  | {u'description': u'elogin1'}                                            |
| last_error             | None                                                                    |
| updated_at             | 2015-12-22T19:34:24+00:00                                               |
| maintenance_reason     | None                                                                    |
| provision_state        | active                                                                  |
| uuid                   | 7baa31f0-07c8-42e9-8743-ae5f147f78c3                                    |
| console_enabled        | True                                                                    |
| target_provision_state | None                                                                    |
| maintenance            | False                                                                   |
| inspection_started_at  | None                                                                    |
| inspection_finished_at | None                                                                    |
| power_state            | power on                                                                |
| driver                 | fuel_rsync_ipmi                                                         |
| reservation            | None                                                                    |
| properties             | {u'memory_mb': 131072, u'cpu_arch': u'x86_64', u'local_gb': 550,        |
|                        | u'cpus': 32}                                                            |
| instance_uuid          | b1e37b80-032f-49e9-8521-39d3edfc8142                                    |
| name                   | elogin1                                                                 |
| driver_info            | {u'ipmi_password': u'******', u'ipmi_address': u'10.142.0.5',           |
|                        | u'deploy_ramdisk': u'2f75c6a2-3259-4929-b29c-1f5bcb194ae4',             |
|                        | u'deploy_kernel': u'84e46274-73d1-4e51-86c2-73c9f81058ab',              |
|                        | u'ipmi_username': u'root'}                                              |
| created_at             | 2015-11-20T18:21:53+00:00                                               |
| driver_internal_info   | {u'clean_steps': None, u'is_whole_disk_image': False}                   |
| chassis_uuid           |                                                                         |
| instance_info          | {u'root_gb': u'100', u'deploy_config_options': {u'instance':            |
|                        | u'deploy_config_elogin2'}, u'image_source': u'371a2726-bdde-            |
|                        | 4a78-bf60-bc17ca0f27fa', u'rsync_root_path':                            |
|                        | u'10.142.0.1::ironic_rsync/7baa31f0-07c8-42e9-8743-ae5f147f78c3/root/', |
|                        | u'driver_actions': u'', u'deploy_driver': u'rsync', u'swap_mb':         |
|                        | u'16384'}                                                               |
+------------------------+-------------------------------------------------------------------------+
```

12 eLogin Configuration Options

This section includes the following eLogin configuration options:

- Enable LiveUpdate Support for eLogin Nodes
- Configure Tagged VLANs for eLogin
- Configure Bonded Interfaces for eLogin
- Configure IPv4 Interface to Include IPv6 Address
- Determine Boot Interface and MAC Address
- Configure SSDs on CDL Nodes
- Deploy An eLogin Node

12.1 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. LiveUpdates provide 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 Cray Management Controller (CMC). 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 *eLogin Image Management*.

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

On eLogin nodes, the booted config set determines if the LiveUpdates service is turned on. With LiveUpdates set to `On`, local respositories are configured to reference the upstream repositories of origin on the SMW.

Perform the following procedure to configure the SMW and CMC for LiveUpdates. This ensures that the package manager invocations function correctly.

Procedure

SMW Configuration

1. Register the LiveUpdates service with the SMW firewall.
   - (For `SuSEFirewall2`) Append the service `liveupdates` to the `/etc/sysconfig/SuSEFirewall2` configuration file and then apply the updated firewall setting to the SMW.

     ```
     smw# vi /etc/sysconfig/SuSEFirewall2
     smw# /sbin/SuSEFirewall2 start
     ```
   - (For other firewalls) Ensure that port traffic is routable over port 2526 through `eth0`.

     ```
     smw# iptables -I INPUT -i eth0 -p tcp --dport 2526 -m state \
     --state NEW,ESTABLISHED -j ACCEPT
     ```

CSMS Configuration

2. Configure the CMC 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 CMC).

   ```
   cmc# ping -I br-em2 elogin_name
   PING elogin2.us.cray.com (172.30.50.174) from 172.30.50.129 br-em2: 56(84) bytes of data.
   64 bytes from elogin2.us.cray.com (172.30.50.174): icmp_seq=1 ttl=64 time=0.408 ms
   64 bytes from elogin2.us.cray.com (172.30.50.174): icmp_seq=2 ttl=64 time=0.349 ms
   64 bytes from elogin2.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 2526 -m state \
   --state NEW,ESTABLISHED -j ACCEPT
   ```

3. Configure and enable `httpd` to serve the Updates repositories.

   Create a config file on the CMC in `/etc/httpd/conf.d` with the following contents:

   ```
   Listen 2526

   <VirtualHost *:2526>
   ServerName your-cmc
   DocumentRoot "/var/opt/cray"
   <Directory "/var/opt/cray/repos">
   AllowOverride FileInfo AuthConfig Limit Indexes
   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
   Require method GET POST OPTIONS
   </Directory>

   </VirtualHost>
   ```

4. Restart the `httpd` service to make the changes take effect.

   ```
   cmc# systemctl restart httpd
   ```

5. Sync the repository directories from the SMW to the CMC.

```
cmc# rsync -lxpr --delete your-smw:/var/opt/cray/repos /var/opt/cray
```

Also, you may choose to put the `rsync` command in `cron`, to run sync automatically every hour, or every day.

> **NOTE:** For automatic sync to work in `cron`, SSH keys must be set to allow the CMC to connect to the SMW without a password.

(Optional) To sync the repository directories automatically every day, do the following:

```
cmc# echo "rsync -lxpr --delete your-smw:/var/opt/cray/repos \
/var/opt/cray" > /etc/cron.daily/

cmc# chmod 755 /etc/cron.daily/syncrepos
```

6. Test to ensure the repositories are available via HTTP from an eLogin node.

```
elogin# wget -O - http://cims-mgmt:2526/repos

--2016-09-12 14:11:04-- http://localhost:2526/repos
Resolving localhost (localhost)... ::1, 127.0.0.1
Connecting to localhost (localhost)|::1|:2526... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://localhost:2526/repos/ [following]
--2016-09-12 14:11:04-- http://localhost:2526/repos/
Reusing existing connection to [localhost]:2526.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘STDOUT’

[<=> ] 0 --.-K/s
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /repos</title>
</head>
<body>
<h1>Index of /repos</h1>

(a list of repo names wrapped in HTML should be shown here)
```

7. Test the commands to list and view the repos:

```
elogin# zypper lr
```

```
#  | Alias                                                  | Name                                                   | Enabled | GPG Check | Refresh
---+--------------------------------------------------------+--------------------------------------------------------+---------+-----------+--------
1  | common_cle_6.0up04_sles_12sp2_x86-64_ari               | common_cle_6.0up04_sles_12sp2_x86-64_ari               | Yes     | (  ) No   | No
2  | common_cle_6.0up04_sles_12sp2_x86-64_ari_updates       | common_cle_6.0up04_sles_12sp2_x86-64_ari_updates       | Yes     | (  ) No   | No
3  | lustre-2.5_cle_6.0up04_sles_12sp2_x86-64_ari           | lustre-2.5_cle_6.0up04_sles_12sp2_x86-64_ari           | Yes     | (  ) No   | No
4  | lustre-2.5_6.0up04_sles_12sp2_x86-64_ari_updates       | lustre-2.5_cle_6.0up04_sles_12sp2_x86-64_ari_updates   | Yes     | (  ) No   | No
5  | passthrough-common_cle_6.0up04_sles_12sp2_x86-64       | passthrough-common_cle_6.0up04_sles_12sp2_x86-64       | Yes     | (  ) No   | No
6  | passthrough-common_cle_6.0up04_sles_12sp2_x86-64_updates | passthrough-common_cle_6.0up04_sles_12sp2_x86-64_updates | Yes     | (  ) No   | No
7  | sle-module_legacy_12sp2_x86-64                         | sle-module_legacy_12sp2_x86-64                         | Yes     | (  ) No   | No
8  | sle-module_legacy_12sp2_x86-64_updates                 | sle-module_legacy_12sp2_x86-64_updates                 | Yes     | (  ) No   | No
9  | sle-sdk_12sp2_x86-64                                   | sle-sdk_12sp2_x86-64                                   | Yes     | (  ) No   | No
10 | sle-sdk_12sp2_x86-64_updates                           | sle-sdk_12sp2_x86-64_updates                           | Yes     | (  ) No   | No
11 | sle-we_12sp2_x86-64                                    | sle-we_12sp2_x86-64                                    | Yes     | (  ) No   | No
12 | sle-we_12sp2_x86-64_updates                            | sle-we_12sp2_x86-64_updates                            | Yes     | (  ) No   | No
13 | sles_12sp2_x86-64                                      | sles_12sp2_x86-64                                      | Yes     | (  ) No   | No
14 | sles_12sp2_x86-64_updates                              | sles_12sp2_x86-64_updates                              | Yes     | (  ) No   | No
```

```
elogin# elogin:/tmp# zypper pa -r  common_cle_6.0up04_sles_12sp2_x86-64_ari
```

```
Building repository 'common_cle_6.0up04_sles_12sp2_x86-64_ari' cache .........done]
Loading repository data...
Reading installed packages...

S | Repository                                  | Name                   | Version          | Arch
--+---------------------------------------------+------------------------+------------------+-------
i | @System                                     | Mesa                   | 10.0.2-90.17     | x86_64
i | @System                                     | Mesa-libEGL-devel      | 10.0.2-90.17     | x86_64
i | @System                                     | Mesa-libEGL1           | 10.0.2-90.17     | x86_64
```

```
i | @System                                   | Mesa-libGL-devel       | 10.0.2-90.17    | x86_64
i | @System                                   | Mesa-libGL1            | 10.0.2-90.17    | x86_64
i | @System                                   | Mesa-libglapi0         | 10.0.2-90.17    | x86_64
i | @System                                   | MyODBC-unixODBC        | 5.1.8-20.1      | x86_64
i | common_cle_6.0up04_sles_12sp2_x86-64_ari  | QConvergeConsoleCLI    | 1.1.03-49       | x86_64
```

8. Start the upgrade on an eLogin node.

```
cmc# ssh elogin "zypper --non-interactive up"
Loading repository data...
Reading installed packages...

The following 3 NEW packages are going to be installed:

cray-dw_wlm-imps_ansible
kernel-source-3.12.60-52.49.1_2.0.62_gd606fea_55.1
kernel-syms-3.12.60-52.49.1_2.0.62_gd606fea_55.1
```

Repeat this step for each of the eLogin nodes at your site.

12.2 Configure Tagged VLANs for eLogin

Prerequisites

- eLogin nodes are deployed
- Config set worksheet (`cray_net_worksheet.yaml`)
- Configurator tool (`cfgset`)

About this task

Virtual Local Area Networks (VLANs) are used to divide a physical network into several broadcast domains. This procedure describes how to configure tagged-packet type VLANs by editing the config-set worksheet (`cray_net_worksheet.yaml`), and then importing it into the config set using the Configurator tool (`cfgset`).

The following section of a config-set worksheet (`cray_net_worksheet.yaml`) illustrates the entries necessary to configure a tagged VLAN on a network interface. The worksheet is easier to implement versus using the Configurator tool to perform the steps interactively. The worksheet file is located on the SMW at: `/var/opt/cray/imps/config/sets/<config_set>/worksheets/cray_net_worksheet.yaml`. This file is updated by the Configurator tool. Each time the `cfgset` command is run to create or update a config set, the Configurator updates the `.../config/*config.yaml` files, and writes a fresh set of `.../worksheets/*worksheet.yaml` files. For this reason, the Configurator tool does not import a worksheet the config set's `worksheets` directory.

Perform the following procedure to create tagged VLANs for eLogin:

Procedure

1. Copy the config-set worksheet file (`cray_net_worksheet.yaml`), located at `/var/opt/cray/imps/config/sets/<config_set>/worksheets/cray_net_worksheet.yaml`, to a new location for editing.

This example makes a directory called `/my/workarea` to copy the worksheet. Use a suitable directory location to perform this step.

```
smw# mkdir -p /my/workarea

smw# cp /var/opt/cray/imps/config/sets/config_set/worksheets/ \
cray_net_worksheet.yaml /my/workarea

smw# cd /my/workarea
```

2. Edit the worksheet and add the following set of data.

```
smw# vi /my/workarea/cray_net_worksheet.yaml
```

Locate the section of the worksheet where the interfaces for the eLogin node are defined, and insert or modify the data there.

> **NOTE:** This example configures the `eth4` interface on the tagged VLAN named `eth4.1234`, for the eLogin named `elogin1`.

```
cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth4: null
cray_net.settings.hosts.data.elogin1.interfaces.eth4.name: eth4
cray_net.settings.hosts.data.elogin1.interfaces.eth4.description: Ethernet connecting network node to customer
network
cray_net.settings.hosts.data.elogin1.interfaces.eth4.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4.network: site
cray_net.settings.hosts.data.elogin1.interfaces.eth4.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth4.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.eth4.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.eth4.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4.params: ''
#cray_net.settings.hosts.data.elogin1.interfaces.eth4.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth4_1234: null
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.name: eth4.1234
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.description: Ethernet connecting the network
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.network: site
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.ipv4_address: 10.236.1.190
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.extra_attributes:
- USERCONTROL='no'
- ETHERDEVICE='eth4'
- VLAN_ID='1234'
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.params: ''
#cray_net.settings.hosts.data.elogin1.interfaces.eth4_1234.unmanaged_interface: false
```

The worksheet edit is now complete.

3. Update the config set, and then push it to the CMC.

This example uses the `p0` config set. Replace `p0` with the name of updated config set.

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

smw# cfgset push -d cmc-name p0
```

4. Create a hosts file with the hostname of each eLogin node intended for update. (One node per line, under the `[update_hosts]` section.)

```
[update_hosts]
elogin1
```

```
elogin2
elogin3
```

5. Log onto the CMC, and run the `add_configset` command to update the running nodes in the hosts file.

   This example uses the default exclude list. Use the list that matches your configuration.

   ```
   cmc# add_configset -c config_set_name -e /etc/opt/cray/elogin/ \
   exclude_lists/elogin/exclude_lists/elogin_cfgset_excludelist -u path_to_host_file
   ```

   This also uploads the updated config set to Swift to ensure readiness for the next deploy.

6. Deploy the eLogin nodes. For instruction, refer to: *Deploy an eLogin Node*.

7. Verify the interfaces come up on the proper tagged VLANs.

   ```
   cmc# cfgset search -t elogin-name -s cray_net config_set
   ```

12.3 Configure Bonded Interfaces for eLogin

Prerequisites

- System is booted; eLogin nodes are not deployed
- Config set worksheet (`cray_net_worksheet.yaml`)
- Configurator tool (`cfgset`)

About this task

Bonded interfaces are configured by editing the config-set worksheet (`cray_net_worksheet.yaml`), and then importing it into the config set with the Configurator tool (`cfgset`).

The following section of the config-set worksheet illustrates the required entries to configure a pair of bonded interfaces. It is easier to implement this via the worksheet than to perform the steps interactively using the Configurator tool.

The worksheet file is located on the SMW at: `/var/opt/cray/imps/config/sets/<config_set>/worksheets/cray_net_worksheet.yaml`. This file is updated by the Configurator tool (`cfgset`). Each time the `cfgset` command is run to create or update a config set, the Configurator updates the `.../config/*config.yaml` files, and writes a fresh set of `.../worksheets/*worksheet.yaml` files. For this reason, the Configurator tool does not import a worksheet from the config set's `worksheets` directory.

Perform the following procedure to configure bonded interfaces for eLogin:

Procedure

1. Copy the config-set worksheet file to a new location for editing.

   The file is located at: `/var/opt/cray/imps/config/sets/<config_set>/worksheets/cray_net_worksheet.yaml`.

This example makes a directory called `/my/workarea` to copy the worksheet. Use a suitable directory location to perform this step.

```
smw# mkdir -p /my/workarea
smw# cp /var/opt/cray/imps/config/sets/config_set/ \
worksheets/cray_net_worksheet.yaml /my/workarea
smw# cd /my/workarea
```

2. Edit the worksheet and add the following set of data to configure the bonded interfaces.

```
smw# vi /my/workarea/cray_net_worksheet.yaml
```

Locate the section of the worksheet where the interfaces for the eLogin node are defined, and insert or modify the data there. This example configures the `eth0` and `eth2` interfaces to be bonded together as the `bond0` interface on `net_224`. Note the use of `slave_eth` for the bonded interfaces.

```
cray_net.settings.networks.data.name.slave_eth: null
cray_net.settings.networks.data.slave_eth.description: Disabled eth device for bond
cray_net.settings.networks.data.slave_eth.ipv4_network: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_netmask: 0.0.0.0
cray_net.settings.networks.data.slave_eth.ipv4_broadcast: ''
cray_net.settings.networks.data.slave_eth.ipv4_gateway: ''
cray_net.settings.networks.data.slave_eth.dns_servers: []
cray_net.settings.networks.data.slave_eth.dns_search: []
cray_net.settings.networks.data.slave_eth.ntp_servers: []
#cray_net.settings.networks.data.slave_eth.fw_external: false

cray_net.settings.networks.data.name.net_224: null
cray_net.settings.networks.data.net_224.description: Customer 10.0.224.0 network
cray_net.settings.networks.data.net_224.ipv4_network: 10.0.224.0
cray_net.settings.networks.data.net_224.ipv4_netmask: 255.255.255.0
cray_net.settings.networks.data.net_224.ipv4_broadcast: ''
cray_net.settings.networks.data.net_224.ipv4_gateway: 10.0.224.1
cray_net.settings.networks.data.net_224.dns_servers:
- 10.0.146.10
- 10.0.199.10
cray_net.settings.networks.data.net_224.dns_search:
- site-example.net
cray_net.settings.networks.data.net_224.ntp_servers:
- 10.0.6.48
- 10.0.42.217
#cray_net.settings.networks.data.net_224.fw_external: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.bond0: null
cray_net.settings.hosts.data.elogin1.interfaces.bond0.name: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.description: bond0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0.network:
cray_net.settings.hosts.data.elogin1.interfaces.bond0.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.bond0.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.bond0.mtu: '9000'
cray_net.settings.hosts.data.elogin1.interfaces.bond0.extra_attributes:
- BONDING_MASTER='yes'
- BONDING_MODULE_OPTS='miimon=100 mode=balance-xor use_carrier=1
xmit_hash_policy=layer2+3'
- BONDING_SLAVE0='eth0'
- BONDING_SLAVE1='eth2'
cray_net.settings.hosts.data.elogin1.interfaces.bond0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth0: null
```

```
cray_net.settings.hosts.data.elogin1.interfaces.eth0.name: eth0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.description: Slave eth0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth0.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth0.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth0.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth0.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth0.unmanaged_interface: false

cray_net.settings.hosts.data.elogin1.interfaces.common_name.eth2: null
cray_net.settings.hosts.data.elogin1.interfaces.eth2.name: eth2
cray_net.settings.hosts.data.elogin1.interfaces.eth2.description: Slave eth2
cray_net.settings.hosts.data.elogin1.interfaces.eth2.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.network: slave_eth
cray_net.settings.hosts.data.elogin1.interfaces.eth2.ipv4_address: 0.0.0.0
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.startmode: hotplug
cray_net.settings.hosts.data.elogin1.interfaces.eth2.bootproto: none
cray_net.settings.hosts.data.elogin1.interfaces.eth2.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.extra_attributes: []
cray_net.settings.hosts.data.elogin1.interfaces.eth2.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.eth2.unmanaged_interface: false
```

To put the `bond0` interface on a tagged VLAN (example, 224), add the following:

```
cray_net.settings.hosts.data.elogin1.interfaces.common_name.bond0_224: null
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.name: bond0.224
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.description: bond0 and vlan 244
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.aliases: []
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.network: net_224
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.ipv4_address: 10.0.224.63
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.mac: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.startmode: auto
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.bootproto: static
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.mtu: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.extra_attributes:
- USERCONTROL='no'
- ETHERDEVICE='bond0'
- VLAN_ID='224'
- REORDER_HDR='no'
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.module: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.params: ''
cray_net.settings.hosts.data.elogin1.interfaces.bond0_224.unmanaged_interface: false
```

The worksheet edit is now complete.

3. Update the config set, and then push it to the CMC.

This example uses the `p0` config set. Replace `p0` with the name of updated config set.

```
smw# cfgset update -w '/my/workarea/cray_net_worksheet.yaml' p0
smw# cfgset push -d cmc-name p0
```

4. Create a hosts file with the hostname of each eLogin node intended for update. (One node per line, under the `[update_hosts]` section.)

```
[update_hosts]
elogin1
```

```
elogin2
elogin3
```

5. Log onto the CMC, and run the `add_configset` command to update the running nodes in the hosts file.

This example uses the default exclude list. Use the list that matches your configuration.

```
cmc# add_configset -c config_set_name -e /etc/opt/cray/ \
elogin/exclude_lists/elogin/exclude_lists/elogin_cfgset_excludelist \
-u path_to_host_file
```

This also uploads the updated config set to Swift to ensure readiness for the next deploy.

6. Deploy the eLogin nodes. Refer to: *Deploy an eLogin Node*.

7. Verify that the bonded interfaces are configured correctly.

```
cmc# cfgset search –t example_eLogin -s cray_net config_set
```

12.4 Configure an IPv4 Interface to Include IPv6 Address

Prerequisites

Assumes a full install of eLogin configured with an IPv4 interface.

About this task

Sites that use the IPv6 interface in addition to IPv4, must configure IPv6 via the config set for eLogin nodes. This configuration is done by editing the config-set worksheet (`cray_net_worksheet.yaml`), located on the SMW at: `/var/opt/cray/imps/config/sets/`*`config_set`*`/worksheets/cray_net_worksheet.yaml`.

Perform this procedure to configure an interface to include both an IPv4 and IPv6 address in the config set:

Procedure

Add IPv6 information to the `extra_attributes` field within the interface of the `cray_net` section of the config set. Replace *`IPV6_ADDR`* with the actual IPv6 address to use.

```
# Example of cfgset section to add ``extra_attributes`` to eth1.

cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.extra_attributes
[<cr>=set 0 entries, +=add an entry, ?=help, @=less] $ +
Add extra_attributes (Ctrl-d to exit) $ IPADDR1='<IPV6_ADDR>'
Add extra_attributes (Ctrl-d to exit) $ PREFIXLEN1='64'
Add extra_attributes (Ctrl-d to exit) $ <Ctrl-d>
```

The following output is produced in the `cray_net` worksheet:

```
cray_net.settings.hosts.data.example_eLogin.interfaces.eth1.extra_attributes:
- IPADDR1='IPV6_ADDR'
- PREFIXLEN1='64'
```

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

NOTE: The config set worksheets do not accommodate more than one route, so the configuration of the IPv6 route must be done via simple_sync or site Ansible play.

12.5 Determine Boot Interface and MAC Address

Prerequisites

CSMS Configuration Worksheet

About this task

This procedure is for determining the Boot Interface and MAC address for an system running eLogin.

Boot Interface

The Boot Interface is dependent upon the hardware being used. The Boot Interface is the first 1GbE interface, per one of the following configurations:

- eth0 on eLogins, with the 4 x 1GbE LOM configuration
- eth2 on eLogins, with the 2 x 10GbE / 2 x 1GbE LOM configuration

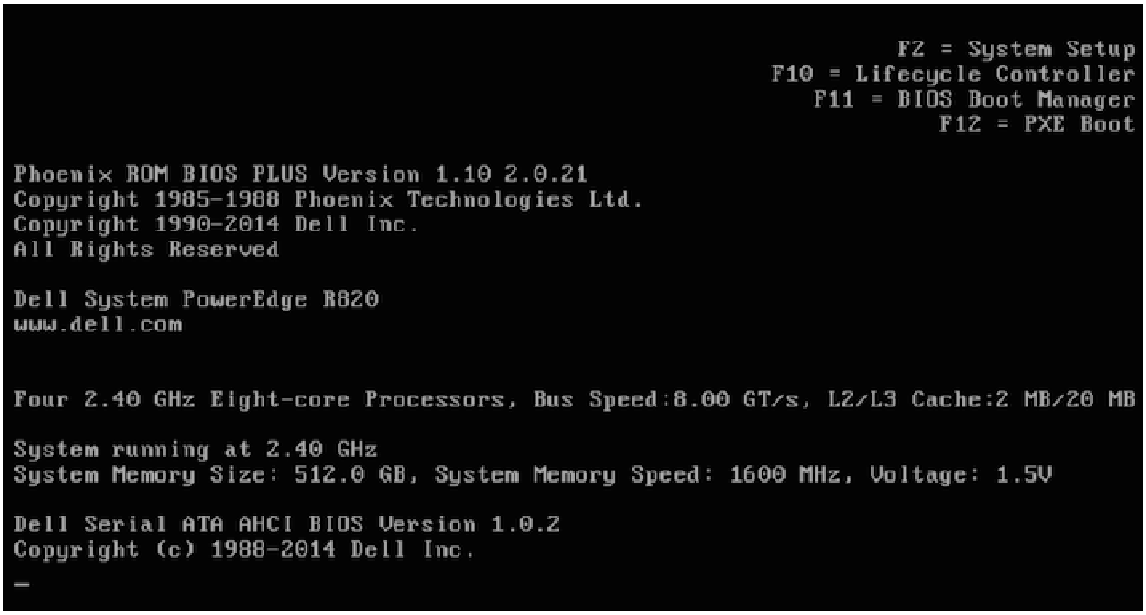
Perform the following procedure to identify the Boot Interface and MAC address for configuring CDL nodes.

Procedure

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

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

Figure 60. BIOS Config Screen



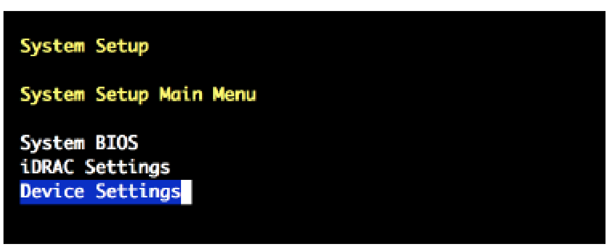
When the **F2** keypress is recognized, the **F2 = System Setup** line changes to **Entering System Setup**.

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

```
System BIOS
iDRAC Settings
Device Settings
```

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

   Figure 61. System Setup Main Menu: Device Settings

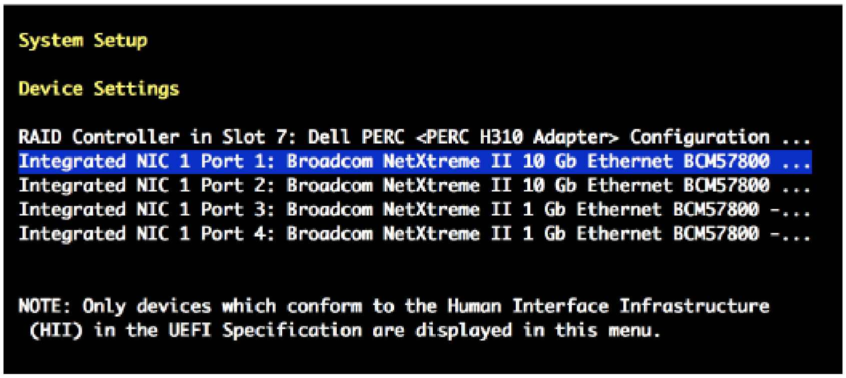
   

3. Select **Integrated NIC 1 Port N ...** in the **Device Settings** window.

   Choose the NIC port number that corresponds to the Ethernet port for the `maint-net` network:

   - Select **Integrated NIC 1 Port 1 ...** if `maint-net` uses the first Ethernet port (`eth0`)
   - Select **Integrated NIC 1 Port 3 ...** if `maint-net` uses the third Ethernet port (`eth2`)

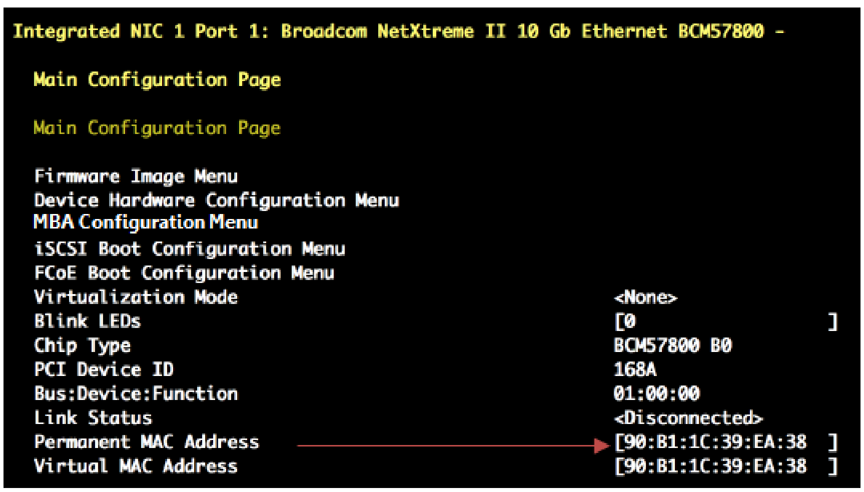
Figure 62. Device Settings: Integrated NIC Port Number



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

   The following figure shows an example MAC address.

   Figure 63. Integrated NIC Port / Main Configuration Page: MAC Address



6. Record the MAC address and the Boot Interface on the CSMS Configuration Worksheet.
7. Press <**Esc**> to exit to the **Device Settings** menu.
8. Select **No** when prompted with the `"Settings have changed"` message, then press **Enter**.
9. Press <**Esc**> to exit the **System Setup Main Menu**.

   The **System Setup Main Menu** screen appears.
10. Press <**Esc**> to exit the **System Setup Main Menu**.

12.6 Configure SSDs on CDL Nodes

Prerequisites

The CMC RAID device must be configured to have two virtual disks (VDs) visible to the operating system. The first VD is configured on 1,920 GB of disk space for the base operating system (O/S), logs, and Cray's Programming Environment (PE). The first VD is configured as VD name '**sda**'. The second VD is configured on the remaining disk space, as VD name '**sdb**'. The VD '**sdb**' is also referred to as 'Swift disk' in eLogin installation documentation.

About this task

This procedure is for recently installed solid-state storage (SSDs) on an XC system. It is intended for SSDs installed after the initial configuration of the CMC hardware BIOS on the system. Refer to *Configure CMC Hardware BIOS* in the *XC Series eLogin Installation Guide CLE 6.0 UP03 Rev C*.

When an SSD is installed after initial BIOS setup, specific configuration setup is required due to the CMC BIOS storage device discovery properties. The CMC by default, uses the Dell PERC controller to manage disk drives. During discovery the Dell PERC controller separates non-RAID devices from RAID devices. The non-RAID devices are presented to the operating system first, followed by the RAID devices.

If SSDs were installed on the CMC as non-RAID devices, then the SSDs are presented to the OS before the required/reserved RAID system devices **sda** and **sdb**. In this case, the SSDs must first be removed from the system, and then the CMC HW BIOS reconfigured to default (without SSDs). After the BIOS is configured, the SSDs are reinstalled and then set up as RAID devices in the BIOS.

For recently installed SSDs to the XC system, perform this procedure to Configure SSDs on CDL Nodes:

Procedure

1. Remove all SSDs from the system.

   Configure CMC Hardware BIOS

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

   Figure 64. Enter RAID Setup: CMC BIOS

   The RAID configuration screen opens.

Figure 65. RAID Configuration Screen: CMC BIOS

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

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

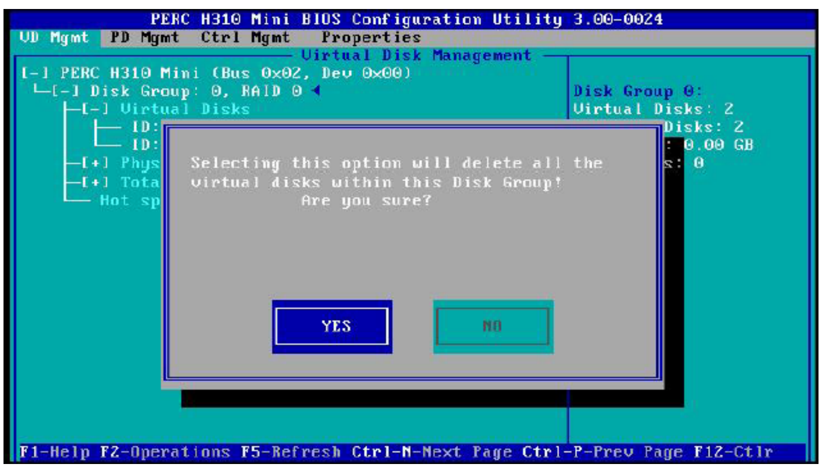
a. Select the disk.

b. Press **F2** key to get a list of operations.

c. Select **Delete Disk Group** and press **Enter**.

Figure 66. Delete Disk Group: CMC BIOS

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



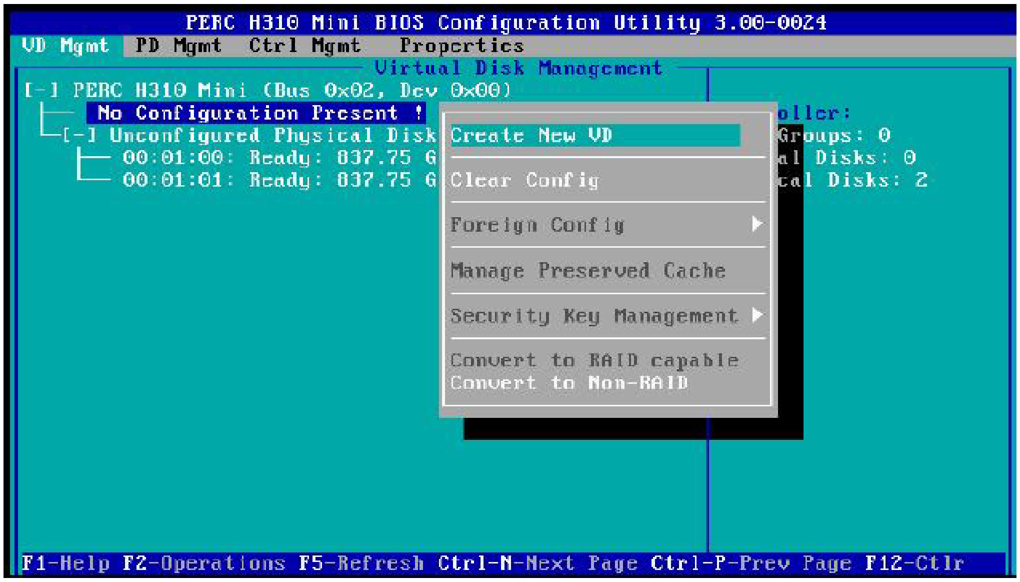
4. Create a new virtual disk (VD) A.

a. In the VD 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.

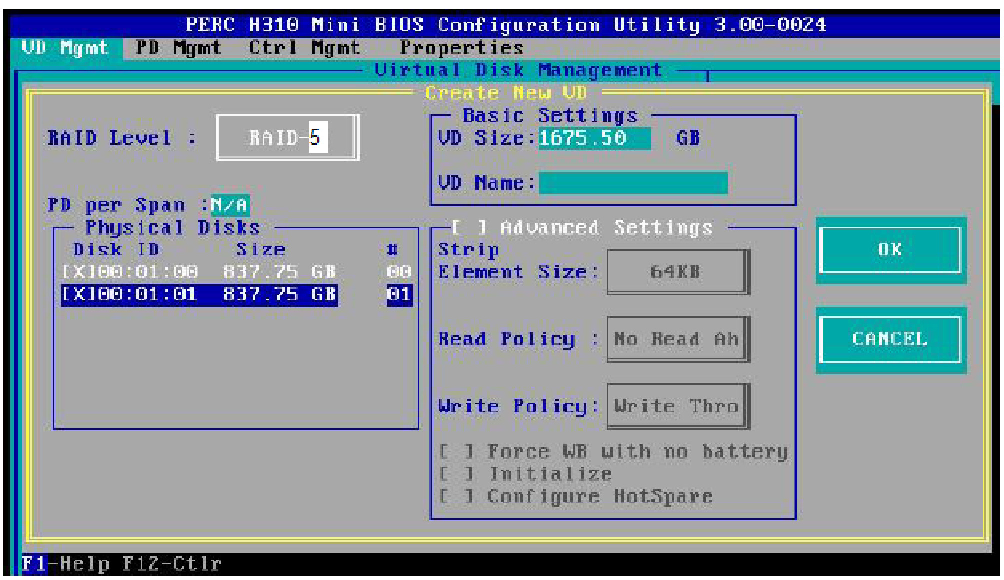
Figure 67. Create Virtual Disk: CMC BIOS



5. Move cursor to select the disk ID, and then press spacebar on keyboard to add disk to RAID.

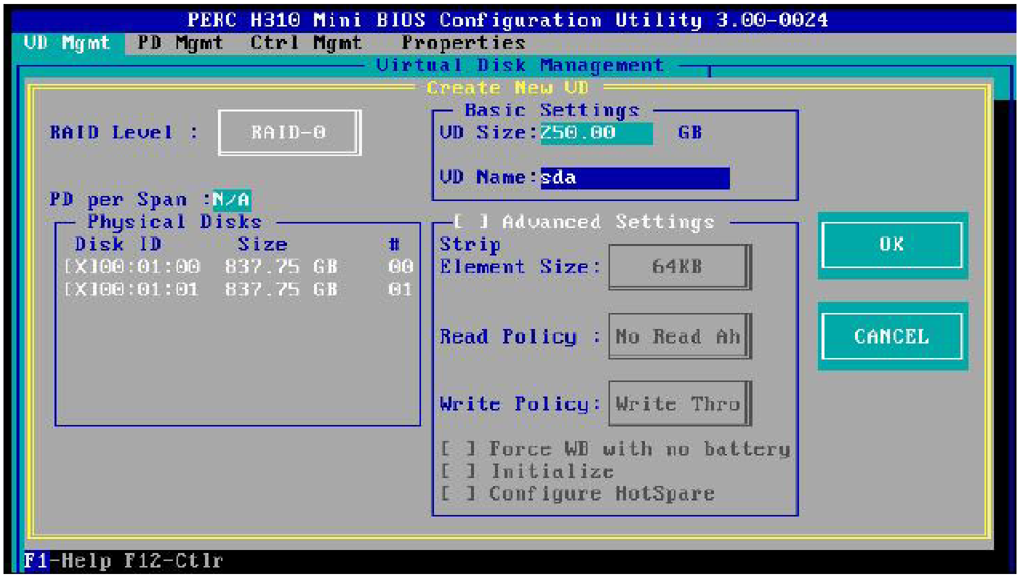
6. Set the RAID Level to **RAID 5**, with hot spare.

Figure 68. Set RAID Level 5: CMC BIOS

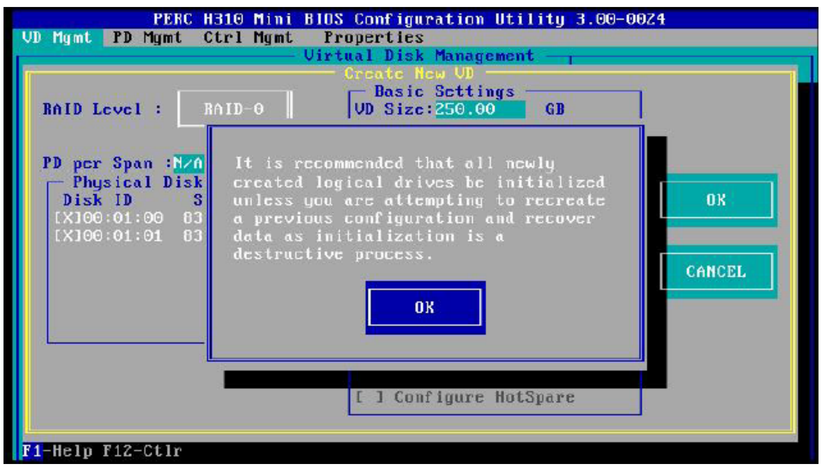


7. Set **VD Size** and **VD Name** for virtual disk A.
   a. Set the **VD Size** of the first disk to **1,920 GB** disk space.
   b. Set the **VD Name** to **sda**.

Figure 69. 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**.



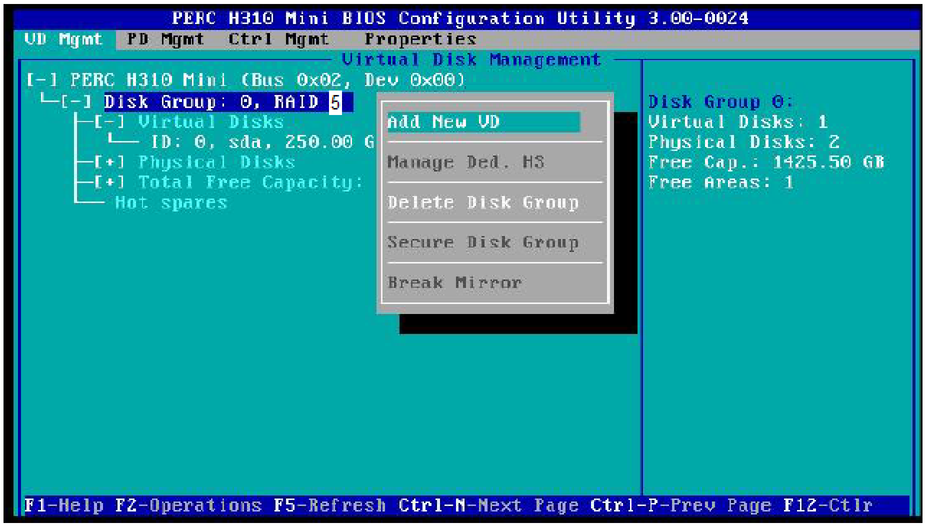
8. Create a new virtual disk (VD) B.

a. In the VD management window, navigate to **Disk Group: 0 RAID-5** using the keyboard up/down arrows.

b. Press **F2** to access the disk creation menu.

c. Select **Add New VD**.

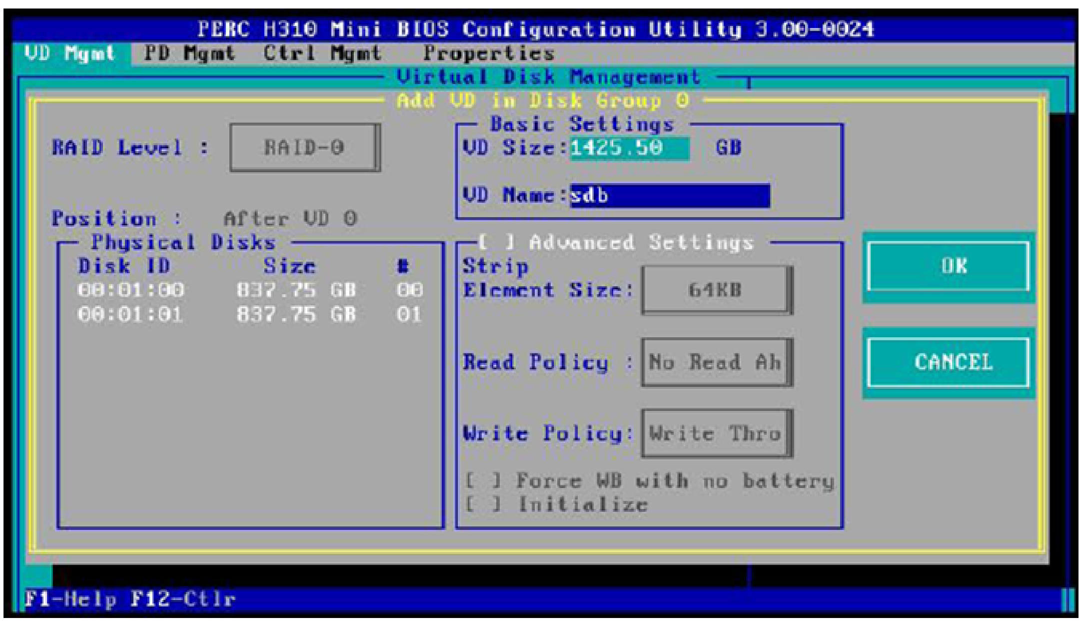
Figure 70. Create New Virtual Disk B: CMC BIOS



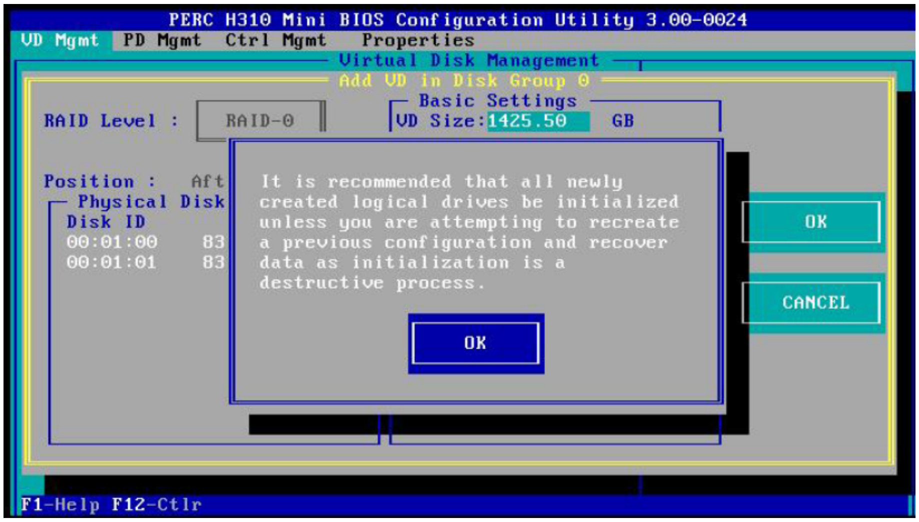
d. Set the **VD Name** to **sdb**.

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

Figure 71. 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**.

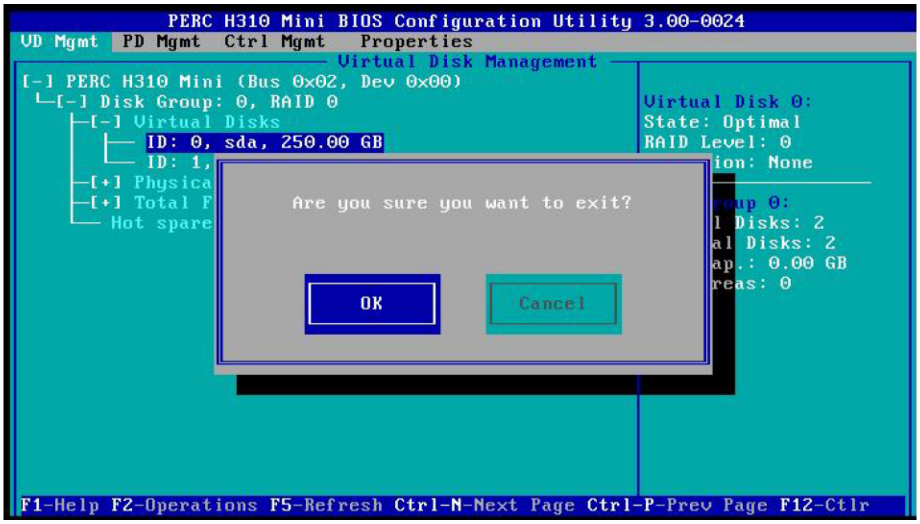


The CMC now has two disks available to install on.

Figure 72. Two Virtual Disks Available: CMC BIOS

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

Figure 73. Exit BIOS Configuration: CMC BIOS



The BIOS configuration utility screen is now closed.

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

11. Plug all SSDs into the system.

12. Enter the BIOS.

13. Configure SSD as a **RAID** device (instead of a non-RAID device).

14. Configure SSD as **RAID0** without an active partner.

12.7 Deploy an eLogin Node

About this task

eLogin nodes are deployed using Heat stacks. Templates are provided to assist in configuring a Heat stack for each node. Each eLogin node requires its own `elogin-env*.yaml` file, which describes the software and configuration elements to use in deploying the eLogin node. There are two template file variations of `elogin-env*.yaml.template`. One template uses a fixed IP address on the management network, while the other uses a system that assigns a new IP address each time a node is deployed.

eLogin-env* Template File	Description
`elogin-env.yaml.template`	eLogin has dynamic management IP address (assigns a new IP address).
`elogin-env-fixed-ip.yaml.template`	eLogin has a fixed management IP address.

These template files reside on the Cray Management Controller (CMC) at:

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

Procedure

1. Copy the appropriate `elogin-env*.yaml.template` file to the following directory, and rename with the eLogin node hostname (for example, `elogin1`).

   ```
   cmc# cd /etc/opt/cray/openstack/elogin/

   cmc# cp /etc/opt/cray/openstack/heat/templates/elogin-env-fixed-
   ip.yaml.template \
   ./example-elogin-env-fixed-ip.yaml
   ```

2. Edit the template file (example, `elogin1-env-fixed-ip.yaml`) to contain the correct parameter information for the node.

   ```
   parameters:
     image_id: elogin_name.qcow2
     host_name: elogin1
     fixed_ip: 10.142.0.100
     instance_flavor: eloginflavor
     cray_config_set: p0
     cims_host_name: cmc-name
     ironic_id: bbc98115-739a-4ee5-b727-cea0a7208fc5
     actions_list: copy_p0
   ```

image_id	Name of the image pushed from the SMW and appended with `.qcow2`. To display the image name, use `glance image-list`.
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 `fixed_ip` address is only available in the `elogin-env-fixed-ip.yaml.template`.
instance_flavor	Nova flavor of the eLogin node being booted. In most cases, use `eloginflavor`.
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 `ironic node-list` 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.

3. Create a Heat template.

   The `elogin-env*.yaml` files work in concert with a base eLogin Heat template file. There are four baseline template variants. It is important to use the correct eLogin Heat template file. The differences between the eLogin Heat templates are whether-or-not to use a fixed IP address on the management network.

   The following table maps eLogin Heat template files with the `elogin-env*.yaml` files.

Table 3. eLogin Heat Template File Mapped with eLogin-env Template File

eLogin-env*.yaml File	eLogin Heat Template File
`elogin-env.yaml.template`	`elogin_template.yaml`
`elogin-env-fixed-ip\.yaml.template`	`elogin_template_fixed_ip.yaml`

a. Create the `elogin-env*.yaml` files for each node.

b. Copy and edit the `deploy-elogin.sh.template` file, located at: `/etc/opt/cray/openstack/heat/templates` to `/etc/opt/cray/openstack/elogin/deploy-<elogin_hostname>.sh`.

4. Set the `TEMPLATE_FILE`, `ENV_FILE`, and `STACK_NAME` to the proper Heat template, env file and hostname, and respectively edit into the deploy script.

   For example, for eLogin named elogin1:

```
cmc# cd /etc/opt/cray/openstack/elogin/

cmc# cp /etc/opt/cray/openstack/heat/templates/ \
deploy-elogin.sh.template ./deploy-elogin1.sh
```

5. Deploy the eLogin by running the deploy script.

```
cmc# cd /etc/opt/cray/openstack/elogin/

cmc# ./deploy-elogin1.sh
```

6. Monitor the deploy to watch Heat and Nova, as follows:

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

cmc# watch -n 5 heat stack-list

cmc# watch -n 5 nova list
```

7. Watch the console as follows (example, using `elogin1`):

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

cmc# ironic_conman elogin1
```

13 eLogin Troubleshooting

This section includes the following eLogin troubleshooting issues:

- Disk Space on CMC and eLogin Node
- Recover from Broken CSMS Installation
- Repeated Cycle Rebooting CentOS Deploy Image
- Configure Fuel Rsync Bandwidth Limit
- Restore Simple Sync UP01 Failures

13.1 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.

13.2 Recover from a Broken CSMS Installation

About this task

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

WARNING: If this is not an initial installation, check with Cray support before proceeding as this will remove all existing OpenStack content.

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

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cinder             |
| glance             |
| heat               |
| hssds              |
| ironic             |
| keystone           |
| mysql              |
| neutron            |
| neutron_ovs        |
| nova               |
| performance_schema |
| swift              |
| test               |
+--------------------+
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
```

13.3 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 Ironic 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 Ironic 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 `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.

   ```
   cmc# source ~/admin.openrc
   cmc# 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** (FFV) 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).

13.4 Configure the Fuel Rsync Bandwidth Limit

Prerequisites

This procedure requires root privileges.

About this task

On systems with a larger number of nodes, limiting the bandwidth may be crucial for a successful deployment. When facing problems during node deployment with Fuel, perfom the following steps to limit the bandwidth.

The Fuel driver is controlled by a mandatory JSON file named `cloud_default_deploy_config` and an optional (but recommended) JSON file named `deploy_config`, which appends the default and overwrites on conflict. Fuel will automatically refer to `cloud_default_deploy_config` for all deployments as specified in /etc/kolla/config/ironic.conf, whereas deploy_config is referenced on a node by node basis using nova boot or via a heat template. Thus it is highly recommended to put any node-specific or image-specific details into deploy_config instead of cloud_default_deploy_config.

Procedure

1. Download the existing Fuel cloud default deploy configuration.

```
cmc# glance image-download cloud_default_deploy_config \
--file /tmp/cloud_default_deploy_config
```

2. Edit the rsync flags.

```
cmc# sed -i.bak \
-e 's/"-a -A -X --timeout 300"/"-a -A -X --timeout 300 --bwlimit 12500"/g' \
/tmp/cloud_default_deploy_config
```

3. Validate the JSON structure.

   The command shouldn't print anything if the file contains valid JSON.

```
cmc# python -m json.tool /tmp/cloud_default_deploy_config > /dev/null
```

4. Delete the old cloud deploy configuration image.

```
cmc# glance image-delete cloud_default_deploy_config
```

5. Upload the new cloud deploy configuration image.

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

6. Download the cloud default deploy configuration image in order to check that the flag has been applied.

```
cmc# glance image-download cloud_default_deploy_config
       {
           "image_deploy_flags": {
              "rsync_flags": "-a -A -X --timeout 300 --bwlimit 12500"
           }
       }
```

   The values given above represent only an example and have to be adjusted to the actual deployment. Limiting the bandwidth too much may result in a timeout, which can make the deployment of all nodes fail. To avoid timeouts due to limited bandwidth, it may be necessary to increase the value of deploy_timeout in the [fuel] section of the Ironic configuration file. After changing this option, restart the Ironic conductor service by running:

```
cmc# systemctl restart openstack-ironic-conductor
```

13.5 Restore Simple Sync UP01 Failures

Prerequisites

This procedure assumes the system has eLogin UP01 release installed.

About this task

This procedure fiixes an existing issue with `simple_sync` in the UP01 release, where two different versions are implemented between the CLE and eLogin. This issue was resolved in UP02 but the following failures are present in UP01 installations during the deployment of eLogin nodes.

```
2016-07-19 13:27:14 TASK: [simple_sync | task main, list types] ***********************************
2016-07-19 13:27:14 failed: [localhost] => {"changed": true, "cmd": "find /etc/opt/cray/config/ \
current/files/roles/simple_sync -mindepth 2 -maxdepth 2", "delta": "0:00:00.016379", "end": }
2016-07-19 13:27:14 stderr: find: '/etc/opt/cray/config/current/files/roles/simple_sync': \
No such file or directory
2016-07-19 13:27:14
2016-07-19 13:27:14 FATAL: all hosts have already failed -- aborting
2016-07-19 13:27:14
2016-07-19 13:27:14 PLAY RECAP ********************************************************************
2016-07-19 13:27:14            to retry, use: --limit @/site.yaml.retry
2016-07-19 13:27:14
2016-07-19 13:27:14 localhost : ok=110  changed=66   unreachable=0    failed=1
2016-07-19 13:27:14
2016-07-19 13:27:14 Failed Ansible configuration
```

The problem stems from a directory structure difference between `simple_sync` v1 and v2. In this case, CLE nodes were moved to v2 before eLogin was compatible with V2. This created a situation where eLogin nodes cannot locate the files to sync. This issue was resolved in the eLogin UP02 release, where `simple_sync` v2 is supported and installed.

Simple Sync v1

The following `simple_sync` v1 directories usable by eLogin, are in the config set rooted at: `/var/opt/cray/imps/config/sets/`*`config_set_name`*`/files`.

- `roles/simple_sync/classes/common`
- `roles/simple_sync/hostnames/`*`hostname`*`/`

All files under the above paths will be synced to the eLogin, rooted at: ``/``. Files under `roles/simple_sync/hostnames/<hostname>` are only synced to the eLogin whose hostname matches <*hostname*>. These files override any files on the same path as those under the `roles/simple_sync/classes/common` directory.

Simple Sync v2

The `simple_sync` v2 directories changed position and name. The following are the usable directories for eLogins located at the same path root as v1: `/var/opt/cray/imps/config/sets/`*`config_set_name`*`/files`.

- `simple_sync/common/files`
- `simple_sync/hostname/`*`hostname`*`/files`

All files under the above paths will be synced to the eLogin, rooted at: ``/``. Files under `roles/simple_sync/hostnames/<hostname>` are only synced to the eLogin whose hostname matches <*hostname*>. These files override any files on the same path as those under the `roles/simple_sync/classes/common` directory.

Restore Directory Structure for Simple Sync

The directory structure for `simple_sync` v1 must be restored, and all files to sync moved to this v1 directory structure.

Perform this procedure to restore the directory structure for Simple Sync:

Procedure

1. Create the `simple_sync` v1 paths in the `config/sets` directory on the SMW.

   The following example uses a config set named `p0`.

   ```
   smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/roles/ \
   simple_sync/classes/common

   smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/ \
   roles/simple_sync/hostnames
   ```

2. Push the config set to the CMC.

   ```
   smw# cfgset push -d cmc-name config_set_name
   ```

3. Log into the CMC, and run `add_configset`.

   ```
   cmc# add_configset -c config_set_name \
   -e /etc/opt/cray/elogin/exclude_lists/elogin_cfgset_excludelist
   ```

4. Place the files you intend to `simple_sync` under these new paths.

   - `/var/opt/cray/imps/config/sets/p0/files/roles/simple_sync/classes/common`
   - `/var/opt/cray/imps/config/sets/p0/files/roles/simple_sync/hostnames`