XC™ Series eLogin Administration Guide (CLE 6.0.UP02) S-2570 Rev A

Contents

1 About the XC Series eLogin Administration Guide

The XC™ Series eLogin Administration Guide provides concepts and tasks for Cray eLogin nodes running externally to Cray XC series systems.

Audience and Scope

This publication is intended for system installers, administrators, and anyone who installs and configures software on a Cray XC Series system. Use of the term *user* throughout refers to the intended audience, not to end users of the system. This publication assumes competence with standard Linux and open source tools.

Record of Revision

Revision	Date	Content Information
XC Series eLogin Administration Guide CLE6.0 UP02 Rev A (this publication)	11/14/2016	**Final UP02 release Rev A**
XC Series eLogin Guide CLE6.0 UP02	10/31/2016	CLE6.0 UP02 - final release
CLE6.0 UP02 (draft v.2)	10/28/2016	UP02 v.2 draft (pre-final)
CLE6.0 UP02 (draft v.1)	10/18/2016	UP02 draft for review. ● CSMS 1.1.3 / CentOS 7.2 (default install) ● Upgrade/update processes included
XC Series eLogin Administration Guide CLE6.0 UP01	06/15/2016	Final UP01 release ● Content aligned with CSMS 1.1.1 / CentOS 7.1 ● Upgrade processes not included
CLE6.0 UP01 v.1	06/2/2016	Pre-release for final review ● Content aligned with CSMS 1.1.1 / CentOS 7.1 ● Upgrade processes not included

This publication was previously titled eLogin Administrtation Guide CLE6.0 UP01. The new title XC-Series eLogin Administration Guide CLE6.0 UP02 complies with the standard titling convention adopted and implemented for all publications within the technical publications department as of June 10, 2016. Previous versions of this publication will not be retitled.

Typographic Conventions

`Monospace`	A `Monospace` font indicates program code, reserved words or library functions, screen output, file names, path names, and other software constructs
`Monospaced Bold`	A **`bold monospace`** font indicates commands that must be entered on a command line.
`Oblique` or *`Italics`*	An *`oblique`* or *`italics`* font indicates user-supplied values for options in the syntax definitions
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

Feedback

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

Trademarks

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.

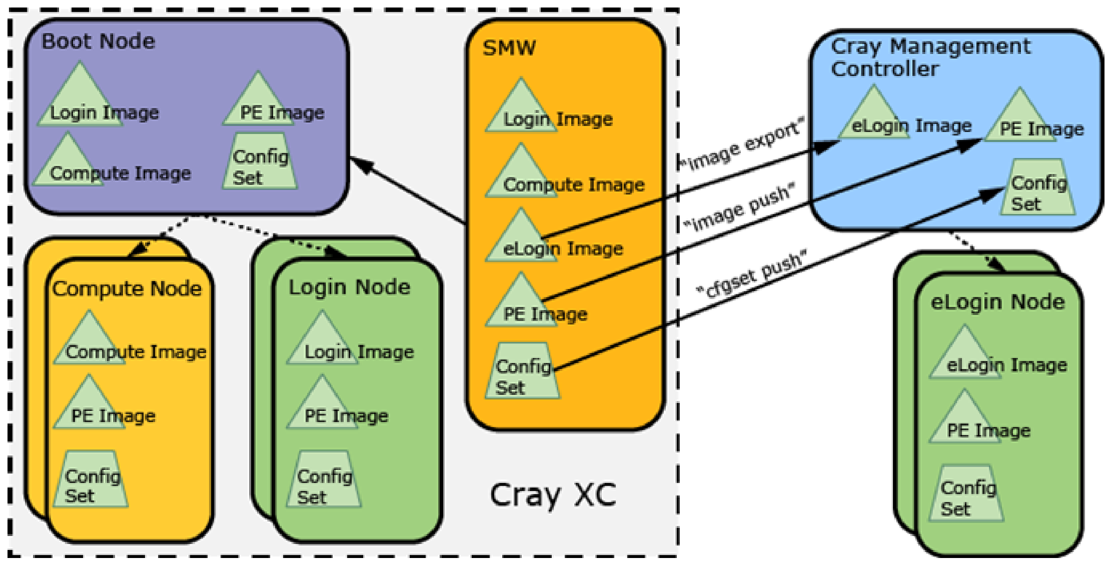
2 eLogin Architecture

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 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, 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.

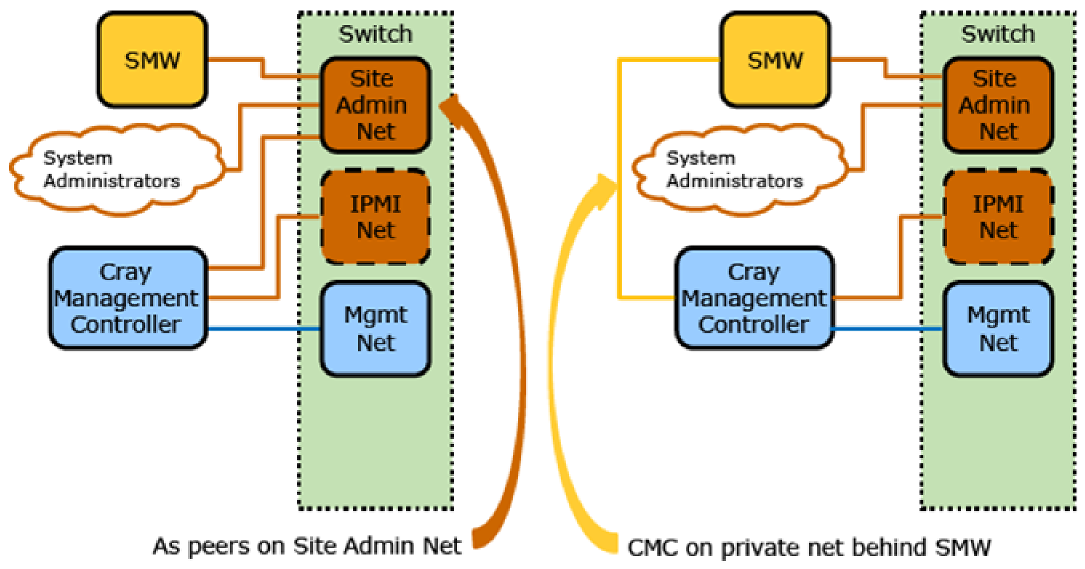
WARNING: For security reasons, configure the CMC behind the SMW on a private network.

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.

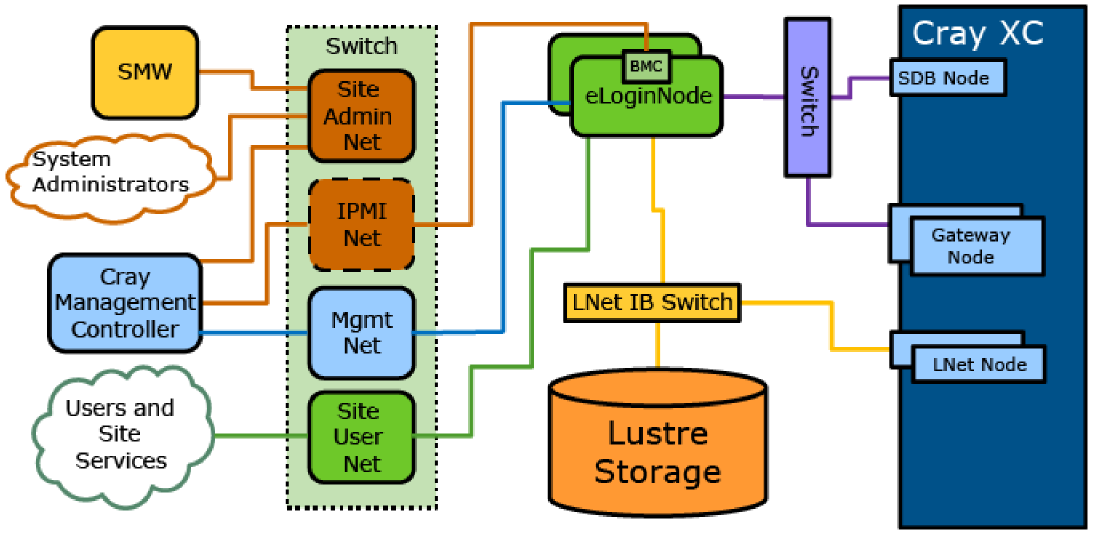
eLogin to Cray Network Attachment

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.

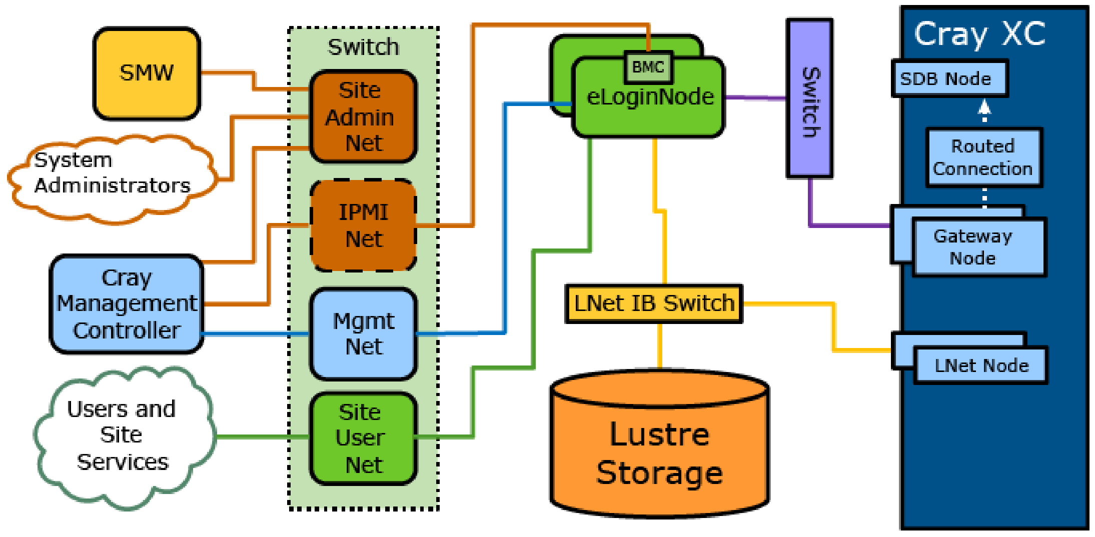
Figure 3. eLogin Nodes Direct Connection to SDB Node: eLogin System Topology



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.

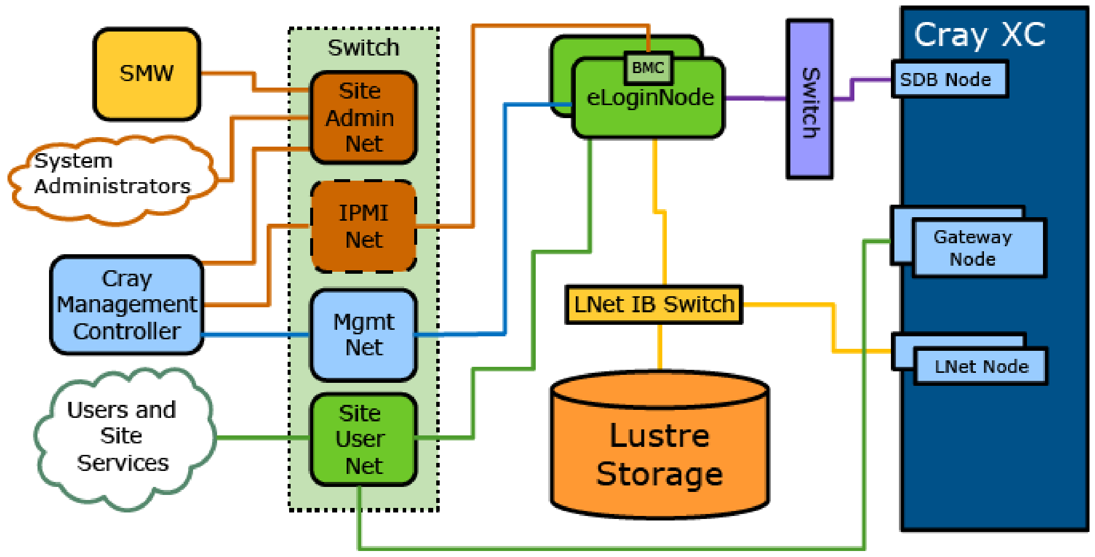
Figure 4. eLogin Nodes Routed Via Gateway to SDB Node: eLogin System Topology



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.

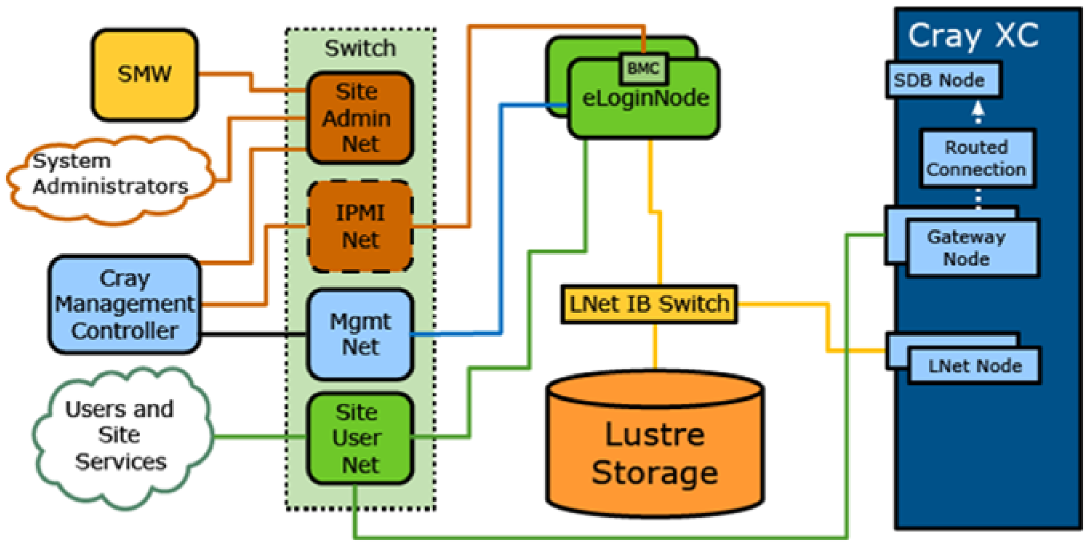
Figure 5. eLogin Nodes Direct Connection to SDB Node with Site User Gateway: eLogin System Topology



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.

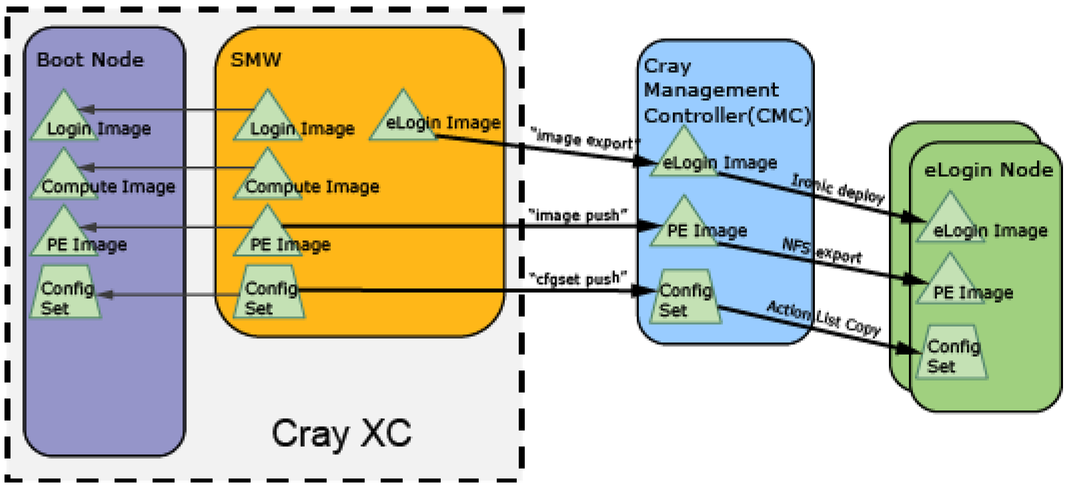
Figure 6. eLogin Nodes Routed to SDB Node via User Accessible Gateway: eLogin System Topology



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

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* section of the *XC Series eLogin Installation Guide CLE 6.0 UP02 Rev A*.

4 Configure and Manage an eLogin Image

Prerequisites

A complete successful eLogin installation.

About this task

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

Cray recommends appending images with '-*YYYYMMDD*'. For example, if generating `elogin-large_cle_6.0up01_sles_12_x86-64_ari` on June 1st, 2015, the image should be named `elogin-large_cle_6.0up01_sles_12_x86-64_ari_20150601`. These image names match the naming scheme of the internal login image, with eLogin prepended.

SMW Image Creation and Export

Procedure

1. Connect to the SMW.

```
# ssh 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 only to be used for test. (In which case, the smaller image allows for shorter boot times.)

3. Optional: Create a custom eLogin image recipe.

Perform this step if either one of these conditions apply:

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

Create a new eLogin image recipe by cloning `elogin-large_cle_6.0up02_sles_12_x86-64_ari_20150601`. Prepend the function of the customization to the original user name of a custom image (example, *`username-function`*).

```
root@smw# recipe create custom-elogin-large_cle_6.0up02_sles_12_x86-64_ari
root@smw# recipe update -r elogin-large_cle_6.0up02_sles_12_x86-64_ari \
custom-elogin-large_cle_6.0up02_sles_12_x86-64_ari
```

4. Build the eLogin image.

```
root@smw# image create -r custom-elogin-large_cle_6.0up02_sles_12_x86-64_ari \
custom-elogin-large_cle_6.0up02_sles_12_x86-64_ari-YYYYMMDD
```

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

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

6. Create two environment variables, one for CMC name, and one for eLogin ISO image.

```
root@smw# CMCNAME=panda-cmc
root@smw# IMAGE=custom-elogin-large_cle_6.0up02_sles_12_x86-64_ari-YYYYMMDD
```

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. For a large image, the estimated time to complete is half an hour.

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

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

```
root@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 step 6.)

   The config set was generated during CLE installation and modified in *Configure Minimum Services Required for eLogin*. Refer to the *XC Series eLogin Installation Guide CLE 6.0 UP02 Rev A*.

```
root@smw# cfgset push -d $CMCNAME global
root@smw# cfgset push -d $CMCNAME <config_set_name>
```

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

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

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.

```
root@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.

```
root@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. Connect to the SMW.

```
# ssh root@smw
```

b. Push the config set changes to the CMC. This allows the CDL to access the changes. (Use environment variable `$CMCNAME`.)

```
root@smw# image push -d $CMCNAME <config_set_name>
```

c. Connect to the CMC.

```
# ssh root@cmc
```

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

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

> **IMPORTANT:** In general, after the config set changes are pushed to the CMC, the config set should 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.

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

```
root@cmc# heat stack-list
```

Run steps (f., g., and h.) only in the circumstance where the Heat stack is deployed.

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

```
root@cmc# heat stack-delete <stack_name>
```

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

```
root@cmc# heat stack-list
```

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

```
root@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.

```
root@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.

```
root@cmc# cp copy_p0.template copy_<config_set_name>
```

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

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

```
"args": "-pxzvf /tmp/<configset_name>_configset.tar.gz -C /mnt/",
"url": "swift:<configset_name>_configset/v<configset_name>_configset.tar.gz",
"target": "/tmp/<configset_name>_configset.tar.gz"
```

d. Add the action list to Glance.

```
root@cmc# glance image-create --is-public True \
--disk-format raw --container-format bare --name copy_<config_set_name>\
--file copy_<config_set_name>
```

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

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`.

```
root@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`.

```
root@cmc# cp <chosen-template> <elogin_name>-env.yaml
```

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

```
root@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`.

```
root@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 13B, 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.

```
root@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.

   ```
   root@cmc# ironic node-list
   ```

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

   ```
   root@cmc# ironic_conman <ironic_name>
   ```

For more details, refer to *eLogin Console Access*, in *XC Series eLogin Installation Guide CLE 6.0 UP02 Rev A*.

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.

```
root@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 "`@.`"

15. Repeat the previous step for each eLogin node.

5 Upgrade Process for CSMS and CentOS

5.1 Prepare to Upgrade

Prerequisites

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

About this task

Before upgrading the CSMS or operating system (OS), specific settings must be made, as described in this procedure.

Procedure

1. 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.

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

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

   ⚠ **CAUTION:** By disabling the CentOS base, extras and update yum repositories, the system is prevented from installing updated CentOS packages during the installation update, or when performing a yum update operation.

4. Ensure all nodes are in a state where deploy images can be updated without problems.

   ```
   root@cmc# ironic node-list
   ```

   Any state other than available or active could potentially lead to problems during update (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:

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

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

5. Execute the `timedatectl` command to verify that in the output the `Universal time` field correctly displays the current time (in UTC), accurate to within a second, and that the `RTC time` field matches `Universal time`.

   The system is configured to read the RTC time in the local time zone, and is not fully supported. The RTC time is never updated, and relies on external facilities to maintain it. This step verifies that the system Real Time Clock (RTC) is synchronized to Universal Time (UTC).

   Following is an example of a valid output:

```
root@cmc# timedatectl
         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:

```
root@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.

6. Create a backup of certificates.

```
root@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`

7. Exit the SSH session.

```
root@cmc# exit
```

Proceed to the upgrade process.

5.2 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.
- 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.

   ```
   root@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:

     ```
     root@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:

     ```
     root@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.

   ```
   root@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

```
root@cmc# cd /etc/opt/cray/openstack/ansible/
root@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:

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

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

```
root@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:

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

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

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

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

```
root@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.

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

9. Unmount the CSMS specific CentOS 7.2 ISO

```
root@cmc# cd /root
root@cmc# umount /mnt/csms
root@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. In addition, it is recommend, 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.

```
root@cmc# find /etc | grep rpmsave
root@cmc# find /etc | grep 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

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

12. Restore all certificates.

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

13. Remove the certificate backup file.

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

14. Restart all OpenStack services.

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

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

- If using an Ansible Vault password file, execute:

```
root@cmc# cd /etc/opt/cray/openstack/ansible/
root@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.

```
root@cmc# openstack-status
```

17. Reboot the CMC.

```
root@cmc# reboot
```

18. Proceed to the *Update Fuel Deployment Images* procedure.

5.3 Upgrade CSMS 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. Refer to *Upgrade CentOS 7.1 to 7.2 for eLogin* procedure.
- *Prepare to Upgrade* procedure is completed.
- 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. Existing deployed workloads will 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.

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

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

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

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

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

```
root@cmc# mkdir /mnt/csms
```

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

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

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

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

8. Run the installer.

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

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

9. Unmount the CSMS specific CentOS ISO.

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

```
root@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` will check that only Cray provided repositories are enabled before proceeding with the installation and configuration of CSMS software

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

12. Restore all certificates.

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

13. Remove the certificate backup file.

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

14. Restart all OpenStack services.

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

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

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

```
root@cmc# openstack-status
```

17. Proceed to the *Update Fuel Deployment Images* procedure.

6 Component Updates

6.1 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 on system using CentOS 7.1 or 7.2
- Upgrade from CentOS 7.1 to CentOS 7.2 on system using CSMS 1.1.3.

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.

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

root@cmc# NODE=node_name
root@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 which use the Fuel driver, skip the remainder of this section.

2. Check if Fuel is in use on the system, and if it is, update the `deploy_config` file to take syntax and feature changes into account.

   If Fuel is NOT used on the system, stop the procedure here. If Fuel IS used, perform the next section and continue to the end.

   Update Deploy Images and Ironic Node Attributes (If Fuel is used on the system)

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

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

4. Source the `admin.openrc` file.

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

5. Create new Glance images for the updated Fuel deployment images.

```
root@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

root@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
```

6. Update existing Ironic nodes to use the patched Fuel deploy images.

   Perform the following command line steps for each Ironic node found in step 1. Replace `KERNEL_ID` and `RAMDISK_ID` with the corresponding site-specific values.

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

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

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

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

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

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

   The Fuel and PXE deploy images are now updated and ready for use the next time the Ironic node(s) are deployed.

   If nodes are deployed via Heat, this occurs on a per-node basis after the `heat stack-create` or `heat stack-update` command is executed.

   If nodes are deployed with Nova, this occurs on a per-node basis after the `nova boot` or `nova rebuild` command is executed.

6.2 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.

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

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

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

3. Unmount the ISO root.

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

4. Run any eLogin-specific Ansible playbooks.

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

7 Component Upgrades

7.1 Upgrade SMW for eLogin

Prerequisites

Cray SMW Software Upgrade/Update procedures. Refer to upgrade/update section of *SMW Software Installation Guide*.

About this task

This procedure describes how to upgrade the SMW software for eLogin.

The SMW stores updates to product (eLogin) ISO images, and then creates the config template, config set, and config worksheet. When the SMW is upgraded, two actions are required:

- The eLogin image must be exported from the SMW via Glance within the Cray System Management Software (CSMS).
- The config set must be pushed to the CMC.

Procedure

1. Upgrade/update the SMW software (if not done). Refer to upgrade/update section of *SMW Software Installation Guide*.

2. Export the updated eLogin image from the SMW to Glance within the CSMS.

   WARNING: Glance allows multiple images with the same name to be stored in the CSMS, but can only deploy an image with a unique name.

   Ensure that the eLogin image being pushed is unique. If reusing an image name, remove that image on the CMC before pushing the new image from the SMW.

   ```
   root@smw# image export --format qcow2 -d glance \
   custom-elogin_cle_6.0up02_sles_12_x86-64_ari-YYYYMMDD
   ```

3. Push the config set to the CMC.

   ```
   root@smw# cfgset push -d <cmc-name> global
   root@smw# cfgset push -d <cmc-name> <config_set_name>
   ```

7.2 Upgrade PE for eLogin Node

Prerequisites

Cray SMW software upgrade/update procedures. Refer to upgrade/update section of *SMW Software Installation Guide*.

About this task

Background info for task.

Procedure

1. Upgrade/update the SMW software (if not done). Refer to upgrade/update section of *SMW Software Installation Guide*.

2. Push the PE image to the CMC.

```
root@smw# image push -d <cmc-name> <pe_image_root>
```

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

```
root@cmc# source admin.openrc
root@cmc# nova reboot <eLogin_name>
```

7.3 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.

```
root@smw# cfgset push -d <cmc-name> global
root@smw# cfgset push -d <cmc-name> <config_set_name>
```

> **ATTENTION:** If only updating config sets, proceed to step 3. If the image is also being updated, skip step 3, and proceed to step 4.

3. Add the config set to the eLogin node.

`NODE_HOSTFILE` is a file with a list of nodes where the config set should be updated.

```
root@cmc# add_configset -u <NODE_HOSTFILE>
```

4. Edit the exclude list file as needed 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. It is likely that key components of the config set are disabled by default. In our experience, it is often necessary to enable munge and SSH keys. These filters can be enacted at a file-by-file level. Any changes should be 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
```

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

```
root@smw# cfgset push -d <cmc-name> <config_set_name>
```

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

```
root@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, the node should first be shut down by deleting the Heat stack, and then redeployed 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.

```
root@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 7.

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

```
root@cmc# heat stack-delete <stack_name>
```

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

```
root@cmc# heat stack-list
```

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

```
root@cmc# /etc/opt/cray/openstack/heat/templates/deploy_<elogin_name>.sh
```

Perform step 7 only if no Heat stack was listed as deployed in step 6a.

7. Create the Heat stack.

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

```
root@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.

```
root@cmc# ironic node-list
```

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

```
root@cmc# ironic_conman <ironic_name>
```

For more details, refer to *eLogin Console Access*, in *XC Series eLogin Installation Guide CLE 6.0 UP02 Rev A*.

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.

```
root@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 "`@.`"

8. Repeat the previous step for each eLogin node.

7.4 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.

   ```
   root@cmc# ./update_<eLogin_name>.sh
   ```

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

8 Administrator Tasks

This section provides information on how to perform the most common system administrative operations for CDL nodes. These tasks are performed on the Cray Management Controller (CMC).

8.1 Change Site Admin IP Address of Management Controller

About this task

This procedure updates the Site-Admin Network and Site-Admin IP address to localize the Cray Management Controller (CMC) on site.

Procedure

1. Gain access to the console of the CMC by either connecting a keyboard, mouse, and monitor to the machine, or through the Baseboard Management Controller (BMC). Note that Serial-over-LAN access is not sufficient.

2. Edit the site-overrides file located in the `/etc/opt/cray/openstack/ansible/config/site` directory.

   Change the following settings:

   a. `site_ip`: Enter the new Site Admin IP address.

      System administrators use the Site Admin IP address to access the CMC.

   b. `site_subnet`: Enter the new Site-Admin-Network information.

   c. Update the prefix and gateway values for the site under the subnets array:

      ```
      subnets:
      - name: site
        physical_network: "{{ physical_networks.site }}"
        address: "{{ site_subnet }}"
        prefix: 22
        gateway: 172.30.12.1
      ```

      WARNING: Do not modify the `default_gateway` value at this time. A change to the `default_gateway` value will cause the update to fail.

3. Remove the `br-int` interface to avoid a possible packet storm:

   ```
   root@cmc# ovs-vsctl del-br br-int
   ```

4. Run `csms_install.sh`.

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

5. Edit the site-overrides file located in the `/etc/opt/cray/openstack/ansible/config/site` directory.

   In the `default_gateway` settings, enter the default gateway address for the new Site-Admin-Network.

6. Remove the `br-int` interface to avoid a possible packet storm:

   ```
   root@cmc# ovs-vsctl del-br br-int
   ```

7. Run `csms_install.sh`.

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

8. Verify that the Admin Logins work on the new IP address.

8.2 Changing Passwords On CMC

The following passwords may be changed after the initial Cray System Management Software (CSMS) installation:

- Admin password
- Ansible Vault files
- Root MariaDB database account
- Linux root user account

Passwords and secrets for the following are auto-generated when the `csms_gen_creds.py` script is executed:

- MariaDB database accounts for OpenStack components
- OpenStack Keystone user accounts for OpenStack components
- RabbitMQ user accounts for OpenStack components
- Neutron metadata proxy shared secret
- Monasca InfluxDB password
- Swift hash path prefix and suffix
- Swift temporary URL key for Ironic Fuel agent deployment

Changing passwords for the aforementioned items is currently not supported. If there is an urgent need to change these passwords, please contact Cray Support.

8.2.1 Change Admin Password After CSMS Installation

Prerequisites

Successful installation of CSMS.

Procedure

1. Update the Keystone admin user password via the Keystone CLI as root on the CMC:

   ```
   root@cmc# keystone user-password-update admin
   ```

2. Switch to the `/etc/opt/cray/openstack/ansible` directory, and update the Keystone MariaDB database account password using the following commands. Enter the new password when prompted.

   ```
   root@cmc# cd /etc/opt/cray/openstack/ansible
   root@cmc# export OS_PASSWORD=$(./getpass.sh "admin")
   Enter admin password:
   root@cmc# ./csms_install.sh
   Vault password:
   ```

3. Verify that keystone is functioning correctly by retrieving a list of users.

   ```
   root@cmc# keystone user-list
   ```

8.2.2 Change Ansible Vault Password After CSMS Installation

Prerequisites

Successful installation of CSMS.

About this task

Ansible Vault is used to encrypt files that contain secure information such as passwords. The following files are encrypted and their passwords should remain synchronized:

```
/etc/opt/cray/openstack/ansible/group_vars/all/service_passwords
/etc/opt/cray/openstack/ansible/vars/credentials.yaml
```

To change the Ansible Vault password, do the following:

Procedure

1. Rekey the Ansible Vault encrypted files as root and enter the old and new passwords when prompted. The default password is `initial0`.

   ```
   root@cmc# cd /etc/opt/cray/openstack/ansible
   root@cmc# ansible-vault rekey group_vars/all/service_passwords vars/
   credentials.yaml
   Vault password:
   New Vault password:
   Confirm New Vault password:
   Rekey successful
   ```

2. Replace `$PASSWORD` with the new password (only if the Vault password was written to a file). This ensures that an Ansible Vault password file (created when the `./csms_gen_creds.py` script is executed) is updated to contain the new password.

```
root@cmc# cd /etc/opt/cray/openstack/ansible
root@cmc# ls vault-password.txt
root@cmc# echo $PASSWORD > vault-password.txt
```

8.2.3 Change Root MariaDB Database Password After CSMS Installation

Prerequisites

Successful installation of CSMS.

About this task

The root MariaDB account is used to manage all other database accounts.

CAUTION: This is an advanced procedure that involves manually editing a YAML file and should be performed with care. Cray recommends to make a backup of the `service_passwords` file before editing it.

To change the root MariaDB database password, do the following:

Procedure

1. Make a backup of the encrypted `service_passwords` file.

```
root@cmc# cd /etc/opt/cray/openstack/ansible
root@cmc# cp group_vars/all/service_passwords service_passwords_backup
```

2. Make a note of the current MariaDB root account password. This is stored via the `mysql_root_password` variable in the service passwords YAML file. Enter the Ansible Vault password when prompted.

```
root@cmc# cd /etc/opt/cray/openstack/ansible
root@cmc# ansible-vault view group_vars/all/service_passwords
Vault password:
```

3. Change the password by using the `mysqladmin` tool. Replace `$PASSWORD` with the old password and `$NEWPASSWORD` with the new password.

```
root@cmc# mysqladmin -u root -p'$PASSWORD' password '$NEWPASSWORD'
```

4. Verify that the password was changed successfully. Replace `$PASSWORD` with the new password.

```
root@cmc# mysqladmin -u root -p'$PASSWORD' ping mysqld is alive
```

5. Edit the Ansible service passwords YAML file by changing the `mysql_root_password` variable to reflect the new password.

```
root@cmc# cd /etc/opt/cray/openstack/ansible
root@cmc# ansible-vault edit group_vars/all/service_passwords
```

Perform the next step only if Monitoring is enabled. Skip the next step if Monitoring is disabled.

6. (Conditional) Configure the OpenStack Monasca Agent to use the new MariaDB root password (if Monitoring is enabled).

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

7. Delete the backup of the encrypted `service_passwords` file.

```
root@cmc# cd /etc/opt/cray/openstack/ansible
root@cmc# rm service_passwords_backup
```

8.2.4 Change the Linux Root User Password After CSMS Installation

Prerequisites

Successful installation of CSMS.

About this task

The Linux root-user account is used to perform many administrative tasks in the CSMS. The password for this account can be changed using the standard Linux method.

To change the Linux root-user password, do the following:

Procedure

1. Execute the `passwd` command as root, and then enter the new password when prompted.

```
root@cmc# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

2. Verify that the password was changed successfully by logging out, logging in, and then entering the new password when prompted. For example, if logged in via SSH:

```
root@cmc# exit
logout
user@my-host# ssh root@cmc
root@cmc’s password:
```

If the password was changed successfully, the preceding command succeeds and the system allows the user to log-on to the Cray management controller (CMC).

8.3 Deploy, Reboot, and Shutdown eLogin Nodes

8.3.1 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:

```
root@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`).

   ```
   root@cmc# cd /etc/opt/cray/openstack/elogin/
   root@cmc# cp /etc/opt/cray/openstack/heat/templates/ \
   elogin-env-fixed-ip.yaml.template ./elogin1-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 1. 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:

```
root@cmc# cd /etc/opt/cray/openstack/elogin/
root@cmc# cp /etc/opt/cray/openstack/heat/templates/ \
deploy-elogin.sh.template ./deploy-elogin1.sh
```

5. Deploy the eLogin by running the deploy script.

```
root@cmc# cd /etc/opt/cray/openstack/elogin/
root@cmc# ./deploy-elogin1.sh
```

6. Monitor the deploy to watch Heat and Nova, as follows:

```
root@cmc# source /root/admin.openrc
root@cmc# watch -n 5 heat stack-list
root@cmc# watch -n 5 nova list
```

7. Watch the console as follows (example, using `elogin1`):

```
root@cmc# source /root/admin.openrc
root@cmc# ironic_conman elogin1
```

8.3.2 Reboot an eLogin Node

Prerequisites

eLogin node is in shut down state.

About this task

Run the `nova reboot` command to reboot an eLogin node (for example, `elogin1`).

```
root@cmc# nova reboot elogin1
```

To change an attribute of the eLogin node, such as the image, shut down and redeploy the eLogin node.

8.3.3 Shut Down eLogin Node

Prerequisites

eLogin node is in powered on state.

Procedure

1. Log into the eLogin node to be shut down.

2. Run the `shutdown -H` command, to shutdown the eLogin node.

   ```
   root@elogin# shutdown -H
   [wait for shutdown to complete]
   ```

3. Run the `heat stack-delete` command from the CMC to delete the Heat stack from the shutdown eLogin node. (Example, use `elogin1` as eLogin name.)

   ```
   root@cmc heat stack-delete elogin1
   ```

8.4 Update Config Set on Running Nodes

Prerequisites

System is running and eLogin nodes are deployed.

About this task

This procedure is a method to update the configuration of a running node, or set of nodes. There are risks involved, however, so this procedure should be used with caution. All configuration changes are handled via the configuration set (config set) mechanism.

Procedure

1. Update the config set on the SMW (if not already completed). For instruction, refer to: *SMW Software Installation Guide*.

2. Push the config set to the CMC.

```
root@smw# cfgset push -d <cmc-name> global
root@smw# cfgset push -d <cmc-name> <config_set_name>
```

3. 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
```

4. Update the running nodes in the hosts file.

```
root@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.

8.5 Maintenance Mode

Prerequisites

- CSMS is installed and configured on system.
- eLogin node is runnning.

About this task

The Cray System Management Software (CSMS), specifically the Ironic service, ensures that a node is in the desired power state. This means the CSMS overrides the manual pressing of the power button switch on an eLogin node. For example, if the the eLogin node is powered on (Regular Service Mode), a user may press the power button on the node to power it down for maintenance. The CSMS (Ironic service) then detects the power is off, and responds by powering the node back on. The same is true for the reverse case. Setting the eLogin node into Maintenance Mode ensures that the eLogin node is ignored by the CSMS.

To set an eLogin node (example, *elogin1*) into Maintenance Mode, do the following:

Procedure

1. Set the maintenance mode **On** for an eLogin node (*elogin1*).

```
root@cmc# ironic node-set-maintenance eLogin1 on
```

   The eLogin node (*elogin1*) is now ignored by the CSMS.

2. Set the maintenance mode **Off** to return the eLogin node (*elogin1*) to regular service.

```
root@cmc# ironic node-set-maintenance eLogin1 off
```

The eLogin node (*`elogin1`*) is now running in Regular Service Mode and detected by the CSMS.

8.6 Connect eLogin Nodes to a Lustre File System

Procedure

1. Ensure that the first InfiniBand interface is physically connected to the Lustre server.

2. Verify that the following settings in the `cray_elogin_lnet` config set are correct for the site.

```
root@smw# cfgset search -s cray_elogin_lnet config_set

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

root@smw# cfgset search -s cray_net <cfgset>
...
cray_net.settings.networks.data.lnet.description: Infiniband network to external Lustre
cray_net.settings.networks.data.lnet.ipv4_network: 10.149.0.0
cray_net.settings.networks.data.lnet.ipv4_netmask: 255.255.0.0
cray_net.settings.networks.data.lnet.ipv4_gateway: # (empty)
cray_net.settings.networks.data.lnet.dns_servers: # (empty)
cray_net.settings.networks.data.lnet.dns_search: # (empty)
cray_net.settings.networks.data.lnet.ntp_servers: # (empty)
...
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.name: ib0
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.description: IB to External
Lustre
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.aliases: # (empty)
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.network: lnet
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.ipv4_address: 10.149.0.123
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.bootproto: static
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.mtu: # (empty)
cray_net.settings.hosts.data.example_elogin.interfaces.ib0.extra_attributes: # (empty)

root@smw# cfgset search -s cray_lustre_client -l advanced <cfgset>

# 9 matches for '.' from cray_lustre_client_config.yaml
#--------------------------------------------------------------------------------
cray_lustre_client.settings.module_params.data.libcfs_panic_on_lbug: True
cray_lustre_client.settings.module_params.data.ptlrpc_at_min: 40
cray_lustre_client.settings.module_params.data.ptlrpc_at_max: 400
cray_lustre_client.settings.module_params.data.ptlrpc_ldlm_enqueue_min: 260
cray_lustre_client.settings.client_mounts.data.rind1.mount_point: /lus/rind1
cray_lustre_client.settings.client_mounts.data.rind1.mgs_lnet_nids: 10.149.0.1@o2ib
cray_lustre_client.settings.client_mounts.data.rind1.mount_options: rw,flock,lazystatfs
cray_lustre_client.settings.client_mounts.data.rind1.mount_at_boot: True
cray_lustre_client.settings.client_mounts.data.rind1.mount_locations: login, compute,
elogin
```

   a. Ensure that the LNet IP wildcard in `cray_elogin_lnet` matches the LNet IPv4 network and netmask in `cray_net`.

   b. Verify that the IPv4 address for the eLogin `ib0` interface is unique within the LNet.

   c. Check that all Lustre mounts include `elogin` in the mount locations list.

   d. Update missing or incorrect settings.

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

e. Proceed to step *4* on page 47 if no updates are needed.

3. Push the config set to the cmc and reboot the affected eLogin nodes if any configuration settings were changed in the previous step.

```
root@smw# cfgset push -d cmc-name config_set
root@smw# ssh cmc-name
root@cmc# source admin.openrc
root@cmc# nova reboot elogin_node
```

4. Verify that Lustre functions correctly.

```
root@elogin# ip addr show ib0
4: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast state UP
group default qlen 256
       link/infiniband 80:00:00:48:fe:
80:00:00:00:00:00:00:00:02:c9:03:00:a4:5d:f1 brd 00:ff:ff:ff:ff:12:40:1b:ff:ff:
00:00:00:00:00:00:ff:ff:ff:ff
       inet 10.149.0.123/16 brd 10.149.255.255 scope global ib0
          valid_lft forever preferred_lft forever
       inet6 fe80::202:c903:a4:5df1/64 scope link
          valid_lft forever preferred_lft forever

root@elogin# lctl list_nids
10.149.0.123@o2ib
root@elogin# mount | grep lustre
10.149.0.1@o2ib:/rind1 on /lus/rind1 type lustre (rw,flock,lazystatfs)
```

8.7 User Authentication

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

LDAP Authentication

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

NIS Authentication

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

8.8 Miscellaneous Configuration Options

PBS License Server Configuration

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

AutoFS

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

8.9 Ironic Inventory File

Define the bare metal nodes in an inventory file before enrolling them. This file can be in one of two formats, either Comma Separated Values (CSV) or Yet Another Markup Language (YAML). The following fields may be specified (shown in CSV format):

Table 2. Ironic Inventory File (CSV format)

Field Name	Description
`NODE_NAME`	Node name.
`BMC_IP`	Baseboard Management Controller (BMC) IP address of the node.
`VIRTUALBOX_HOST`	IP or hostname of the VirtualBox web service used for the node.
`VIRTUALBOX_VMNAME`	Virtual machine name of the VirtualBox used for the node.
`SNMP_POWER_DRIVER`	SNMP power driver name for the node.
`SNMP_POWER_ADDRESS`	SNMP power IP address for the node.
`SNMP_POWER_OUTLET`	SNMP power outlet number for the node.
`SNMP_POWER_COMMUNITY`	SNMP power community for the node.
`PXE_OVERLAY`	PXE overlay name for the node.
`MAC_ADDR`	Node MAC address(es) (one or more). Multiple MAC addresses are specified by providing multiple MAC address fields in the header (and just as many MAC addresses in each node definition). Each of these fields may be prefixed (example, with `Management` or `HighSpeed`), where each prefix must match one of the MAC field `RegExs`. The MAC field `RegExs` are dyamically generated from the mapping given to the `-network` option of `inventory_convert.py`.
`N_CPUs`	Number of CPUs on the node.
`ARCH`	CPU architecture of the node.
`RAM_MB`	Size of node's RAM in MB.

Field Name	Description
`DISK_GB`	Size of the node’s primary disk in GB.
`NODE_CAPS`	A list of comma-separated `key:value` capability pairs. Use double quotes to avoid CSV intepreting commas.
`NODE_DESC`	Short description of the node.

The fields can be used in the following way:

Table 3. Ironic Inventory File (CSV format)

Field Name	Required	Multiple Allowed	Example Value
`NODE_NAME`	False	False	`My Node`
`BMC_IP`	False	False	`10.2.3.4`
`VIRTUALBOX_HOST`	False	False	`my.virtualbox.host`
`VIRTUALBOX_VMNAME`	False	False	`My Virtual Machine`
`SNMP_POWER_DRIVER`	False	False	`apc`
`SNMP_POWER_ADDRESS`	False	False	`10.3.0.3`
`SNMP_POWER_OUTLET`	False	False	`1`
`SNMP_POWER_COMMUNITY`	False	False	`private`
`PXE_OVERLAY`	False	False	`overlay_solarflare`
`MAC_ADDR`	True	True	`00:11:22:33:44:55`
`N_CPUs`	True	False	`8`
`ARCH`	True	False	`x86_64`
`RAM_MB`	True	False	`8192`
`DISK_GB`	True	False	`1024`
`NODE_CAPS`	False	False	`“oss:true,name:bert”`
`NODE_DESC`	False	False	`chassis 1 node 2`

The field name is a variable itself. Each field name must match its RegEx given below:

Table 4. Ironic Inventory File Syntax

`Field Name`	Name RegEx (case sensitive)
`NODE_NAME`	`^(node)?[ _]?name$`
`BMC_IP`	`^bmc[ _]?ip[ _]?(address)?$`
`VIRTUALBOX_HOST`	`^(virtualbox[ _]?)?host$`
`VIRTUALBOX_VMNAME`	`^(virtualbox[ _]?)?vmname$`
`SNMP_POWER_DRIVER`	`^(snmp[ _]?)?(power[ _]?)?driver$`

`Field Name`	Name RegEx (case sensitive)
`SNMP_POWER_ADDRESS`	`^(snmp[ _]?)?(power[ _]?)?address$`
`SNMP_POWER_OUTLET`	`^(snmp[ _]?)?(power[ _]?)?outlet$`
`SNMP_POWER_COMMUNITY`	`^(snmp[ _]?)?(power[ _]?)?community$`
`PXE_OVERLAY`	`^(pxe[ _]?)?overlay$`
`MAC_ADDR`	`^(((<regex>))[ _]?)?mac([ _]?addr(ess)?)?$`
`N_CPUs`	`^(n[ _]?)?cpus$`
`ARCH`	`^(cpu[ _]?)?arch(itecture)?$`
`RAM_MB`	`^(ram|memory)[ _]?([(]?mb[)]?)?$`
`DISK_GB`	`^(disk|local)[ _]?([(]?gb[)]?)?$`
`NODE_CAPS`	`^(node[ _]?)?(capabilities|caps)$`
`NODE_DESC`	`^(node[ _]?)?desc(ription)?$`

8.10 Use of Simple Sync by eLogin Nodes

Simple Sync is a convenient way to copy user defined content to various Cray nodes. It is documented as a part of the SMW and IMPS. While eLogin nodes support Simple Sync, there are a few minor differences.

eLogin nodes can use Simple Sync to sync files from these directories:

- `common` classes directory: Since eLogin nodes do not have a specific node class associated with them, Simple Sync does not sync files from any of the more specific class directories.
- `nodegroups` directory: Node groups must be configured in the config set for this to be functional.

The eLogin node does not have a `cname` and hence does not support syncing files from the `cname` directories. The eLogin node syncs files from the directory matching the host name of the eLogin server.

8.11 How To Use eLogin

8.11.1 Set Up Passwordless SSH

About this task

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

Procedure

1. Generate an SSH key pair.

```
elogin$ ssh-keygen
```

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

```
elogin$ ssh-copy-id <login_name>
```

8.11.2 Use eswrap Utility

About this task

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

Procedure

1. List the available wrapped commands:

```
elogin$ eswrap
eswrap version 2.0.3
Will connect to host 'elogin1'
Usage: eswrap [--install] | [--check]
Environment variables:
        ESWRAP_LOGIN:   Forces eswrap to ssh to named host.
        ESWRAP_DEBUG:   Turns on internal debug output.
        ESWRAP_KEYFILE: Optional ini file.
                        Default /opt/cray/elogin/eswrap/etc/eswrap.ini
        ESWRAP_ENVFILE: Environment variable configuration file.
                        Default /opt/cray/elogin/eswrap/etc/eswrap.env
        ESWRAP_ROOT:    Allows root to execute command.
        ESWRAP_USER:    Login node user name.
        ESWRAP_CWD:     Login node working directory.
        ESWRAP_PREFIX:  Command to execute before wrapped commands.

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

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

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

8.12 eLogin Firewall

Prerequisites

eLogin software is installed and configured on the system.

About this task

Setting up firewalls on eLogin nodes is accomplished by editing the `/etc/opt/cray/openstack/elogin/templates/elogin_file`, and copying it to the proper `simple_sync` directory for deployment to the eLogin node(s).

Setup Firewall to All eLogin Nodes

Copy the `elogin_iptables` file to the following directory on the CMC to apply the file to all eLogin nodes in a node group:

```
/var/opt/cray/imps/config/sets/<config_set>/files/\
/simple_sync/nodegroups/<node_group>/files/etc/sysconfig/elogin_iptables
```

where: <*config_set*> is the name of the config set used. The *node_group* group must be created and membership must be configured in the config set to contain the desired eLogin nodes. The file must be named `elogin_iptables`.

Setup Firewall to Specific eLogin Node

Copy the `elogin_iptables` file to the following directory to apply the file to a specific eLogin node:

```
/var/opt/cray/imps/config/sets/<config_set>/files/\
/simple_sync/hostname/<elogin_hostname>/files/etc/sysconfig/elogin_iptables
```

where: <*config_set*> is the config set used, and <*elogin_hostname*> is the host name of the targeted eLogin node. The file must be named `elogin_iptables`.

8.13 Ironic Fuel Driver: Deploy Configuration JSON File

The Ironic Fuel driver is controlled by a JSON file that contains key attributes and elements to control various Ironic Fuel deployment parameters.

> **NOTICE:** This topic is intended as reference information. We recommend the user become familiar with the key attributes and schema of the Deploy Config JSON file to assist in performing the procedures: *Resize Disk Partitions for eLogin Nodes* and *Create Disk Partitions with Persistent Data for eLogin Nodes*.

The Ironic Fuel deploy config JSON file for eLogin is located at: `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin.json`

The `deploy_config_elogin.json` file contains multiple attributes on the 1st level:

```
"image_deploy_flags": ...
"partitions_policy": ...
"partitions": ...
```

image_deploy_flags

The attribute `"image_deploy_flags"` is composed in JSON, and is used to set flags in the deployment tool (Optional). Currently `"image_deploy_flags"` is used only by `rsync`. The default values for `rsync_flags` attribute are: `-a`, `-A`, and `-X`.

```
"image_deploy_flags": {"rsync_flags": "-a -A -X --timeout 300"}
```

partitions_policy

The attribute `"partitions_policy"` is optional and defaults to `"verify"`. The partitions policy accepts one of the following values:

- `"verify"` - applied in two scenarios:
  - Compare partitions schema with existing partitions on the disk(s). If the schema matches the ondisk partition layout (including registered`fstab` mount points) then deployment succeeds. If the schema does not match the on-disk layout, deployment fails and the node is returned to the pool. No modification to the on-disk content is made in this case. Any disks present on the target node that are not mentioned in the schema are ignored.
  - Clean data on filesystems marked as: `keep_data=False`
- `"clean"` - applied in one scenario:
  - Ignore existing partitions on the disk(s). Clean the disk and create partitions according to the schema. Any disks present on the target node not mentioned in the schema are ignored.

```
"partitions_policy": "verify"
```

partitions

The `"partitions"` attribute holds a partitioning schema that is applied to the node during deployment. The `"partitions"` attribute is required for the deploy config JSON file.

The following is the general structure and schema flow of the `"partitions"` attribute within the deploy config JSON file:

```
"partitions": [
        {
                "type": "disk",
                "id": {
                    "type": "name",
                    "value": "sda"
                },
                "size": "150 GB",
                "volumes": [
                        {
                                "type": "partition",
                                ...
                        },
                        ...,
                        {
                                "type": "pv",
                                ...
                        },
                        ...
                ]
        },
        {
                "type": "vg",
                ...
                "volumes": [
                        {
                                "type": "lv",
                                ...
                        },
                        ...
                ]
        },
]
```

The following table lists the attributes for `partitions` with requirements.

Table 5. Attributes for Partitions ("`partitions`") with Requirements

Attribute of **"partitions"**	Attribute Value Description	Requirement
`type`	`"disk"`	Required

id	ID used to find a disk. For example:
	`"id":{"type": "scsi", "value": "6:1:0:0"}`
	`"id":{"type": "path",`
	`          "value" : "disk/by-path/pci-0000:00:07.0-virtio-pci-virtio3"}`
	`"id":{"type": "path",`
	`           "value" : "disk/by-id/ata-SomeSerialNumber"}`
	`"id":{"type": "name", "value": "vda"}`
		Required

`size`	Size of the disk.	Required
`volumes`	Array of partitions / physical volumes.	Required

- All "`size`" values are strings containing either an integer number and size unit (example, `100 MiB` or `100MiB`). In the case of partitions, a value relative to the size of the disk (example, 40%) may also be used.
- Available measurement units are: MB, GB, TB, PB, EB, ZB, YB, MiB, GiB, TiB, PiB, EiB, ZiB, and YiB. Relative values use the size of the containing device or physical volume as a base. For example, specifying "`40%`" for a 100MiB device would result in a 40MiB partition. Relative sizes cannot be used for disks.
- User may specify "`remaining`" as a size value for "`volumes`" in a disk or "`volume group`". When "`remaining`" is specified, assuming that allocations are made for all other "`volumes`", all remaining free space on the drive will be used for this volume.

volumes

The following highlights the "`volumes`" attribute of the "`partitions`" schema flow within the deploy config JSON file:

```
"partitions": [
        {
                "type": "disk",
                "id": {
                    "type": "name",
                    "value": "sda"
                },
                "size": "150 GB",
                "volumes": [
                        {
                                "type": "partition",
                                "size": "32 GB",
                                "mount": "/",
                                "file_system": "ext4",
                                "name": "slash",
                                "disk_label": "disklabel",
                                "partition_guid": partitionguid",
                                "fstab_enabled": true,
                                "fstab_optons" "fstaboptions":,
                                "keep_data": true
                        },
                        ...,
                        {
                                "type": "pv",
                                ...
                        },
                        ...
                ]
        },
        {
                "type": "vg",
                ...
                "volumes": [
                        {
                                       "type": "lv",
                                       ...
                        },
                        ...
                ]
        },
]
```

The following table lists the attributes for `volumes` with requirements.

Table 6. Attributes for Volumes ("volumns") with Requirements

Attribute of "volumes"	Attribute Value Description	Requirement
`type`	`"partition"`	Required
`size`	Size of partition. Values: MB, GB, TB, PB, EB, ZB, YB, MiB, GiB, TiB, PiB, EiB, ZiB, and YiB.	Required
`mount`	Mount point (example, `"/"` and `"/usr"`).	Optional (not mounted by default)
`file_system`	File system type. Passed down to `mkfs` call.	Optional (`xfs` by default)
`name`	String name of the partition.	Required
`disk_label`	Filesystem label.	Optional (empty by default)
`partition_guid`	GUID to be assigned to partition.	Optional
`fstab_enabled`	Boolean value that specifies whether the partition will be included in `/etc/fstab` and mounted.	Optional (`true` by default)
`fstab_options`	String to specify `fstab` mount options.	Optional (`"defaults"` by default)

Attribute of "volumes"	Attribute Value Description	Requirement
`keep_data`	Boolean flag specifying whether or not to preserve data on this partition. Applied when `"verify"` of `"partitions_policy"` is used.	Optional (`true` by default)

IMPORTANT: If you are using the Fuel Swift deployment driver (`fuel_swift_*`), care must be taken when declaring mount points in your deployment configuration file that may conflict with those that exist in the tenant image. This conflict causes the mount points defined in the deployment configuration to mask the corresponding directories in the tenant image when the deployment completes. For example, if your deployment configuration file contains a definition for `/etc/`, the deployment creates an empty filesystem on disk and mounts it on `/etc` in the tenant image. This results in hiding contents of `/etc/` from the original tenant image with the on-disk filesystem created during deployment.

Physical volume (`"pv"`)

The following highlights the physical volume (`"pv"`) attribute of the `"partitions"` schema flow within the deploy config JSON file:

```
"partitions": [
        {
                "type": "disk",
                "id": {
                    "type": "name",
                    "value": "sda"
                },
                "size": "150 GB",
                "volumes": [
                        {
                                "type": "partition",
                                "size": "32 GB",
                                "mount": "/",
                                "file_system": "ext4",
                                "name": "slash",
                                "disk_label": "disklabel",
                                "partition_guid": partitionguid",
                                "fstab_enabled": true,
                                "fstab_optons" "fstaboptions":,
                                "keep_data": true
                        },
                        ...,
                        {
                                "type": "pv",
                                "size":"32 GB",
                                "vg": "volumegroupid",
                                "lvm_meta_size": "64 MB"
                        },
                        ...
                ]
        },
        {
                "type": "vg",
                ...
                "volumes": [
                        {
                                "type": "lv",
                                ...
                        },
                        ...
                ]
        },
]
```

The following table lists the attributes for physical volume with requirements.

Table 7. Attributes for Physical Volume ("pv") with Requirements

Attribute of "pv"	Attribute Value Description	Requirement
type	"pv"	Required
size	Size of the physical volume.	Required
vg	ID of the volume group this physical volume is associated with.	Required
lvm_meta_size	A size given to lvm to store metadata.	Optional (64 MiB by default) Minimum allowable value: 10 MiB.

Volume group ("vg")

The following highlights the volume group ("vg") attribute of the "partitions" schema flow within the deploy config JSON file:

```
"partitions": [
        {
                "type": "disk",
                "id": {
                    "type": "name",
                    "value": "sda"
                },
                "size": "150 GB",
                "volumes": [
                        {
                                "type": "partition",
                                "size": "32 GB",
                                "mount": "/",
                                "file_system": "ext4",
                                "name": "slash",
                                "disk_label": "disklabel",
                                "partition_guid": partitionguid",
                                "fstab_enabled": true,
                                "fstab_optons" "fstaboptions":,
                                "keep_data": true
                        },
                        ...,
                        {
                                "type": "pv",
                                "size":"32 GB",
                                "vg": "volumegroupid",
                                "lvm_meta_size": "64 MB"
                        },
                        ...
                ]
        },
        {
                "type": "vg",
                "id": "volumegroupname",
                "volumes": [
                        {
                                "type": "lv",
                                ...
                        },
                        ...
                ]
        },
]
```

The following table lists the attributes for volume group with requirements.

Table 8. Attributes for Volume Group ("vg") with Requirements

Attribute of **"vg"**	Attribute Value Description	Requirement
`type`	"vg"	Required
`id`	Volume group name. Should be refered at least once from "pv".	Required
`volumes`	Array of logical volumes.	Required

Logical volume (**"lv"**)

The following highlights the logical volume ("lv") attribute of the "partitions" schema flow within the deploy config JSON file:

```
"partitions": [
        {
                "type": "disk",
                "id": {
                    "type": "name",
                    "value": "sda"
                },
                "size": "150 GB",
                "volumes": [
                        {
                                "type": "partition",
                                "size": "32 GB",
                                "mount": "/",
                                "file_system": "ext4",
                                "name": "slash",
                                "disk_label": "disklabel",
                                "partition_guid": partitionguid",
                                "fstab_enabled": true,
                                "fstab_optons" "fstaboptions":,
                                "keep_data": true
                        },
                        ...,
                        {
                                "type": "pv",
                                "size":"32 GB",
                                "vg": "volumegroupid",
                                "lvm_meta_size": "64 MB"
                        },
                        ...
                ]
        },
        {
                "type": "vg",
                "id": "volumegroupname",
                "volumes": [
                        {
                                "type": "lv",
                                "name": "logicalvolumename",
                                "size": "64 GB",
                                "mount": "mountpoint",
                                "file_system": "filesystemtype",
                                "disk_label": "filesystemlabel"
                        },
                        ...
                ]
        },
]
```

The following table lists the attributes for logical volume with requirements.

Table 9. Attributes for Logical Volume (`"lv"`) with Requirements

Attribute of `"lv"`	Attribute Value Description	Requirement
`type`	`"lv"`	Required
`name`	Name of the logical volume.	Required
`size`	Size of the logical volume.	Required
`mount`	Mount point: example, `"/"` and `"/usr"`).	Optional
`file_system`	File system type. Passed down to `mkfs` call.	Optional (`xfs` by default).
`disk_label`	Filesystem label.	Optional (empty by default)

NOTE: Putting a `"/"` partition on the Logical Volume Management (LVM) requires a standalone `"/boot"` partition defined in the schema. In this case, the node should be managed by the `fuel_rsync` Ironic driver.

8.13.1 Resize Disk Partitions for eLogin Nodes

Prerequisites

- eLogin is installed and configured on the system.
- For reference, user should be familiar with the *Ironic Fuel Driver: Deploy Configuration JSON File*.

About this task

This procedure offers guidance on how to resize disk partitions for eLogin nodes. The default disk size and disk partitions for the eLogin node are defined in the Ironic Fuel driver JSON file, located at: `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin.json`.

Procedure

1. Run these commands to observe the default configuration (`deploy_config_elogin.json`) file.

The default configuration defines a 150 GB disk with five partitions: `/`, `/var/crash`, `/boot`, `swap` and `/tmp`; with partition sizes: 32 GB, 24 GB, 512 MB, 64 GB and 4 GB respectively.

```
root@cmc# cd /etc/opt/cray/openstack/fuel/deploy_config
root@cmc# cat deploy_config_elogin.json
```

```
{
    "partitions_policy": "clean",
    "partitions": [
        {
            "type": "disk",
            "id": {
                "type": "name",
                "value": "sda"
            },
            "size": "150 GB",
            "volumes": [
                {
                    "type": "partition",
                    "mount": "/",
                    "file_system": "ext4",
                    "size": "32 GB",
                    "name": "slash"
                },
                {
                    "type": "partition",
                    "mount": "/var/crash",
                    "file_system": "ext4",
                    "size": "24 GB",
                    "name": "varcrash"
                },
                {
                    "type": "partition",
                    "mount": "/boot",
                    "file_system": "ext4",
                    "size": "512 MB",
                    "name": "boot"
                },
                {
                    "type": "partition",
                    "mount": "swap",
                    "file_system": "swap",
                    "size": "64 GB",
                    "name": "swap"
                },
                {
                    "type": "partition",
                    "mount": "/tmp",
                    "file_system": "ext4",
                    "size": "4 GB",
                    "name": "tmp"
                }
            ]
        }
    ]
}
```

Resize a Disk Partition

In the scenario where a larger root partition is required, perform the following steps to change the size of the root partition (for example, double the size of the root partition from 32 to 64 GB).

2. Go to the `/etc/opt/cray/openstack/fuel/deploy_config` directory.

```
root@cmc# cd /etc/opt/cray/openstack/fuel/deploy_config
```

3. Copy the `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin.json` file to `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin_root64gb.json`.

```
root@cmc# cp deploy_config_elogin.json deploy_config_elogin_root64gb.json
```

4. Edit the /etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin_root64gb.json file:

a. Identify the current partitions policy: `“partitions_policy”: “clean”`. To resize any partition, ensure that the `partitions_policy` attribute type is set to `‘clean’`.

b. Identify the current disk size: `“size”: “150 GB”`

c. Identify the current size of the root partition: `“size”: “32 GB”`

d. Change the disk size to accomodate the sum of the resized root partition and of all other paritions. Ensure the size is within the allowable limits of the actual physical disk space: for example, **`“size”: “180 GB”`**

e. Change the root partition size to 64 GB: for example, **`“size”: “64 GB”`**

```
{
    "partitions_policy": "clean",
    "partitions": [
        {
            "type": "disk",
            "id": {
                "type": "name",
                "value": "sda"
            },
            "size": "180 GB",
            "volumes": [
                {
                    "type": "partition",
                    "mount": "/",
                    "file_system": "ext4",
                    "size": "64 GB",
                    "name": "slash"
                },
                {
                    "type": "partition",
                    "mount": "/var/crash",
                    "file_system": "ext4",
                    "size": "24 GB",
                    "name": "varcrash"
                },
                {
                    "type": "partition",
                    "mount": "/boot",
                    "file_system": "ext4",
                    "size": "512 MB",
                    "name": "boot"
                },
                {
                    "type": "partition",
                    "mount": "swap",
                    "file_system": "swap",
                    "size": "64 GB",
                    "name": "swap"
                },
                {
                    "type": "partition",
                    "mount": "/tmp",
                    "file_system": "ext4",
                    "size": "4 GB",
                    "name": "tmp"
                }
            ]
        }
    ]
}
```

5. Create a new Glance deployment config image called `deploy_config_elogin_root64gb`.

```
root@cmc# glance image-create --is-public True --disk-format raw \
--container-format bare --name deploy_config_elogin_root64gb \
--file /etc/opt/cray/openstack/fuel/deploy_config/
deploy_config_elogin_root64gb.json
```

NOTE: Glance is an OpenStack image service.

6. Go to the `/etc/opt/cray/openstack/heat/templates` directory.

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

7. Edit the `/etc/opt/cray/openstack/heat/templates/elogin_template.yaml` file, or the site specific file (if file was renamed).

   a. Locate the `deployment_config_image` section:

   ```
   deployment_config_image:
   type: string
   ```

```
label: DeployConfigImage
description: Deployment config image
default: deploy_config_elogin
```

b. Change the `deployment_config_image` default value to the Glance deployment config image created in step 5.

```
deployment_config_image:
type: string
label: DeployConfigImage
description: Deployment config image
default: deploy_config_elogin_root64gb
```

8. (Conditional) Quiesce and shutdown the currently running eLogin node (only if necessary).

```
root@cmc# heat stack-delete <elogin-name>
```

9. Deploy the eLogin node, using the new deployment config image.

```
root@cmc# ./deploy_elogin.sh
```

When booted, the eLogin node is expected to have the resized partition size.

8.13.2 Create Disk Partitions with Persistent Data for eLogin Nodes

Prerequisites

- eLogin is installed and configured on the system.
- For reference, user should be familiar with the *Ironic Fuel Driver: Deploy Configuration JSON File*.
- Completed procedure to *Resize Disk Partitions for eLogin Nodes* .

About this task

This procedure offers instruction on how to create eLogin node disk partitions with persistent data across boots. To do this, the `"keep_data"` attribute must be used in the partition definitions of the `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin_root64gb.json` file.

Procedure

1. Run these commands to observe the `deploy_config_elogin_root64gb.json` file.

```
root@cmc# cd /etc/opt/cray/openstack/fuel/deploy_config
root@cmc# cat deploy_config_elogin_root64gb.json
```

```
{
    "partitions_policy": "clean",
    "partitions": [
        {
            "type": "disk",
            "id": {
                "type": "name",
                "value": "sda"
            },
            "size": "180 GB",
            "volumes": [
                {
                    "type": "partition",
                    "mount": "/",
                    "file_system": "ext4",
                    "size": "64 GB",
                    "name": "slash"
                },
                {
                    "type": "partition",
                    "mount": "/var/crash",
                    "file_system": "ext4",
                    "size": "24 GB",
                    "name": "varcrash"
                },
                {
                    "type": "partition",
                    "mount": "/boot",
                    "file_system": "ext4",
                    "size": "512 MB",
                    "name": "boot"
                },
                {
                    "type": "partition",
                    "mount": "swap",
                    "file_system": "swap",
                    "size": "64 GB",
                    "name": "swap"
                },
                {
                    "type": "partition",
                    "mount": "/tmp",
                    "file_system": "ext4",
                    "size": "4 GB",
                    "name": "tmp"
                }
            ]
        }
    ]
}
```

2. Copy the `deploy_config_elogin_root64gb.json` file to: `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin_root64gb_keepdata.json`

```
root@cmc# cd /etc/opt/cray/openstack/fuel/deploy_config/
root@cmc# cp deploy_config_elogin_root64gb.json \
deploy_config_elogin_root64gb_keepdata.json
```

3. Edit the `/etc/opt/cray/openstack/fuel/deploy_config/deploy_config_elogin.json` file, or if renamed, use the site specific name.

a. Change the partitions policy to `verify`: for example, `“partitions_policy”: “verify”`

b. Add the boolean `keep_data` attribute to each `partition` as required. For example, the `"/"`, `/var/crash`, and `/boot` partitions are defined with `"keep_data": true`, and `"swap"` and `"/tmp"` are defined with `"keep_data": false`. Setting `"keep_data": true` for a partition, means that data in the partition will be persistent across boots.

Changing the `"partitions_policy"` attribute to `"verify"` for persistent data partitions, first requires a `"clean"` `"partitions_policy"` eLogin node boot, to create the defined paritions on disk. The `"verify"` `"partitions_policy"` configuration can then be configured and the eLogin booted, otherwise a mismatch occurs between the configured schema and the physical disk partitions, thus causing the boot to fail.

NOTE: The attribute `"keep_data"` is boolean data type, and must only be set to: `true` or `false`, all lower case letters, and not enclosed in double quotes such that the value is a string.

```
 {
    "partitions_policy": "verify",
    "partitions": [
        {
            "type": "disk",
            "id": {
                "type": "name",
                "value": "sda"
            },
            "size": "180 GB",
            "volumes": [
                {
                    "type": "partition",
                    "mount": "/",
                    "file_system": "ext4",
                    "size": "64 GB",
                    "keep_data": true,
                    "name": "slash"
                },
                {
                    "type": "partition",
                    "mount": "/var/crash",
                    "file_system": "ext4",
                    "size": "24 GB",
                    "keep_data": true,
                    "name": "varcrash"
                },
                {
                    "type": "partition",
                    "mount": "/boot",
                    "file_system": "ext4",
                    "size": "512 MB",
                    "keep_data": true,
                    "name": "boot"
                },
                {
                    "type": "partition",
                    "mount": "swap",
                    "file_system": "swap",
                    "size": "64 GB",
                    "keep_data": false,
                    "name": "swap"
                },
                {
                    "type": "partition",
                    "mount": "/tmp",
                    "file_system": "ext4",
                    "size": "4 GB",
                    "keep_data": false,
                    "name": "tmp"
                }
            ]
        }
    ]
}
```

CAUTION: When using the `"verify"` `"partitions_policy"` and the `"keep_data"` partition attribute, if the actual physical disk partitions do not match the schema, as defined in the JSON file, then the eLogin node will fail to boot. The error `"Failed To Provision Instance"` will be logged.

4. Create a new Glance deployment config image called: `deploy_config_elogin_root64gb_keepdata`.

```
root@cmc# glance image-create --is-public True --disk-format raw \
--container-format bare --name deploy_config_elogin_root64gb_keepdata \
--file /etc/opt/cray/openstack/fuel/deploy_config/ \
deploy_config_elogin_root64gb_keepdata.json
```

5. Go to the `/etc/opt/cray/openstack/heat/templates` directory.

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

6. Edit the file: `/etc/opt/cray/openstack/heat/templates/elogin_template.yaml`, or the site specific file if renamed.

a. Locate the `deployment_config_image` section:

```
deployment_config_image:
type: string
label: DeployConfigImage
description: Deployment config image
default: deploy_config_elogin
```

b. Change the `deployment_config_image` default value to the Glance deployment config image created in step 4.

```
deployment_config_image:
type: string
label: DeployConfigImage
description: Deployment config image
default: deploy_config_elogin_root64gb_keepdata
```

7. (Conditional) Quiesce and shutdown the currently running eLogin node (only if necessary).

```
root@cmc# heat stack-delete <elogin-name>
```

8. Deploy the eLogin node, using the new deployment config image.

```
root@cmc# ./deploy_elogin.sh
```

When booted, the eLogin node is expected to have partitions with persistent data configured.

8.13.3 Create Dump File Using kdump Service

Prerequisites

- Must have available space on hard drive to store a core file.
- The *Create Disk Partitions with Persistent Data for eLogin Nodes* procedure has been completed. The partition where to create the dump files is configured for persistent data across boots via the `"keep_data"` partition attribute, in the associated `deploy_config_elogin.json` file.

About this task

`kdump` is a Linux utility used to capture the system core dump in the event of system crashes. When triggered, `kdump` exports a memory image (known as vmcore) that can be analyzed for the purposes of debugging and determining the cause of a crash.

This procedure instucts how to create a dump file using the `kdump` service. This must be done on each eLogin node in order to catch a crash and have `kdump` create a core file.

Procedure

1. Identify total memory on the eLogin node. This is roughly the amount of free space required on a partition for a `dumplevel 0` crash file.

```
root@elogin# cat /proc/meminfo | grep MemTotal
MemTotal: 24200620 kB
root@elogin#
```

2. Identify partition for saving dump file.

The disk partition must have enough available space for the dump files. Note the available space and the `/dev` name for the partition. In the following example, the `"/"` partition, `/dev/sda2`, with 48 GB of available space is used for `kdump`.

> **NOTE:** The partition where to create the dump files must be configured for persistent data across boots via the `"keep_data"` partition attribute, in the associated `deploy_config_elogin.json` file.

```
root@elogin# df -h
Filesystem              Size  Used Avail Use%  Mounted on
/dev/sda2                59G  8.5G   48G  16%  /
tmpfs                    12G   10M   12G   1%  /run
devtmpfs                 12G     0   12G   0%  /dev
/dev/sda3                22G   45M   21G   1%  /var/crash
/dev/sda4               465M   55M  382M  13%  /boot
/dev/sda6               3.7G   15M  3.4G   1%  /tmp
/dev/sdb1               932G  120G  812G  13%  /home
tmpfs                    12G     0   12G   0%  /dev/shm
tmpfs                    12G     0   12G   0%  /sys/fs/cgroup
10.149.0.1@o2ib:/rind1   44T  727G   41T   2%  /lus/rind1
root@elogin#
```

3. Verify the system was booted with memory reserved for the `kdump` crash kernel.

Run the following command line to ensure that the system running on the eLogin node was started with the `crashkernel` paramater: `crashkernel=512M`. The command line with the kernel parameters are located in: `/proc/cmdline`.

```
root@elogin# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.12.51-52.39-default root=UUID=b9b0532d-663c-46bb-91f5-c6a9eeefc488
nofb nomodeset vga=normal console=tty0 crashkernel=512M console=ttyS1,115200n8
pci=bfsort root=/dev/sda2 BOOTIF=01-78-2b-cb-33-4b-b7 quiet splash=silent
root@elogin#
```

4. Edit the `kdump` configuration file located at: `/etc/sysconfig`.

The two configuration options to modify are: `KDUMP_IMMEDIATE_REBOOT` and `KDUMP_VERBOSE`.

a. Set `KDUMP_IMMEDIATE_REBOOT` to `"no"`.

b. Set `KDUMP_VERBOSE` to `31`.

```
root@elogin# cd /etc/sysconfig
root@elogin# ls
SuSEfirewall2       console        keyboard  nfs      rpcbind   windowmanager
SuSEfirewall2.d     cron           language  nginx    scripts   ypbind
atftpd              dmraid         mail      ntp      security
autofs              fonts-config   munge     opensm   services
backup              irqbalance     network   postfix  ssh
clock               kdump          news      proxy    syslog

root@elogin# vi kdump
change:
KDUMP_IMMEDIATE_REBOOT="yes"
```

```
to: KDUMP_IMMEDIATE_REBOOT="no"

change:
KDUMP_VERBOSE=3
to: KDUMP_VERBOSE=31

Write the kdump config file.
```

5. Enable and restart the `kdump` service.

```
root@elogin# systemctl enable kdump
root@elogin# systemctl restart kdump
```

6. Verify the `kdump` service has successfully started.

```
root@elogin# service kdump status
kdump.service - Load kdump kernel on startup
    Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled)
    Active: active (exited) since Fri 2016-09-09 11:34:59 CDT; 7s ago
   Process: 7003 ExecStop=/etc/init.d/boot.kdump stop (code=exited, status=0/SUCCESS)
   Process: 7014 ExecStart=/etc/init.d/boot.kdump start (code=exited, status=0/SUCCESS)
Main PID: 7014 (code=exited, status=0/SUCCESS)

Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: getCanonicalPathRoot(/boot/vmlinuz-3.12.51-52.39-default, /)
Sep 09 11:34:45 elogin boot.kdump[7014]: DEBUG: Trying /boot/config-3.12.51-52.39-default for config
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: Kconfig::readFromConfig(/boot/config-3.12.51-52.39-default)
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: FindKernel::findInitrd(/boot/vmlinuz-3.12.51-52.39-default)
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: getCanonicalPathRoot(/boot/vmlinuz-3.12.51-52.39-default, /)
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: FindKernel::isKdumpKernel(3.12.51-52.39-default)=false
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: getErrorCode: Returning error code of 0
Sep 09 11:34:45 elogin boot.kdump[7014]: TRACE: KdumpTool::~KdumpTool()
Sep 09 11:34:59 elogin boot.kdump[7014]: Loading kdump kernel: /sbin/kexec -p /boot/vmlinuz-3.12.51-52.39-default \
--append="nofb nomodeset vga=normal console=tty0 console=ttyS1,115200n8 pci=bfsort BOOTIF=01-78-2b-cb-33-4b-b7 \
quiet elevator=deadline sysrq=yes reset_devices acpi_no_memhotplug cgroup_disable=memory irqpoll nr_cpus=1 \
root=kdump disable_cpu_apicid=16   panic=1" --initrd=/boot/initrd-3.12.51-52.39-default-kdump
Sep 09 11:34:59 elogin boot.kdump[7014]: ..done
root@elogin#
```

Forcing a Kernal Panic

At this point, you wait for an OOPS or a kernel panic to trigger the `kdump` service.

7. Test `kdump` by forcing a kernel panic.

To test the `kdump` configuration, specific values are written to `/proc` files that force or trigger a kernel panic.

```
root@elogin# echo 1 > /proc/sys/kernel/sysrq
root@elogin# echo c > /proc/sysrq-trigger
```

Kernal Panic Occurs

When a kernel panic occurs, you must have console access (either physical access or via `ironic_conman`) to the eLogin node. Kernel headers and the kernel log buffer are dumped to the console. These messages are difficult to read and capture as they scroll by, but can be reproduced via the `makedumpfile` command. After the scrolling kernel log buffer, the `dmesg` text is extracted. Next, an attempt is made to save the dump file to the `"/"` partiton of the crash kernel's local environment. Due to a SLES issue, the dump file is not written to a location specified in the `kdump` config file. Instead, it is written to the crash kernel's limited space `"/"` partition. In this case, there is not sufficient space to save the dump file to the crash kernel's `"/"` partition, and the copy will fail and abort the `makedumpfile` process. Most important, is that a shell prompt will be presented for the administrator to run a small set of system utilities and commands.

Example output: The following section of example output, down to the `bash-4.2#` prompt, is meant to show the console display prior to the `makedumpfile` failure and abort.

```
<dump headers>
<log messages>
The dmesg log is saved to /kdump/mnt1/var/crash/2016-09-09-17:25/dmesg.txt.
```

```
Extracting dmesg
--------------------------------------------------------------------------------
makedumpfile Completed.
--------------------------------------------------------------------------------
Saving dump using makedumpfile
--------------------------------------------------------------------------------
Copying data                       : [100.0 %] |
The dumpfile is saved to /kdump/mnt1/var/crash/2016-09-09-17:25/vmcore.
makedumpfile Completed.
--------------------------------------------------------------------------------
Dump too large. Aborting. Check KDUMP_FREE_DISK_SIZE.
Error in fopen for /kdump/mnt1/var/crash/2016-09-09-17:25/README.txt (No such file or directory)
No System.map found in /kdump/boot
Last command failed (1).

Something failed. You can try to debug that here.
Type 'reboot -f' to reboot the system or 'exit' to
boot in a normal system that still is running in the
kdump environment with restricted memory!
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.2#
```

8. Mount partition to save dump files.

Mount the partition identified in step 2, for having adequate space to save a core file. Again, in this example, the root filesystem or `/dev/sda2` had enough available space. After mounting the partition, create a directory to create the core files. In the following example, the directory `KDUMP` is used.

```
bash-4.2# mount /dev/sda2 /mnt
bash-4.2# ls /mnt
bin   dal  etc   lib     lost+found  mnt  proc  run   selinux  srv  tmp var
boot  dev  home  lib64   lus         opt  root  sbin  share    sys  usr

bash-4.2# mkdir /mnt/KDUMP
bash-4.2# ls /mnt
KDUMP  boot dev  home    lib64       lus  opt   root  sbin     share sys usr
bin   dal   etc  lib     lost+found  mnt  proc  run   selinux  srv   tmp var
bash-4.2#
```

9. Extract display message (`dmesg`) log.

Use the `makedumpfile` command to extract the `dmesg` log to a file. In this example, the `dmesg` file is extracted to `/mnt/KDUMP/dmesgfile`. The `/mnt/KDUMP/dmesgfile` contains the kernel log buffer that scrolled quickly off the screen immediatley after the kernel panic occurred.

```
bash-4.2# makedumpfile --dump-dmesg /proc/vmcore /mnt/KDUMP/dmesgfile

The dmesg log is saved to /mnt/KDUMP/dmesgfile.
makedumpfile Completed.
bash-4.2#
```

10. Create a dump file.

Use the `makedumpfile` comand to create a core file.

The `-E` option creates the file in ELF format. The default format for the dump file is the `kdump` format. The crash command is able to read the `kdump` format dump file, but GDB (GNU debugger) does not. Saving the dump file in ELF format allows for analysis by both crash and GDB.

IMPORTANT: Cray recommends the use of `dumplevel 27` when creating the core file.

For details on the `makedumpfile` command and `dumplevels`, refer to the `makedumpfile` man page.

```
bash-4.2# makedumpfile -E -d 27 /proc/vmcore /mnt/KDUMP/vmcorefile
Copying data                      : [100.0 %] \

The dumpfile is saved to /mnt/KDUMP/vmcorefile.
makedumpfile Completed.

bash-4.2# ls -al /mnt/KDUMP
total 3137208
drwxr-xr-x  2 root 0       4096 Sep 13 14:22 .
drwxr-xr-x 27 root 0       4096 Sep 13 14:19 ..
-rw-------  1 root 0      70716 Sep 13 14:20 dmesgfile
-rw-------  1 root 0 3212411532 Sep 13 14:22 vmcorefile
bash-4.2#
```

11. (Optional) Dump levels

The `makedumpfile -d 0` option creates the core file at dump level `0`. The following is an example of the use of `dumplevel 0` and included for comparison purposes. Reference the `makedumpfile` man page, if necessary, for the use of different dump level values that result in smaller core files. Using the `-d 0` option (a dumplevel of zero), is the same as a complete copy of `/proc/vmcore` to a file.

```
bash-4.2# makedumpfile -E -d 0 /proc/vmcore /mnt/KDUMP/vmcore-d0
Copying data                        : [100.0 %] /

The dumpfile is saved to /mnt/KDUMP/vmcore-d0.
makedumpfile Completed.

bash-4.2# ls -al /mnt/KDUMP
total 27783676
drwxr-xr-x  2 root 0        4096 Sep 13 14:33 .
drwxr-xr-x 27 root 0        4096 Sep 13 14:19 ..
-rw-------  1 root 0       70716 Sep 13 14:20 dmesgfile
-rw-------  1 root 0 25237978772 Sep 13 14:36 vmcore-d0
-rw-------  1 root 0  3212411532 Sep 13 14:22 vmcorefile
bash-4.2#
```

12. Unmount the partition.

Unmount the partition that the core files were saved to. It is not necessary to run `sync` prior to the unmount since the `sync` command does not exist in this environement.

```
bash-4.2# umount /dev/sda2
```

13. Reboot the eLogin node.

Run the `nova reboot` command at the command line to reboot an eLogin node (for example, `elogin1`).

```
root@cmc# nova reboot <elogin1>
```

After a reboot of the eLogin node, the core files are available for analysis.

Dump File Analysis (Example)

The following commands offer an example of a dump file analysis. Note that the crash command is not available in the default eLogin environment, but GDB is available as follows:

```
root@elogin# cd /KDUMP
root@elogin# cp /boot/vmlinux-3.12.51-52.39-default.gz
root@elogin# gunzip vmlinux-3.12.51-52.39-default.gz
```

```
root@elogin# ls
dmesgfile procvmcorefile vmcorefile vmlinux-3.12.51-52.39-default
```

```
root@elogin# gdb ./vmlinux-3.12.51-52.39-default ./vmcorefile
:
(gdb) bt
#0 0xffffffff8130d751 in intel_idle ()
#1 0xffffffff813eae50 in cpuidle_enter_state ()
#2 0xffffffff813eafa2 in cpuidle_idle_call ()
#3 0xffffffff8100b9ca in arch_cpu_idle ()
#4 0xffffffff810ac4b1 in cpu_startup_entry ()
#5 0xffffffff81d4cea1 in start_kernel ()
#6 0xffffffff81d4c6a3 in x86_64_start_kernel ()
#7 0x0000000000000000 in ?? ()
(gdb) i r
rax 0x20 32
rbx 0x8 8
rcx 0x1 1
rdx 0x0 0
rsi 0xffffffff81c01fd8 -2118115368
rdi 0x46 70
rbp 0x4 0x4 <irq_stack_union+4>
rsp 0xffffffff81c01e78 0xffffffff81c01e78 <init_thread_union+7800>
r8 0x18 24
r9 0xe8c 3724
r10 0x3922 14626
r11 0x1 1
r12 0x20 32
r13 0x3 3
r14 0x4 4
r15 0xffffffff81cd4e58 -2117251496
rip 0xffffffff8130d751 0xffffffff8130d751 <intel_idle+193>
eflags 0x46 [ PF ZF ]
cs 0x10 16
ss 0x18 24
ds 0x0 0
es 0x0 0
fs 0x0 0
gs 0x0 0
(gdb)
```

8.14 Managed Shutdown of CMC and Deployed Instances

This section includes instruction on how to ensure a proper (clean) system shutdown (powered off).

A clean system shutdown requires the following tasks be perfomed (in order):

1. Shut down deployed instances.
2. Shut down the Cray Management Controller (CMC).

Perform these procedures to manage the shutdown of the CMC and deployed instances:

1. *Determine If A Nova Instance Supports Soft Power Off*
2. *Shutting Down Deployed Instances*
3. *Shutting Down The CMC*

To shut down the CMC only (not the full system), the CMC may be shut down without powering down deployed instances. In this case, perform the procedure: *Shutting Down The CMC*.

8.14.1 Determine If A Nova Instance Supports Soft Power Off

Prerequisites

System is configured and powered on.

About this task

This procedure determines if the driver used by Ironic to back a Nova instance, supports Soft Power Off.

Procedure

1. Obtain the Ironic node UUID from the Nova instance.

```
root@cmc# nova list
```

```
+----------------------------+---------+--------+------------+-------------+--------------------------+
| ID                         | Name    | Status | Task State | Power State | Networks                 |
+----------------------------+---------+--------+------------+-------------+--------------------------+
| ee390bfe-...-f95889fd7467 | elogin  | ACTIVE | -          | Running     | management=10.142.0.116 |
+----------------------------+---------+--------+------------+-------------+--------------------------+
[root@cmc#
```

The full ID listed in the above output is:

```
ee390bfe-2e9f-4316-a376-f95889fd7467
```

2. Run the following commands using the Nova Ironic ID to determine if the Ironic node driver supports Soft Power Off.

```
root@cmc# nova show ee390bfe-2e9f-4316-a376-f95889fd7467 | grep hypervisor | awk '{print $4}' \
2746d51d-f4bf-45fb-88d8-e80346b45cd4
root@cmc# ironic node-show 2746d51d-f4bf-45fb-88d8-e80346b45cd4 | grep 'driver ' | awk '{print $4}' \
fuel_rsync_ipmi
```

The Fuel driver name must contain the string `ipmi` to support Soft Power Off.

Choose one of the following results of running Nova Ironic ID:

- Soft Power Off is supported. The Fuel driver name contains the string `ipmi`.

  Run `nova stop` at the command line for a managed stop of a node.

  ```
  root@cmc# nova stop NODE_NAME
  ```

  ATTENTION: Skip the remaining steps in this section, and continue next to performing the procedure: *Shutting Down Deployed Instances*.

- Soft Power Off is NOT supported. The Fuel driver name DOES NOT contain the string `ipmi`.

  Run `nova stop` at the command line to produce a hard power off.

  ```
  root@cmc# nova stop NODE_NAME
  ```

(Optional) Manual Soft Power Off

To produce a manual Soft Power Off, do the following steps:

3. Log-in to the server.

```
root@cmc# ssh root@SERVER_IP
```

4. Shut down the server without powering off.

```
root@node# sudo shutdown -H
```

> **NOTE:** The command `shutdown -H` is used to ensure that power is not turned off after the shutdown. This is to prevent Ironic's attempt to restart the node.

Wait until shutdown is complete.

5. Stop the instance on the CMC.

```
root@cmc# nova stop NODE_NAME
```

8.14.2 Shutting Down Deployed Instances

Prerequisites

The procedure *Determine If A Nova Instance Supports Soft Power Off* is completed.

About this task

Deployed instances are created by using two principal methods:

- Directly (using the Nova CLI or API)
- Under the control of a Heat stack. (Product specific management tools commonly make use of Heat internally.)

For both methods, the instances must be stopped using Nova because Heat does not provide a feature to shut down or stop instances.

> **NOTE:** Most Cray products provide tools or scripts to assist with this process. These instructions are provided to assist in authoring scripts or for use as a manual fall-back.

Perform the following procedure to shut down deployed instances using Nova.

Procedure

1. Ensure that system users are informed of the shutdown.

2. Follow any product specific shutdown procedures.

3. Run `nova list` at the command line, to list the current deployed instances.

```
root@cmc # nova list
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
| ID                                   | Name    | Status  | Task State | Power State | Networks              |
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
| 18beae14-2280-4ded-958d-e536c4f0b803 | node000 | ACTIVE  | -          | Running     | management=10.0.0.100 |
| 002a054f-b46e-4938-88b1-45dc990b7ce1 | node001 | ACTIVE  | -          | Running     | management=10.0.0.101 |
...
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
```

4. Run `nova stop` for each deployed instance in the output of Nova list. (Example, `node000` and `node001`.)

> **NOTE:** If Soft Power Off is not available, you may manually shut down the instance before running `nova stop`. (For further information, and when to use Soft Power Off, see `nova stop`.)

```
root@cmc# nova stop node000
root@cmc# nova stop node001
```

5. Verify that the deployed instances are stopped, and the power is Off.

```
root@cmc# nova list
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
| ID                                   | Name    | Status  | Task State | Power State | Networks              |
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
| 18beae14-2280-4ded-958d-e536c4f0b803 | node000 | SHUTOFF | -          | Shutdown    | management=10.0.0.100 |
| 002a054f-b46e-4938-88b1-45dc990b7ce1 | node001 | SHUTOFF | -          | Shutdown    | management=10.0.0.101 |
...
+--------------------------------------+---------+---------+------------+-------------+-----------------------+
```

8.14.3 Shutting Down The CMC

Prerequisites

A clean system shutdown requires the completion of the following procedures before perfoming this procedure:

1. *Determine If A Nova Instance Supports Soft Power Off*
2. *Shutting Down Deployed Instances*

About this task

NOTE: Most Cray products provide tools or scripts to assist with this process. These instructions are provided to assist in authoring scripts or for use as a manual fall-back.

Perform the following procedure to shut down the Cray Management Controller (CMC).

Procedure

1. Ensure that no deployment operations are in progress, and that all external resources dependent upon Cray Software Management System (CSMS) services are shutdown.

2. Log in to the CMC as the root user, either on the console or using SSH.

3. Run the `poweroff` command.

```
root@cmc# poweroff
```

The power off process may take several minutes to complete.

⚠ **WARNING:** Do not cut power to the CMC while disk activity is occurring to avoid data corruption.

8.15 Starting Up CMC And Deployed Instances

About this task

This procedure describes how to power up the Cray Management Controller (CMC) and managed instances after planned or unplanned shutdown. For shutting down instances, the process of restarting is the same both for instances created directly and for those created by Heat.

NOTE: Most Cray products provide tools or scripts to assist with this process. These instructions are provided to assist in authoring scripts or for use as a manual fall-back.

Perform the following procedure to start up the CMC and deployed instances.

Procedure

1. Restore power to infrastructure components, such as: network switches, rack controllers, and so on.

   If possible, do not restore power to deployed instances; as once CSMS has started, any nodes managed by Ironic will have their power state set to match the state in the Ironic database. This may result in nodes having their power cut - if the nodes previously had power restored outside CSMS.

2. Restore power to the CMC running the CSMS.

   When startup is complete, the log-in prompt is displayed on the console allowing log-in using SSH.

   Ironic adjusts the power state of its managed nodes to match the state in its database. This may result in managed nodes starting up without further manual intervention, or nodes shut down if they were powered up but Ironic has their power state set to `Off`.

3. Log-in to the CMC, and run `csms-svc-verify` to verify that all services have started.

   ```
   root@cmc# csms-svc-verify
   ```

   All the services should be either `Ok` or `disabled`. See Checking Service Status using `csms-svc-verify` for further information.

4. List any required deployed instances with current state information using `nova list`.

   ```
   root@cmc# nova list
   ```

   ```
   +--------------------------+---------+---------+------------+-------------+-----------------------+
   | ID                       | Name    | Status  | Task State | Power State | Networks              |
   +--------------------------+---------+---------+------------+-------------+-----------------------+
   | 18beae14-...-e536c4f0b803 | node000 | SHUTOFF | -          | Shutdown    | management=10.0.0.100 |
   | 002a054f-...-45dc990b7ce1 | node001 | SHUTOFF | -          | Shutdown    | management=10.0.0.101 |
   ...
   +--------------------------+---------+---------+------------+-------------+-----------------------+
   ```

5. Start any required deployed instances using `nova start`.

   ```
   root@cmc# nova start node000
   root@cmc# nova start node001
   ```

8.16 Using Heat to Create and Mangage Nodes

About this task

Heat provides orchestration support and allows for complex compute, storage, and/or network environments to be created and managed. Heat templates are used to define the resources that comprise the stack and environment files used to pass parameters for the template. Heat may be used in the Cray environment to boot tenant instances with the desired resources and metadata to support various products. Heat templates may provide a more concise and manageable interface for booting nodes, as opposed to the Nova CLI.

A number of reference Heat templates are provided, located in: `/etc/opt/cray/openstack/heat/templates/`. Cray provides Heat template use-case examples for the user to modify for a given use case.

Perform this procedure to create and manage nodes using Heat:

Procedure

1. Create a Heat stack.

   Stack parameters are specified on the command line or via environment files. All corresponding servers will be booted and resources provisioned.

   ```
   root@cmc# heat stack-create -f /etc/opt/cray/openstack/heat/templates/ \
   heat_deploy_template.yaml -e /tmp/heat_env_file.yaml test_stack_2
   ```

2. List all Heat stacks with minimal details.

   ```
   root@cmc# heat stack-list
   +--------------------------------------+--------------+--------------------+----------------------+
   | id                                   | stack_name   | stack_status       | creation_time        |
   +--------------------------------------+--------------+--------------------+----------------------+
   | 9437da6e-4338-4131-8973-6fdb4a43995b | test_stack_2 | CREATE_COMPLETE    | 2015-08-17T18:45:26Z |
   +--------------------------------------+--------------+--------------------+----------------------+
   ```

3. Delete a specified Heat stack.

   All corresponding servers will be shutdown, destroyed, and resources unprovisioned.

   ```
   root@cmc# heat stack-delete test_stack_2
   ```

8.17 Required BIOS Configuration Settings

Prerequisites

- eLogin installation is complete.
- System is running.
- Keyboard and monitor connected to the server.

  To complete this procedure, the monitor must be connected to the serial port. In lieu of a connected keyboard and monitor, an iDRAC connection allowing console access to the server is required.

About this task

Prior to configuring an eLogin node, you must change the BIOS and remote access controller (iDRAC) settings. Several of the following BIOS settings are required for `ironic_conman` (a console manager) to function properly:

- Nodes must be cabled to both the CMC node administration network (`maint-net`) and IPMI network (`ipmi-net`).

- A node running a workload manager such as Moab® or TORQUE must be cabled directly to the SDB on the Cray system over the `wlm-net` network.
- All nodes must be configured to provide IPMI Serial Over LAN (SOL) for remote console support.
- A node must be configured to PXE boot from the CMC node (embedded NIC) before attempting to boot from the local disk.
- Configure all nodes to remain powered off after a site power failure. Cray recommends that the CMC node be powered on first and become operational before each node is powered on.

In the BIOS System Setup utility, use the **Tab** key to navigate to fields on the screen. To select an item, use the up/down-arrow keys to highlight the item, then press the **Enter** key. Press the <**Esc**> key to exit a sub-menu and return to the previous screen.

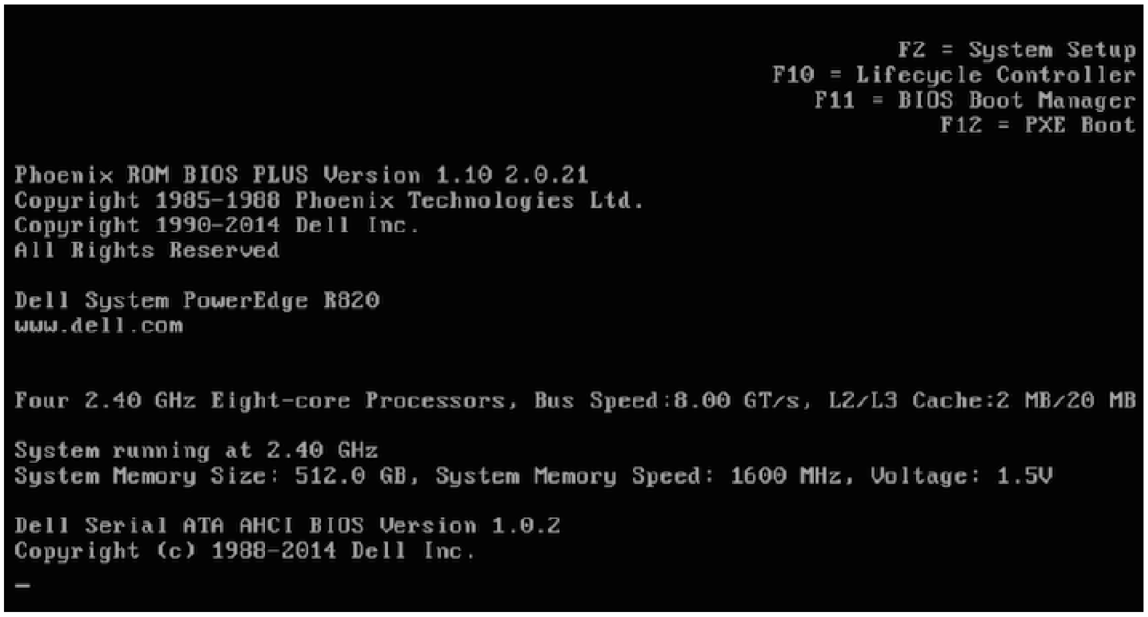
Perform the following procedure to change the BIOS and iDRAC settings for an eLogin node.

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 8. BIOS Config Screen



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

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

Dell System PowerEdge R820
www.dell.com


Four 2.40 GHz Eight-core Processors, Bus Speed:8.00 GT/s, L2/L3 Cache:2 MB/20 MB

System running at 2.40 GHz
System Memory Size: 512.0 GB, System Memory Speed: 1600 MHz, Voltage: 1.5V

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

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

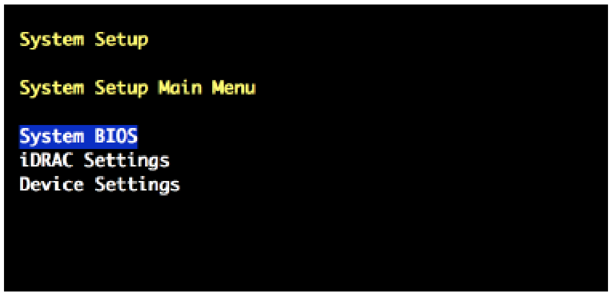
After the post process completes and all disk and network controllers have been initialized, the **Dell System Setup** screen appears with the following sub menus:

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

2. Change the system BIOS settings.

   a. Select **System BIOS**, then press **Enter** on the keyboard.

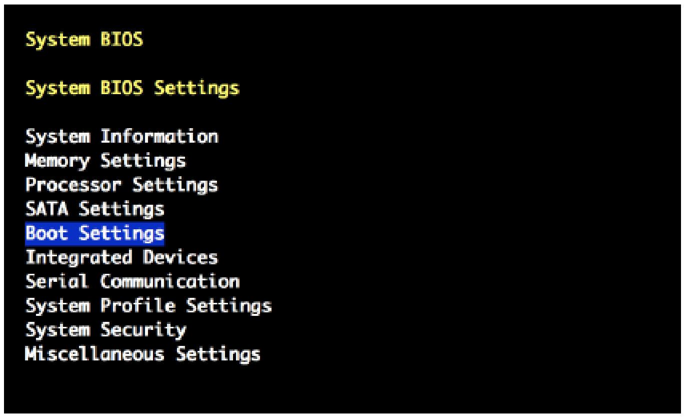
      Figure 9. System Setup Main Menu: System BIOS

      ```
      System Setup

      System Setup Main Menu

      System BIOS
      iDRAC Settings
      Device Settings
      ```

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

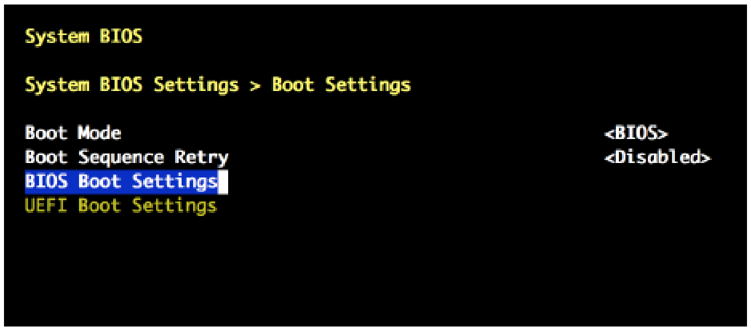
      Figure 10. System BIOS Boot Settings

      ```
      System BIOS

      System BIOS Settings

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

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

      Figure 11. BIOS Boot Settings

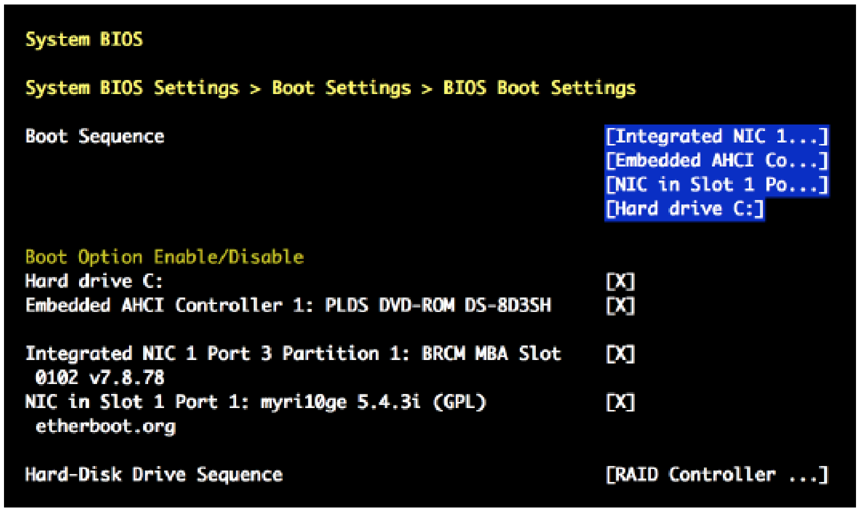
      ```
      System BIOS

      System BIOS Settings > Boot Settings

      Boot Mode                                            <BIOS>
      Boot Sequence Retry                                  <Disabled>
      BIOS Boot Settings
      UEFI Boot Settings
      ```

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

   e. Set the boot sequence:

      - Change the boot order so that the integrated NIC appears first, before the optical (DVD) drive. Ensure that the hard drive is last on the list.

Figure 12. System BIOS Settings: Set Boot Sequence



```
System BIOS

System BIOS Settings > Boot Settings > BIOS Boot Settings

Boot Sequence                                           [Integrated NIC 1...]
                                                        [Embedded AHCI Co...]
                                                        [NIC in Slot 1 Po...]
                                                        [Hard drive C:]

Boot Option Enable/Disable
Hard drive C:                                           [X]
Embedded AHCI Controller 1: PLDS DVD-ROM DS-8D3SH       [X]

Integrated NIC 1 Port 3 Partition 1: BRCM MBA Slot      [X]
 0102 v7.8.78
NIC in Slot 1 Port 1: myri10ge 5.4.3i (GPL)             [X]
 etherboot.org

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

- Ensure that the **Integrated NIC Port** is enabled. This allows the node to PXE boot from the CMC (embedded NIC) before attempting to boot from the local disk.

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

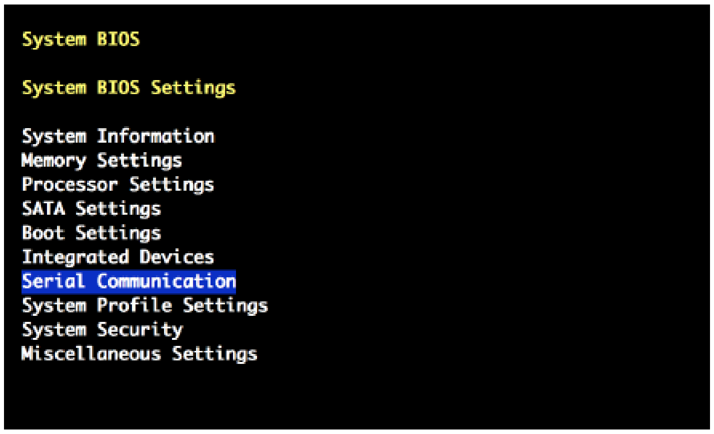
g. Press <**Esc**> to exit **BIOS Boot Settings**.

h. Press <**Esc**> to exit **Boot Settings** and return to the **System BIOS Settings** screen.

3. Change the serial communication settings.

a. Select **Serial Communication** on the **System BIOS Settings** screen.

Figure 13. System BIOS Settings: Serial Communication Selection

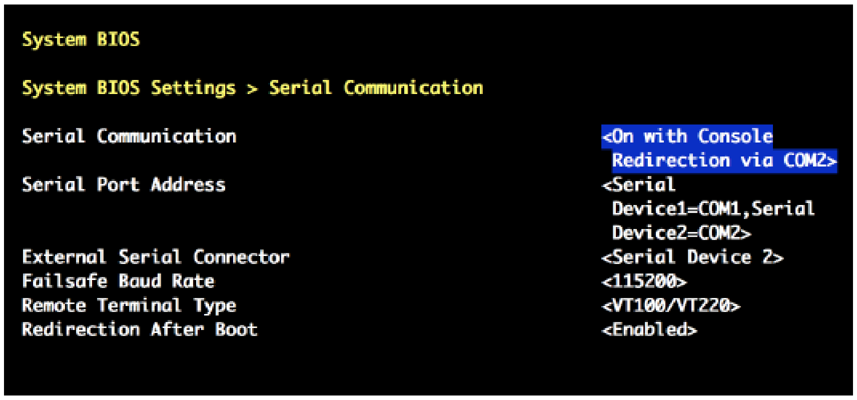


```
System BIOS

System BIOS Settings

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

A pop-up window displays the available **Serial Communication** setting options.

b. Select **On with Console Redirection via COM2** from the **Serial Communication** setting options, then press **Enter**.

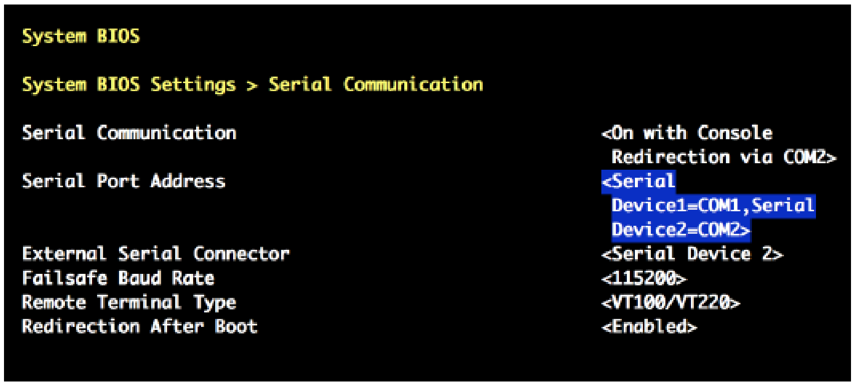
Figure 14. Serial Communication: Console Redirection



```
System BIOS

System BIOS Settings > Serial Communication

Serial Communication                                   <On with Console
                                                        Redirection via COM2>
Serial Port Address                                    <Serial
                                                        Device1=COM1,Serial
                                                        Device2=COM2>
External Serial Connector                              <Serial Device 2>
Failsafe Baud Rate                                     <115200>
Remote Terminal Type                                   <VT100/VT220>
Redirection After Boot                                 <Enabled>
```

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

Figure 15. Serial Communication: Serial Port Address



```
System BIOS

System BIOS Settings > Serial Communication

Serial Communication                                   <On with Console
                                                        Redirection via COM2>
Serial Port Address                                    <Serial
                                                        Device1=COM1,Serial
                                                        Device2=COM2>
External Serial Connector                              <Serial Device 2>
Failsafe Baud Rate                                     <115200>
Remote Terminal Type                                   <VT100/VT220>
Redirection After Boot                                 <Enabled>
```

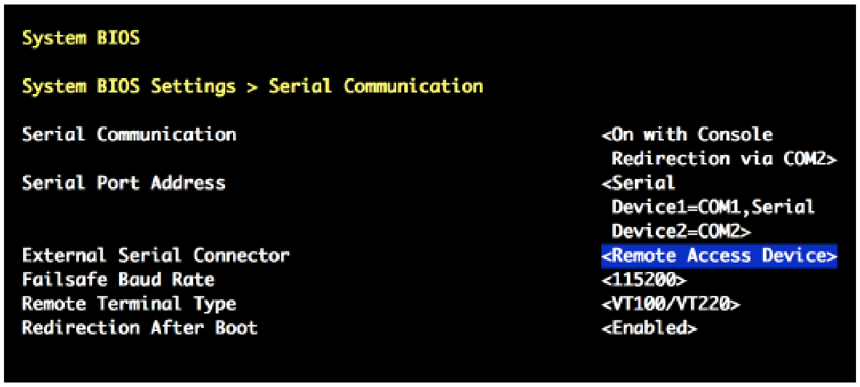
This setting enables the remote console. If this setting is incorrect, you cannot use a remote console to access the node.

To change the **Serial Port Address** to correct setting, do the following:

1. Press **Enter** to display the available options.
2. Change the setting to **Serial Device1=COM1**, **Serial Device2=COM2**.
3. Press **Enter** to return to the **Serial Communication** screen.

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

Figure 16. External Serial Connector: Remote Access Device

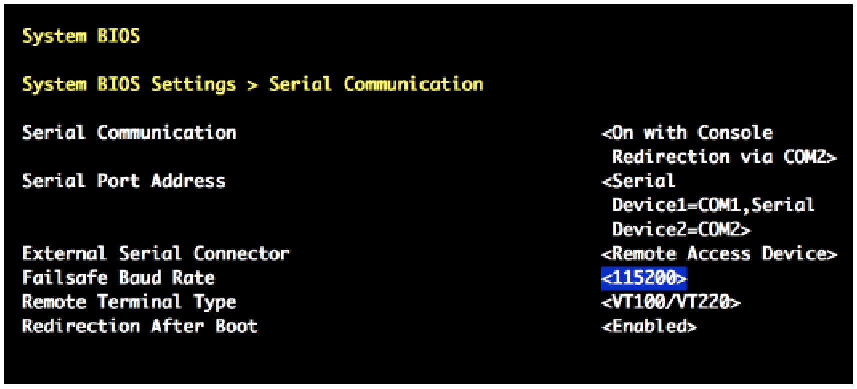


```
System BIOS

System BIOS Settings > Serial Communication

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

e. Select **Remote Access Device** from the pop-up window, then press **Enter** to return to the previous screen.

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

Figure 17. Serial Communication: Failsafe Baud Rate



```
System BIOS

System BIOS Settings > Serial Communication

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

g. Select **115200** in the pop-up window, then press **Enter** to return to the previous screen.

h. Press the <**Esc**> key to exit the **Serial Communication** screen.

i. Press the <**Esc**> key to exit the **System BIOS Settings** screen.

j. Press the <**Esc**> key to exit the **BIOS Settings** screen.

k. Select **Yes** to save changes when prompted with the "`Settings have changed`" message.

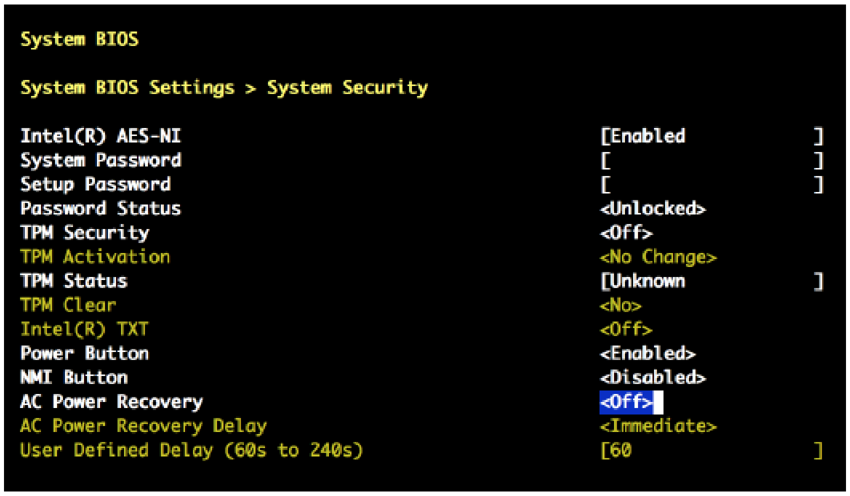
l. Select **OK** when prompted with the "`Settings saved successfully`" message.

4. Navigate to the **System BIOS Settings** menu, and then select **System Security**.

5. Set the **AC Power Recovery** to **Off** from the **System Security** menu.

This sets the node to remain powered off after a system power failure.

Figure 18. System Security: AC Power Recovery Off



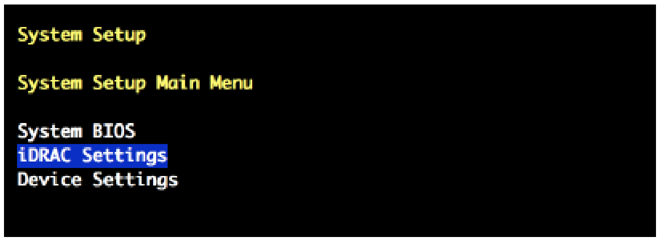
```
System BIOS

System BIOS Settings > System Security

Intel(R) AES-NI                                [Enabled          ]
System Password                                [                 ]
Setup Password                                 [                 ]
Password Status                                <Unlocked>
TPM Security                                   <Off>
TPM Activation                                 <No Change>
TPM Status                                     [Unknown          ]
TPM Clear                                      <No>
Intel(R) TXT                                   <Off>
Power Button                                   <Enabled>
NMI Button                                     <Disabled>
AC Power Recovery                              <Off>
AC Power Recovery Delay                        <Immediate>
User Defined Delay (60s to 240s)               [60               ]
```

NOTE: Cray recommends that the CMC node power up and become operational before all client nodes.

6. Press the <**Esc**> key to return to the **System Setup Main Menu**.

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

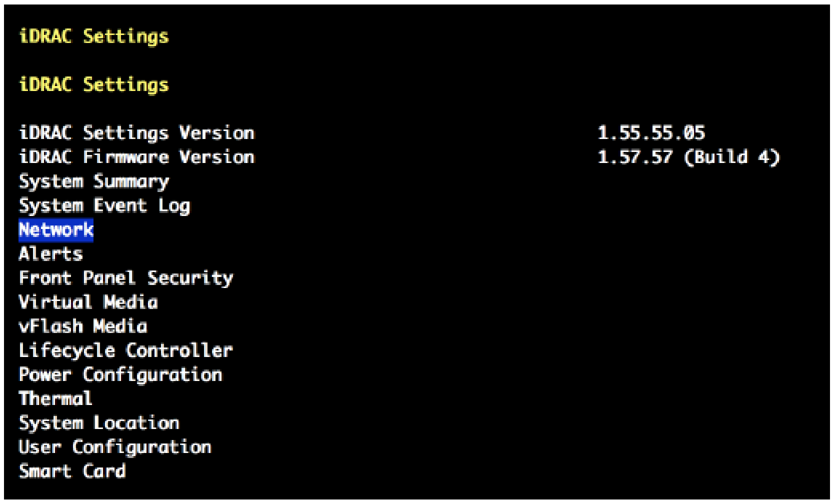
Figure 19. System Setup Main Menu: iDRAC Settings



```
System Setup

System Setup Main Menu

System BIOS
iDRAC Settings
Device Settings
```

8. Select **Network** from the **iDRAC Settings** menu, then press **Enter**. A long list of network settings is displayed.

Figure 20. iDRAC Settings: Network



```
iDRAC Settings

iDRAC Settings

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

9. Change the **IPMI Settings** in the **Network** menu to enable the **Serial Over LAN (SOL)** console.

a. Scroll to the **IPMI Settings** list using the down-arrow key.

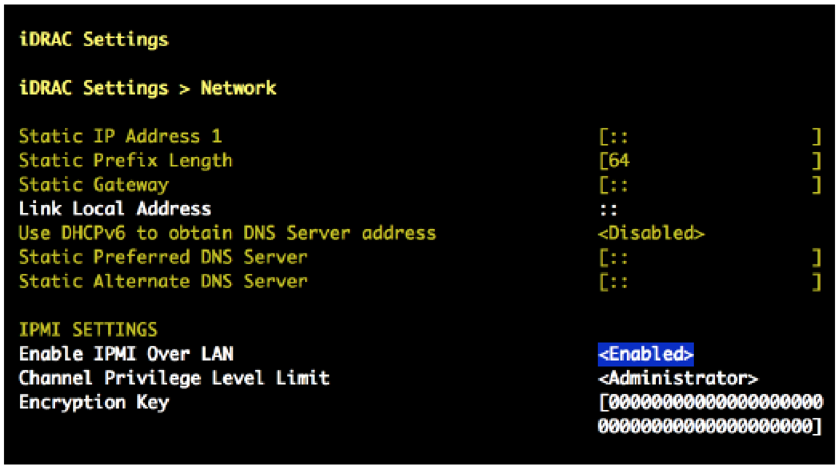
b. Ensure that **IPMI over LAN** is set to **Enabled**.

To change the setting to **Enabled**, do this:

1. Select **IPMI over LAN**, then press **Enter**.

2. Select **<Enabled>** in the pop-up window.

Figure 21. Network IPMI Settings: IPMI Over LAN



```
iDRAC Settings

iDRAC Settings > Network

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

IPMI SETTINGS
Enable IPMI Over LAN                                <Enabled>
Channel Privilege Level Limit                       <Administrator>
Encryption Key                                      [00000000000000000000
                                                    00000000000000000000]
```

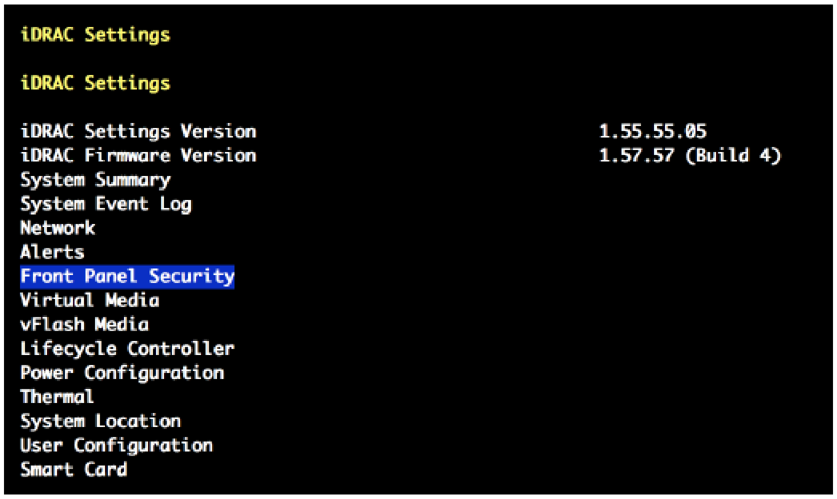
3. Press **Enter** to return to the previous screen.

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

10. Change the **Front Panel Security** (LCD configuration) to show the hostname in the LCD display.

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

Figure 22. iDRAC Settings: Front Panel Security



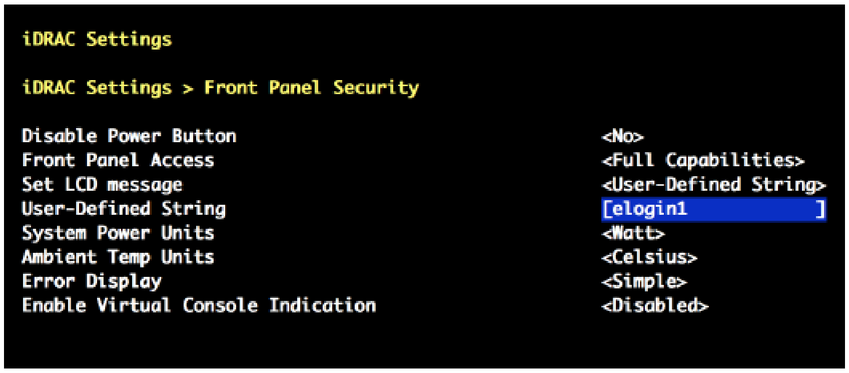
```
iDRAC Settings

iDRAC Settings

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

b. Select **Set LCD message** from the **Front Panel Security** menu. A pop-up window opens.

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

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

Figure 23. Front Panel Security:



```
iDRAC Settings

iDRAC Settings > Front Panel Security

Disable Power Button                                    <No>
Front Panel Access                                      <Full Capabilities>
Set LCD message                                         <User-Defined String>
User-Defined String                                     [elogin1            ]
System Power Units                                      <Watt>
Ambient Temp Units                                      <Celsius>
Error Display                                           <Simple>
Enable Virtual Console Indication                       <Disabled>
```

e. Enter the hostname (such as: `elogin1`) in the text pop-up window for **User-Defined String**, then press **Enter**.

f. Press the <**Esc**> key to exit the **Front Panel Security** screen.

g. Press the <**Esc**> key to exit the **Network** screen.

h. Press the <**Esc**> key to exit the **iDRAC Settings** screen.

i. Select **Yes** to save changes when prompted with the "`Settings have changed`" message.

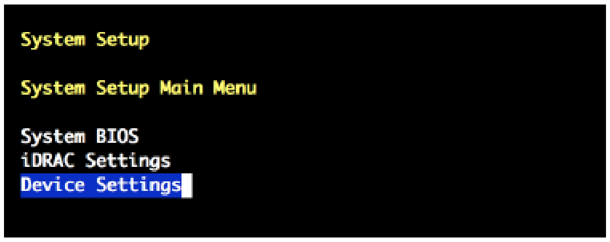
j. Select **OK** when prompted with the "`Settings saved successfully`" message.

11. Navigate to the **System Setup Main Menu** screen.

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

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

Figure 24. System Setup Main Menu: Device Settings



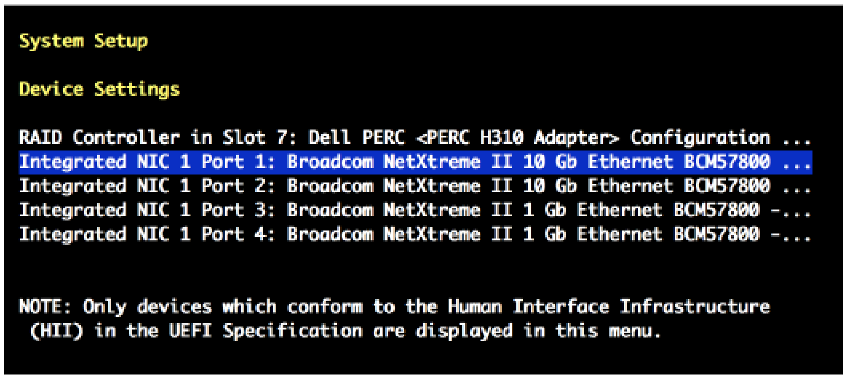
```
System Setup

System Setup Main Menu

System BIOS
iDRAC Settings
Device Settings
```

b. 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 25. Device Settings: Integrated NIC Port Number



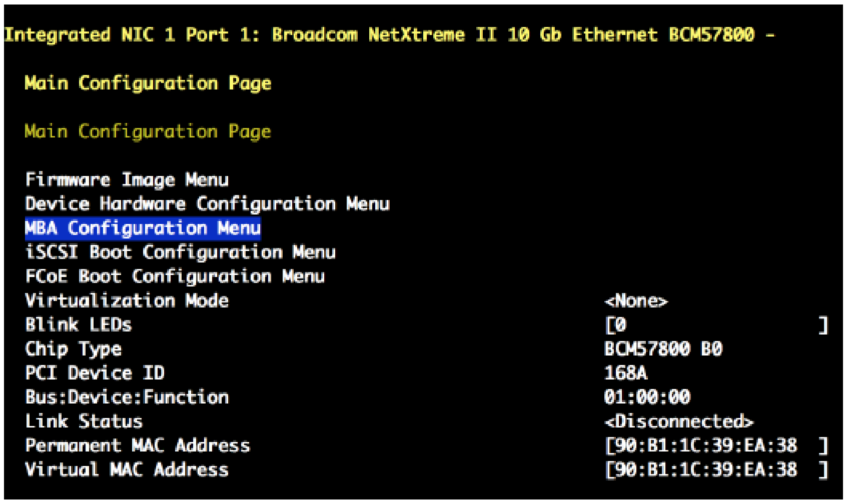
```
System Setup

Device Settings

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


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

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

After the correct NIC port number is selected, press **Enter** to open the **Main Configuration Page**.

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

Figure 26. Integrated NIC Port: MBA Configuration Menu



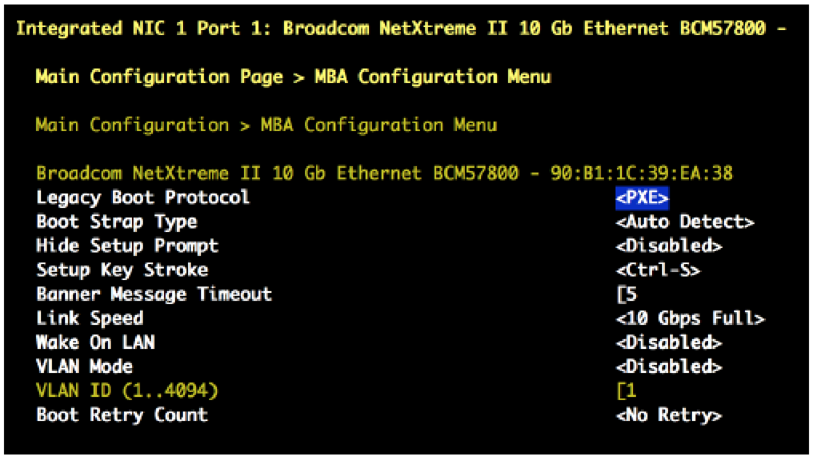
```
Integrated NIC 1 Port 1: Broadcom NetXtreme II 10 Gb Ethernet BCM57800 -

  Main Configuration Page

  Main Configuration Page

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

d. Select **Legacy Boot Protocol** from the **MBA Configuration Menu** screen, then press **Enter**.

   A pop-up window displays the available options.

e. Highlight **<PXE>** option for **Legacy Boot Protocol** in the pop-up window using the down-arrow key, then press **Enter**.

Figure 27. MBA Configuration Menu: Legacy Boot Protocol



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

  Main Configuration Page > MBA Configuration Menu

  Main Configuration > MBA Configuration Menu

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

f. Press the <**Esc**> 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, repeat the previous steps to change the setting for these three ports.

h. Press the <**Esc**> key to exit the **Device Settings** screen.

i. Select **Yes** to save changes when prompted with the "`Settings have changed`" message, then press **Enter**.

j. Select **OK** when prompted with the "`Settings saved successfully`" message, then press **Enter**.

   The **System Setup Main Menu** screen appears.

13. Exit and reboot

a. Press <**Esc**> to exit the **System Setup Main Menu**.

b. Select **Yes** when prompted with the `"Are you sure you want to exit and reboot?"` message.

8.18 Validate an eLogin Node

Prerequisites

Successful completion of an eLogin image deployment or update.

About this task

This procedure tests basic eLogin node functionality. If the commands run successfully, eLogin is functional and can be released to users.

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

Procedure

1. Log on to the eLogin node.

2. Generate an SSH key pair.

```
elogin$ ssh-keygen
```

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

```
elogin$ ssh-copy-id <login_name>
```

4. Test the `eswrap` utility.

```
elogin$ cnselect
20-27,32-43,48-51,60-63
elogin$ xtprocadmin
  NID    (HEX)    NODENAME     TYPE    STATUS        MODE
    1      0x1  c0-0c0s0n1  service        up interactive
    2      0x2  c0-0c0s0n2  service        up interactive
    5      0x5  c0-0c0s1n1  service        up interactive
    6      0x6  c0-0c0s1n2  service        up interactive
   20     0x14  c0-0c0s5n0  compute        up interactive
   21     0x15  c0-0c0s5n1  compute        up interactive
   22     0x16  c0-0c0s5n2  compute        up interactive
...
elogin$ xtnodestat

     C0-0
  n3      ;; ;;;X;  ;
  n2 SS   ;;S;;;X;  ;
  n1 SS   ;;S;;;X;  ;
c0n0      ;; ;;;X;  ;
    s0123456789abcdef
```

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

Available compute nodes:         28 interactive,          0 batch
```

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

```
elogin$ **aprun hostname**
nid00020
Application 21221 resources: utime ~0s, stime ~0s, Rss ~4256, inblocks ~0,
outblocks ~0
```

6. Test PBS or Moab/Torque if installed on the system.

```
elogin$ pbsnodes -a
percival-p1_305
    Mom = nid00008,nid00043
    ntype = PBS
    state = free
    pcpus = 8
    resv_enable = True
    sharing = force_exclhost
    resources_available.arch = XT
    resources_available.host = percival-p1_305
    resources_available.mem = 67108864kb
    ...
elogin$ qstat
Job id            Name             User              Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
2034657.sdb       STDIN            crayadm           00:00:00 R workq
elogin$ qsub -I
qsub: waiting for job 2034657.sdb to start
qsub: job 2034657.sdb ready
```

7. Test Slurm if installed on the system.

```
elogin$ squeue
  JOBID     USER ACCOUNT      NAME  ST REASON  START_TIME     TIME  TIME_LEFT
NODES CPUS
 131669      xmp  (null)   testMPI   R None    10:37:09       1:36
8:24     5   10
 131543      ymp  (null)       sst   R None    09:27:32    1:11:13
2:48:47     2   32
 131534      c90  (null)      bash   R None    09:23:31    1:15:14
4:44:46     1   24
elogin$ sinfo
PARTITION AVAIL JOB_SIZE  TIMELIMIT   CPUS  S:C:T   NODES STATE      NODELIST
workq*    up    1-infini   infinite     48 2:12:2       1 drained    nid00022
workq*    up    1-infini   infinite     48 2:12:2       5 mixed
nid000[13-15,20-21]
workq*    up    1-infini   infinite     48 2:12:2       5 allocated
nid000[08-12]
workq*    up    1-infini   infinite     48 2:12:2      41 idle
nid000[23-63]
```

```
elogin$ salloc
salloc: Granted job allocation 131674
```

9 eLogin Diagnostic Tools

9.1 Connect to an eLogin Console

About this task

Use `ironic_conman` to connect to an eLogin console.

Procedure

1. List the nodes to find the Universally Unique IDentifier (UUID) for the host.

```
root@cmc# ironic node-list
+-----------------------+---------+-----------+-------------+--------------------+-------------+
| UUID                  | Name    | Inst UUID | Power State | Provisioning State |Maintenance  |
+-----------------------+---------+-----------+-------------+--------------------+-------------+
| 98bc4fea-351c-57d4... |elogin1  | None      | power off   | available          | False       |
| e2f15f21-d217-4332... |elogin2  | None      | power off   | available          | False       |
+-----------------------+---------+-----------+-------------+--------------------+-------------+
```

2. Attach to the desired console.

```
root@cmc# ironic_conman ironic_node_name
```

```
root@cmc# ironic_conman elogin1
```

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

9.2 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.

```
root@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.

```
root@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
```

9.3 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.

9.4 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.

9.5 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:

```
root@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.
```

9.5.1 Configure eLogin Cray_dumpsys Plugin

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

For example:

```
 [cray_dumpsys]
 # List of enabled Cray_dumpsys plugins.
 plugins =
   process,
   networking,
   memory,
   openvswitch,
   mysql,
   openstack_cinder,
   openstack_horizon,
   openstack_keystone,
   openstack_neutron,
   openstack_nova,
   openstack_swift,
   newtplugin,
   elogin

 # List of Cray_dumpsys plugin options.
 options =
   openstack_cinder.log=off
   openstack_horizon.log=off
   openstack_keystone.log=off
   openstack_nova.cmds=on
   openstack_nova.log=off
   openstack_swift.log=off
   elogin.nodes="elogin1 elogin2"
```

To override the configured node list, use the `cray_dumpsys` option: `--extra-option elogin.nodes="elogin1 elogin2"`.

9.6 kdump and crash

The `kdump` utility is installed and enabled by default to ensure that data is saved in the event of a server kernel panic.

For details about creating a dump file, refer to *Create Dump File Using kdump Service* on page 65, in the *XC Series eLogin Administration Guide CLE 6.0 UP02 Rev A*.

For details on customizing `kdump`, refer to the *RedHat 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 also responsible for copying and/or removing `kdump` data from tenant nodes.

9.7 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 10. 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`

OpenStack Service	Log File Location
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.

9.8 OpenStack Diagnostic Commands

9.8.1 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`

```
root@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.

```
root@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                                                                   |
+-----------------------+------------------------------------------------------------------------+
```

9.8.2 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.

```
root@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.

```
root@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                          |
+--------------------------------------+-----------------------------------------------------------+
```

9.8.3 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.

```
root@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.

```
root@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'}                                                               |
+------------------------+-------------------------------------------------------------------------+
```

10 Common Issues

10.1 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.

10.2 Recovering from Broken CSMS Installation

Prerequisites

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

About this task

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

Procedure

1. Stop all OpenStack services.

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

2. Drop schemas directly related to OpenStack.

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

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

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

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

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

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

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

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

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

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

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

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

10.3 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.

   ```
   root@cmc# source ~/admin.openrc
   root@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).

10.4 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 is 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 is 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>` will only be 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>` will only be 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 must be 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`.

   ```
   root@smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/roles/ \
   simple_sync/classes/common

   root@smw# mkdir -p /var/opt/cray/imps/config/sets/p0/files/ \
   roles/simple_sync/hostnames
   ```

2. Push the config set to the CMC.

   ```
   root@smw# cfgset push -d <cmc-name> <config_set_name>
   ```

3. Log into the CMC, and run `add_configset`.

   ```
   root@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`