XC™ Series eLogin Administration Guide

(CLE 6.0.UP01)

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. It assumes competence with standard Linux and open source tools.

Record of Revision

Revision	Date	Content Information
XC Series eLogin Administration Guide CLE6.0 UP02 v.1 draft (this publication)	July 13, 2016	Revised content for UP02
XC Series eLogin Administration Guide CLE6.0 UP01	June 15, 2016	● Final UP01 release ● Content aligned with Alpaca 1.1.1 ● Upgrade processes not included
XC Series eLogin Administration Guide CLE6.0 UP01 v.1	June 2, 2016	● Content aligned with Alpaca 1.1.1 ● Upgrade processes not included

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

Typographic Conventions

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

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

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

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.

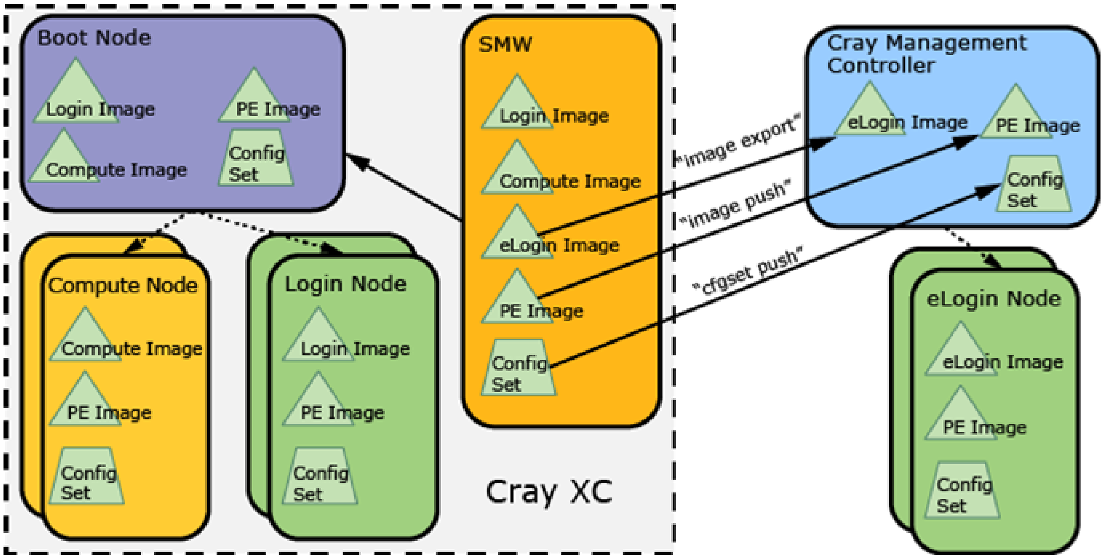
2 eLogin Architecture

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 Software Management Workstation (SMW). The SMW provides shared image and configuration services.

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

Figure 1. 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.

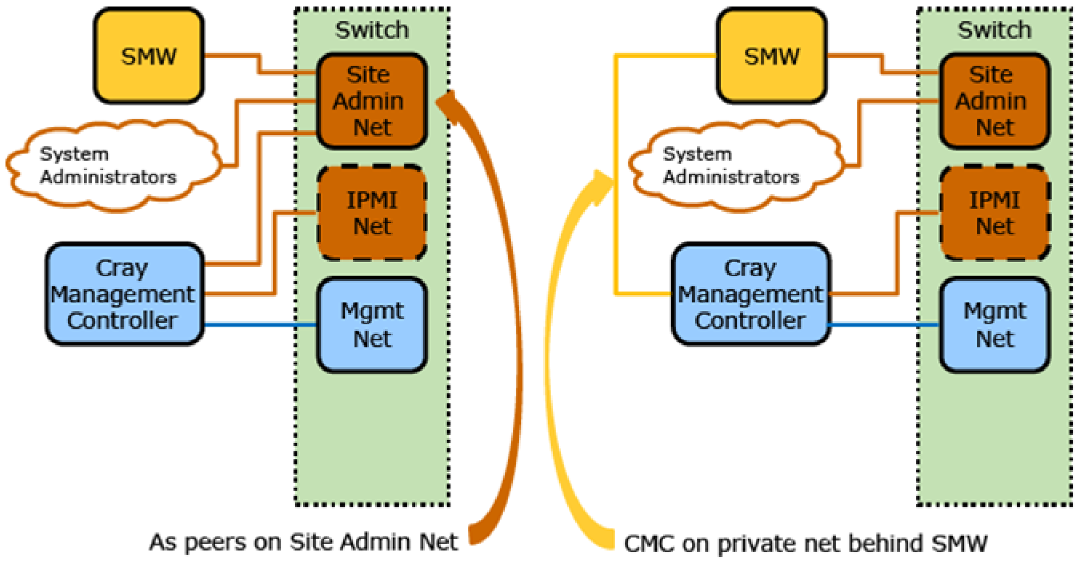
WARNING: For security reasons, configure the CMC behind the SMW on a private network.

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.

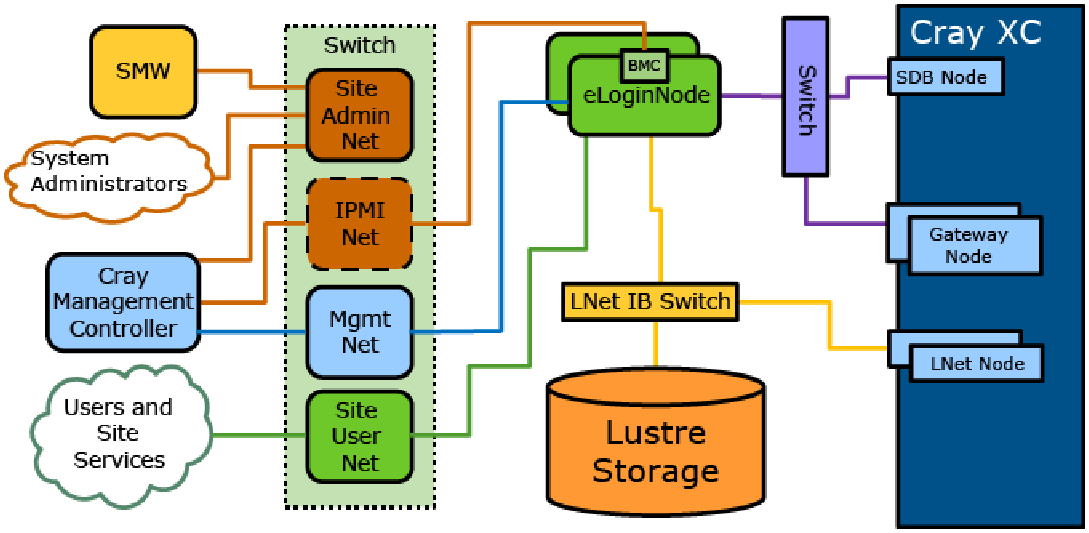
eLogin to Cray Network Attachment

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 or 10GbE depending on site infrastructure.

eLogin Nodes Direct Connection to SDB Node

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

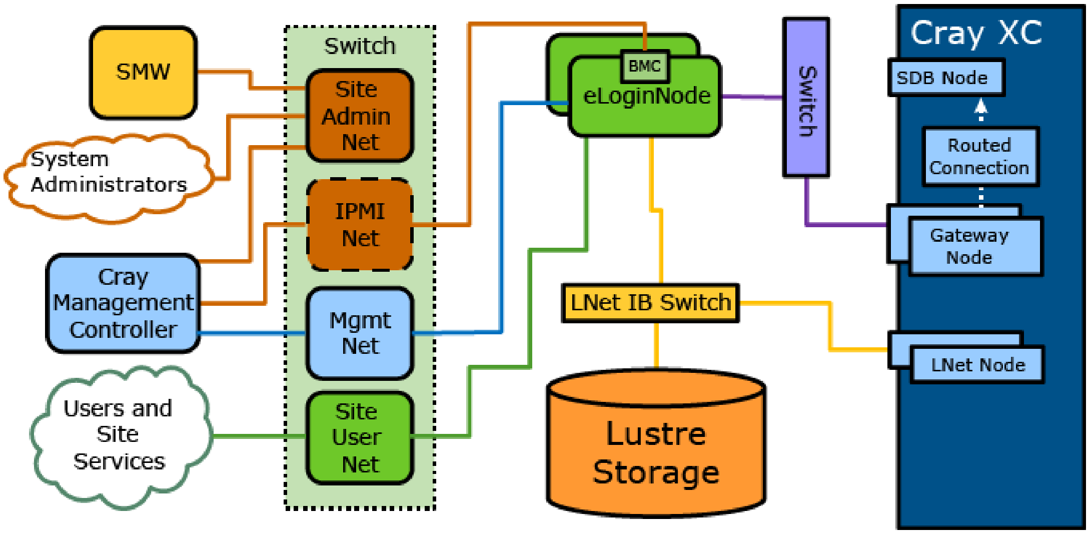
Figure 3. eLogin Nodes Direct Connection to SDB Node: eLogin System Topology



eLogin Nodes Routed Via Gateway to SDB Node

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

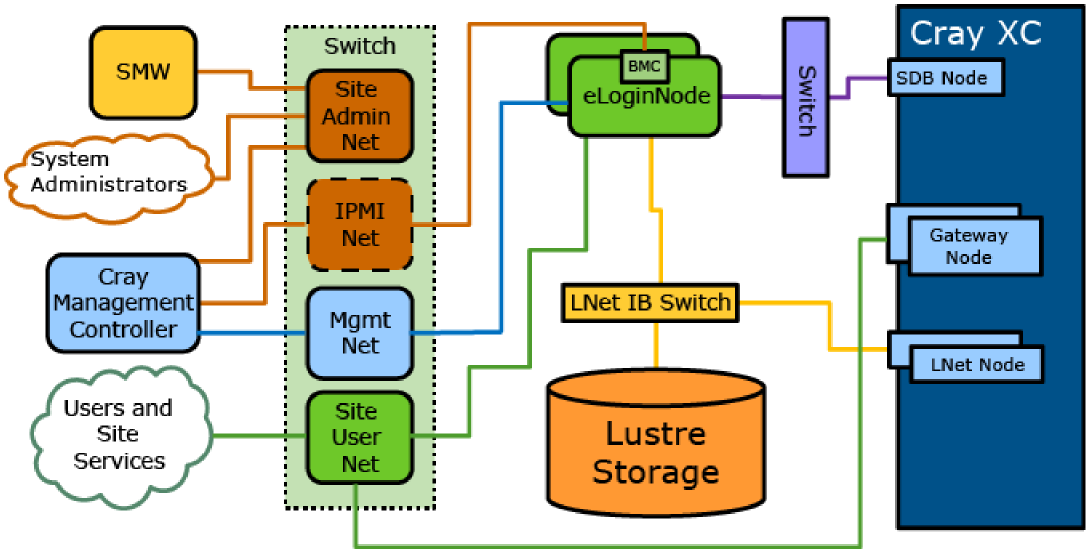
Figure 4. eLogin Nodes Routed Via Gateway to SDB Node: eLogin System Topology



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

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

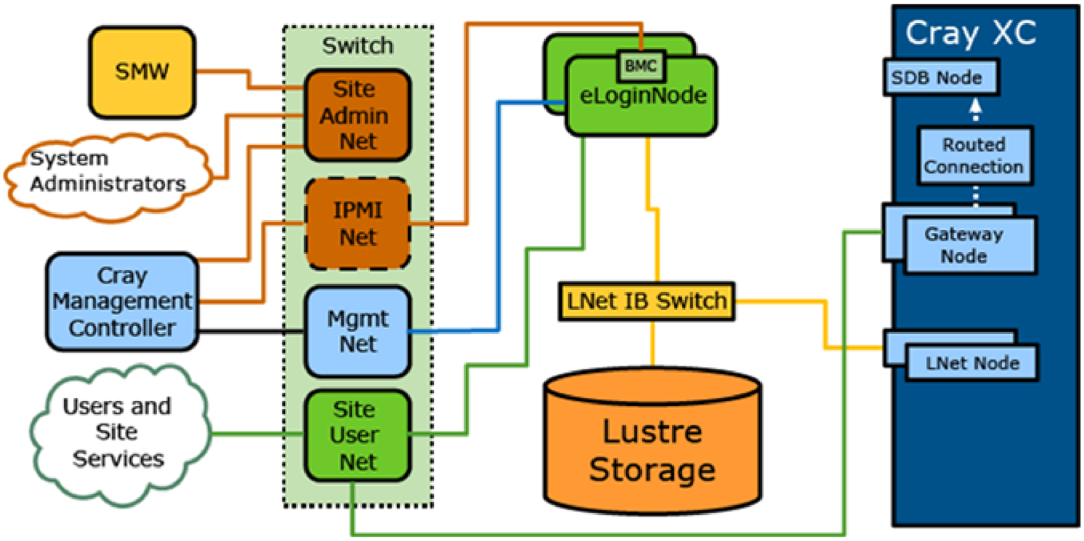
Figure 5. eLogin Nodes Direct Connection to SDB Node with Site User Gateway: eLogin System Topology



eLogin Nodes Routed to SDB Node with User Accessible Gateway

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

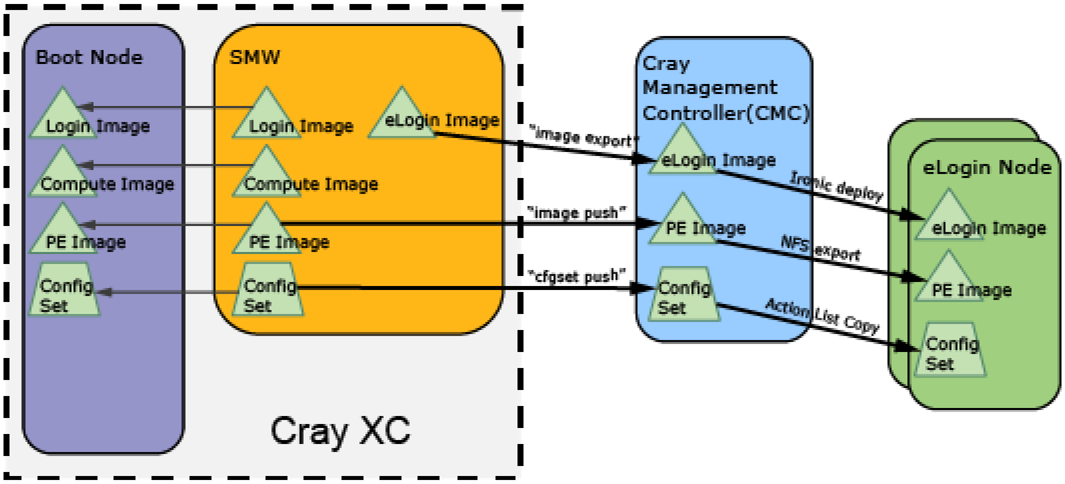
Figure 6. eLogin Nodes Routed to SDB Node via User Accessible Gateway: eLogin System Topology



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 boot process, the PE image is exported to the eLogin node with `nfs_export`.

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	

Each CDL node also needs the following information:

Item	Value
BMC IP address	
Boot interface MAC address	

Common Configuration Options

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

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

Table 1. Common Configuration Options

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

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

4 Configure and Manage an eLogin Image

Prerequisites

A complete successful CSMS/eLogin installation.

About this task

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

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

SMW Image Creation and Export

Procedure

1. Connect to the SMW.

   ```
   # ssh smw
   ```

2. Select an eLogin image type.

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

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

3. Optional: Create a custom eLogin image recipe.

   Perform this step if either one of these conditions apply:

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

   Create a new eLogin image recipe by cloning `elogin-large_cle_6.0up01_sles_12_x86-64_ari_20150601`. Prepend the function of the customization to the original user name of a custom image (example, *`username-function`*).

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

4. Build the eLogin image.

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

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

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

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

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

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

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

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

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

7. Push the config set to the CMC.

   The config set was generated during CLE installation and modified in *Create a Minimum eLogin Config Set*.

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

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

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

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

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

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

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

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

CSMS Image Deployment

9. Connect to the CMC node.

```
# ssh cmc
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

12. Create the config set action list:

a. Move to the Heat stack template directory.

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

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

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

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

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

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

d. Add the action list to Glance.

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

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

13. Configure the deployment of images and deploy.

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

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

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

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

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

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

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

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

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

`image_id` Name of the image pushed from the SMW and appended with `.qcow2`. To display the image name, use `glance image-list`.

`host_name` The host name of the node to be deployed.

`fixed_ip` The static IP address of the management interface on this eLogin node. This must be an IP address in the management network that is unique to the node. The `fixed_ip` address is only available in the `elogin-env-fixed-ip.yaml.template`.

`instance_flavor` Nova flavor of the CDL being booted. In most cases, use `eloginflavor`.

`cray_config_set` Name of config set to use.

`cims_host_name` Host name of the management controller (not an alias).

`ironic_id` `UUID` of the node being booted by this stack. To determine the `UUID`, use the `ironic node-list` command. This is used to target specific hardware.

`actions_list` A list of additional actions to take. This list must have the value of the config set action list uploaded above for the appropriate config set.

d. Create a Heat template.

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

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

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

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

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

- `elogin_template.yaml`: If `elogin-env.yaml.template` was used.

or

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

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

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

e. Create the Heat stack.

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

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

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

To access the console of an eLogin node:

1. Find the Ironic name of the eLogin node.

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

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

   ```
   cmc# ironic_conman ironic_name
   ```

For more details, refer to *eLogin Console Access*.

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

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

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

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

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

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

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

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

14. Repeat the previous step for each eLogin node.

5 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).

5.1 Update Site Admin IP Address of Management Controller

About this task

Change 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:

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

4. Run `csms_install.sh`.

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

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

7. Run `csms_install.sh`.

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

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

5.2 Changing Passwords On CMC

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

- Linux root user account
- Ansible Vault files
- Root MariaDB database account
- Keystone admin user account
- Keystone MariaDB database 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.

5.2.1 Change Admin Password After CSMS Installation Prerequisites

Prerequisites

Successful installation of CSMS.

Procedure

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

   ```
   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 listed and enter the new password when prompted.

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

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

   ```
   cmc# keystone user-list
   ```

5.2.2 Change Ansible Vault Password After CSMS Installation Prerequisites

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`.

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

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

5.2.3 Change Root MariaDB Database Password After CSMS Installation Prerequisites

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.

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

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

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

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

```
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.

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

6. (Conditional) If monitoring is enabled, configure the OpenStack Monasca Agent to use the new MariaDB root password.

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

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

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

5.2.4 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 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.

```
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:

```
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 management controller.

5.3 Deploy, Reboot, and Shutdown eLogin Nodes

5.3.1 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:

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

Procedure

1. Copy the appropriate `elogin-env*.yaml.template` file to the following directory, and rename with the eLogin node hostname (for example, `elogin1`).

```
cmc# cd /etc/opt/cray/openstack/elogin/
cmc# cp /etc/opt/cray/openstack/heat/templates/\
elogin-env-fixed-ip.yaml.template ./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: 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 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 password with rsync or a fixed IP address on the management network.

   The following table maps eLogin Heat template files with the `elogin-env*.yaml` files.

   Table 2. eLogin Heat Template File Mapped with eLogin-env Template File

eLogin-env Template File	eLogin Heat Template File
`elogin-env.yaml.template`	`elogin_template.yaml_elogin_template_rsync_password.yaml`

`elogin-env-fixed-ip\`
`.yaml.template`
	`elogin_template_fixed_ip.yaml\`
	`_elogin_template_rsync_password_fixed_ip.yaml`

5.3.2 Reboot An eLogin Node

Prerequisites

eLogin node is in shut down state.

About this task

Run the `nova reboot` command at the command line to reboot an eLogin node.

```
cmc# nova rebootelogin_node_name
```

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

5.3.3 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 `shutdown -H` at the command line, to shutdown the eLogin node.

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

3. Run the following command from the CMC to delete the Heat stack from the shut down eLogin node.

   ```
   cmc heat stack-delete elogin_node_name
   ```

5.4 Update Config Set on Running Nodes

Prerequisites

System is running and eLogin nodes are deployed.

About this task

All configuration changes are handled via the configuration set (config set) mechanism.

Procedure

1. Modify the config set on the SMW, and push to the CMC using the normal procedures described for handling config sets.

2. Create a hosts file with the hostname of each eLogin node intended for update: one per line under the `[update_hosts]` section.

   ```
    [update_hosts]
   elogin1
   elogin2
   elogin3
   ```

3. Update the running nodes in the hosts file.

   ```
   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.

5.5 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.

To ensure that the CSMS ignores an eLogin node, set the node into maintenance mode.

Procedure

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

   ```
   cmc# ironic node-set-maintenance eLogin_node_name on
   ```

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

   ```
   cmc# ironic node-set-maintenance eLogin_node_name off
   ```

5.6 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.

   ```
   smw# cfgset search -s cray_elogin_lnet config_set

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

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

```
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.

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

e. Proceed to step *4* on page 29 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.

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

4. Verify that Lustre functions correctly.

```
elogin# ip addr show ib0
4: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast state UP 
group default qlen 256
       link/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

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

5.7 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.

5.8 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.

5.9 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 3. Ironic Inventory File (CSV format)

Field Name	Description
`NODE_NAME`	Node name.
`BMC_IP`	Baseboard Management Controller (BMC) IP address of the node.

Field Name	Description
`VIRTUALBOX_HOST`	IP or hostname of the VirtualBox webservice 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, e.g. 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.
`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 4. 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`

Field Name	Required	Multiple Allowed	Example Value
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 5. Ironic Inventory File Syntax

<table>
<tr><th>Field Name</th><th>Name RegEx (case sensitive)</th></tr>
<tr><td>NODE_NAME</td><td>^(node)?[_]?name$</td></tr>
<tr><td>BMC_IP</td><td>^bmc[_]?ip[_]?(address)?$</td></tr>
<tr><td>VIRTUALBOX_HOST</td><td>^(virtualbox[_]?)?host$</td></tr>
<tr><td>VIRTUALBOX_VMNAME</td><td>^(virtualbox[_]?)?vmname$</td></tr>
<tr><td>SNMP_POWER_DRIVER</td><td>^(snmp[_]?)?(power[_]?)?driver$</td></tr>
<tr><td>SNMP_POWER_ADDRESS</td><td>^(snmp[_]?)?(power[_]?)?address$</td></tr>
<tr><td>SNMP_POWER_OUTLET</td><td>^(snmp[_]?)?(power[_]?)?outlet$</td></tr>
<tr><td>SNMP_POWER_COMMUNITY</td><td>^(snmp[_]?)?(power[_]?)?community$</td></tr>
<tr><td>PXE_OVERLAY</td><td>^(pxe[_]?)?overlay$</td></tr>
<tr><td>MAC_ADDR</td><td>^(((<regex>))[_]?)?mac([_]?addr(ess)?)?$</td></tr>
<tr><td>N_CPUs</td><td>^(n[_]?)?cpus$</td></tr>
<tr><td>ARCH</td><td>^(cpu[_]?)?arch(itecture)?$</td></tr>
<tr><td>RAM_MB</td><td>^(ram|memory)[_]?([(]?mb[)]?)?$</td></tr>
<tr><td>DISK_GB</td><td>^(disk|local)[_]?([(]?gb[)]?)?$</td></tr>
<tr><td>NODE_CAPS</td><td>^(node[_]?)?(capabilities|caps)$</td></tr>
<tr><td>NODE_DESC</td><td>^(node[_]?)?desc(ription)?$</td></tr>
</table>

5.10 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.

5.11 How to Use eLogin

5.11.1 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
   ```

5.11.2 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
        dwci
        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
```

5.12 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.

5.13 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
```

6 eLogin Diagnostic Tools

6.1 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.

```
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.

```
cmc# ironic_conman ironic_node_name
```

```
cmc# ironic_conman elogin1
```

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

6.2 The journalctl Command

`systemd` (on both the management controller and eLogin nodes) forgoes traditional logging mechanisms, and instead stores the following messages in a custom database:

- `syslogd` messages
- Kernel log messages
- Initial RAM disk and early boot messages
- Messages written to `stderr`/`stdout` for all services

Access to the information in the database is through the `journalctl` tool.

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

```
elogin# journalctl -a
-- Logs begin at Mon 2015-06-08 19:28:53 UTC, end at Thu 2015-06-11 22:15:01 UTC. --
Jun 08 19:28:53 example-elogin2 systemd-journal[1681]: Runtime journal is using 8.0M\
   (max allowed 4.0G, trying to leave 4.0G free of 252.4G available → current limit 4.0G).
Jun 08 19:28:53 example-elogin2 systemd-journal[1681]: Runtime journal is using 8.0M \
   (max allowed 4.0G, trying to leave 4.0G free of 252.4G available → current limit 4.0G).
Jun 08 19:28:53 example-elogin2 kernel: Initializing cgroup subsys cpuset
Jun 08 19:28:53 example-elogin2 kernel: Initializing cgroup subsys cpu
Jun 08 19:28:53 example-elogin2 kernel: Initializing cgroup subsys cpuacct
Jun 08 19:28:53 example-elogin2 kernel: Linux version 3.12.28-4-default \
   (geeko@buildhost) (gcc version 4.8.3 20140627 [gcc-4_8-branch revision 212064] \
   (SUSE Linux) ) #1 SMP Thu Sep 25 17:02:34 UTC 2014 (9879bd4)
Jun 08 19:28:53 example-elogin2 kernel: Command line: \
   initrd=/var/lib/tftpboot/e79e85cd-57f5-4fcd-ba43-14ccea0375e7/ramdisk \
   root=UUID=f09a21f4-1bb1-4b1e-8a12-c5329e4b9073 ro text nofb nomodeset vga=normal \
   BOOT_IMAGE=/var/lib/tftpboot/e79e85cd-57f5-4fcd-ba43-14ccea0375e7/kernel BOOTIF=01-90-
b1-1c-39-ea-3c
```

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

```
elogin# journalctl  -f /usr/sbin/ntpd
-- Logs begin at Mon 2015-06-08 19:28:53 UTC, end at Thu 2015-06-11 22:15:01 UTC. --
Jun 08 19:30:00 example-elogin2 ntpd[3436]: ntpd 4.2.6p5@1.2349-o Wed Oct  8 14:41:40 UTC
2014 (1)
Jun 08 19:30:00 example-elogin2 ntpd[3437]: proto: precision = 0.103 usec
Jun 08 19:30:00 example-elogin2 ntpd[3437]: ntp_io: estimated max descriptors: 1024, initial
socket boundary: 16
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen and drop on 1 v6wildcard :: UDP 123
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen normally on 3 eth2 10.142.0.111 UDP 123
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen normally on 4 lo ::1 UDP 123
Jun 08 19:30:00 example-elogin2 ntpd[3437]: Listen normally on 5 eth2
fe80::92b1:1cff:fe39:ea3c UDP 123
```

6.3 The /var/log Directory

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

6.4 Ansible Install Logs

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

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

6.5 The cray_dumpsys Command

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

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

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

sosreport (version 3.2)

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

Creating compressed archive...

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

The checksum is: bb87d9323f88813e07659e53aebb16b6

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

6.6 OpenStack Log File Locations

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

Table 6. 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

OpenStack Service	Log File Location
Neutron	`/var/log/neutron`
Nova	`/var/log/nova`
Swift	`/var/log/swift`

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

6.7 OpenStack Diagnostic Commands

6.7.1 Heat Diagnostic Commands

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

- `heat stack-list`

```
cmc# heat stack-list
+--------------------------------------+-------------------+-----------------+----------------------+
| id                                   | stack_name        | stack_status    | creation_time        |
+--------------------------------------+-------------------+-----------------+----------------------+
| 4452df3e-46f1-4345-8b61-c489bbbc863f | eLogin1           | CREATE_COMPLETE | 2015-06-11T20:52:39Z |
+--------------------------------------+-------------------+-----------------+----------------------+
```

- `heat stack-show` *`stack_name_or_id`*

  This command describes the stack.

```
cmc# heat stack-show elogin1
+-----------------------+------------------------------------------------------------------------+
| Property              | Value                                                                  |
+-----------------------+------------------------------------------------------------------------+
| capabilities          | []                                                                     |
| creation_time         | 2015-06-11T20:52:39Z                                                   |
| description           | Simple deploy template with parameters                                 |
| disable_rollback      | True                                                                   |
| id                    | 4452df3e-46f1-4345-8b61-c489bbbc863f                                   |
| links                 | http://172.30.50.129:8004/v1/acc067874bfd45dcbce9f44d1516910a/ \       |
|                       |       stacks/eLogin1/4452df3e-46f1-4345-8b61-c489bbbc863f (self)       |
| notification_topics   | []                                                                     |
| outputs               | [                                                                      |
|                       |   {                                                                    |
|                       |     "output_value": {                                                  |
|                       |       "management": [                                                  |
|                       |         "10.149.0.157"                                                 |
|                       |       ]                                                                |
|                       |     },                                                                 |
|                       |     "description": "IP assigned to the instance",                      |
|                       |     "output_key": "instance_ip"                                        |
|                       |   }                                                                    |
|                       | ]                                                                      |
| parameters            | {                                                                      |
|                       |   "network_id": "management",                                          |
|                       |   "OS::stack_id": "4452df3e-46f1-4345-8b61-c489bbbc863f",              |
|                       |   "OS::stack_name": "eLogin1",                                         |
|                      |   "cray_config_set": "sta_p2",                                             |
|                      |   "key_name": "default",                                                   |
|                      |   "instance_flavor": "eloginflavor",                                       |
|                      |   "cray_cims_ip": "10.149.0.1",                                            |
|                      |   "image_id": "whale_eLogin2.qcow2",                                       |
|                      |   "host_name": "elogin1"                                                   |
|                      | }                                                                          |
| parent               | None                                                                       |
| stack_name           | eLogin1                                                                    |
| stack_owner          | admin                                                                      |
| stack_status         | CREATE_COMPLETE                                                            |
| stack_status_reason  | Stack CREATE completed successfully                                        |
| template_description | Simple deploy template with parameters                                     |
| timeout_mins         | None                                                                       |
| updated_time         | None                                                                       |
+----------------------+----------------------------------------------------------------------------+
```

6.7.2 Nova Diagnostic Commands

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

- `nova list`

  This command lists active servers.

```
cmc# nova list
+-----------------------+------------+--------+------------+-------------+-------------------------+
| ID                    | Name       | Status | Task State | Power State | Networks                |
+-----------------------+------------+--------+------------+-------------+-------------------------+
| ac6384e2-...-4c9e1885 | elogin1    | ACTIVE | -          | Running     | management=10.142.0.156 |
+-----------------------+------------+--------+------------+-------------+-------------------------+
```

- `nova show server_name_or_id`

  This command displays details about the given server.

```
cmc# nova show example-elogin
+--------------------------------------+------------------------------------------------------------+
| Property                             | Value                                                      |
+--------------------------------------+------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                     |
| OS-EXT-AZ:availability_zone          | nova                                                       |
| OS-EXT-SRV-ATTR:host                 | cims                                                       |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | e63ffc33-029f-44ac-8808-c55909f85f2f                       |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000050                                          |
| OS-EXT-STS:power_state               | 1                                                          |
| OS-EXT-STS:task_state                | -                                                          |
| OS-EXT-STS:vm_state                  | active                                                     |
| OS-SRV-USG:launched_at               | 2015-06-11T21:01:16.000000                                 |
| OS-SRV-USG:terminated_at             | -                                                          |
| accessIPv4                           |                                                            |
| accessIPv6                           |                                                            |
| config_drive                         |                                                            |
| created                              | 2015-06-11T20:52:40Z                                       |
| flavor                               | eloginflavor (012435a2-54f7-458b-8734-6cdefe58b52e)        |
| hostId                               | 9e184dc6993ac9954652611f13f3faaaa797b5ff1625869be0edeb80   |
| id                                   | ac6384e2-4ca0-421f-9e6e-4c9e138f8785                       |
| image                                | eLogin2_new.qcow2 (1cc535c0-9f71-446a-8f4e-66aacc2617fe)   |
| key_name                             | default                                                    |
| management network                   | 10.149.0.156                                               |
| metadata                             | {"cray_config_set": "sta_p2", "cray_cims_ip": "10.147.0.1",
|                                      | "cray_cims_rsync_password": "fab9--679b47aca4",            |
|                                      | "cray_cims_rsync_username": "eLogin"}                      |
| name                                 | elogin1                                                    |
| os-extended-volumes:volumes_attached | []                                                         |
| progress                             | 0                                                          |
| security_groups                      | default                                                    |
| status                               | ACTIVE                                                     |
| tenant_id                            | acc067874bfd45dcbce9f44d1516910a                           |
| updated                              | 2015-06-11T21:01:16Z                                       |
| user_id                              | 762d33ecbeb64356a933e27bce688579                           |
+--------------------------------------+------------------------------------------------------------+
```

6.7.3 Ironic Diagnostic Commands

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

- `ironic node-list`

  Lists nodes that are registered with the Ironic service.

```
cmc# ironic node-list
+------------------------+-------------------------+-------------+--------------------+-------------+
| UUID                   | Instance UUID           | Power State | Provisioning State | Maintenance |
+------------------------+-------------------------+-------------+--------------------+-------------+
| e6300c33-...-c55952f9f | ac6384e2-...-4c9e138f85 | power on    | active             | False       |
+------------------------+-------------------------+-------------+--------------------+-------------+
```

- `ironic node-show identifier`

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

```
cmc# ironic node-show e63ffc33-029f-44ac-8808-c55909f85f2f
+------------------------+---------------------------------------------------------------------------+
| Property               | Value                                                                     |
+------------------------+---------------------------------------------------------------------------+
| instance_uuid          | ac6384e2-...--4c9e138f85                                                  |
| target_power_state     | None                                                                      |
| properties             | {u'memory_mb': 131072, u'cpu_arch': u'x86_64', u'local_gb': 550,          |
|                        | u'cpus': 32}                                                              |
| maintenance            | False                                                                     |
| driver_info            | {u'pxe_deploy_ramdisk': u'd867d80f-8847-40f0-a499-50de8a31a997',          |
|                        | u'pxe_deploy_kernel': u'a46d386c-3307-45fe-a395-8fc1d5c2285c',            |
|                        | u'ipmi_address': u'10.142.0.5', u'ipmi_username': u'root',                |
|                        | u'ipmi_password': u'initial0'}                                            |
| extra                  | {u'description': u'elogin1'}                                              |
| last_error             | None                                                                      |
| created_at             | 2015-05-18T21:25:54+00:00                                                 |
| target_provision_state | None                                                                      |
| driver                 | pxe_ipmitool_cm                                                           |
| updated_at             | 2015-06-12T14:54:29+00:00                                                 |
| maintenance_reason     |                                                                           |
| instance_info          | {u'ramdisk': u'960017eb-ef0b-439a-9dd1-331c9ed449ac', u'kernel': u        |
|                        | '9a51857a-008c-47cb-9768-92622fad7314', u'root_gb': u'100',               |
|                        | u'image_source': u'1cc535c0-9f71-446a-8f4e-66aacc2617fe', u'deploy_key':  |
|                        | u'J0O47487N9F4CLNMFWHSS9QLHHBTNLPF', u'swap_mb': u'10240'}                |
| chassis_uuid           | None                                                                      |
| provision_state        | active                                                                    |
| reservation            | None                                                                      |
| power_state            | power on                                                                  |
| console_enabled        | False                                                                     |
| uuid                   | e6300c33-...-c55952f0f                                                    |
+------------------------+---------------------------------------------------------------------------+
```

6.8 Common Issues

6.8.1 Disk Space On CMC and eLogin Node

Disk Space Issues On CMC

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

- Images fill up space in `/var/lib/glance`.

  Solution: Remove using Glance commands only.
- Images fill up space in `/var/lib/tftpboot`.

  Solution: These are removed automatically following a successful deployment. If they remain, remove manually.
- PE, config sets, and repositories fill up space in subdirectories of `/var/opt/cray`.

  Solution: Remove manually.

Disk Space on eLogin Node

The eLogin node is partitioned into two virtual disks:

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

6.8.2 Recovering from Broken CSMS Installation

Prerequisites

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

About this task

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

Procedure

1. Stop all OpenStack services.

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

2. Drop schemas directly related to OpenStack.

   ```
   cmc# mysql
   Welcome to the MariaDB monitor. Commands end with ; or \g.
   Your MariaDB connection id is 34562
   Server version: 5.5.41-MariaDB MariaDB Server
   +--------------------+
   ```

```
14 rows in set (0.00 sec)
MariaDB [(none)]> drop database cinder;
Query OK, 21 rows affected (0.08 sec)

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

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

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

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

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

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

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

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

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

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

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

6.8.3 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.

   ```
   csms# source ~/admin.openrc
   csms# ironic node-show percival-elogin3 | grep ipmi_address \
   | driver_info | {u’ipmi_password’: u’******’, u’ipmi_address’: u’10.142.0.7’
   ```

2. Open a browser on the CMC, and enter the `ipmi_address` value for the URL. (10.142.0.7, in this example.)

3. Enter the credentials for the iDRAC (`root/initial0`).

4. Locate **Hardware** directory from left-hand side of window, and click to expand.

5. Click on **Network Devices** under **Hardware**.

6. Click on **Integrated NIC1** in the main window.

7. Click on **+** to expand the information for **Port 3**.

8. Browse to **Port Properties** and verify the **Family Firmware Version** is `7.2.20` or later.

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