eLogin Made Easy

Jeff Keopp and Blaine Ebeling

Agenda

- What is eLogin?
- Introduction to eLogin
- Introduction to Cray System Management Software (CSMS)
- Managing eLogin nodes
- Summary
- Q&A

What is eLogin?

- **Next Generation Cray Development and Login server (CDL)**
  - Provides a login, job submission, and development environment for the Cray XC
  - External to the Cray XC, therefore it is available to users independent of Cray XC availability
  - Replaces previous esLogin-based CDL for Cray XC systems running CLE 6.x
- **eLogin vs esLogin:**

	eLogin	esLogin
Image Management	Prescriptively built on the SMW from CLE, eLogin and any customer provided sources	ESL image ISO built and released by Cray
Programming Environment	Shared Cray PE image synchronized to each eLogin node	Cray PE installed inside of the ESL image
System Management	Cray System Management Software	Bright Cluster Manager

What is eLogin?

- **eLogin hardware**
  - 1U or 2U Rack Mounted server
  - 20 or 24 processor cores
  - 256 or 512 GB Memory
  - 2x 1.2 TB SAS drives
    - 1 drive for the operating system
    - 1 drive for Cray PE and other persistent data
  - 2x 10 GbE LOM ports
  - 2x 1 GbE LOM ports
  - 3 or 6 PCIe slots (1U or 2U format)
    - 1 dual-port IB card for LNet connection

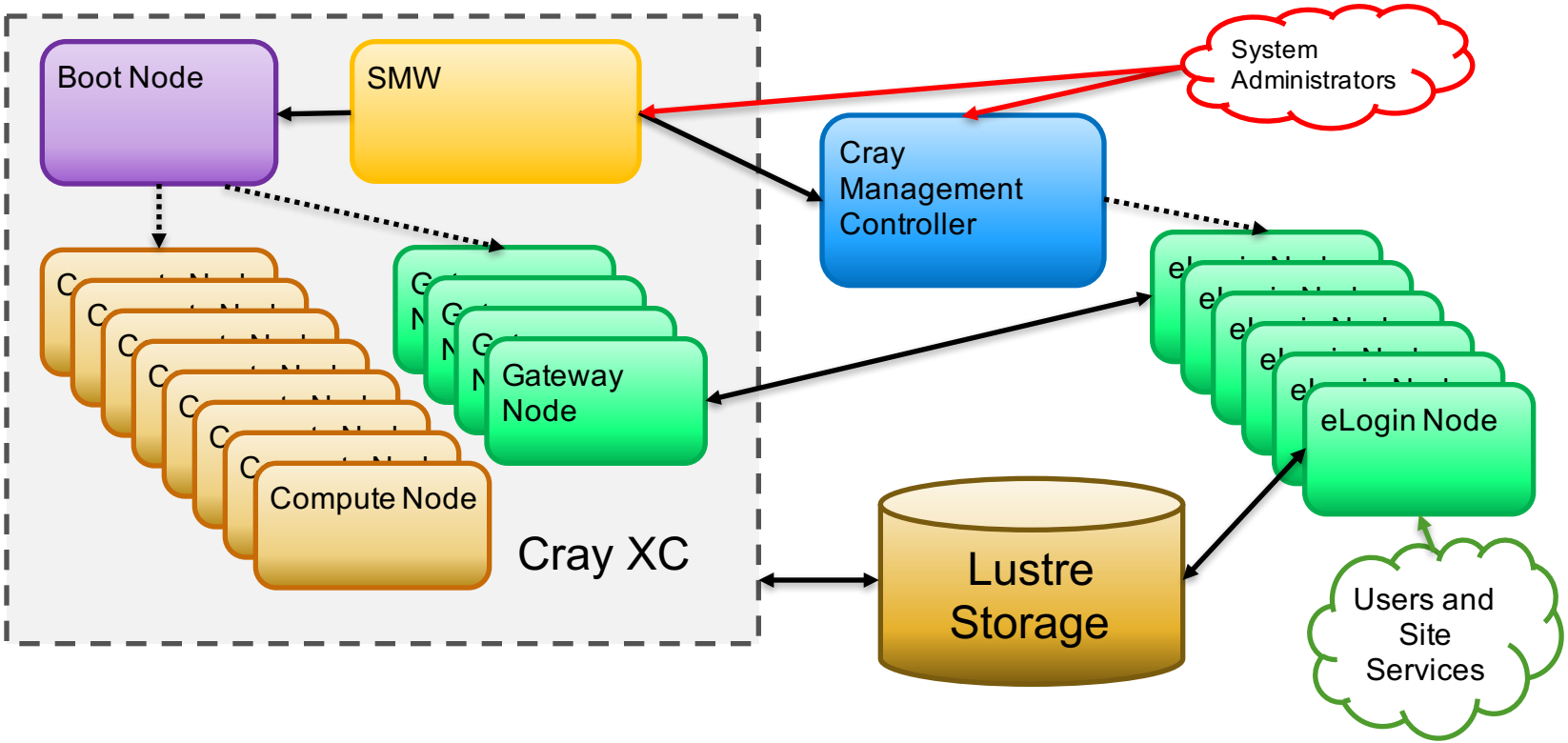
Agenda

- What is eLogin?
- Introduction to eLogin
- Introduction to Cray System Management Software (CSMS)
- Managing eLogin nodes
- Summary
- Q&A

Introduction to eLogin

- System Topology
- Software Images
- Cray Programming Environment (Cray PE)
- eLogin Configuration

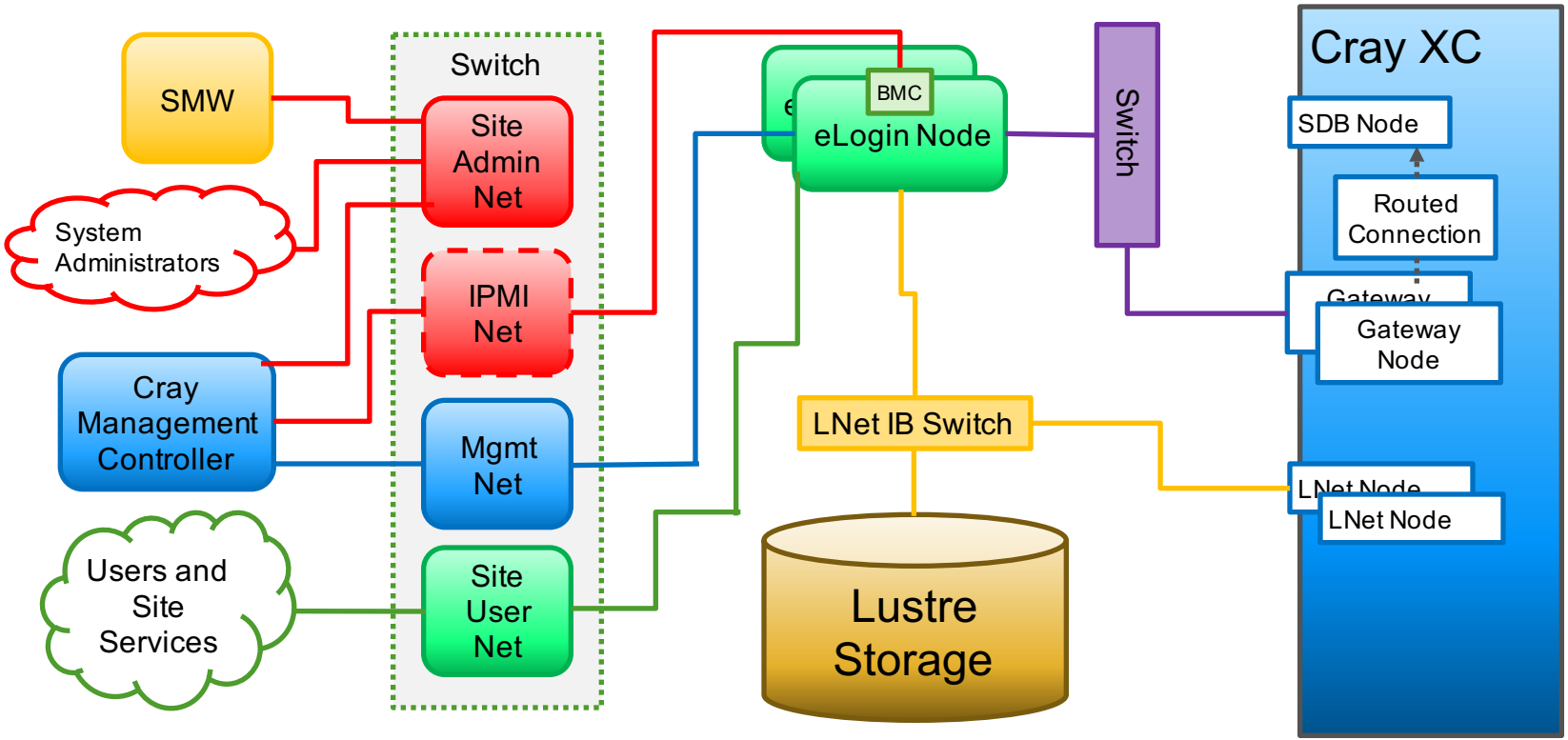
Introduction to eLogin: System Topology



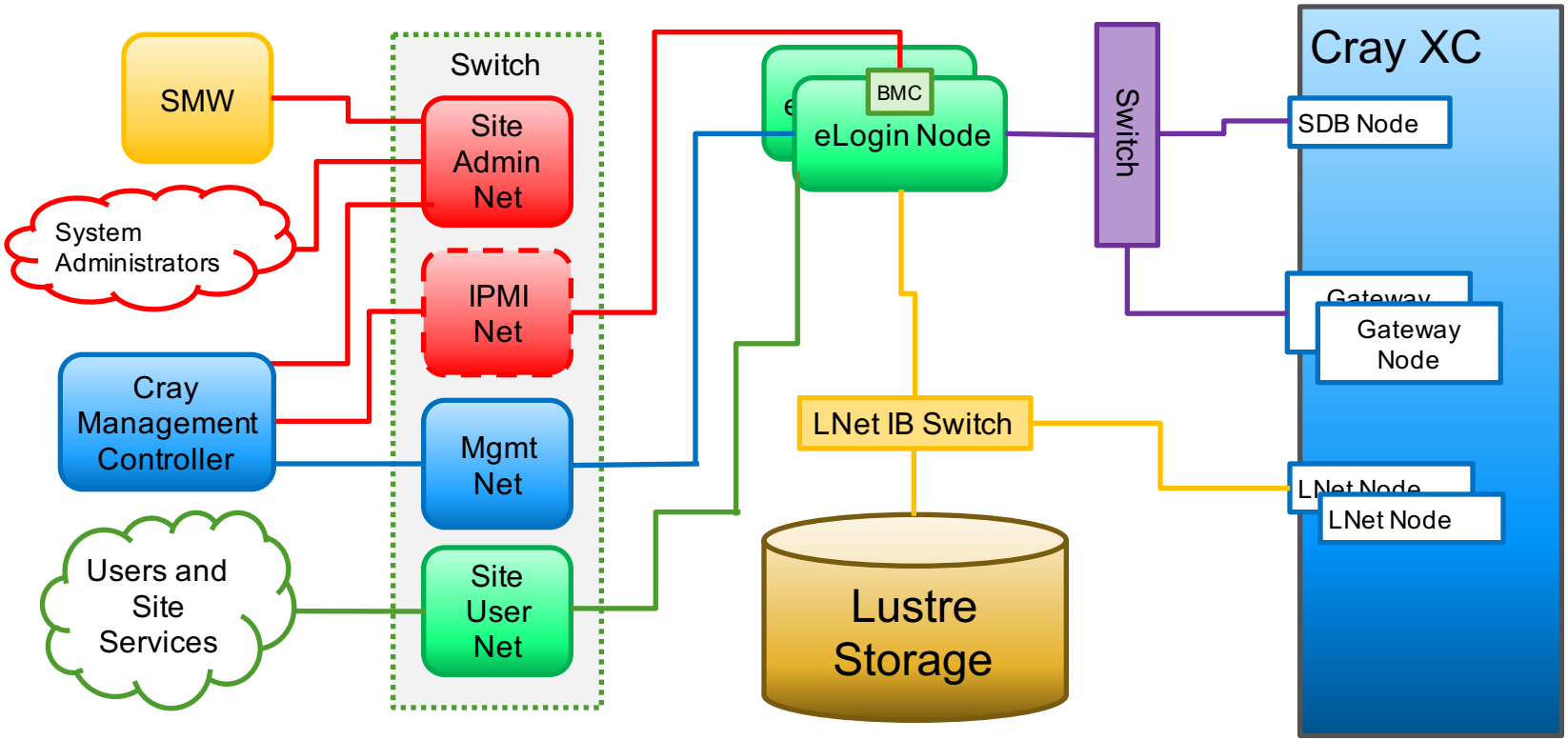
Introduction to eLogin: System Topology

- **4 eLogin to XC Network Configurations**
  - Differing on two networks
    - eLogin to XC SDB Node
      - Routed via XC Gateway Node
      - Direct connect over private network
    - eLogin to XC Gateway Node
      - Direct connect private network
      - On Site User network

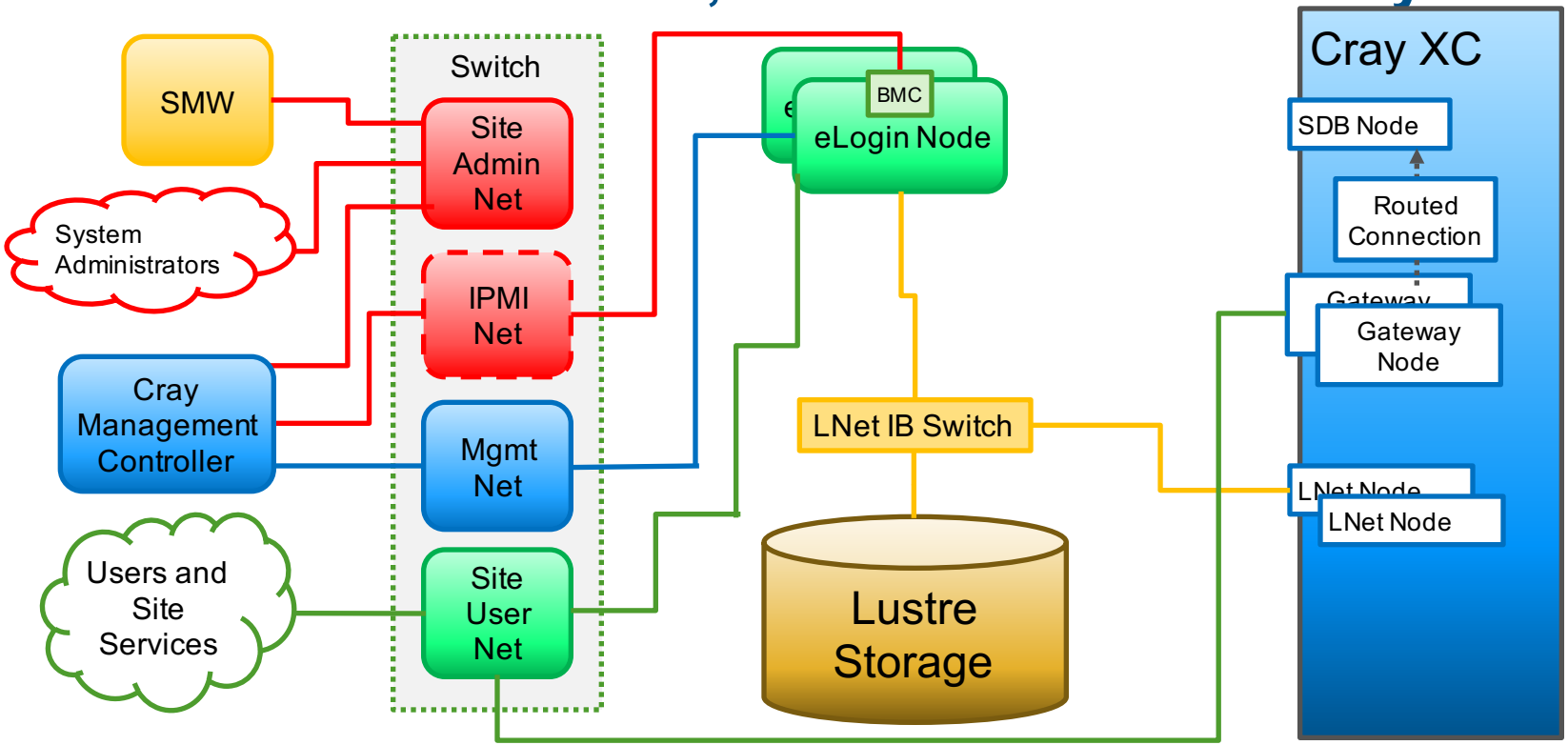
Introduction to eLogin: System Topology Routed SDB Connection



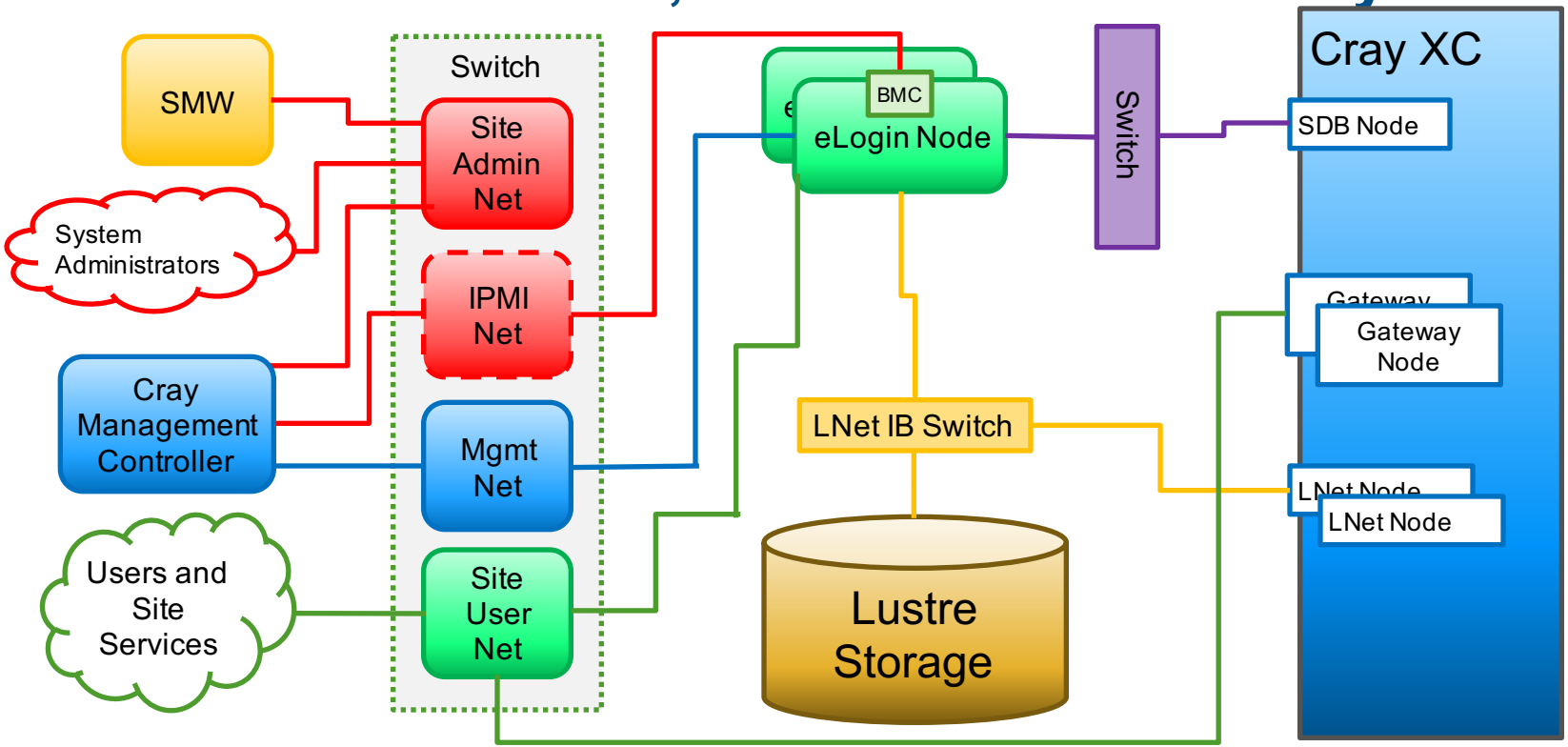
Introduction to eLogin: System Topology Direct SDB Connection



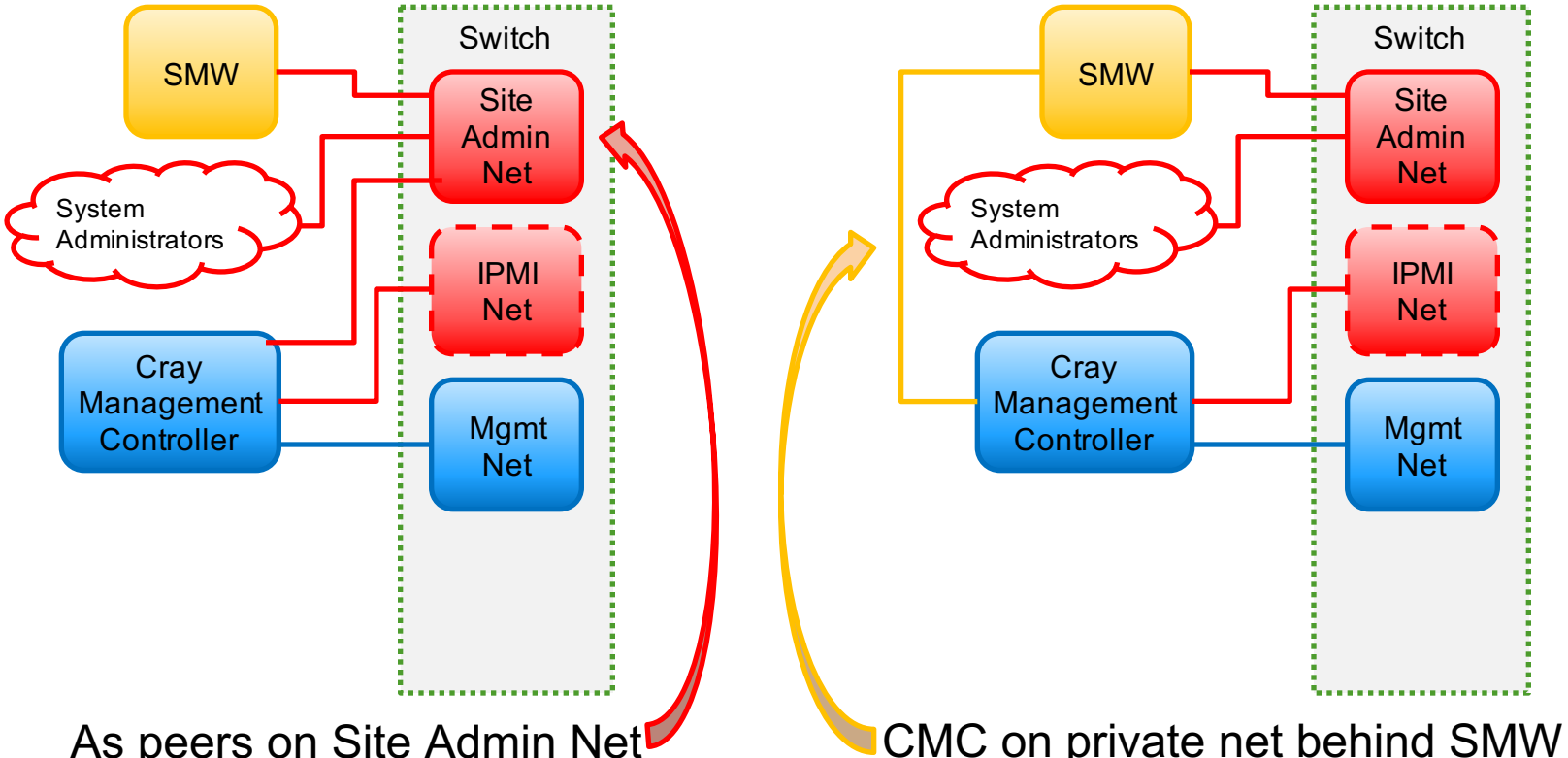
Introduction to eLogin: System Topology Routed SDB Connection, Site Connected Gateway



Introduction to eLogin: System Topology Direct SDB Connection, Site Connected Gateway



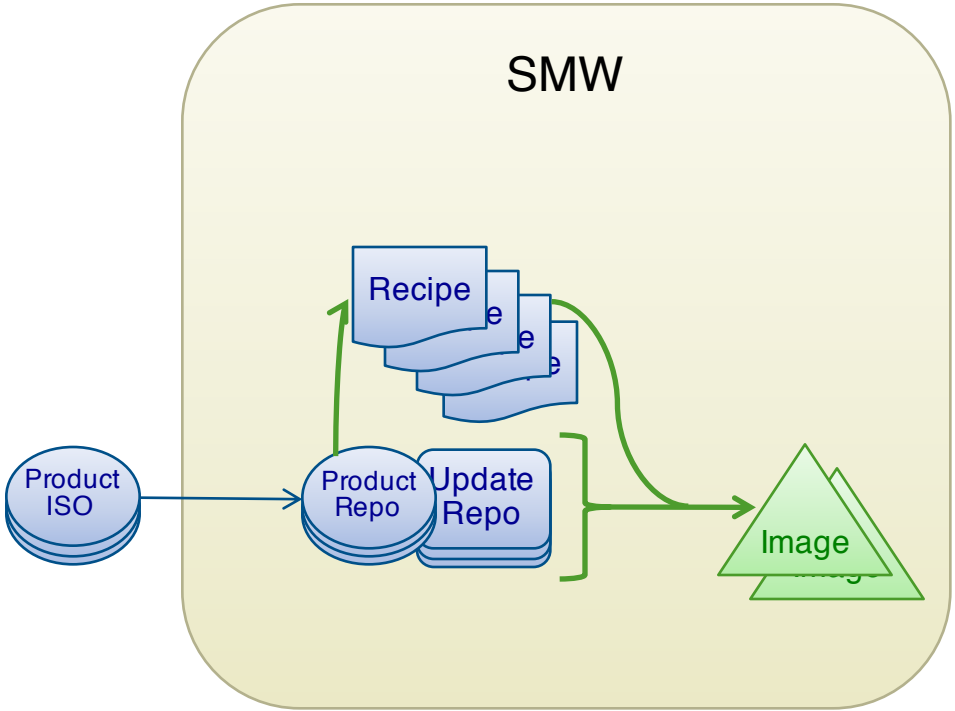
Introduction to eLogin: Topology
SMW to Cray Management Controller (CMC) options



Introduction to eLogin: Software Images

- **Based on the Cray Linux Environment (CLE)**
  - SLES 12
  - Lustre Client
  - Cray eswrap
    - Wraps a set of Cray XC and WLM commands for execution on the Cray XC
- **Separate from the image, but part of the eLogin environment**
  - Cray Programming Environment (Cray PE)

Introduction to eLogin: Software Images



- **Images are built on the SMW**
  - Package repositories are located under: `/var/opt/cray/repos`
  - All rpm dependencies are resolved from these repositories
- **Prescribed by Image Recipes**
  - CLE and eLogin images share recipes
  - eLogin recipes
    - Based on CLE recipes plus eLogin packages
  - Recipes may be cloned and modified
  - Images are rooted under: `/var/opt/cray/imps/image_roots`
  - Images must be exported to the Cray Management Controller for eLogin deployment

Introduction to eLogin: Image Recipes

- **Each image recipe is in a JSON file**
  - Contains name and description
  - Includes package collections, packages (rpms), and repositories
- **JSON file may contain more than one image recipe**
  - Versioned JSON file(s) for each Cray software release
- **Everything has a rationale**
  - Description explaining why each package collection, package, or repository is required
- **Custom image recipes can be created to serve specific purposes**
- **SMW location:**
  - `/etc/opt/cray/imps/image_recipes.d/`

Introduction to eLogin: Image Recipe Example

```
{
  "elogin-large_cle_6.0up01_sles_12_x86-64_ari": {
    "description": "A more complete image recipe for elogin nodes",
    "package_collections": {
      "login-large-external_cle_6.0up01_sles_12": {
      "rationale": "Default packages for external login nodes."
      }
    },
    "packages": {},
    "postbuild_chroot": [
      "dracut --kver $(rpm -ql kernel-default | grep /boot/vmlinuz- | sed
's_/boot/vmlinuz-__')"
    ],
    "recipes": [
      "seed_common_6.0up01_sles_12_x86-64"
    ],
"repositories": {
       "common_cle_6.0up01_sles_12_x86-64_ari": {
         "rationale": "Cray-provided CLE-related packages"
       },
```

Introduction to eLogin: Package Collections

- Represent logical groupings of packages (rpms)
- Contain versioned and unversioned package names
- CLE Installed package collections are read only
- Package collections can include packages
- Package collections can include other package collections
- SMW location:
  - `/etc/opt/cray/imps/package_collections.d/`

Introduction to eLogin: Package Collections (1 of 2)

```
"storage-generic_cle_6.0up01_sles_12": {
    "description": "Generic packages that allow the management of any
type of storage device.",
    "package_collections": {
        "storage-base_cle_6.0up01_sles_12": {
            "rationale": "Packages that are required on KNL compute
nodes with SSD”
        }
    },
```

Introduction to eLogin: Package Collections (2 of 2)

```
"packages": {
        "btrfsprogs": {
            "rationale": "Manage BTRFS filesystems."
        },
        "multipath-tools": {
            "rationale": "Useful for mapping persistent names to sd names in a multipath
environment."
        },
        "parted": {
            "rationale": "Utility to partition disks."
        },
        "targetcli": {
            "rationale": "Target software for iscsi"
        },
        "xfsprogs": {
            "rationale": "Manage XFS filesystems."
        }
    }
},
```

Introduction to eLogin: Recipe Tool - Manages Recipes

```
smw: # recipe --help
usage: recipe [-h] [-V] [-v] [-q]
              {create,list,remove,show,update,validate} ...

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         perint version header
  -v, --verbose         increase the verbosity of the command
  -q, --quiet           suppresses unnecessary output

subcommands:
  {create,list,remove,show,update,validate}

details:
    Detailed usage for subcommands can be displayed by providing -h or
    --help anywhere after the subcommand. For example, 'recipe list -h'
    will display detailed usage information for recipe list.
```

Introduction to eLogin: Listing Recipes

```
smw: # recipe list
compute-large_cle_6.0up01_sles_12_x86-64_ari
compute_cle_6.0up01_sles_12_x86-64_ari
elogin-large_cle_6.0up01_sles_12_x86-64_ari
elogin_cle_6.0up01_sles_12_x86-64_ari
login-large_cle_6.0up01_sles_12_x86-64_ari
login-lustre-master_cle_6.0up01_sles_12_x86-64_ari
login_cle_6.0up01_sles_12_x86-64_ari
pe_image_cle_6.0up01_sles_12
seed_common_6.0up01_sles_12_x86-64
```

Introduction to eLogin: Viewing Recipes

```
smw:~ # recipe show
elogin_cle_6.0up01_sles_12_x86-64_ari
elogin_cle_6.0up01_sles_12_x86-64_ari:
  name: elogin_cle_6.0up01_sles_12_x86-64_ari
  repositories:
    sle-sdk_12_x86-64
    sle-module_legacy_12_x86-64_updates
    sle-we_12_x86-64_updates
    lustre-2.5_cle_6.0up01_sles_12_x86-64_ari
    sles_12_x86-64_updates
    sles_12_x86-64
    lustre-2.5_cle_6.0up01_sles_12_x86-64_ari_updates
    sle-sdk_12_x86-64_updates
    sle-module_legacy_12_x86-64
    common_cle_6.0up01_sles_12_x86-64_ari
```

```
common_cle_6.0up01_sles_12_x86-64_ari_updates
    sle-we_12_x86-64
    passthrough-common_cle_6.0up01_sles_12_x86-64_updates
    passthrough-common_cle_6.0up01_sles_12_x86-64
  recipes:
    seed_common_6.0up01_sles_12_x86-64
  package_collections:
    login-small-external_cle_6.0up01_sles_12
  path: /etc/opt/cray/imps/image_recipes.d/elogin_cle_6.0up01_sles12_ari.json
  description: A base image recipe for elogin nodes
```

Introduction to eLogin: Recipe Creation

```
smw:~ # recipe create --help
usage: recipe create [-h] [-V] [-v] [-q] [--clone RECIPE] [-f] RECIPE

positional arguments:
  RECIPE          name of recipe to create

optional arguments:
  -h, --help      show this help message and exit
  -V, --version   print version header
  -v, --verbose   increase the verbosity of the command
  -q, --quiet     suppresses unnecessary output
  --clone RECIPE  copy recipe content from RECIPE
  -f, --force     force the overwrite of an existing named recipe

examples:
  # recipe create login_cle_cle6.0up01_sles_12_x86-64_ari
  # recipe create --clone login_cle_cle6.0up01_sles_12_x86-64_ari \
        login_cle_6.0up01_sles_12_x86-64_ari_Test
```

Introduction to eLogin: Cloned Recipe Info

```
smw:~ # recipe show elogin-
large_cle_6.0up01_sles_12_x86-64_ari-test
elogin-large_cle_6.0up01_sles_12_x86-64_ari-test:
  name: elogin-large_cle_6.0up01_sles_12_x86-64_ari-
test
  created: 2016-03-17T21:37:28
  repositories:
    sle-sdk_12_x86-64
    sle-module_legacy_12_x86-64_updates
    sle-we_12_x86-64_updates
    lustre-2.5_cle_6.0up01_sles_12_x86-64_ari
    sles_12_x86-64_updates
    sles_12_x86-64
    lustre-2.5_cle_6.0up01_sles_12_x86-64_ari_updates
    sle-sdk_12_x86-64_updates
    sle-module_legacy_12_x86-64
    common_cle_6.0up01_sles_12_x86-64_ari
    common_cle_6.0up01_sles_12_x86-64_ari_updates
    sle-we_12_x86-64
    passthrough-common_cle_6.0up01_sles_12_x86-
64_updates
    passthrough-common_cle_6.0up01_sles_12_x86-64
  recipes:
    seed_common_6.0up01_sles_12_x86-64
  package_collections:
    login-large-external_cle_6.0up01_sles_12
  path:
/etc/opt/cray/imps/image_recipes.d/image_recipes.local
.json
  history:
    2016-03-17T21:37:28: Locally cloned from Recipe
'elogin-large_cle_6.0up01_sles_12_x86-64_ari'.
  description: A more complete image recipe for elogin
nodes
```

Introduction to eLogin: Image Tool - Manages and Creates Images

```
smw:~ # image --help
usage: image [-h] [-V] [-v] [-q]
             {create,diff,list,export,push,remove,show,validate} ...

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         print version header
  -v, --verbose         increase the verbosity of the command
  -q, --quiet           suppresses unnecessary output

subcommands:
  {create,diff,list,export,push,remove,show,validate}

details:
    Detailed usage for subcommands can be displayed by providing -h or
    --help anywhere after the subcommand. For example, 'image list -h'
    will display detailed usage information for image list.
```

Introduction to eLogin: List Images

```
smw:~ # image list
elogin_cle_6.0.UP00-build6.0.19_sles_12-created20151113
elogin_cle_6.0.UP01-build6.0.44_sles_12-created20160106
elogin_cle_6.0.UP01-build6.0.45_sles_12-created20160113
elogin_cle_6.0.UP01-build6.0.47_sles_12-created20160120
elogin_cle_6.0.UP01-build6.0.50_sles_12-created20160127
elogin_cle_6.0.UP01-build6.0.52_sles_12-created20160210
elogin_cle_6.0.UP01-build6.0.52_sles_12-created20160210-test
elogin_cle_6.0.UP01-build6.0.62_sles_12-created20160303
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316-test
```

Introduction to eLogin: List Images

```
smw:~ # image list
elogin_cle_6.0.UP00-build6.0.19_sles_12-created20151113
elogin_cle_6.0.UP01-build6.0.44_sles_12-created20160106
elogin_cle_6.0.UP01-build6.0.45_sles_12-created20160113
elogin_cle_6.0.UP01-build6.0.47_sles_12-created20160120
elogin_cle_6.0.UP01-build6.0.50_sles_12-created20160127
elogin_cle_6.0.UP01-build6.0.52_sles_12-created20160210
elogin_cle_6.0.UP01-build6.0.52_sles_12-created20160210-test
elogin_cle_6.0.UP01-build6.0.62_sles_12-created20160303
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316-test
```

Cloned?

Introduction to eLogin: View Image Info

```
smw:~ # image show elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316:
  name: elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316
  created: 2016-03-16T09:13:57
  history:
    2016-03-16T09:14:07: Successful build of Recipe 'seed_common_6.0up01_sles_12_x86-64'
into Image 'elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316'.
    2016-03-16T09:18:11: Successful build of Recipe 'elogin_cle_6.0up01_sles_12_x86-64_ari'
into Image 'elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316'.
    2016-03-16T09:18:41: Successful rebuild of RPM database.
    2016-03-16T12:56:24: Locally cloned to IMPS Image 'elogin_cle_6.0.UP01-
build6.0.69_sles_12-created20160316-test'.
    2016-03-17T14:31:14: Image 'elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316'
registered to Glance Service 'example-cmc' ('http://111.222.333.444:9292') as bare:qcow2
named 'elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316'.
  path: /var/opt/cray/imps/image_roots/elogin_cle_6.0.UP01-build6.0.69_sles_12-
created20160316
```

Cloning record

Export to CMC record

Introduction to eLogin: Creating an Image

```
smw:~ # image create --help
usage: image create [-h] [-V] [-v] [-q] (--clone IMAGE | -r RECIPE) [-f] IMAGE

positional arguments:
  IMAGE                 name of image to create

optional arguments:
  -h, --help            show this help message and exit
  -V, --version         print version header
  -v, --verbose         increase the verbosity of the command
  -q, --quiet           suppresses unnecessary output
  -f, --force           force the overwrite of an existing named image

required named arguments:
  --clone IMAGE         copy settings and content from IMAGE
  -r RECIPE, --recipe RECIPE
                        create the image using RECIPE

examples:
  # image create my_login -r login_cle_rhine_sles_12_x86-64_ari
  # image create my_login2 --clone my_login
```

Introduction to eLogin: View Cloned Image

```
smw:~ # image show elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316-
test
elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316-test:
  name: elogin_cle_6.0.UP01-build6.0.69_sles_12-created20160316-test
  created: 2016-03-16T09:13:57
  history:
    2016-03-16T09:14:07: Successful build of Recipe
'seed_common_6.0up01_sles_12_x86-64' into Image 'elogin_cle_6.0.UP01-
build6.0.69_sles_12-created20160316'.
    2016-03-16T09:18:11: Successful build of Recipe
'elogin_cle_6.0up01_sles_12_x86-64_ari' into Image 'elogin_cle_6.0.UP01-
build6.0.69_sles_12-created20160316'.
    2016-03-16T09:18:41: Successful rebuild of RPM database.
    2016-03-16T12:56:24: Locally cloned from Image 'elogin_cle_6.0.UP01-
build6.0.69_sles_12-created20160316'.
  path: /var/opt/cray/imps/image_roots/elogin_cle_6.0.UP01-
build6.0.69_sles_12-created20160316-test
```

Introduction to eLogin: Image Diff

```
smw:~ # image diff elogin_cle_6.0up01_sles_12_x86-64_ari-20160226
elogin_cle_6.0up01_sles_12_x86-64_ari-20160226-test
INFO - Calculating RPM differences between Image 'elogin_cle_6.0up01_sles_12_x86-64_ari-
20160226' and 'elogin_cle_6.0up01_sles_12_x86-64_ari-20160226-test'...
WARNING - Cannot query installed RPMs, falling back to disk based diff.
INFO - Calculating path entry differences..
INFO - Unique entries to IMPS Image 'elogin_cle_6.0up01_sles_12_x86-64_ari-20160226':
          - etc/opt/cray/pre-pivot.d/01Ping.sh
Unique entries to IMPS Image 'elogin_cle_6.0up01_sles_12_x86-64_ari-20160226-test':
          - etc/opt/cray/pre-pivot.d/11Ping.sh
          - etc/opt/cray/pre-pivot.d/32ConfigNetworkUdevRules.sh
          - etc/opt/cray/pre-pivot.d/33HostbasedAuthClientSetup.sh
          - etc/opt/cray/pre-pivot.d/hostbased-auth-client-setup.yaml
          - tmp/cray-dracut-pre-pivot-elogin-1.0.4-0.noarch.rpm
          - tmp/cray-dracut-pre-pivot-elogin-1.0.5-0.noarch.rpm
          - tmp/cray-dracut-pre-pivot-elogin-1.0.6-0.noarch.rpm
INFO - None
```

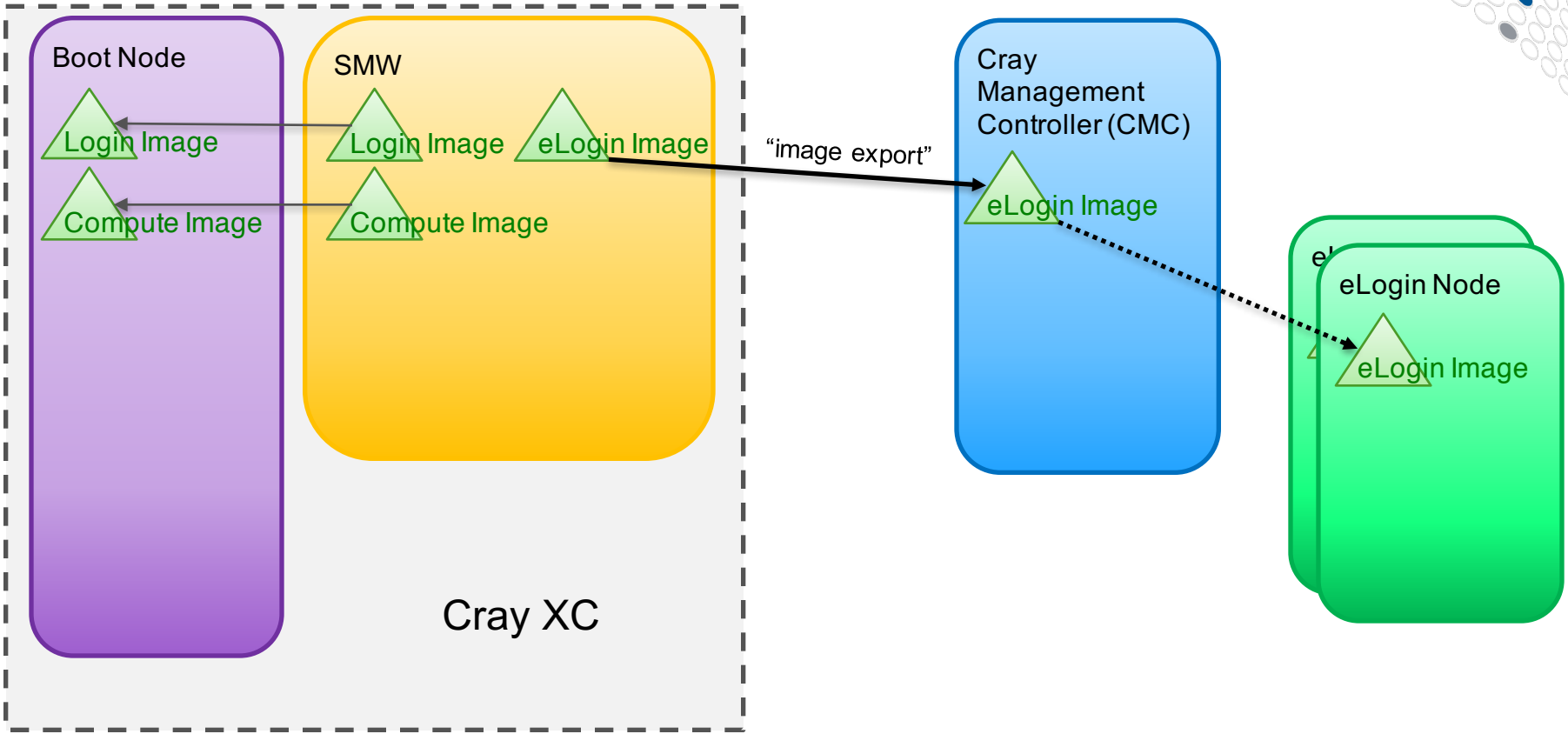
Introduction to eLogin: Export Image to the Cray Management Controller

- **Images must be exported from the SMW to the CMC for deployment to eLogin nodes**

```
smw:~ # image export elogin_image --format qcow2 \
        -d glance:example-cmc:elogin_image_20160319
```

  - The image named "elogin_image" will be exported to qcow2 format and registered in the CSMS image service (glance) on the Cray Management Controller named "example-cmc" as "elogin_image_20160319"
- This image is essentially unconfigured

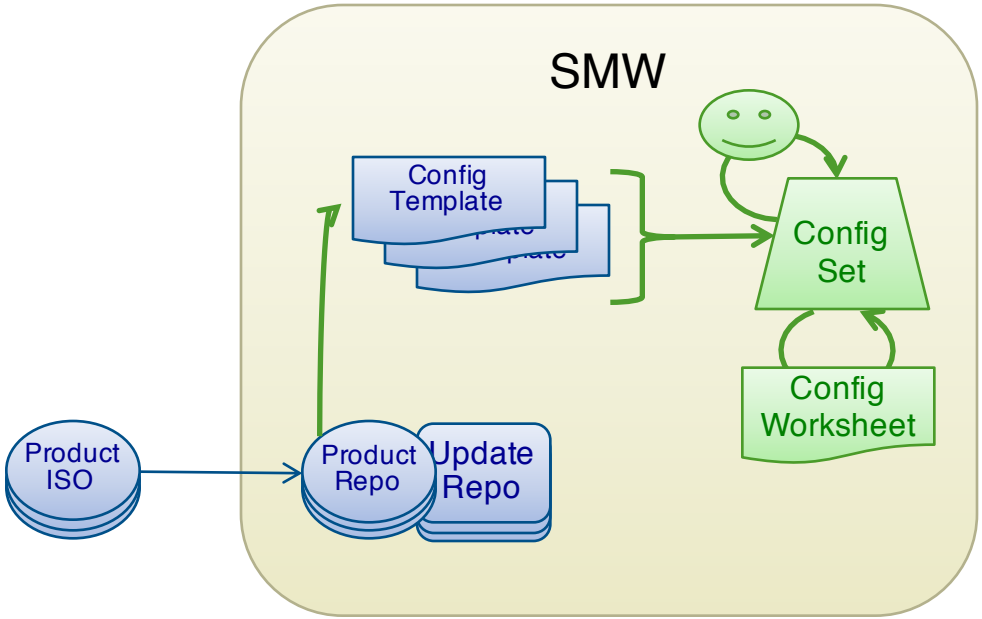
Introduction to eLogin: Image Export



Introduction to eLogin: Image Configuration

- **Configuration data is separate from the image**
  - All configuration information needed to operate the eLogin system is stored in a "configuration set" or "config set"
  - eLogin shares same configuration set as the Cray XC
    - eLogin specific configuration data is added to the Cray XC config set
- **Configuration sets are created and maintained on the SMW then pushed to the Cray Management Controller**
- **More than one config set can exist to support alternative configurations**
  - Only one config set is active on a given eLogin node at a time

Introduction to eLogin: Config Sets



- **Config Set container**
  - Centralized configuration
  - Contains configuration files for different areas
  - YAML format allows direct editing
  - Other (non-YAML) content can be added as necessary
- **Configurator tool (cfgset)**
  - Validates values, prompts for missing content
  - Allows quick updating of specific values
  - Creates and consumes configuration worksheets
- **Configuration Worksheets**
  - For fresh installs, big changes
  - Alternative display & input method
  - Editable
    - Read back in by Configurator

Introduction to eLogin: Configurator Tool

- **The Configurator Tool (cfgset)**
  - Completely data driven by templates
  - Merge existing configuration data with new templates
- **Configuration templates**
  - Provide useful documentation for the value
  - Provide useful defaults
  - Provide value and syntax checking to be used by configurator
- **Run configurator to collect new data**
  - Will automatically prompt/merge new data elements
  - System administrator's "answers" to questions become new default

Introduction to eLogin: Configurator Tool (cfgset)

- **cfgset create**
  - Create new empty config sets or clone existing config sets or prepare configuration worksheets
- **cfgset diff**
  - Show changes between files in config sets
- **cfgset push**
  - Copies contents of a config set to remote host
- **cfgset remove**
  - Remove config set
- **cfgset search**
  - Searches configuration data entries in config sets
- **cfgset update**
  - Modify the attributes or contents of a config set or rename the config set
- **cfgset validate**
  - Checks a config set for syntax, structure, and configuration status of required-level items

Introduction to eLogin: Configurator Tool (cfgset)

- **Iterate on the configurator as necessary**
  - Admin can configure a specific service

```
smw # cfgset update -s cray_eswrap -S unset -l basic p0
smw # cfgset update -s cray_elogin_networking -S all -l advanced p0
```

- **3 different modes**
  - All modes
    - Merge in new templates
    - Run pre/post-configuration scripts (unless suppressed with --no-scripts)
    - Create configuration worksheets based on current settings
  - **auto** – Asks questions based on filters (level and state) from command line
  - **interactive** – View or change any setting in any service
  - **prepare** – Creates configuration worksheets for each service

Introduction to eLogin: Config Set Templates

- **Design of template schema drives how information is gathered**
  - YAML format
  - Cray-provided templates are named: ”cray_<service_name>_config.yaml”

- **Template sections**
  - Service
    - Describes the service
    - Initial question about whether the service should be configured further
  - Settings
    - Contains questions to be answered to configure service

Introduction to eLogin: Configurator Template Schema

```
---
cray_service_name:
    ...
    [service metadata]
    ...
    settings:
        ...
        [service settings]
        ...
...
```

Introduction to eLogin: Configurator Template Fields for Service

- **Fields in template for service**
  - **Title** - explanation of the service
  - **Guidance** – description to aid in enable/disabling service
  - **Enabled** – boolean decision to configure service (or not)
  - **Configured** – whether this has been configured already
  - **Level** – required, basic, or advanced
  - **Config_after** – this should be configured after these other services
  - **Template_type** – CLE or global

Introduction to eLogin: Configurator Template Example for Service

```
cray_elogin_networking:
    enabled: true
    configurator:
        configured: true
        config_after: []
        guidance: Cray eLogin Networking defines the number and key network
            attribute for the various eLogin nodes connected to the system.
            If you have one or more eLogin nodes attached to this system, it
            is necessary to define these attributes in order to produce a
            functional system. If you do not have any external login nodes,
            disable this service.
        level: basic
        template_type: cle
        title: Configure Cray eLogin Networking
        type: boolean
```

Introduction to eLogin: Configurator Template Fields for Settings

- **Fields in template for settings**
  - **Title** - explanation of the class
  - **Guidance** – description to aid in setting the value(s)
  - **Members** – values of the class
  - **Regex** – regular expression to validate input
  - **Configured** - whether this has been configured already
  - **Level** – required, basic, or advanced
  - **Argspec** – one or more values to be configured
  - **Data** - one or more values which have been configured

Introduction to eLogin: Configurator Template Example for Settings

```
settings:
    elogin_networking:
        data:
        -   key: test-elogin1
            postfix_relay_host: cmc-mgmt
        -   key: test-elogin2
            postfix_relay_host: cmc-mgmt
        configurator:
            argspec:
                hostname:
                    allow_none: false
                    configured: true
                    unconfigured_keys: []
                    default_value: ''
                    guidance: The hostname of the
                        eLogin node to configure.
                    level: required
                    multival_key: true
                    purge: false
                    regex: ^(?=.{1,255}$)[0-9A-Za-
z](?:(?:[0-9A-Za-z]|-){0,61}[0-9A-Za-z])?(?:\.[0-9A-
Za-z](?:(?:[0-9A-Za-z]|-){0,61}[0-9A-Za-z])?)*\.?$
                    title: hostname of this eLogin
                        node
                    type: string
                postfix_relay_host:
                    allow_none: false
                    configured: true
                    unconfigured_keys: []
                    default_value: cims-mgmt
                    guidance: The hostname of the
                        postfix relay to use for this
                        eLogin node.
                    level: basic
                    multival_key: false
                    purge: false
                    regex: ^(?=.{1,255}$)[0-9A-Za-
z](?:(?:[0-9A-Za-z]|-){0,61}[0-9A-Za-z])?(?:\.[0-9A-
Za-z](?:(?:[0-9A-Za-z]|-){0,61}[0-9A-Za-z])?)*\.?$
                    title: Postfix Relay Host
                    type: string
                    scope_type: multival
                    guidance: eLogin host
                        configuration values for each
                        eLogin host attached to the
                        system.
                    purge: false
```

Introduction to eLogin: cfgset

- **Updating a config set – actions by the configurator**
  - Clone the config set as a backup
  - Run pre-configuration scripts
  - Validate templates and configuration data
    - YAML syntax validation check
    - Schema validation check
    - Merge templates
  - Prompt for all information to be configured
  - Write changelog entry
    `/var/opt/cray/imps/config/sets/p0/changelog/changelog_2015-09-29T12:00:37.yaml`
  - Run post-configuration scripts
  - Remove backup config set

Introduction to eLogin: Config Sets

- **Config sets have two types**
  - **global** config set covers both the management domain ("SMW" and/or "CMC") as well as truly global data
  - **CLE/eLogin** config set (for p0, or on a partitioned system for p1, p2, p3, etc.)

- **Configuration sets must be pushed to the Cray Management Controller for use by eLogin nodes**
  - Each eLogin node gets its config set data from the Cray Management Controller during node deployment

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

  - On the CMC, the config set is filtered for eLogin usage and added the config_set storage area (OpenStack Swift container) for deployment

```
cmc # add_configset <config_set_name> /etc/opt/cray/elogin/exclude/<exclude_list>
```

Introduction to eLogin: Config Sets on the SMW and Cray Management Controller

- Config sets reside on the SMW (and Cray Management Controller after being pushed)

```
smw|cmc:/var/opt/cray/imps/config/sets/<config_set_name>
```

- The global config set is also available on the SMW as a link to the /var/opt/cray/imps/config/sets/global

```
smw:/etc/opt/cray/config/global
```

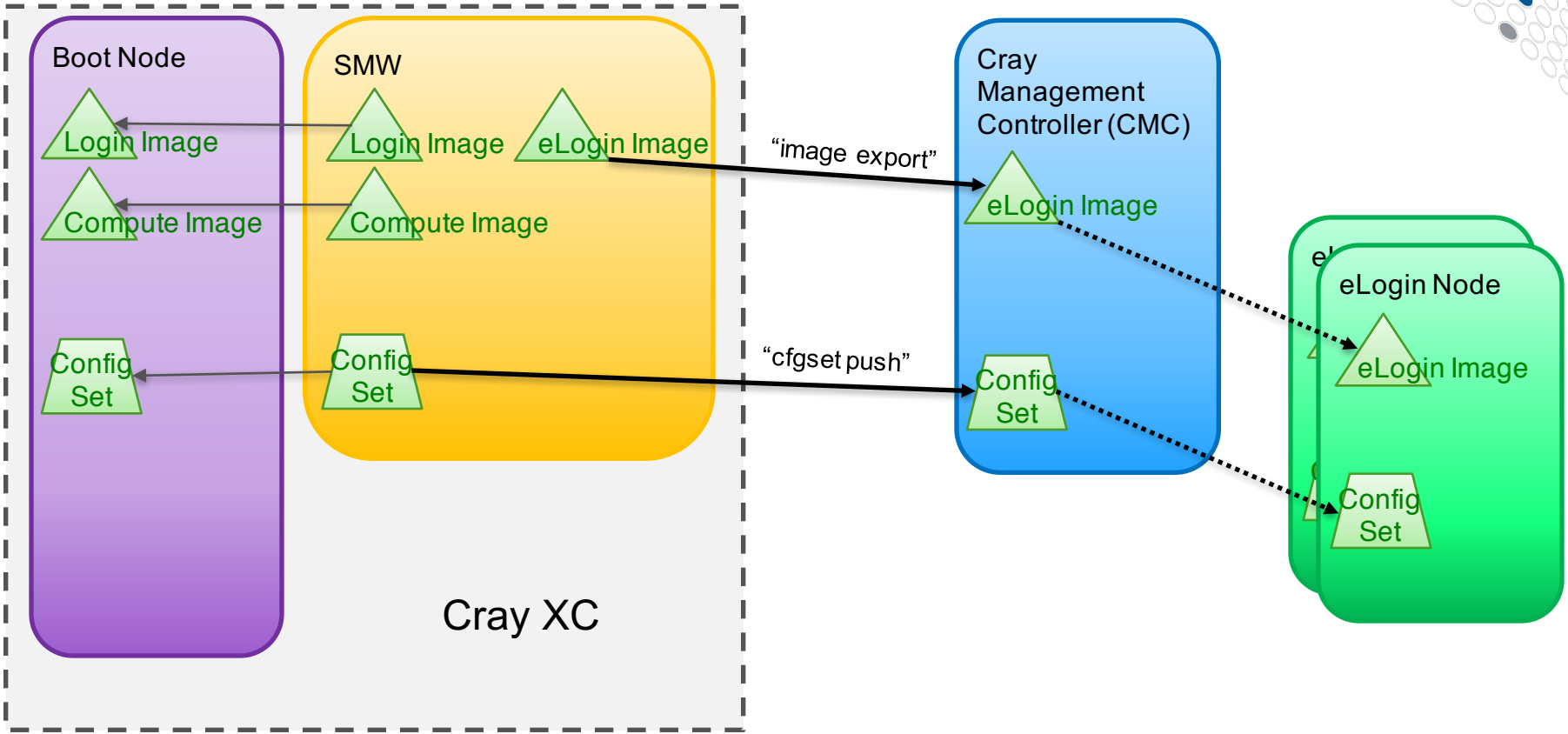
Introduction to eLogin: Config Sets on the node

- From the eLogin node's perspective, a config set is just a directory of config files for the current and global config sets

```
elogin:/etc/opt/cray/config/current
elogin:/etc/opt/cray/config/global
```

- /etc/opt/cray/config/current contains the following subdirectories
  - ansible, config, dist, files
    - config subdirectory contains the configuration files

Introduction to eLogin: Config Set



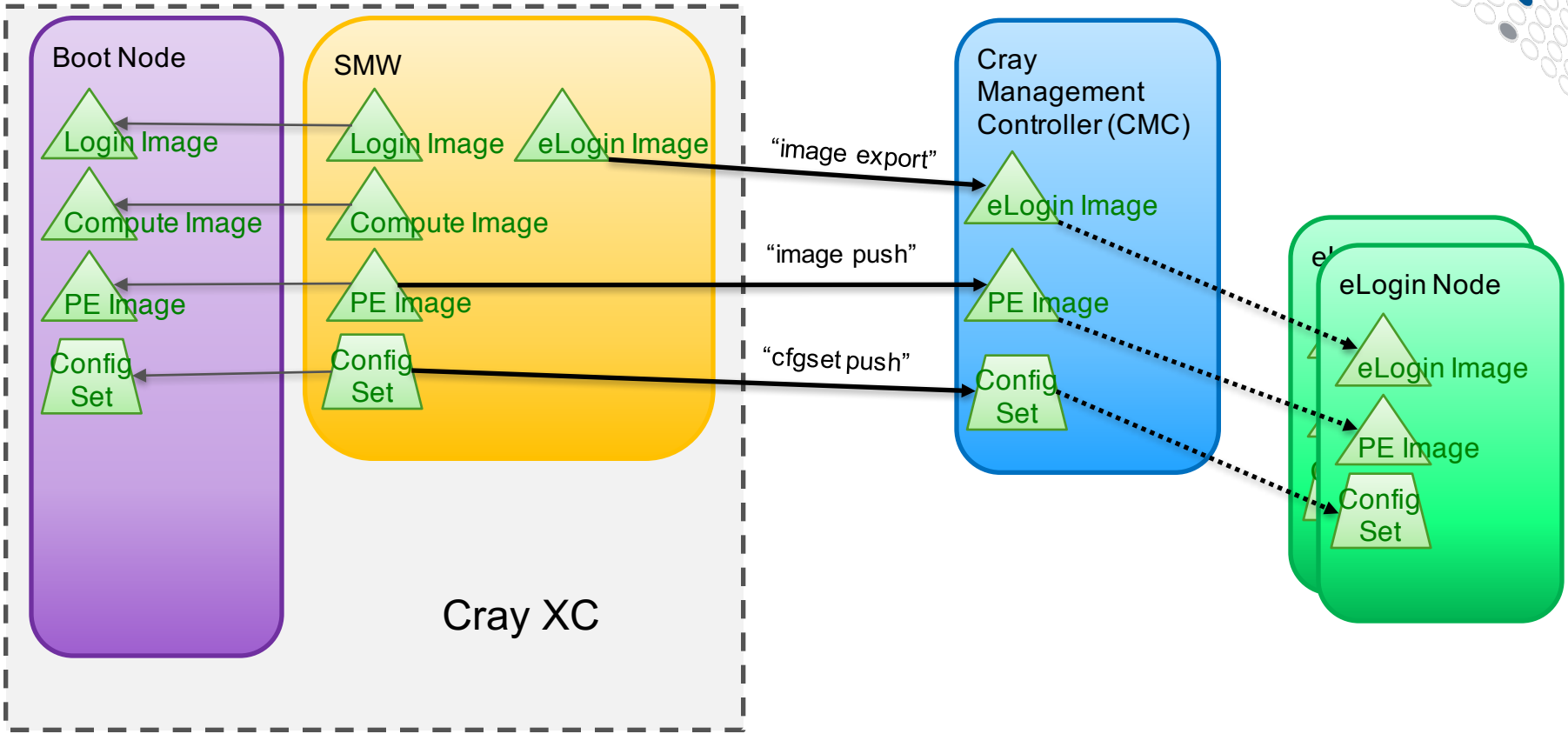
Introduction to eLogin: Cray PE

- **Cray Programming Environment (Cray PE)**
  - Pushed to the CMC from the SMW
  - Same exact Cray PE image that the XC is using!

```
smw # image push -d <cmc-name> <pe_compute_image>
```

- **eLogin syncs Cray PE from the CMC server**
  - Differs from esLogin where Cray PE was installed directly to the esLogin image which caused these images to grow very large

Introduction to eLogin: Image Export Flow



Agenda

- What is eLogin?
- Introduction to eLogin
- **Introduction to Cray System Management Software (CSMS)**
- **Managing eLogin nodes**
- **Summary**
- **Q&A**

Introduction to Cray System Management Software (CSMS)

- **Overview**
  - Base Operating System is CentOS 7
    - Installed via Cray Bootable CentOS ISO image
  - Cray System Management Software
    - Installed via Cray System Management Software ISO image
  - eLogin support software
    - Installed via eLogin Installation ISO image
  - Leverages OpenStack services
    - Keystone – Authentication between services
    - Nova – Node lifecycle management
    - Ironic with Fuel - Bare metal provisioning
    - Glance - Image service
    - Swift – Object storage (backs Glance)
    - Neutron – Networking service
    - Heat – Orchestration
    - Cinder – block storage (not used by eLogin)
  - Provides remote console and console logging
  - eLogin syslogs are forwarded to the Cray Management Controller

Managing eLogin Nodes: CSMS operation

- **Image Registration –** *glance and swift*
- **Node Configuration –** *Ansible and config sets and swift*
- **Node Inventory –** *ironic, nova, glance and swift*
- **Node Deployment –** *heat, nova, ironic, glance and swift*
- **Node Updates –** *heat, nova, ironic, glance and swift*
- **Troubleshooting**

Managing eLogin Nodes: Image Registration

- **eLogin images are registered on the CMC (*Glance*)**
  - Images consist of three entities
    - Image in qcow2 format
    - Kernel image in AKI (Amazon Kernel Image) format
    - Initramfs image in ARI (Amazon Ramdisk Image) format
  - "image export" command takes care of image registration from the SMW
- **Local Disk configuration is separate from the image**
  - eLogin disk partitioning information is registered in the Glance service as a raw image type
    - deploy_config_elogin - JSON file stored in raw image format
    - Installed as part of the eLogin installation process

Managing eLogin Nodes: Node Configuration

- **Configuration performed locally on the node**
  - Uses config set data and Ansible plays
  - **cray-ansible** finds all Ansible plays and executes them
  - Ansible plays are packaged with their application software
    - eLogin plays get packaged with the eLogin software
  - Sites may add their own Ansible plays

- **cray-ansible runs in the pre-pivot init phase of the eLogin image boot process**

Managing eLogin Nodes: Config Set

- **eLogin Ansible plays reference data from several Cray CLE config set services:**
  - cray_local_users - local users configuration
  - cray_time - settings for various aspects of time including ntp, timezone
  - cray_user_settings - user environment settings
  - cray_auth - user authentication settings - LDAP, NIS, etc.
  - cray_ssh - SSH settings
  - cray_lustre_client - Lustre Client settings
  - cray_net - management, site, and LNet network settings
  - cray_simple_sync - a generic method of distributing files to targeted locations on the eLogin nodes

Managing eLogin Nodes: Config Set

- **eLogin Ansible plays reference data from the following eLogin specific config sets:**
  - cray_elogin_lnet – Lustre Network (LNet) settings for eLogin
  - cray_elogin_networking – eLogin postfix networking
  - cray_eswrap – settings for executing certain Cray XC and WLM commands

Managing eLogin Nodes: Extending Config

- **Cray's Configuration Management Framework (CMF) allows additional configuration tasks**
  - Add site-specific tasks in concert with Cray-provided Ansible boot-time execution
- **For example, site Ansible plays may perform the following tasks**
  - Start/stop services
  - Change crontab entries
  - Modify files
  - Copy files

Managing eLogin Nodes: Site Ansible Plays

- **Site Ansible plays**
  - Can be either in the image or in the config set
    - **cray-ansible** will find plays in both locations and include them automatically
    - config set is the optimal location
      - Avoids tinkering with custom images
      - Site changes are kept separate from the image

Managing eLogin Nodes: Site Ansible Plays

- **Integrating site Ansible plays in config set**
  - Place site Ansible plays in the following directory

```
smw:/var/opt/cray/imps/config/sets/<config_set>/ansible/
```

  - On the CMC, the config set needs to be added the config_set storage area after any changes are made

```
cmc # add_configset <config_set_name> /etc/opt/cray/elogin/exclude/<exclude_list>
```

- **Site Ansible plays will run automatically with all Cray provided plays**

Managing eLogin Nodes: Ansible Best Practices

- **Ansible expects that all tasks are idempotent**
  - (action performed only once, even if play is run more than once)
  - Care should be taken to ensure that tasks prescribe the desired state of the running system, making changes only when necessary
  - See http://docs.ansible.com/ansible/glossary.html#resource-model

- **When modifying files on a running system**
  - Keep in mind that other services may access the file
  - Take the appropriate measures to ensure the modifications do not interfere with other operations
  - Leave a breadcrumb that the file is updated by an automated process
    - The “insertbefore” or “insertafter” options in the Ansible “lineinfile” module are well-suited for this

Managing eLogin Nodes: Ansible Best Practices

- **If you find that you are trying to do something that is difficult to achieve in a few simple steps…**
  - It is likely that Ansible already has a module that provides the functionality
  - See http://docs.ansible.com/ansible/modules_by_category.html
- **The Ansible "ini_file" module was specially created for modifying INI-style configuration files**
  - Use this instead of "lineinfile" when applicable

Managing eLogin Nodes: Simple Sync

- **Alternative method of installing files to eLogin nodes other than using a site Ansible play**
  - Simple rsync of files from a root in the config set container to / on the eLogin node
  - Files can be installed on all eLogin nodes or by hostname
  - To install a file to all eLogin nodes, place it under the following directory on the CMC

```
/var/opt/cray/imps/config/sets/<config_set>/files/roles/simple_sync/classes/common/<path_to_file_on_elogin>/<file>
```

  - To install a file to a specific eLogin node, place it under the following directory on the CMC

```
/var/opt/cray/imps/config/sets/<config_set>/files/roles/simple_sync/classes/hostnames/<host_name>/<path_to_file_on_elogin>/<file>
```

Managing eLogin Nodes: Node Inventory

- Nodes are registered with CSMS using an inventory file
  `/etc/opt/cray/openstack/ansible/inventory.csv`
- Template can be found at:
  `/etc/opt/cray/openstack/ansible/roles/ironic_enrollment/files/example_inventory.csv`
- Node inventory example:

```
NODE_NAME, BMC_IP, MAC_ADDR, N_CPUs, ARCH, RAM_MB, DISK_GB, NODE_DESC
  elogin1,10.142.0.5,14:fe:b5:ca:b7:00,32,x86_64,131072,550,elogin1
  elogin2,10.142.0.6,e0:db:55:0a:25:a8,32,x86_64,131072,550,elogin2
  elogin3,10.142.0.7,78:2b:cb:33:4b:b7,32,x86_64,131072,550,elogin3
```

Managing eLogin Nodes: Node Inventory

- **Register nodes with the csms_ironic_enrollment script**

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

  - Registers each node with the ironic service (bare-metal nodes)
  - Assigns the node installer image to the node
  - Creates a nova "flavor" named "ironic_flavor" matching the hardware configuration
    - Nova is the hardware scheduler for tenant instances (deployed nodes)
    - Flavors define the minimum required hardware specifications
      - *"What flavor of hardware would you like to deploy to?"*

Managing eLogin Nodes: Node Inventory

- List registered eLogin nodes

```
example-cmc # ironic node-list
+--------------------------------------+---------+--------------+-------------+--------------------+-------------+
| UUID                                 | Name    | Instance UUID | Power State | Provisioning State | Maintenance |
+--------------------------------------+---------+--------------+-------------+--------------------+-------------+
| 692e40b2-ff8c-4842-b9bf-8c6957256b23 | elogin1 | None          | power off   | available          | False       |
| 3c0a3cd6-eb76-4ab1-85e3-615d867a66a0 | elogin2 | None          | power off   | available          | False       |
| c0386c4d-9410-4113-a71b-2a770b6239df | elogin3 | None          | power off   | available          | False       |
+--------------------------------------+---------+--------------+-------------+--------------------+-------------+
```

- List node “ports” (network interfaces) for elogin3
  - This is the interface that elogin3 will boot over

```
example-cmc # ironic node-port-list elogin3
+--------------------------------------+-------------------+
| UUID                                 | Address           |
+--------------------------------------+-------------------+
| e23aff8f-25af-4b9d-89ff-a68f9ed54bf8 | 78:2b:cb:33:4b:b7 |
+--------------------------------------+-------------------+
```

Managing eLogin Nodes: Node Info
Example: Ironic Ports (network interfaces)

```
example-cmc # ironic port-show e23aff8f-25af-4b9d-89ff-a68f9ed54bf8
+------------+--------------------------------------+
| Property   | Value                                |
+------------+--------------------------------------+
| node_uuid  | c0386c4d-9410-4113-a71b-2a770b6239df |
| uuid       | e23aff8f-25af-4b9d-89ff-a68f9ed54bf8 |
| extra      | {}                                   |
| created_at | 2016-03-17T18:59:23+00:00            |
| updated_at | 2016-03-30T21:56:08+00:00            |
| address    | 78:2b:cb:33:4b:b7                    |
+------------+--------------------------------------+
```

elogin3

Managing eLogin Nodes: Node Info (1 of 2)

```
example-cmc # ironic node-show elogin3
+-----------------------+--------------------------------------------------------------------+
| Property              | Value                                                              |
+-----------------------+--------------------------------------------------------------------+
| target_power_state    | None                                                               |
| extra                 | {u'description': u'elogin3'}                                       |
| last_error            | None                                                               |
| updated_at            | 2016-03-31T21:18:16+00:00                                          |
| maintenance_reason    | None                                                               |
| provision_state       | available                                                          |
| uuid                  | c0386c4d-9410-4113-a71b-2a770b6239df                               |
| console_enabled       | True                                                               |
| target_provision_state| None                                                               |
| maintenance           | False                                                              |
| inspection_started_at | None                                                               |
| inspection_finished_at| None                                                               |
| power_state           | power off                                                          |
| driver                | fuel_rsync_ipmi                                                    |
```

Managing eLogin Nodes: Node Info (2 of 2)

```
| reservation            | None                                                                |
| properties             | {u'memory_mb': 131072, u'cpu_arch': u'x86_64', u'local_gb': 550,    |
|                        | u'cpus': 32}                                                        |
| instance_uuid          | None                                                                |
| name                   | elogin3                                                             |
| driver_info            | {u'ipmi_password': u'******', u'ipmi_address': u'10.142.0.7',       |
|                        | u'deploy_ramdisk': u'e59491e5-d4da-4956-99c8-be662f6ea8c7',         |
|                        | u'deploy_kernel': u'60c99670-7512-4372-8102-84d94bdb5b50',          |
|                        | u'ipmi_username': u'root'}                                          |
| created_at             | 2016-03-17T18:59:18+00:00                                           |
| driver_internal_info   | {u'clean_steps': None, u'is_whole_disk_image': False}               |
| chassis_uuid           |                                                                     |
| instance_info          | {}                                                                  |
+------------------------+---------------------------------------------------------------------+
```

Managing eLogin Nodes: Nova Flavors

```
example-cmc # nova flavor-list
+--------------------------------------+--------------+-----------+------+-----------+-------+-------+-------------+-----------+
| ID                                   | Name         | Memory_MB | Disk | Ephemeral | Swap  | VCPUs | RXTX_Factor | Is_Public |
+--------------------------------------+--------------+-----------+------+-----------+-------+-------+-------------+-----------+
| 7b26a552-d079-4635-a4a6-1bec1cd7b902 | ironic_flavor | 131072    | 550  | 0         |       | 32    | 1.0         | True      |
| aac33543-4d49-44c0-8e22-22a02537f2ee | eloginflavor  | 65536     | 100  | 0         | 16384 | 16    | 1.0         | True      |
+--------------------------------------+--------------+-----------+------+-----------+-------+-------+-------------+-----------+
```

Managing eLogin Nodes: Node Deployment

- **Heat Orchestration service is used for node deployment**
  - Orchestrates node stack deployments
  - Uses a template plus an environment file to deploy eLogin stacks
    - Template file is common to all eLogin nodes
    - Environment file is node specific
      - Provides values for the heat stack template
    - CMC location:
      - `/etc/opt/cray/openstack/heat/templates`

Managing eLogin Nodes: Node Deployment

- **Heat template files for eLogin**
  - **elogin_template.yaml**
    - Dynamic management IP address
  - **elogin_template_fixed_ip.yaml**
    - Fixed management IP address

Managing eLogin Nodes: Node Deployment

- **Heat environment template files for eLogin**
  - 2 templates depending on whether you want to use a fixed or dynamic management IP address

    - **elogin-env.yaml.template**
      - For use with **elogin_template.yaml**

    - **elogin-env-fixed-ip.yaml.template**
      - For use with **elogin_template_fixed_ip.yaml**

Managing eLogin Nodes: Node Deployment heat stack commands

- **heat stack-create**
  - Deploys the node with the prescribed software stack
- **heat stack-delete**
  - Tears down the software stack and powers off the node
- **heat stack-update**
  - Updates the software stack on the node
- **heat stack-list**
  - Lists all heat stacks

Managing eLogin Nodes: Node Deployment

- Heat environment template file for eLogin

```
# cat elogin-env-fixed-ip.yaml.template
---
# An example env.yaml file to use with the Heat templates when a fixed
# management IP address is desired.
# Copyright 2016 Cray Inc. All Rights Reserved.

parameters:
  image_id: elogin_image.qcow2
  host_name: elogin1
  fixed_ip: 10.142.0.100
  instance_flavor: eloginflavor
  cray_config_set: p0
  cims_host_name: example-cmc
  ironic_id: 7baa31f0-07c8-42e9-8743-ae5f147f78c3
  actions_list: copy_p0
```

Managing eLogin Nodes: Node Deployment

- Create <hostname>-env.yaml from the env template
  - **image_id:** elogin_image.qcow2
    - The image to deploy to the node
  - **host_name:** elogin1
    - The hostname of the node
  - **fixed_ip:** 10.142.0.100
    - The management IP address
  - **instance_flavor:** eloginflavor
    - The nova flavor to use – typically set to 'eloginflavor'
  - **cray_config_set:** p0-eLogin
    - The config set to use in configuring this eLogin node
  - **cims_host_name:** example-cmc
    - The CMC hostname
  - **ironic_id:** 7baa31f0-07c8-42e9-8743-ae5f147f78c3
    - The UUID of the ironic node to deploy
  - **actions_list:** copy_p0
    - The script that installs the config set on the eLogin node

Managing eLogin Nodes: Node Deployment

- Copy deploy_elogin.sh.template to deploy_<elogin_name>.sh and edit with appropriate settings –TEMPLATE_FILE, ENV_FILE, STACK_NAME

```
# cat deploy_elogin1.sh
#!/bin/bash
#
# A template eLogin deploy script.
# Copyright 2015 Cray Inc. All Rights Reserved.
# Edit these values to the correct values for the node to be deployed
# Use full paths only
TEMPLATE_FILE=/etc/opt/cray/openstack/heat/templates/elogin_template_rsync_password_fixed_ip.yaml
ENV_FILE=/etc/opt/cray/openstack/heat/templates/elogin1-env.yaml
STACK_NAME=elogin1

source ~/admin.openrc
heat stack-create -f $TEMPLATE_FILE -e $ENV_FILE $STACK_NAME
```

Managing eLogin Nodes: Node Deployment Checking heat, nova and ironic data

- Deploy elogin1 by running deploy_elogin.sh

```
# ./deploy_elogin1.sh
# heat stack-list
+--------------------------------------+------------+-----------------+----------------------+
| id                                   | stack_name | stack_status    | creation_time        |
+--------------------------------------+------------+-----------------+----------------------+
| 956e7974-7557-4091-b749-2833827718f3 | elogin1    | CREATE_COMPLETE | 2016-03-01T22:45:33Z |
+--------------------------------------+------------+-----------------+----------------------+
# nova list
+--------------------------------------+---------+--------+------------+-------------+------------------------+
| ID                                   | Name    | Status | Task State | Power State | Networks               |
+--------------------------------------+---------+--------+------------+-------------+------------------------+
| d72227dc-bfd6-4c60-988b-b152a7bd821a | elogin1 | ACTIVE | -          | Running     | management=10.142.0.176 |
+--------------------------------------+---------+--------+------------+-------------+------------------------+
# ironic node-list
+--------------------------------------+---------+--------------------------------------+-------------+--------------------+-------------+
| UUID                                 | Name    | Instance UUID                        | Power State | Provisioning State | Maintenance |
+--------------------------------------+---------+--------------------------------------+-------------+--------------------+-------------+
| 50698545-ce51-41b3-b873-a3d2dbaf8d79 | elogin1 | d72227dc-bfd6-4c60-988b-b152a7bd821a | power on    | active             | False       |
+--------------------------------------+---------+--------------------------------------+-------------+--------------------+-------------+
```

Managing eLogin Nodes: Node Deployment

- Link between nova and ironic

```
# ./deploy_elogin1.sh
# heat stack-list
+--------------------------------------+------------+-----------------+----------------------+
| id                                   | stack_name | stack_status    | creation_time        |
+--------------------------------------+------------+-----------------+----------------------+
| 956e7974-7557-4091-b749-2833827718f3 | elogin1    | CREATE_COMPLETE | 2016-03-01T22:45:33Z |
+--------------------------------------+------------+-----------------+----------------------+
# nova list
+--------------------------------------+---------+--------+------------+-------------+------------------------+
| ID                                   | Name    | Status | Task State | Power State | Networks               |
+--------------------------------------+---------+--------+------------+-------------+------------------------+
| d72227dc-bfd6-4c60-988b-b152a7bd821a | elogin1 | ACTIVE | -          | Running     | management=10.142.0.176 |
+--------------------------------------+---------+--------+------------+-------------+------------------------+
# ironic node-list
+--------------------------------------+---------+--------------------------------------+------------+--------------------+-------------+
| UUID                                 | Name    | Instance UUID                        | Power State | Provisioning State | Maintenance |
+--------------------------------------+---------+--------------------------------------+------------+--------------------+-------------+
| 50698545-ce51-41b3-b873-a3d2dbaf8d79 | elogin1 | d72227dc-bfd6-4c60-988b-b152a7bd821a | power on   | active             | False       |
+--------------------------------------+---------+--------------------------------------+------------+--------------------+-------------+
```

Managing eLogin Nodes: Node Console

- **Remote console can be accessed by the ironic_conman command**

```
# ironic_conman <hostname>
```

- **Console traffic is logged to /var/log/conman on the CMC**
  - Logs are named by ironic node UUID
    - ironic-c0386c4d-9410-4113-a71b-2a770b6239df.log

Managing eLogin Nodes: Deploy Process

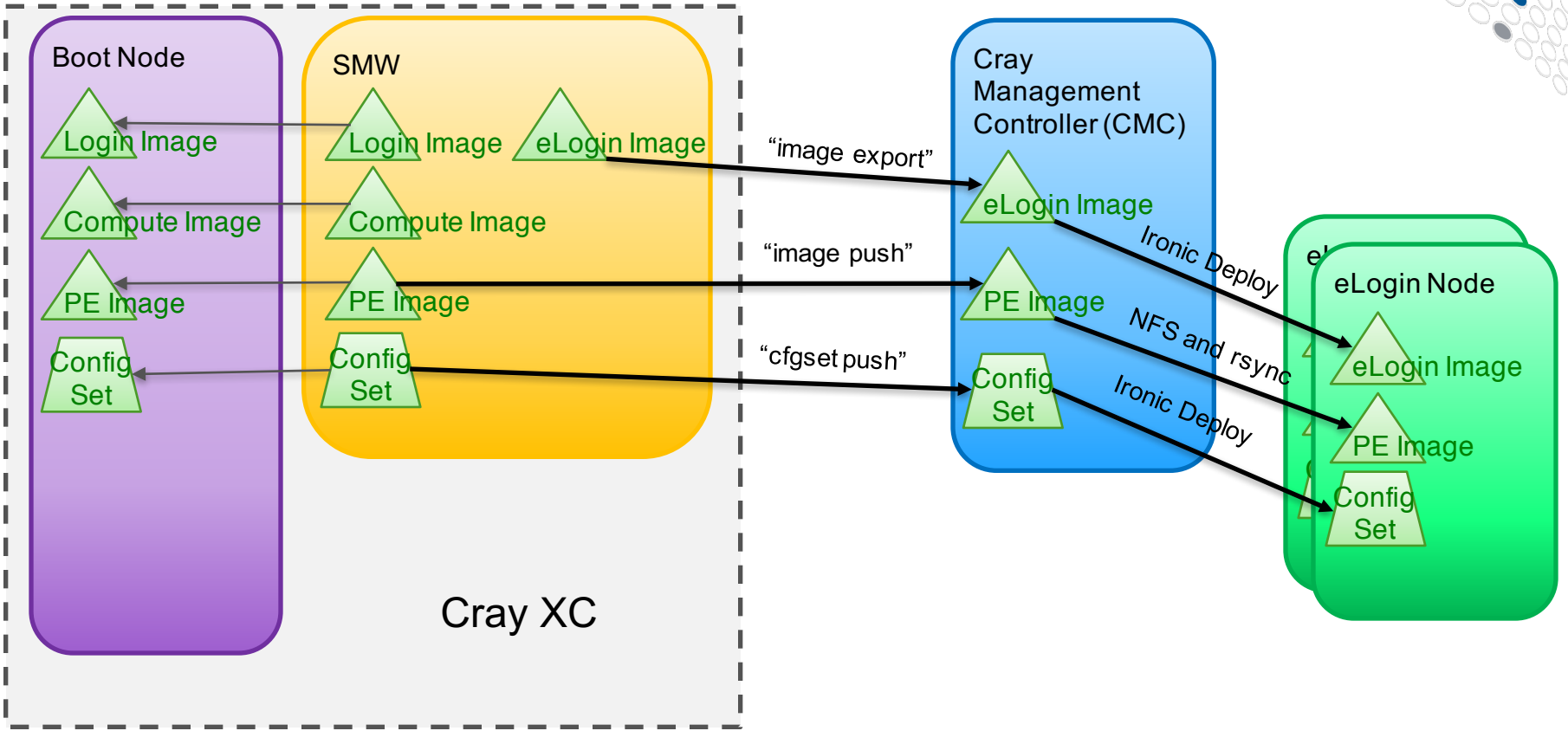
- **During the node deployment process a common deploy image is booted**
  - Checks local disk partitioning and repartitions to match that described in the deploy_config.json file
  - Mounts the local disk and rsyncs the eLogin image to the node
  - Config Set is transferred to node
  - Reboots the node from the eLogin image on the local disk

Managing eLogin Nodes: Boot Process

- **Node boots into a dracut pre-init environment**
  - **cray-ansible** is run to configure the node
  - Cray PE is synchronized to persistent storage on the eLogin node
    - First sync can be long (all data transferred)
    - Subsequent syncs are image diffs only
      - Sync progress may be monitored on the eLogin node

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

Introduction to eLogin: Node Deploy Flow



Managing eLogin Nodes: Node Shutdown

- **Shutdown using 'heat stack-delete <stack_name>'**
  - Tears down the stack
  - Powers off the node
  - Removes the nova instance
  - Removes the heat stack

```
# heat stack-delete elogin1
```

  - **It is important to use this method when shutting down eLogin nodes**

Managing eLogin Nodes: Things to Know

- **CSMS will enforce the power state of the node**
  - Manually powering on a node that is set to “power off” in CSMS (*ironic*) will result in the node being powered off. The reverse is also true.
  - Set the node to “maintenance mode” in ironic to avoid this enforcement
- **eLogin images must be pushed to the CMC after being created or edited on the SMW**
- **Config Sets must be pushed to the CMC after being created or edited on the SMW**

Managing eLogin Nodes: Things to Know

- **Changing eLogin hardware requires updating the inventory.csv file and the elogin-env.yaml file**
  - Copy original inventory.csv file to inventory.csv.back
  - Delete all entries except for the hardware being updated
  - Run `csms_ironic_enrollment.sh`
  - Update the environment template for this node with the new ironic node UUID

Managing eLogin Nodes: Troubleshooting

- **Dealing with systemd (SLES 12 and CentOS 7)**
  - `systemd` forgoes traditional logging mechanisms and stores the following messages in a custom database
    - `syslogd` messages
    - Kernel log messages
    - Initial ram and early boot messages
    - Messages written to stdout/stderr for all services
  - `journalctl` is used to access this information
    - '**`journalctl -a`**' displays all kernel messages and other available information
    - '**`journalctl -f`**' behaves like '**`tail -f`**', displaying updates as they happen

Managing eLogin Nodes: Troubleshooting

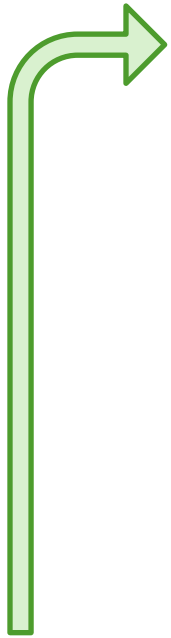
- **cray_dumpsys**
  - Gathers data to help debug Cray System Management Software (CSMS) problems
  - Dumps the state of the OpenStack services, various configuration and log files plus background information about the system
  - Files are compressed and the results are stored in /var/tmp/
  - By default, only recent logs are dumped
    - Use --all-logs option to dump all rotated logs
    - Use --days option to dump logs up to a certain number of days

Managing eLogin Nodes: Troubleshooting

- **cray_dumpsys eLogin plugin**
  - Includes information from eLogin nodes in the cray_dumpsys report
  - Edit /etc/cray_tools/cray_tools.conf
    - Add elogin to the list of enabled plugins, and a space-separated list of eLogin node names in the elogin.nodes option.
  - To override the configured node list, use the cray_dumpsys option --extra-option elogin.nodes="elogin1 elogin2"

Managing eLogin Nodes: Troubleshooting cray_tools.conf example enabling eLogin plugin

```
[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"
```

Managing eLogin Nodes: Troubleshooting cray_dumpsys example

```
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.
```

Managing eLogin Nodes: Troubleshooting

- **Logs**
  - /var/log/messages
  - OpenStack Logs (on the CMC node)
    - /var/log/cinder – block storage service
    - /var/log/glance – image service
    - /var/log/heat – orchestration service
    - /var/log/ironic – bare metal provisioning service
    - /var/log/keystone – identity and authentication service
    - /var/log/neutron – networking service
    - /var/log/nova – node scheduling service
    - /var/log/swift – object storage service (backs glance in CSMS)
  - Ansible Install Logs (on the eLogin node)
    - /var/opt/cray/log/ansible/sitelog-init
    - /var/opt/cray/log/ansible/sitelog-booted

Managing eLogin Nodes: Troubleshooting Diagnostics: Heat Stacks (1 of 2)

- **Heat Stacks**
  - We use heat stacks for deploying nodes

```
example-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"                                                                      |
|                     |   }                                                                                                  |
|                     | ]                                                                                                    |
```

Managing eLogin Nodes: Troubleshooting Diagnostics: Heat Stacks (2 of 2)

- ## Heat Stacks

```
example-cmc # heat stack-show elogin1


| parameters            | {                                                                  |
|                       |   "network_id": "management",                                      |
|                       |   "OS::stack_id": "4452df3e-46f1-4345-8b61-c489bbbc863f",          |
|                       |   "OS::stack_name": "elogin1",                                     |
|                       |   "cray_config_set": “p0-elogin",                                  |
|                       |   "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                                                               |
+-----------------------+--------------------------------------------------------------------+
```

Managing eLogin Nodes: Troubleshooting Diagnostics: Nova (1 of 2)

- **Nova (active servers)**
  - Use nova show to look for details about the eLogin in question

```
example-cmc # nova show elogin1
+--------------------------------------+----------------------------------------------------------+
| Property                             | Value                                                    |
+--------------------------------------+----------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                   |
| OS-EXT-AZ:availability_zone          | nova                                                     |
| OS-EXT-SRV-ATTR:host                 | csms                                                     |
| 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                                     |
```

Managing eLogin Nodes: Troubleshooting Diagnostics: Nova (2 of 2)

- **Nova (active servers)**
  - Use nova show to look for details about the eLogin in question

```
| 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": “p0-elogin", "cray_cims_ip": "10.142.0.1",   |
|                                     | "cray_cims_rsync_password": "daf5ba09-6be4-4e50-bf43-7ba54394aca4", |
|                                     | "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                                 |
+-------------------------------------+------------------------------------------------------------------+
```

Managing eLogin Nodes: Troubleshooting Diagnostics: Ironic

```
example-cmc # ironic node-show elogin3
+-----------------------+--------------------------------------------------------------------+
| Property              | Value                                                              |
+-----------------------+--------------------------------------------------------------------+
| target_power_state    | None                                                               |
| extra                 | {u'description': u'elogin3'}                                       |
| last_error            | None                                                               |
| updated_at            | 2016-03-31T21:18:16+00:00                                          |
| maintenance_reason    | None                                                               |
| provision_state       | available                                                          |
| uuid                  | c0386c4d-9410-4113-a71b-2a770b6239df                               |
| console_enabled       | True                                                               |
| target_provision_state| None                                                               |
| maintenance           | False                                                              |
| inspection_started_at | None                                                               |
| inspection_finished_at| None                                                               |
| power_state           | power off                                                          |
| driver                | fuel_rsync_ipmi                                                    |
```

Managing eLogin Nodes: Troubleshooting Diagnostics: Ironic

```
| reservation            | None                                                                |
| properties             | {u'memory_mb': 131072, u'cpu_arch': u'x86_64', u'local_gb': 550,    |
|                        | u'cpus': 32}                                                        |
| instance_uuid          | None                                                                |
| name                   | elogin3                                                             |
| driver_info            | {u'ipmi_password': u'******', u'ipmi_address': u'10.142.0.7',       |
|                        | u'deploy_ramdisk': u'e59491e5-d4da-4956-99c8-be662f6ea8c7',         |
|                        | u'deploy_kernel': u'60c99670-7512-4372-8102-84d94bdb5b50',          |
|                        | u'ipmi_username': u'root'}                                          |
| created_at             | 2016-03-17T18:59:18+00:00                                           |
| driver_internal_info   | {u'clean_steps': None, u'is_whole_disk_image': False}               |
| chassis_uuid           |                                                                     |
| instance_info          | {}                                                                  |
+------------------------+---------------------------------------------------------------------+
```

Managing eLogin Nodes: Troubleshooting

- **There are multiple places where there may be pressure on the file system on the management server**
  - Images may fill up space in `/var/lib/glance`
    - Remove these using Glance commands only
  - Images may fill up space in `/var/lib/tftpboot`
    - These are removed automatically following a successful deployment
      - If they remain, remove manually
  - PE, config sets and repositories may fill up space in subdirectories of `/var/opt/cray`
    - Remove manually

Managing eLogin Nodes: Troubleshooting

- **The eLogin node is partitioned into two disks:**
  - /dev/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.
  - /dev/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 config set data is resynchronized upon reboot. Administrators can safely delete data here.

Managing eLogin Nodes: Troubleshooting

- **Problem:** Lack of disk space on the management server to store the glance images used to boot.
- **Signature:** A 'heat stack-create' fails. 'heat stack-show' displays "No valid host was found".
- **Log messages:** From /var/log/nova/nova-conductor.log

```
2015-06-16 11:37:34.171 5202 ERROR nova.scheduler.utils [req-d7f3e9fa-f87f-44e0-b615-b288f3de02cd
None] [instance: 43de8ff4-d746-49e6-9226-2a3c159552db] Error from last host: example-cmc (node
e63ffc33-029f-44ac-8808-c55909f85f2f): [u'Traceback (most recent call last):\n', u'  File
"/usr/lib/python2.7/site-packages/nova/compute/manager.py", line 2053, in
_do_build_and_run_instance\n    filter_properties)\n', u'  File "/usr/lib/python2.7/site-
packages/nova/compute/manager.py", line 2184, in
_build_and_run_instance\n    instance_uuid=instance.uuid, reason=six.text_type(e))\n',
u"RescheduledException: Build of instance 43de8ff4-d746-49e6-9226-2a3c159552db was re-scheduled:
Failed to provision instance 43de8ff4-d746-49e6-9226-2a3c159552db: Failed to deploy. Error: Disk
volume where '/var/lib/ironic/master_images/tmpr4qVSW' is located doesn't have enough disk space.
Required 3646 MiB, only 784 MiB available space present.\n"]
```

- **Action:** Free up space for the file system that provides /var/lib/tftpboot/, which is where ironic copies glance images prior to deploy. Perhaps there are leftover ISO files in /root/isos/ that can be deleted.

Managing eLogin Nodes: Troubleshooting

- **Problem:** Inability to communicate with a BMC prevents the admin from performing any operations on the ironic node as well as the corresponding nova server and heat stack. There may be other ways to get into this same state.
- **Signature:** A 'heat stack-delete' fails with a "Provision state still 'deleting'" message.
- **Log messages:** From `/var/log/heat-engine.log`:

```
2015-06-19 08:01:20.911 1743 TRACE heat.engine.resource Error: Server elogin1 delete failed: (500)
Error destroying the instance on node e79e85cd-57f5-4fcd-ba43-14ccea0375e7. Provision state still
'deleting'.
```

- **Action:** Determine why the BMC fails to respond and address that issue. Use 'ironic node-set-provision-state $UUID delete' to clear the Provisioning State. Use 'nova reset-state $SERVER' to clear the server state. At this point, you should be able to delete the heat stack.

Summary

- **eLogin differences over the previous esLogin product.**
  - Prescriptive image builds based on the same image recipes used for CLE nodes
  - Package collections are shared between CLE and eLogin images
  - Software releases with CLE
  - Cray Programming Environment is separate from the base eLogin image
    - Exactly the same Cray PE image as used on the XC
  - Managed by the new Cray System Management Software

Q&A

Jeff Keopp (keopp@cray.com)
Blaine Ebeling (bce@cray.com)

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: ~~ACE~~, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHILEX, LIBSCI, NODEKARE, THREADSTORM. 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.