Hewlett Packard
Enterprise

HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX Systems (22.06 Rev A) (S-8003)

Part Number: S-8003
Published: July 2022

HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX Systems (22.06 Rev A) S-8003

Contents

1 Copyright and Version

CPE: 22.06-Rev_A

Doc git hash: 89217f7b3a37d1903fb5ee57766cd3086e078bd8

Generated: Tue Jul 05 2022

2 About the HPE CPE Installation Guide: CSM on HPE Cray EX Systems

The *HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX Systems (S-8003)* contains procedures for installing the HPE Cray Programming Environment (CPE), workload managers, and third-party programming environment components, including TotalView, Forge, and AMD and Intel compilers.

This publication is intended for system administrators who want to perform an install or reinstall of all CPE components, install additional licensed components, or customize the programming environment files prior to use. It assumes some familiarity with standard Linux and open source tools, including Ansible, YAML, and (optionally) Kubernetes.

- This guide assumes access to a method of obtaining the most current CPE tar files.
- See the *HPE Cray Programming Environment User Guide: CSM on HPE Cray EX Systems (S-8005)* for a complete list of components and modules installed as part of the Programming Environment.

2.1 Release Information

This publication supports the installation of CPE 22.06 on HPE Cray EX systems with:

- HPE Cray (EX) Supercomputer 22.07 recipe for CSM: CSM 1.2.0 (or later) and COS 2.3 (based on SLES 15 SP3)
- HPE Cray (EX) Supercomputer 22.03 recipe for CSM: CSM 1.0.11 (or later) and COS 2.2.X (based on SLES 15 SP3)
- HPE Cray (EX) Supercomputer 22.02 recipe for CSM: CSM 1.0.10 (or later) and COS 2.1.X (based on SLES 15 SP2)

IMPORTANT: Use the following variable substitutions throughout the included procedures.

Variables for all system installations:

- <CPE_RELEASE> = `22.06`
- <CPE_VERSION> = `22.6.X`
- <WLM_RELEASE> = `22.06`
- <SLURMBLOB_VERSION> = `1.1.10`
- <PBSBLOB_VERSION> = `1.1.6`
- <spX> or <SPX> = `sp2` or `SP2` (systems running COS 2.1.X)
- <spX> or <SPX> = `sp3` or `SP3` (systems running COS 2.2.X or COS 2.3.X)

2.2 Record of Revision

New in the CPE 22.06 Rev A publication

This revision of S-8003 for CPE 22.06 adds support for the HPE Cray (EX) Supercomputer 22.07 recipe for CSM: CSM 1.2.0 (or later) and COS 2.3 (based on SLES 15 SP3).

It also includes information and workarounds for the following critical issues:

- **WARNING:** Two critical issues exist on systems using Slurm or PBS Professional (PBS) workload management systems (WLM) with CPE 22.04, 22.05, or 22.06. Workarounds for these issues **must** be applied. See ***Apply Critical Issue Workarounds***.

  The issues are:

  1. Network traffic generated by WLM k8 pods on Cray System Management (CSM) based systems causes the transmission of duplicate network packets on Slingshot fabrics.
  2. After configuring UAIs for HSN access during WLM installation, UAIs have access to previously forbidden areas of the node management network (NMN).

- **WARNING:** On systems running a version of Slurm or PBS prior to 22.04, a networking change within the upgrade from CSM 1.0 to CSM 1.2 can result in Slurm's `slurmctld` and `slurmdbd` pods or PBS's `pbs` pod not starting when migrated to a system running CSM-1.2 (i.e., HPE Cray (EX) Supercomputer 22.07 recipe for CSM).

  Follow the instructions in ***Prevent WLM Pod Issue*** to resolve this issue if the system is running HPE Cray (EX) Supercomputer 22.07 recipe for CSM and using a version of Slurm or PBS prior to 22.04.

- Variables <spX> and <SPX> added to support installation on nodes with COS systems based on either SLES 15 SP2 or SLES 15 SP3.

New in the CPE 22.06 publication

- The following procedures are updated: The Slurm accounting database changes in this release from a MariaDB instance to Percona XtraDB cluster. As such the following procedures are updated:

  - *Run the Upgrade Slurm Installation Script*
  - *Back up Slurm Accounting Database*
  - *Restore Slurm Accounting Database from Backup*

  The slurm-backup.yaml file is now included in the wlm-slurm- directory, in `kubernetes/slurm-backup.yaml`. The following procedures are updated to no longer create this file.

  - *Back up Slurm Spool Directory*
  - *Restore Slurm Spool Directory from Backup*

New in the CPE 22.05 publication

- CPE no longer supports CSM 0.8.2 (or later) and COS 2.0.X (based on SLES 15 SP1). All references are removed.
- Added *Configure Slurm for Systems with Slingshot Networks*.
- Added Totalview to the list of supported third-party products in *Install or Upgrade CPE*.

New in the CPE 22.04 publication

- Added procedures:
  - *Modify PBS Configuration*
  - *Configure PBS GPU Scheduling*
  - *Check PBS Server or Scheduling Logs*
  - *Configure UAIs for HSN Connectivity*
  - *Configure Slurm for Low Noise Mode*
- Added steps to support HSN communication in the following procedures:
  - *Install Slurm Workload Manager*
  - *Upgrade Slurm Workload Manager*
  - *Install PBS Professional Workload Manager*
  - *Upgrade PBS Professional Workload Manager*
- Added a step to remove nodes configured for Node Management Network (NMN) traffic to *Upgrade PBS Professional Workload Manager*
- Variables <spX> and <SPX> added to support installation on nodes with COS systems based on either SLES 15 SP2 or SLES 15 SP3.

New in the CPE 22.03 publication

- Numerous steps within the *Install PBS Professional Workload Manager* and *Upgrade PBS Professional Workload Manager* procedures are replaced by the installation script `post-bringup-install.sh`
- Added the *Replace an Installed CPE Release squashfs File for Redeployment* procedure for replacing an installed CPE image
- Removed troubleshooting topics *Slurm Config Import Pod in Error State*, *PBS Config Import Pod in Error State*, and *pbsnodes No Node List Error*, as the underlying issues are resolved

New in the CPE 22.02 publication

- Numerous steps within the *Install Slurm Workload Manager* and *Upgrade Slurm Workload Manager* procedures are replaced by the installation script `post-bringup-install.sh`
- Added troubleshooting procedures for *Slurm pods stuck in ContainerCreating state* and *Slurm config import pod in Error state*
- Added troubleshooting procedures for *PBS pods stuck in ContainerCreating state* and PBS config import pod in Error state
- Updated variable values in *Release Information*
- Corrected publication title for CPE 21.12 release in *Record of Revision* Publication Title table

New in the CPE 21.12 publication

- Procedural changes to *Install Slurm Workload Manager*, *Install PBS Professional Workload Manager*, and *Upgrade PBS Professional Workload Manager*
- Added information for deploying multiple versions of a third-party product via `pe_deploy` to *Configure CPE Using CFS*

New in the CPE 21.11 Rev A publication

- Added support for systems with COS 2.1.X installed
- Procedural differences based on the COS version installed
- Variables added to *Release Information*
- Installation, upgrade, and troubleshooting instructions added for Slurm and PBS Professional workload managers, see *Install a Workload Manager*
  - Note that for systems with COS 2.0.X installed, the installation instructions for Slurm and PBS Professional workload managers are included in *HPE Cray EX Systems Installation and Configuration Guide (1.4) S-8000*

New in the CPE 21.11 publication

- Removed 1.4 from the publication title, as it is a misnomer.
- Added CSM and COS version requirements to *CPE Installation Prerequisites*.
- Procedural changes due to increased automation.
- Removed information regarding support for AMD ROCm that was incorrectly added in the CPE 21.10 publication.

Publication Title	Date
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.06 Rev A) S-8003	July 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.06) S-8003	June 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.05) S-8003	May 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.04) S-8003	April 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.03) S-8003	March 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (22.02) S-8003	February 2022
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (21.12) S-8003	December 2021
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (21.11) S-8003 Rev A	November 2021
HPE Cray Programming Environment Installation Guide: CSM on HPE Cray EX (21.11) S-8003	November 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.10) S-8003	October 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.09) S-8003	September 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.08) S-8003	August 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.07) S-8003 Rev A	July 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.07) S-8003	July 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.06) S-8003	June 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.05) S-8003	May 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.04) S-8003	April 2021
HPE Cray Programming Environment Installation Guide: CSM 1.4 on HPE Cray EX (21.03) S-8003	March 2021
HPE Cray Asynchronous Installer Guide (21.03) S-8003	March 2021
HPE Cray Asynchronous Installer Guide (20.11) S-8003	November 2020
HPE Cray Asynchronous Installer Guide (20.10) S-8003	October 2020
HPE Cray Asynchronous Installer Guide (20.09) S-8003	September 2020
HPE Cray Asynchronous Installer Guide (20.08) S-8003	August 2020
Cray Asynchronous Installer Guide (20.06) S-8003	June 2020
Cray Asynchronous Installer Guide (20.05) S-8003	May 2020
Cray Asynchronous Installer Guide (20.04) S-8003	April 2020
Cray Asynchronous Installer Guide (20.03) S-8003	March 2020
Cray Asynchronous Installer Guide (20.02) S-8003	February 2020
Cray Shasta Asynchronous Installer Guide (20.01) S-8003	January 2020

2.3 Typographic Conventions

`This style` indicates program code, reserved words, library functions, command-line prompts, screen output, file/path names, variables, and other software constructs. \ (backslash) At the end of a command line, indicates the Linux shell line continuation character (lines joined by a backslash are parsed as a single line).

2.4 Command Prompt Conventions

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

- The root or super-user account always has the # character at the end of the prompt.
- Any non-root account is indicated with `account@hostname>`. A non-privileged account is referred to as `user`.

Node abbreviations: The following list contains abbreviations for nodes used in command prompts.

- CN - Compute Nodes
- NCN - Non Compute Nodes
- AN - Application Node (special type of NCN)
- UAN - User Access Node (special type of AN)

Command prompts: The following list contains command prompts used in this guide.

- `ncn-m001#` - Run the command as root on the specific NCN-M (NCN that is a Kubernetes master node) with hostname `ncn-m001`.
- `ncn-w001#` - Run the command as root on the specific NCN-W (NCN that is a Kubernetes worker node) with hostname `ncn-w001`.
- `uan01#` - Run the command on a specific UAN.
- `cn#` - Run the command as root on any CN. Note that a CN has a hostname of the form `nid123456` (i.e., "nid" and a six digit, zero padded number).
- `pod#` - Run the command as root within a Kubernetes pod.

2.5 Copying and Pasting from a PDF

Using copy/paste from a PDF is notoriously unreliable. Although copying/pasting a command line typically works, copying/pasting formatted file content (e.g., JSON, YAML) typically fails. To ensure that file content is copied and pasted correctly while performing the procedures in this guide:

1. Copy the content from the PDF.
2. Paste it to a neutral editing form and add the necessary formatting.
3. Copy the content from the neutral form and paste it into the console.

TIP: It is always a good idea to double-check copied/pasted commands for correctness, as some commands may not render correctly in the PDF.

3 About Ansible

Ansible is an open-source software provisioning and configuration management tool. More information can be found at https://www.ansible.com/. The CPE Installer leverages Ansible playbooks and roles to install CPE components.

Here's an example of the pe_deploy.yml playbook:

```
---
- hosts: uai:Application_UAN:Application:Compute
  any_errors_fatal: true
  gather_facts: no
  remote_user: root

  pre_tasks:
    - name: Unmount any overlays first
      command: bash /etc/cray-pe.d/pe_overlay.sh cleanup
      when:
        - not cray_cfs_image
        - forcecleanup | default(false)
      ignore_errors: yes

  roles:
    - { role: cray.pe_deploy, cray_pe_pkg: aocc, when: not cray_cfs_image }
    - { role: cray.pe_deploy, cray_pe_pkg: intel, when: not cray_cfs_image }
    - { role: cray.pe_deploy, when: not cray_cfs_image }

  post_tasks:
    - name: Run mount overlay setup script
      command: bash /etc/cray-pe.d/pe_overlay.sh
      when:
        - not cray_cfs_image
        - not forcecleanup | default(false)
```

3.1 Note on Updating Ansible Files

Custom installation instructions require updating Ansible .yml files. These files should be updated with great caution. The syntax of Ansible files does not support using tabs for editing, only spaces. See https://docs.ansible.com/ for more information about Ansible syntax.

4 CPE Installation Prerequisites

The following prerequisites must be satisfied before installing HPE Cray Programming Environment on HPE Cray EX systems running Cray System Management (CSM):

- COS 2.1.X, COS 2.2.X, or COS 2.3.X must be installed

For systems running COS 2.1.X:

- CSM 1.0.X (or later) must be installed

For systems running COS 2.2.X:

- CSM 1.0.11 (or later) must be installed

For systems running COS 2.3.X:

- CSM 1.2.0 (or later) must be installed

For all systems:

- Root administrator access permissions are required to properly run the CPE Installer. Ansible needs these permissions to create the directory structure and install various elements of the CPE. Root access is ***not*** required to run the CPE, only to install or upgrade it.
- Familiarity with the following technologies is required:
  - Linux - To properly run the CPE Installer, an understanding of Linux file system basics is necessary.
  - Ansible - Knowledge of running Ansible and using Ansible playbooks is required. See https://docs.ansible.com/ for more information.
  - YAML - YAML is a human-readable data-serialization language. Ansible playbooks are stored in `.yml` format. Knowledge of YAML is not necessary to run Ansible playbooks but is useful for image customization.
  - Kubernetes (optional) - If installing CPE on containerized User Access Instance (UAI) nodes, an understanding of Kubernetes could be helpful, but such knowledge is not necessary to install or use the nodes.

5 Install or Upgrade CPE

PREREQUISITES

- See *CPE Installation Prerequisites*.
- CPE does not distribute third-party compilers (e.g., AOCC, Intel); they must be downloaded from their respective websites.

OBJECTIVE

Install or upgrade the base Cray Programming Environment on an HPE Cray EX system. The same instructions are followed whether installing CPE for the first time or upgrading CPE on a previously installed system.

IMPORTANT: Throughout this procedure, replace instances of the following variables:

- <CPE_RELEASE>
- <CPE_VERSION>
- <spX> or <SPX>

with the values specified in *Release Information*.

PROCEDURE

1. SSH into the management node.

   ```
   user@hostname> ssh root@<system>-ncn-m001
   ```

2. Create a staging directory for install. This example uses `/var/tmp/cpe`, but any path with at least 10 GB will suffice.

   ```
   ncn-m001# mkdir -p /var/tmp/cpe && cd /var/tmp/cpe
   ```

3. Download the CPE tar file.

   ```
   ncn-m001# wget urlpath/cpe-<CPE_RELEASE>-sles15-<spX>-csm-<CPE_VERSION>.tar.gz
   ```

4. Extract the tar file.

   ```
   ncn-m001# tar xvf cpe-<CPE_RELEASE>-sles15-<spX>-csm-<CPE_VERSION>.tar.gz
   ```

5. Run the `install.sh` script.

   ```
   ncn-m001# cpe-<CPE_RELEASE>-sles15-<spX>/install.sh
   ```

6. Verify that CPE installed successfully. The `install.sh` prints all CPE versions in the product catalog, double check the latest version is also in the list. **TIP:** The latest CPE version is likely not at the end of the output; scroll up a bit to find it.

   ```
   <CPE_VERSION>:
     configuration:
       clone_url: https://vcs.hostname.com/vcs/cray/cpe-config-management.git
       commit: 341017e953c3c57dd46ddbccec168ca28af9199a
       import_branch: cray/cpe/<CPE_VERSION>
       import_date: 2021-09-24 20:10:42.950742
       ssh_url: git@host.com:cray/cpe-config-management.git
   ```

7. (Optional) Upload third party artifact(s) to a Nexus repository.

   The CPE release tar file contains a script, `install-3p.sh`, that uploads third party packages to Nexus repositories. New repositories are automatically created if they do not already exist. The script has two modes of operation:

   a. Upload a file.

      ```
      ncn-m001# install-3p.sh <FILE> <REPO_NAME>
      ```

   b. Upload RPM files, where <RPM_DIR> is a directory of RPMs.

      ```
      ncn-m001# install-3p.sh <RPM_DIR> <REPO_NAME>
      ```

      TIP: The second mode of operation automatically generates RPM repository metadata required for installing via zypper.

   Specific product examples:

   AMD AOCC Compiler

```
ncn-m001# cpe-<CPE_RELEASE>-sles15-<spX>/install-3p.sh \
aocc-compiler-3.2.0.tar aocc-compiler-3.2.0-linux-x86_64-raw
```

ARM Forge

```
ncn-m001# cpe-<CPE_RELEASE>-sles15-<spX>/install-3p.sh \
arm-forge-21.1.2-linux-x86_64.tar arm-forge-21.1.2-linux-x86_64-raw
```

Intel oneAPI

Note that the oneAPI 2022.2.0 release uses the version string “2022.1.0” for RPM versions and installation paths; therefore, it is the version number needed for installation scripts.

```
ncn-m001# cpe-<CPE_RELEASE>-sles15-sp1/install-3p.sh \
intel-oneapi-2022.1.0/ intel-oneapi-2022.1.0
```

Totalview

```
ncn-m001# cpe-<CPE_RELEASE>-sles15-<spX>/install-3p.sh \
totalview-2022.1.11-0.x86_64.rpm totalview-2022.1.11-linux-x86_64-yum
```

Installation of CPE is now complete. If other HPE Cray EX software products are being installed or upgraded in conjunction with CPE, refer to the *HPE Cray EX System Software Getting Started Guide S-8000* to determine which step to execute next. Otherwise, continue to the next sections of this document for operations to configure and deploy new CPE images.

5.1 Optional Third Party Product Image Customization

PREREQUISITES

- The CPE package is installed and third-party artifacts are available in a Nexus repository, see the ***Install or Upgrade CPE***.

OBJECTIVE

Configure third-party compilers AOCC, Forge, and Intel oneAPI into a new CPE image for deployment with CPE deploy.

HPE provides Ansible customization roles for the AMD AOCC Compiler, Intel oneAPI, and Forge. Some steps in this procedure use the AOCC customization as an example; however, the procedure is similar for the other products.

Product ansible roles:

- AMD AOCC Compiler: `cray.pe_aocc_customize`
- Intel oneAPI: `cray.pe_intel_customize`
- Forge: `cray.pe_forge_customize`
- Totalview: `cray.pe_totalview_customize`

For Nvidia HPC SDK, see the ***Enable Nvidia GPU Support*** section of ***HPE Cray Operating System Administration Guide: CSM on HPE CrayEX Systems (S-8024)*** for details

IMPORTANT: Throughout this procedure, replace instances of the following variables:

- <CPE_RELEASE>
- <CPE_VERSION>

with the values specified in ***Release Information***.

PROCEDURE

1. The CPE `install.sh` script executed earlier cloned a local VCS repo. Change directory into the new path and continue.

   ```
   ncn-m001# cd /var/tmp/cpe/cpe-config-management
   ```

2. Verify the image customization role's default variables match the values used earlier for uploading to Nexus.

   ```
   ncn-m001# vi roles/cray.pe_aocc_customize/defaults/main.yml
   ```

   For the `cray.pe_intel_customize` role, `intel_pkgs` can be modified to install a different set of oneAPI components.

3. (Forge Only) Copy the license file to `roles/cray.pe_forge_customization/files/License.dat` or populate the existing empty `License.dat` file with license information.

4. (Totalview only) Copy the license file (`License.dat` or `tv_license_file`) to `roles/cray.pe_totalview/files/` and update `roles/cray.pe_totalview/defaults/main.yml` based on the type of license.

   Example using FNP license:

   ```
   totalview:
   ...
   license_path: "/opt/toolworks/FNP_license"
   license_file: "License.dat"
   ```

   Example using FNE license:

   ```
   totalview:
   ...
   license_path: "/opt/toolworks/FNE_license"
   icense_file: "tv_license_file"
   ```

5. Add, commit, and push changes.

   ```
   ncn-m001# git commit -am "Add customizations to install the AMD AOCC compiler"
   ncn-m001# git push -u origin cpe-<CPE_RELEASE>-integration
   ```

6. Run the CPE image customization script with a parameter (`aocc`, `intel`, `forge`, `totalview`) specifying which built-in playbook to use.

   ```
   ncn-m001# ./cpe-custom-img.sh aocc
   ```

7. After the CFS session completes, a new image (deployable with the provided CPE image) is created. Record the `result_id` for use when preparing the CPE deployment.

   ```
   ncn-m001# cray cfs sessions describe cpe-aocc-customization \
   --format json | jq -r .status.artifacts[].result_id

   0e54050a-c43c-4534-ba38-7191838e348d
   ```

Repeat the steps above for each third party product image that needs CPE support customization. Then continue to the next section in this document, ***Configure CPE Using CFS***, to prepare the CPE deployment.

5.2 Configure CPE Using CFS

PREREQUISITES

- The CPE package is installed; see the *Install or Upgrade CPE*.
- Any optional image customization or third-party product installation is complete, see *Optional Third Party Product Image Customization*.

OBJECTIVE

Prepare a CPE CFS layer for product integration.

PROCEDURE

1. The CPE `install.sh` script executed earlier cloned a local VCS repo and created a new local integration branch. Change directory into the new path and continue.

   ```
   ncn-m001# cd /var/tmp/cpe/cpe-config-management
   ```

2. Determine which images to deploy; the order of roles matter. The first is the top-most layer and also the default image. Lower layers and non-default images must follow.

   The `cray_pe_pkg` parameter takes the following values:

   - `base`: contains the base CPE content including `PrgEnv-cray` and `PrgEnv-gnu`; this is the default value if`cray_pe_pkg` is not set
   - `intel`: contains `PrgEnv-intel` for Intel OneAPI support
   - `aocc`: contains `PrgEnv-aocc` for AMD Optimizing C/C++ Compiler support
   - `amd`: contains `PrgEnv-amd` for AMD ROCm support; HPE recommends installing this for AMD GPU-enabled systems
   - `nvidia`: contains `PrgEnv-nvhpc` for Nvidia HPC SDK support; HPE recommends installing this for Nvidia GPU-enabled systems

   To deploy a customized image, set `img_id` to the IMS image ID of the customized image (recorded during *Optional Third Party Product Image Customization*) and give the image a unique name with `img_name`.

   In the following example, the CPE `base` and `aocc` images are deployed with the current, and a previous, version of PE along with two versions of the AOCC Compiler.

   ```
   ncn-m001# vim pe_deploy.yml
   roles:
     - { role: cray.pe_deploy, when: not cray_cfs_image }
     - { role: cray.pe_deploy, cray_pe_version: "21.10", when: not cray_cfs_image }
     - { role: cray.pe_deploy, cray_pe_pkg: aocc, when: not cray_cfs_image }
     - { role: cray.pe_deploy, cray_pe_pkg: aocc, cray_pe_version: "21.10", when: not cray_cfs_image }
    - { role: cray.pe_deploy, img_name: "aocc-compiler-3.1.0", img_id: "1f506586-e447-4c2a-b38d-
    1158cb29e4f8", when: not cray_cfs_image }
    - { role: cray.pe_deploy, img_name: "aocc-compiler-3.0.0", img_id: "0e54050a-c43c-4534-ba38-
    7191838e348d", when: not cray_cfs_image }
   ```

   TIPS:

   - Use `git diff origin/cpe-<prev_release>-integration..` to check differences between latest and previous integration branches.
   - Use `git checkout origin/cpe-<prev_release>-integration -- pe_deploy.yml` to pick up previously customized files, as needed.

3. (Optional) Customize site modules. The `cray-pe-configuration.csh` and `cray-pe-configuration.sh` scripts contained in `roles/cray.pe_deploy/files` can be modified to meet a site's specific needs.

   - `module_prog`: defines the default module handling system, either Lmod or Environment Modules (TCL)
   - `default_prgenv`: defines the default programming environment
   - `mpaths`: defines any site specific paths to be added to `MODULEPATH` to make site modules available
   - 'init_module_list': defines the modules to be loaded on login
   - `prgenv_module_list`: defines modules to be swapped as part of the PrgEnv module
   - `one_off_set_defaults`: defines a list of paths to `set_default` scripts to be run at deploy time. This enables setting default versions at the component level.

For example, to set Lmod as the default module handling system in the image.

a. In `roles/cray.pe_deploy/files/cray-pe-configuration.csh`, change: "`set module_prog = environment modules`" to "`set module_prog = lmod`".

b. In `roles/cray.pe_deploy/files/cray-pe-configuration.sh`, change: "`module_prog = environment modules`" to "`module_prog = lmod`".

4. Commit and push the changed files to git.

```
ncn-m001# git commit -am "Update CPE packages and image layers"
ncn-m001# git push -u origin cpe-<CPE_RELEASE>-integration
```

5. CPE includes an operation automation script that creates a new CFS configuration with the latest CPE version and commit ID. The script takes two optional parameters:

```
ncn-m001# cpe-cfs.sh [CFS_name] [apply]
```

- When no parameters are specified, the script updates the latest `cpe-yy.mm-integration` config in CFS.
  - It also outputs a section of yaml code for use in a `sat bootprep` input file for integration with other products. **TIP:** `sat bootprep` is a SAT version 2.2.16+ feature (`sat showrev` displays the installed version). See the ***SAT Bootprep*** section of the ***HPE Cray EX System Admin Toolkit (SAT) Guide (S-8031)*** for more information.
- When the `CFS_name` parameter is specified, the script proposes a new `.json` file that adds or replaces any existing CPE layer.
- When the `apply` parameter is specified, the script modifies the CFS config using the proposed `.json` file. HPE recommends a trial run without the `apply` parameter to verify the results, then rerun with the `apply` parameter to incorporate the changes.

```
ncn-m001# ./cpe-cfs.sh cos-config-2.1.27 [apply]
...
----------------------------------------
Updating new CPE CFS configuration ...
{
  "lastUpdated": "2021-10-13T21:05:40Z",
  "layers": [
    {
      "cloneUrl": "https://api-gw-service-nmn.local/vcs/cray/cpe-config-management.git",
      "commit": "4194bd87979f876400fa9159a60985dacee06a3b",
      "name": "cpe-21.11-integration",
      "playbook": "pe_deploy.yml"
    }
  ],
  "name": "cpe-21.11-integration"
}

----------------------------------------
Generating new layers for cos-config-2.0.27 ...
Proposed new layers for cos-config-2.0.27:
{
  "layers": [
...
```

6. If `sat bootprep` is not used (e.g., for NCN-personalization of UAI hosts or if SAT version 2.2.16+ in not installed), then the `cpe-cfs.sh [CFS_name]` parameter must be specified.

a. HPE recommends a trial run without the `apply` parameter to verify the results, then rerun with the `apply` parameter to incorporate the changes.

b. Re-run the script with the `apply` parameter for COS, UAN, and NCN personalization CFS configurations as necessary. Finally update BOS session templates to ensure the latest CPE CFS configs are included on all nodes after reboots. For details, see the ***Configuration Management*** section of the ***CSM Administration Guide***.

TIP: After CFS completes on a compute or UAN node, run `module list` to check if PE is ready to use. Note that module versions listed are examples only and may differ from those currently loaded on the system. For current CPE release product versions, see the release announcement.

```
nid000001# module list
```

```
Currently Loaded Modulefiles:
1) craype-x86-rome    4) perftools-base/21.12.0 7) cray-dsmml/0.2.2 10) PrgEnv-cray/8.3.0
2) libfabric/1.13.0.0 5) cce/13.0.0             8) cray-mpich/8.1.12
3) craype-network-ofi 6) craype/2.7.13          9) cray-libsci/21.08.1.2
```

5.3 Enable CPE in UAIs

PREREQUISITES

- HPE CPE is installed on an HPE Cray EX system running CSM; for version requirements, see *CPE Installation Prerequisites*.
- A WLM is customized into a compute image; for installation procedures, see *Install a Workload Manager*.

OBJECTIVE

After CPE is installed, this procedure ensures that UAIs also run CPE.

PROCEDURE

1. If CPE was previously installed, clear out any UAS projection paths that CPE previously created by running a script from the earlier CPE branch.

   ```
   ncn-m001:~/cpe-config-management# git checkout cray/cpe/<prev_cpe_version>
   ncn-m001:~/cpe-config-management# bash roles/cray.pe_deploy/files/uas_setup_pe.sh clean
   ```

2. Run the `uas_compute_init.sh` script to set up a new or changed compute image for CPE. Where `<BOS_session_template>` corresponds to a template that includes COS & WLM. Running the script without this parameter shows a list of available choices.

   ```
   ncn-m001:~/cpe-config-management# git checkout cray/cpe/<latest_cpe_version>
   ncn-m001:~/cpe-config-management# bash roles/cray.pe_deploy/files/uas_compute_init.sh \
   <BOS_session_template>
   ```

   Proceed based on the outcome of the execution of `uas_compute_init.sh`:

   a. If the script runs to completion and there are no errors, then add CPE to the ncn-personalization CFS layer and run it again on the worker (UAI) nodes. This procedure is done, and the rest of its steps can be skipped.

   b. If there are errors, continue through this procedure, which breaks down the script and references the documentation on which it is based.

3. Run the `uas_setup_pe.sh` script to update the UAS projection paths for CPE. This script can also be run any time the CPE paths are reset in the UAS.

   ```
   ncn-m001:~/cpe-config-management# bash roles/cray.pe_deploy/files/uas_setup_pe.sh
   ```

4. Check that the HSM group labeled **uai** exists. It should contain all worker nodes designated as UAI hosts.

   ```
   ncn-m001# cray hsm groups describe uai
   ```

5. If group label **uai** does not exist, run the helper script below. For further information, see the ***User Access Service*** section of the *Cray System Managment Administration Guide*.

   ```
   ncn-m001# /opt/cray/csm/scripts/node_management/make_node_groups -u
   ```

6. Check that there is a UAI image based on compute nodes.

   ```
   ncn-m001# cray uas admin config images list
   [[results]]
   default = true
   image_id = "85c7fd74-c410-4920-a452-bd84d27d238e"
   imagename = "registry.local/cray/cray-uai-compute:latest"
   ```

7. If no similar image name exists, see the ***User Access Service*** section of the *Cray System Managment Administration Guide* for how to create and register a custom UAI image. Note the BOS session template specific name may vary. CPE requires one built for computes (COS) with a workload manager included.

8. If the compute image is not set as default, use this command to set it.

   ```
   ncn-m001# cray uas admin config images update --default yes \
   85c7fd74-c410-4920-a452-bd84d27d238e
   ```

CPE is now enabled for UAIs; use the `cpe-cfs.sh` script (described in *Configure CPE Using CFS*), or add CPE to the ncn-personalization CFS layer manually, and run it again on the worker (UAI) nodes. For details, see *Perform NCN Personalization*

After successfully adding CPE to the ncn-personalization CFS layer and running it again on the worker (UAI) nodes, configuration of the HPE Cray HPE Cray Programming Environment is complete. Refer to *HPE Cray EX System Software Getting Started Guide S-8000* for further installation instructions.

6 Install Previously Released CPE Packages for CSM on HPE Cray EX

PREQUISITES

- HPE CPE is installed on an HPE Cray EX system running CSM; for version requirements, see *CPE Installation Prerequisites*.

OBJECTIVE

Install a previously released CPE package, <PREV_RELEASE>, after installing the latest CPE.

Previously released CPE packages must use the installer that comes with the latest CPE package. This procedure installs the package along with the latest release.

IMPORTANT: Throughout this procedure, replace instances of:

- <PREV_RELEASE> with the desired previous release's `YY.MM` value

PROCEDURE

1. Download the old CPE tar file, extract it into a path <untar_path>, and then run the following commands.

```
ncn-m001# SQFS=CPE-base.x86_64-<PREV_RELEASE>.squashfs
ncn-m001# cray artifacts create boot-images PE/$SQFS <untar_path>/squashfs/$SQFS
```

2. Check out the integration branch from the VCS git repo, and update two files to include the <PREV_RELEASE> package for deployment.

```
ncn-m001# git checkout integration
ncn-m001# vi pe_deploy.yml
roles:
    - { role: cray.pe_deploy, when: not cray_cfs_image }
    - { role: cray.pe_deploy, cray_pe_version: <PREV_RELEASE>, when: not cray_cfs_image }
```

3. Update the CFS configuration layers (compute/COS, ncn-personalization, and UAN) to point to the new integration branch commit ID.

7 Create Modulefiles for Third Party Products

PREREQUISITES

- Third-party packages are downloaded and installed

OBJECTIVE

These instructions use `crypkg-gen` to create a modulefile for a specific version of a supported third-party product. This allows a site to set a specific version as default.

The following tasks are necessary and can be embedded in a script where a third-party product is being installed.

PROCEDURE

1. Load `craypkg-gen` module.

   ```
   ncn-w001# source /opt/cray/pe/modules/default/init/bash
   ncn-w001# module use /opt/cray/pe/modulefiles
   ncn-w001# module load craypkg-gen
   ```

2. Generate module and set default scripts for products. Where:

   AMD Optimizing C/C++ Compiler: (requires `craypkg-gen` >= 1.3.16)

   ```
   ncn-w001# craypkg-gen -m /opt/AMD/aocc-compiler-<MODULE_VERSION>/
   ```

   Nvidia HPC SDK (requires `craypkg-gen` >= 1.3.16)

   ```
   ncn-w001# craypkg-gen -m /opt/nvidia/hpc_sdk/Linux_x86_64/<MODULE_VERSION>/
   ```

   Intel oneAPI

   ```
   ncn-w001# craypkg-gen -m /opt/intel/oneapi/compilers/<MODULE_VERSION>/
   ```

3. Run a `set default` script.

   ```
   ncn-w001# /opt/admin-pe/set_default_craypkg/set_default_<MODULE_NAME>_<MODULE_VERSION>
   ```

8 Lmod Custom Dynamic Hierarchy

Lmod enables a user to dynamically modify their user environment through Lua modules. CPE's implementation of Lmod capitalizes on its hierarchical structure including Lmod's module auto swapping functionality. This means that the module dependencies determine the branches of the tree-like hierarchy. Lmod allows static and dynamic hierarchical module paths. Lmod provides full support for static paths, which build the hierarchy based on the current set of modules loaded. Alongside static paths, CPE implemented dynamic paths for a subset of the Lmod hierarchy (compilers, networks, CPUs, and MPIs). Dynamic paths give an advanced level of flexibility for detecting multiple dependency paths and allow custom paths to join CPE's existing Lmod hierarchy without modifying customer modulefiles.

8.1 Static Lmod Hierarchy

Modules dependent on one or more modules being loaded are not visible to a user until their prerequisite modules are loaded. When the prerequisite modules are loaded, it adds the static paths of the dependent modules to the `MODULEPATH` environment variable, thereby exposing the dependent modules to the user. For more detailed information on Lmod's static module hierarchy, please consult ***User Guide for Lmod***.

8.2 Dynamic Lmod Hierarchy

CPE's custom dynamic Lmod hierarchy abbreviates the overall Lmod hierarchy tree by relying on compatibility and not directly on a prerequisite's version. Therefore, dependent modules do not need to exist in a new branch every time their prerequisite modules change versions. Instead, dynamic paths use a compatibility version that increases when a new prerequisite module version breaks compatibility in some way. The number following the module's path alias (e.g., `1.0` in `x86-rome/1.0` and `ofi/1.0`) identifies the compatible version.

8.3 Module Path Aliases and Current Compatibility Versions

Compiler	Module Alias/Compatible Version
`cce`	crayclang/10.0
`gcc`	gcc/8.0
`aocc`	aocc/3.0
`intel`	intel/19.0
`nvidia`	nvidia/20

Network	Module Alias/Compatible Version
`craype-network-none`	none/1.0
`craype-network-ofi`	ofi/1.0
`craype-network-ucx`	ucx/1.0

CPU	Module Alias/Compatible Version
`craype-x86-milan`	x86-milan/1.0
`craype-x86-rome`	x86-rome/1.0
`craype-x86-trento`	x86-trento/1.0

MPI	Module Alias/Compatible Version
`cray-mpich`	cray-mpich/8.0
`cray-mpich-abi`	cray-mpich/8.0
`cray-mpich-abi-pre-intel-5.0`	cray-mpich/8.0
`cray-mpich-ucx`	cray-mpich/8.0
`cray-mpich-ucx-abi`	cray-mpich/8.0
`cray-mpich-ucx-abi-pre-intel-5.0`	cray-mpich/8.0

8.4 Custom Dynamic Hierarchy

CPE's custom dynamic hierarchy extension allows custom module paths to join CPE's existing Lmod hierarchy implementation without modifying customer modulefiles. The custom dynamic module types CPE supports include:

- Compiler
- Network
- CPU
- MPI
- Compiler/Network
- Compiler/CPU
- Compiler/Network/CPU/MPI

As each custom dynamic module type loads, a handshake occurs using special pre-defined environment variables. When all hierarchical prerequisites are met, the paths of the dependent modulefiles are added to the `MODULEPATH` environment variable, thereby exposing the dependent modules to the user.

TIPS

- For Lmod to assist a user optimally, it is recommended that a compiler, network, CPU, and MPI module are loaded. Lmod cannot detect modules hidden in dynamic paths without one of each type of module being loaded.
- The `cray-gcc` compiler does not currently support the Custom Dynamic Hierarchy.

8.5 Create a Custom Dynamic Hierarchy

PREREQUISITES

- Lmod is set as the default module handling system

OBJECTIVE

For the CPE Custom Dynamic Hierarchy to detect the desired Lmod module path, one or more custom dynamic environment variables must be created according to the requirements defined within this procedure.

PROCEDURE

To create a custom dynamic environment variable:

1. The environment variable name begins with `LMOD_CUSTOM_`.
2. Append the descriptor of the module type that the environment variable will represent. The module types and descriptors are:

Module Type	Descriptor
Compiler	`COMPILER_`
Network	`NETWORK_`
CPU	`CPU_`
MPI	`MPI_`
Compiler/Network	`COMNET_`
Compiler/CPU	`COMCPU_`
Compiler/Network/CPU/MPI	`CNCM_`

   Example: The custom dynamic environment variable for the combined compiler and CPU module begins with `LMOD_CUSTOM_COMCPU_`.

3. Following the descriptor, append all prerequisite module aliases along with their respective compatible versions. See *Module Path Aliases and Current Compatibility Versions*. The format of the module path alias/compatible version string for each module type is shown below. Note that due to publishing issues, long module alias/compatible version strings are split across two lines as indicated below.

   Module Type: Module Path Alias/Compatible Version String

   Compiler: <compiler_name>/<compatible_version>

   Network: <network_name>/<compatible_version>

CPU: <cpu_name>/<compatible_version>

MPI: String definition is split across two lines

<compiler_name>/<compatible_version>/<network_name>/<compatible_version>/

<mpi_name>/<compatible_version>

Compiler/Network: <compiler_name>/<compatible_version/<network_name>/<compatible_version>

Compiler/CPU: <compiler_name>/<compatible_version>/<cpu_name>/<compatible_version>

Compiler/Network/CPU/MPI: String definition is split across two lines

<compiler_name>/<compatible_version>/<network_name>/<compatible_version>/

<cpu_name>/<compatible_version>/<mpi_name>/<compatible_version>

TIP: To create an acceptably formatted environment variable name, all slashes and dots in the module alias/compatible version string must be replaced with underscores and all letters must be uppercase.

Example Module Path Alias/Compatible Version Strings

- **Compiler** = `cce`

  The path alias/compatible version string (values found in *Module Path Aliases and Current Compatibility Versions*) is `crayclang/10.0`; therefore, the text added to the environment variable name is `CRAYCLANG_10_0`.

- **Network** = `craype-network-ofi`

  The path alias/compatible version string is `ofi/1.0`; therefore, the environment variable text is `OFI_1_0`.

- **CPU** = `craype-x86-rome`

  The path alias/compatible version string is `x86-rome/1.0`; therefore, the environment variable text is `X86_ROME_1_0`.

- **MPI** = `cray-mpich`

  `cray-mpich` has two prerequisite module types (compiler and network). Therefore, the environment variable must include the alias/compatible version for the desired compiler, network, and MPI. For a `cray-mpich` module dependent on `cce` and `craype-network-ofi` the path alias/compatible version string is `crayclang/10.0/ofi/1.0/cray_mpich/8.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0_CRAY_MPICH_8_0`.

- **Compiler/Network** = `cce` with `craype-network-ofi`

  The path alias/compatible version string is `crayclang/10.0/ofi/1.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0`.

- **Compiler/CPU** = `cce` with `craype-x86-rome`

  The path alias/compatible version string is `crayclang/10.0/x86-rome/1.0`; therefore, the environment variable text is `CRAYCLANG_10_0_X86_ROME_1_0`.

- **Compiler/Network/CPU/MPI** = `cce`, `craype-network-ofi`, `craype-x86-rome`, and `cray-mpich`

  The path alias/compatible version string is `crayclang/10.0/ofi/1.0/x86-rome/1.0/cray-mpich/8.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0_X86_ROME_1_0_CRAY_MPICH_8_0`.

4. Append the text: `_PREFIX` following the final module/compatibility text instance. Creation of the custom dynamic environment variable is now complete.

   Example: Network = `craype-network-ofi`

   The custom dynamic environment variable is `LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX`.

NEXT: Add the custom dynamic environment variable to the user environment by exporting it with its value set to the Lmod module path.

Example: Network = `craype-network-ofi`

After executing the command below, all modulefiles in `<lmod_module_path>` are shown to users whenever `craype-network-ofi` is loaded

```
# export LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX=<lmod_module_path>
```

9 Troubleshooting Common Issues

Check here for various troubleshooting topics, which will be added as necessary.

9.1 Some nodes see errors when CFS configurations are applied while updating CPE, and the logs show `pe_overlay.sh` failed

This is often seen when a process, or interactive shell, has a lock on a path within the CPE overlay mounted paths. A reboot of the affected nodes should clear it up, and then a re-run of CFS should work on those nodes. There are a couple of things to try before rebooting:

1. If it's an NCN node, make sure there are no UAIs still running on that worker node.

   ```
   ncn-m001# cray uas uais list
   ```

2. Run these commands manually on the affected node(s), i.e, a UAI host node, UAN, or compute node. This example uses `uan01`.

   ```
   uan01# bash /etc/cray-pe.d/pe_overlay.sh cleanup
   uan01# find /var/opt/cray/pe/pe_images -maxdepth 1 -exec umount -f {} \;
   uan01# find /var/opt/cray/pe -maxdepth 1 -exec umount -f {} \;
   uan01# mount | grep pe_image
   ```

   The `mount` command should list no mounts; otherwise `lsof pe_overlay_path` might help narrow down which process may need to be terminated to free up the path.

When there are no previous CPE mounts active, a re-run of CFS on the affected node(s) should be successful.

9.2 Replace an Installed CPE Release squashfs File for Redeployment

If an incorrect CPE image is installed for a release (e.g., service pack 2 instead of service pack 3), follow this procedure to delete both the CPE base image in S3 storage and the CPS cache, and then redeploy the correct image. Repeat as necessary for optional packages such as `amd`, `aocc`, `intel`, or `nvidia`.

1. Delete the CPE base image in S3 storage and the CPS cache.

   ```
   ncn-m001# cray artifacts delete boot-images PE/CPE-base.x86_64-<CPE_RELEASE>.squashfs
   ncn-m001# cray cps contents delete \
   --s3path s3://boot-images/PE/CPE-base.x86_64-<CPE_RELEASE>.squashfs
   ```

2. Rerun CPE `install.sh` from the correct `.tar` package.

   ```
   ncn-m001# cpe-<CPE_RELEASE>-sles15-<spX>/install.sh
   ```

3. Finally, either:

   a. Rerun CFS on affected nodes.

      ```
      ncn-m001# cray cfs components update --enabled true --state '[]' --error-count 0 <xnode>
      ```

   Or:

   b. Reboot all or a limited number (using `--limit` parameter) of nodes.

      ```
      ncn-m001# cray bos session create --template-uuid cos-sessionTemplate-x.y.z \
      --operation reboot [--limit xnode]
      ```

10 Install a Workload Manager

CPE supports the Slurm and PBS Professional workload managers.

10.1 Install Slurm Workload Manager

PREREQUISITES

- The following system components must be installed:
  - CSM, which includes:
    - Loftsman
    - Helm
  - COS 2.1.X or later
  - UAN

OBJECTIVE

Install Slurm as the system's workload manager.

IMPORTANT: Throughout this procedure, replace instances of:

- <CPE_RELEASE>
- <SLURMBLOB_VERSION>
- <spX> or <SPX>

with the values specified in *Release Information*.

PROCEDURE

1. Start a typescript to capture the commands and output from this installation.

```
ncn-m001# script -af product-slurm.$(date +%Y-%m-%d).txt
ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
```

2. Copy Slurm release tarball onto the system and unpackage the release.

```
ncn-m001# tar -xf cpe-slurm-<CPE_VERSION>-sles15-<SLURMBLOB_VERSION>.tar.gz
ncn-m001# cd wlm-slurm-<SLURMBLOB_VERSION>
```

10.1.1 Update Settings for Slurm Installation

Important: Do not skip any steps in this procedure. Skipping a single step can result in installation failure.

1. Get `customizations.yaml` from git or the `site-init` secret.

   a. If `customizations.yaml` is managed in an external Git repository, then clone a local working tree.

   ```
   ncn-m001# git clone <URL> /root/site-init
   ncn-m001# cd /root/site-init
   ```

   b. Otherwise, extract `customizations.yaml` from the `site-init` secret.

   ```
   ncn-m001# kubectl -n loftsman get secret site-init -o \
   jsonpath='{.data.customizations\.yaml}' | base64 -d - > customizations.yaml
   ```

2. Update settings in `customizations.yaml`.

   a. Set the Slurm `ClusterName` setting.

   ```
   ncn-m001# yq w -i customizations.yaml spec.wlm.cluster_name <ClusterName>
   ```

   b. Set LDAP server information for user lookup.

   - To disable LDAP user lookup for Slurm:

     ```
     ncn-m001# yq w -i customizations.yaml spec.kubernetes.services.cray-sssd.domains []
     ```

     TIP: If LDAP user lookup is disabled, job launch will fail unless either `LaunchParameters=disable_send_gids` is set in `slurm.conf` or `/etc/passwd` and `/etc/group` are configured in the `slurmctld` container.

- To configure LDAP user lookup:

  ```
  ncn-m001# yq w -i customizations.yaml \
  spec.kubernetes.services.cray-sssd.domains[0].name LDAP
  ncn-m001# yq w -i customizations.yaml \
  spec.kubernetes.services.cray-sssd.domains[0].ldapSchema rfc2307
  ncn-m001# yq w -i customizations.yaml \
  spec.kubernetes.services.cray-sssd.domains[0].ldapURI <LDAP_server_URI>
  ncn-m001# yq w -i customizations.yaml \
  spec.kubernetes.services.cray-sssd.domains[0].ldapSearchBase <LDAP_search_base>
  ncn-m001# yq w -i customizations.yaml \
  spec.kubernetes.services.cray-sssd.domains[0].ldapTLSReqcert allow
  ```

  For more information about the possible values for these parameters, see the `sssd-ldap(5)` man page.

c. Configure network settings.

The `update-customizations.sh` script provided in this release updates Slurm network configuration in `customizations.yaml` to support the transition to HSN communication.

TIP: the script is located in `/<TARFILE_DIR>/wlm-slurm-<SLURMBLOB_VERSION>`, where `<TARFILE_DIR>` is the directory path where the release tarfile was downloaded. Ensure location in the correct working directory before proceeding.

- To see a listing of default settings: `./update-customizations.sh`
  - The default settings assume a HSN network of 10.253.0.0/16 and must be overridden if a different HSN network is configured on the system.
- To check the HSN network: `yq r customizations.yaml spec.network.high_speed`
- To override the default settings, export environment variables matching the usage message:
  - For example, `export SLURMCTLD_ADDR=10.253.123.2` to override the `slurmctld` IP address.

```
ncn-m001# ./update-customizations.sh customizations.yaml
```

3. Update the `site-init` secret.

```
ncn-m001# kubectl delete secret -n loftsman site-init
ncn-m001# kubectl create secret -n loftsman generic site-init --from-file=customizations.yaml
```

4. If `customizations.yaml` is managed in an external Git repository, commit the changes.

```
ncn-m001# git add customizations.yaml
ncn-m001# git commit -m "Configure Slurm"
ncn-m001# git push
```

10.1.2 Run the Installation Script

1. Run the `install.sh` script. Use the `-f` option to force a fresh install; use the `-r` option to continue execution after correcting any issues reported by the installation script.

   Option: to use a Slurm RPM other than the provided distribution, copy it into the `rpms/cray-sles15-<spX>-cn/x86_64` directory. The RPM must be built for SLES 15 <SPX> using the `slurm.spec` file from the Slurm distribution. Do this prior to running `install.sh`.

   ```
   ncn-m001# ./install.sh [-f] [-r]
   ```

2. Validate Kubernetes deployments.

   ```
   ncn-m001# kubectl get deployment -n user slurmdb
   NAME      READY   UP-TO-DATE   AVAILABLE   AGE
   slurmdb   1/1     1            1           0d0h

   ncn-m001# kubectl get deployment -n user slurmdbd
   NAME       READY   UP-TO-DATE   AVAILABLE   AGE
   slurmdbd   1/1     1            1           0d0h

   ncn-m001# kubectl get deployment -n user slurmctld
   ```

```
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
slurmctld   1/1     1            1           0d0h
```

10.1.3 Apply Critical Workarounds Relevant to Slurm Installation

WARNING: Critical issues were found in CPE 22.06 WLM installations after the release was distributed. To avoid these issues, complete the procedures in *Apply Critical Issue Workarounds* before proceeding further with this installation.

10.1.4 Configure UAIs for HSN Connectivity

If User Access Instances (UAIs) are used to launch jobs or applications with PBS or Slurm, the UAI network attachment definition must be updated to connect to the high speed network (HSN) rather than the node management network (NMN).

1. Edit the configuration.

   ```
   ncn-m001# kubectl edit net-attach-def -n user macvlan-uas-nmn-conf
   ```

2. Update the `master`, `subnet`, `rangeStart`, `rangeEnd`, and `routes` settings

   ```
   spec:
        config: '{ "cniVersion": "0.3.0", "type": "ipvlan", "master": "hsn0", "mode": "l2",
          "ipam": { "type": "host-local", "subnet": "10.253.0.0/16", "rangeStart": "10.253.124.10",
          "rangeEnd": "10.253.125.254",
          "routes": [{"dst": "10.106.0.0/17", "gw": "10.253.255.254"},
          { "dst": "10.92.100.0/24", "gw": "10.253.255.254" },
          { "dst": "10.103.3.0/25", "gw": "10.253.255.254" }] } }'
   ```

   TIP: this example assumes a HSN network `10.253.0.0/16`; the `subnet`, `rangeStart`, and `rangeEnd` values must be adjusted if a different network is configured. Only newly-created UAIs get the new network settings; existing UAIs are not affected.

10.1.5 Customize Slurm

- To update Slurm configuration templates, see *Configure Slurm During or Post Installation*.
- To update the Ansible configuration, see *Update Slurm Ansible Configuration During or Post Installation*.
- To configure Slurm for systems with Slingshot networks, see *Configure Slurm for Systems with Slingshot Networks*.

10.1.6 Prepare Computes and UANs for Bringup During Slurm Installation

PREREQUISITES

- COS 2.1.X (or later) is running
- System Admin Toolkit (SAT) version 2.2.16 (or later) is installed
  - Use `sat showrev` to determine which version of SAT is installed on the system.
- If these prerequisites cannot be met, see the *Run the Slurm Bringup Script* to complete the Slurm installation.

This procedure adds Slurm software and configuration data to the `sat bootprep` input file. The `sat bootprep` command streamlines the process to build, configure, and boot COS compute and UAN images.

To include Slurm software and configuration data in these operations, ensure that the `sat bootprep` input file includes content similar to that described in the following steps. **TIP:** the `sat bootprep` input file contains content for additional HPE Cray EX software products. The following examples focus on Slurm entries only.

1. Configure COS – SLURM Compute Node Content.

   Customize Slurm Ansible content and add the Slurm layer to COS. This step can be performed in conjunction with steps for `cos-compute` image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` Slurm layer COS configuration settings are as shown below (this is needed during the install or update of COS compute nodes). Replace `<slurm_version>` with the desired supported version of Slurm.

   ```
   - name: slurm-master-<slurm_version>
     playbook: site.yml
     product:
   ```

```
        name: slurm
        version: <slurm_version>
        branch: master
```

2. Configure UAN – SLURM UAN Content

   Customize Slurm Ansible content and add Slurm layer to UAN. This step can be performed in conjunction with steps for the UAN image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` Slurm layer UAN configuration settings are shown below (this is needed during install or update of UAN nodes). Replace `<slurm_version>` with the desired supported version of Slurm.

```
 - name: slurm-master-<slurm_version>
   playbook: site.yml
   product:
     name: slurm
     version: <slurm_version>
     branch: master
```

3. The remaining steps utilize procedures in other publications:

   a. Run `sat bootprep`; refer to the ***SAT Bootprep*** section of the ***HPE Cray EX System Admin Toolkit (SAT) Guide (S-8031)***.

   b. Boot compute nodes; refer to ***HPE Cray Operating System Administration Guide: CSM on HPE Cray EX Systems (S-8025)***.

   c. Boot UAN nodes; refer to ***HPE Cray User Access Node (UAN) Software Administration Guide S-8033***.

   d. Refer to the *HPE Cray EX System Software Getting Started Guide (S-8000)* to determine the next steps for installing additional HPE Cray EX software products or beginning the operational activities required to complete CPE installation.

10.1.7 Run the Slurm Bringup Script

This procedure is only required for installations that do not have the `sat bootprep` command, which is included in System Admin Toolkit (SAT) version 2.2.16, or later. **IMPORTANT:** HPE recommends using `sat bootprep` if it is available.

1. Determine the COS and UAN CFS configuration names.

```
ncn-m001# cray cfs configurations list --format=json | jq -r .[].name
```

2. Determine the COS and UAN BOS template names.

```
ncn-m001# cray bos sessiontemplate list --format=json | jq -r .[].name
```

3. Run the `post-bringup-install.sh` script, providing the CFS configuration and BOS template names acquired above. Use the `-f` option to force a fresh install; use the `-r` option to continue execution after correcting any issues reported by the script.

```
ncm-m001# ./post-bringup-install.sh -c <COS_CFS_config> -s <COS_BOS_template> \
-n <UAN_CFS_config> -u <UAN_BOS_template> [-f] [-r]
```

   TIPS:

   - If the error message ***UAN image ID to customize is empty Please create new UAN image manually and run post-bringup-install.sh -r to resume*** occurs, the UAN image is corrupted. Note the values of `<UAN_CFS_config>` and `<SLURMBLOB_VERSION>` and see ***Recover after UAN Image Corruption during Slurm Installation or Upgrade*** below to resolve this issue.

   - If the OS version (e.g., SLES 15 SP2, SLES 15 SP3) does not match the version supported by Slurm, a fatal error occurs when the script attemps to create of a new compute image. Rerun the script providing CFS configuration and BOS template names for matching OS versions.

10.1.8 Validate the Content Post Slurm Installation

1. Validate compute node content.

   a. Check that `slurmd` is running.

```
ncn-m001# ssh ncn-w001
ncn-w001:~ # ssh nid000001
nid000001# systemctl status slurmd
```

If slurmd is not running, check the logs for errors (see *Retrieve Slurm Logs*).

b. Check the state of Slurm.

```
nid000001# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
workq*       up   infinite      4   idle nid[000001-000004]
```

If the state is down or drain, this indicates a problem with the installation. Check the logs for errors, see *Retrieve Slurm Logs*.

c. Run hostname on every available compute node; <num_computes> is the number of available compute nodes as indicated by the sinfo command.

```
nid000001# srun -N <num_computes> <hostname>
```

For example:

```
nid000001# srun -N 4 <hostname>
nid000003
nid000001
nid000004
nid000002
nid000001# exit
ncn-w001:~ # exit
```

If the results for the system's <num_computes> are not similar to the above, check the logs for errors, see *Retrieve Slurm Logs*.

2. Validate UAN content.

a. Check the state of Slurm.

```
ncn-m001# ssh uan01
uan01# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
workq*       up   infinite      4   idle nid[000001-000004]
```

If the state is down or drain, this indicates a problem with the installation. Check the logs for errors, see *Retrieve Slurm Logs*.

b. Validate Slurm by running hostname on every available compute node; <num_computes> is the number of available compute nodes as indicated by the sinfo command.

```
uan01# srun -N <num_computes> <hostname>
```

For example:

```
uan01# srun -N 4 <hostname>
nid000003
nid000001
nid000004
nid000002
uan01# exit
```

If the results for the system's <num_computes> are not similar to the above, check the logs for errors, see *Retrieve Slurm Logs*.

3. Close the typescript file started at the beginning of this procedure.

```
ncn-m001# exit
```

Successfully reaching this point indicates that Slurm is running and ready for use. Next, proceed to *Enable CPE in UAIs* to ensure UAIs are running CPE with Slurm as the workload manager.

Post installation: to modify the Slurm configuration, see *Configure Slurm During or Post Installation*.

10.2 Upgrade Slurm Workload Manager

PREREQUISITES

- The following system components must be installed:
  - CSM, which includes:
    - Loftsman
    - Helm
  - COS 2.1.X or later
  - UAN
- The Slurm spool directory is backed up, see *Back up Slurm Spool Directory*

OBJECTIVE

On a system running Slurm, upgrade to a newer version. **TIP:** HPE recommends upgrading Slurm while the system is idle (i.e., no Slurm jobs are running) to avoid job failures.

IMPORTANT: Throughout this procedure, replace instances of:

- <CPE_RELEASE>
- <SLURMBLOB_VERSION>
- <spX> or <SPX>

with the values specified in *Release Information*.

PROCEDURE

1. Start a typescript to capture the commands and output from this installation.

   ```
   ncn-m001# script -af product-slurm.$(date +%Y-%m-%d).txt
   ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
   ```

2. Copy Slurm release tarball onto the system and extract the tarball contents.

   ```
   ncn-m001# tar xf cpe-slurm-<CPE_RELEASE>-sles15-<SLURMBLOB_VERSION>.tar.gz
   ncn-m001# cd wlm-slurm-<SLURMBLOB_VERSION>
   ```

3. To be safe, scale down `slurmctld` and `slurmdbd` deployments before upgrading SLURM. Warning, this causes the Slurm service to go down until the next step is completed.

   a. Back up the Slurm accounting database, see *Back up Slurm Accounting Database*.

   b. Scale down `slurmctld` and `slurmdbd` deployments to 0.

   ```
   ncn-m001# kubectl scale deployment -n user slurmdbd --replicas 0
   ncn-m001# kubectl scale deployment -n user slurmctld --replicas 0
   ```

10.2.1 Configure Network Settings to support HSN communication

IMPORTANT: this procedure is only necessary if upgrading from a CPE Slurm 22.03 or earlier release to a CPE Slurm 22.04 or later release. Otherwise, proceed to *Run the Upgrade Slurm Installation Script*.

1. Get `customizations.yaml` from git or the `site-init` secret.

   a. If `customizations.yaml` is managed in an external Git repository, then clone a local working tree.

   ```
   ncn-m001# git clone <URL> /root/site-init
   ncn-m001# cd /root/site-init
   ```

   b. Otherwise, extract `customizations.yaml` from the `site-init` secret.

   ```
   ncn-m001# kubectl -n loftsman get secret site-init -o \
   jsonpath='{.data.customizations\.yaml}' | base64 -d - > customizations.yaml
   ```

2. Configure network settings; run `update-customizations.sh`.

   The `update-customizations.sh` script updates Slurm network configuration in `customizations.yaml` to support the transition to HSN communication.

TIP: the script is located in /<TARFILE_DIR>/wlm-slurm-<SLURMBLOB_VERSION>, where <TARFILE_DIR> is the directory path where the release tarfile was downloaded. Ensure location in the correct working directory before proceeding.

- To see a listing of default settings: ./update-customizations.sh
  - The default settings assume a HSN network of 10.253.0.0/16 and must be overridden if a different HSN network is configured on the system.
- To check the HSN network: yq r customizations.yaml spec.network.high_speed
- To override default settings, export environment variables matching the usage message:
  - For example, export SLURMCTLD_ADDR=10.253.123.2 to override the slurmctld IP address.

```
ncn-m001# yq customizations.yaml
```

3. Update the site-init secret.

```
ncn-m001# kubectl delete secret -n loftsman site-init
ncn-m001# kubectl create secret -n loftsman generic site-init --from-file=customizations.yaml
```

4. If customizations.yaml is managed in an external Git repository, commit the changes.

```
ncn-m001# git add customizations.yaml
ncn-m001# git commit -m "Configure Slurm"
ncn-m001# git push
```

10.2.2 Run the Upgrade Slurm Installation Script

1. Run the installation script.

   - If updating from CPE Slurm 22.06 or earlier, use the -m option to migrate accounting data from the previous MariaDB instance to the new Percona XtraDB high-availability cluster.

   - Use the -f option to force a fresh install.

   - Use the -r option to allow continuation after correcting any issues reported by the installation script.

   - To use a Slurm RPM other than the provided distribution, copy it into the rpms/cray-sles15-<spX>-cn/x86_64 directory. The RPM must be built for SLES 15 <SPX> using the slurm.spec file from the Slurm distribution. Do this prior to running install.sh.

     ```
     ncn-m001# ./install.sh [-f] [-m] [-r]
     ```

2. Validate Kubernetes deployments.

```
ncn-m001# kubectl get deployment -n user slurmdb
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
slurmdb   1/1     1            1           0d0h

ncn-m001# kubectl get deployment -n user slurmdbd
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
slurmdbd   1/1     1            1           0d0h

ncn-m001# kubectl get deployment -n user slurmctld
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
slurmctld   1/1     1            1           0d0h
```

10.2.3 Apply Critical Workarounds Relevant to Slurm Upgrade

WARNING: Critical issues were found in CPE 22.04, 22.05, and 22.06 WLM installations after the releases were distributed. To avoid these issues, complete the procedures in *Apply Critical Issue Workarounds* before proceeding further with this upgrade.

10.2.4 Configure UAIs for HSN Connectivity

If User Access Instances (UAIs) are used to launch jobs or applications with PBS or Slurm, the UAI network attachment definition must be updated to connect to the high speed network (HSN) rather than the node management network (NMN).

1. Edit the configuration.

```
ncn-m001# kubectl edit net-attach-def -n user macvlan-uas-nmn-conf
```

2. Update the `master`, `subnet`, `rangeStart`, `rangeEnd`, and `routes` settings

```
spec:
     config: '{ "cniVersion": "0.3.0", "type": "ipvlan", "master": "hsn0", "mode": "l2",
       "ipam": { "type": "host-local", "subnet": "10.253.0.0/16", "rangeStart": "10.253.124.10",
       "rangeEnd": "10.253.125.254",
       "routes": [{"dst": "10.106.0.0/17", "gw": "10.253.255.254"},
       { "dst": "10.92.100.0/24", "gw": "10.253.255.254" },
       { "dst": "10.103.3.0/25", "gw": "10.253.255.254" }] } }'
```

TIP: this example assumes a HSN network `10.253.0.0/16`; the `subnet`, `rangeStart`, and `rangeEnd` values must be adjusted if a different network is configured. Only newly-created UAIs get the new network settings; existing UAIs are not affected.

10.2.5 Customize Slurm

- To update Slurm configuration templates, see *Configure Slurm During or Post Installation*.
- To update the Ansible configuration, see *Update Slurm Ansible Configuration Post Installation*.
- To configure Slurm for systems with Slingshot networks, see *Configure Slurm for Systems with Slingshot Networks*.

10.2.6 Prepare Computes and UANs for Bringup during Slurm Upgrade

PREREQUISITES

- COS 2.1.X (or later) is running
- System Admin Toolkit (SAT) version 2.2.16 (or later) is installed
  - Use `sat showrev` to determine which version of SAT is installed on the system.
- If these prerequisites cannot be met, see the *Run the Slurm Bringup Script* to complete the Slurm installation.

This procedure adds Slurm software and configuration data to the `sat bootprep` input file. The `sat bootprep` command streamlines the process to build, configure, and boot COS compute and UAN images.

To include Slurm software and configuration data in these operations, ensure that the `sat bootprep` input file includes content similar to that described in the following steps. **TIP:** the `sat bootprep` input file contains content for additional HPE Cray EX software products. The following examples focus on Slurm entries only.

1. Configure COS – SLURM Compute Node Content.

   Customize Slurm Ansible content and add the Slurm layer to COS. This step can be performed in conjunction with steps for `cos-compute` image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` Slurm layer COS configuration settings are as shown below (this is needed during the install or update of COS compute nodes). Replace `<slurm_version>` with the desired supported version of Slurm.

```
- name: slurm-master-<slurm_version>
  playbook: site.yml
  product:
    name: slurm
    version: <slurm_version>
    branch: master
```

2. Configure UAN – SLURM UAN Content

   Customize Slurm Ansible content and add Slurm layer to UAN. This step can be performed in conjunction with steps for the UAN image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` Slurm layer UAN configuration settings are shown below (this is needed during install or update of UAN nodes). Replace `<slurm_version>` with the desired supported version of Slurm.

```
- name: slurm-master-<slurm_version>
  playbook: site.yml
  product:
   name: slurm
   version: <slurm_version>
   branch: master
```

3. The remaining steps utilize procedures in other publications:

   a. Run `sat bootprep`; refer to the ***SAT Bootprep*** section of the ***HPE Cray EX System Admin Toolkit (SAT) Guide (S-8031)***.

   b. Boot compute nodes; refer to ***HPE Cray Operating System Administration Guide: CSM on HPE Cray EX Systems (S-8025)***.

   c. Boot UAN nodes; refer to ***HPE Cray User Access Node (UAN) Software Administration Guide S-8033***.

   d. Refer to the ***HPE Cray EX System Software Getting Started Guide (S-8000)*** to determine the next steps for installing additional HPE Cray EX software products or beginning the operational activities required to complete CPE installation.

10.2.7 Run the Upgrade Slurm Bringup Script

This procedure is only required for installations that do not have the `sat bootprep` command, which is included in System Admin Toolkit (SAT) version 2.2.16, or later. **IMPORTANT:** HPE recommends using `sat bootprep` if it is available.

1. Determine the COS and UAN CFS configuration names.

   ```
   ncn-m001# cray cfs configurations list --format=json | jq -r .[].name
   ```

2. Determine the COS and UAN BOS template names.

   ```
   ncn-m001# cray bos sessiontemplate list --format=json | jq -r .[].name
   ```

3. Run the post-bringup installation script, providing the CFS configuration and BOS template names acquired above. Use the `-f` option to force a fresh install; use the `-r` option to continue execution after correcting any issues reported by the script.

   ```
   ncm-m001# ./post-bringup-install.sh -c <COS_CFS_config> -s <COS_BOS_template> \
   -n <UAN_CFS_config> -u <UAN_BOS_template> [-f] [-r]
   ```

   TIPS:

   - If the error message ***UAN image ID to customize is empty Please create new UAN image manually and run post-bringup-install.sh -r to resume*** occurs, the UAN image is corrupted. Note the values of `<UAN_CFS_config>` and `<SLURMBLOB_VERSION>` and see *Recover after UAN Image Corruption during Slurm Installation or Upgrade* below to resolve this issue.

   - If the OS version (e.g., SLES 15 SP2, SLES 15 SP3) does not match the version supported by Slurm, a fatal error occurs when the script attemps to create of a new compute image. Rerun the script providing CFS configuration and BOS template names for matching OS versions.

10.2.8 Validate the Content Post Slurm Upgrade

1. Validate compute node content.

   a. Check that `slurmd` is running.

   ```
   ncn-m001# ssh ncn-w001
   ncn-w001:~ # ssh nid000001
   nid000001# systemctl status slurmd
   ```

   If `slurmd` is not running, check the logs for errors (see *Retrieve Slurm Logs*).

   b. Check the state of Slurm.

   ```
   nid000001# sinfo
   PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
   workq*       up   infinite      4   idle nid[000001-000004]
   ```

   If the state is `down` or `drain`, this indicates a problem with the installation. Check the logs for errors, see *Retrieve Slurm Logs*.

   c. Run `hostname` on every available compute node; `<num_computes>` is the number of available compute nodes as indicated by the `sinfo` command.

   ```
   nid000001# srun -N <num_computes> <hostname>
   ```

For example:

```
nid000001# srun -N 4 <hostname>
nid000003
nid000001
nid000004
nid000002
nid000001# exit
ncn-w001:~ # exit
```

If the results for the system's `num_computes` are not similar to the above, check the logs for errors, see ***Retrieve Slurm Logs***.

2. Validate UAN content.

   a. Check the state of Slurm.

   ```
   ncn-m001# ssh uan01
   uan01# sinfo
   PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
   workq*       up   infinite      4   idle nid[000001-000004]
   ```

   If the state is `down` or `drain`, this indicates a problem with the installation. Check the logs for errors, see ***Retrieve Slurm Logs***.

   b. Validate Slurm by running `hostname` on every available compute node; `<num_computes>` is the number of available compute nodes as indicated by the `sinfo` command.

   ```
   uan01# srun -N <num_computes> <hostname>
   ```

   For example:

   ```
   uan01# srun -N 4 <hostname>
   nid000003
   nid000001
   nid000004
   nid000002
   uan01# exit
   ```

   If the results for the system's `num_computes` are not similar to the above, check the logs for errors, see ***Retrieve Slurm Logs***.

3. Close the typescript file started at the beginning of this procedure.

   ```
   ncn-m001# exit
   ```

   Successfully reaching this points indicates that Slurm is upgraded, running, and ready for use. Next, proceed to ***Enable CPE in UAIs*** to ensure CPE is using the latest Slurm.

10.3 Configure Slurm During or Post Installation

PREREQUISITES

- CPE is installed, i.e., the `install.sh` script has completed
- Compute nodes are deployed and booted
- The system's workload manager is Slurm

OBJECTIVE

These procedures are optional based on site needs and system circumstances.

10.3.1 Add a New or Configure an Existing Slurm Template

1. Log on to an NCN (`ncn-m001`) as root.

2. To add a new Slurm configuration template (e.g., a LUA script):

   a. Get current `slurm-config-templates.yaml` file.

   ```
   ncn-m001# kubectl get configmap -n services slurm-config-templates \
   -o yaml >slurm-config-templates.yaml
   ```

b. Copy the LUA script to current working directory.

```
ncn-m001# cp <LUA_script> .
```

c. Add LUA script to `slurm-config-templates.yaml`.

```
ncn-m001# yq w -i slurm-config-templates.yaml -- \
'data."burst_buffer.lua"' "$(cat ./<LUA_script>)"
```

d. Skip to step 4.

3. To update the existing Slurm configuration template:

   a. For minor, or simple, updates:

   ```
   ncn-m001# kubectl edit configmap -n services slurm-config-templates
   ```

   b. For more complex updates, such as when the YAML formatting of `kubectl edit` makes editing difficult (e.g., updating `slurm.conf`):

   ```
   ncn-m001# kubectl get configmap -n services slurm-config-templates \
   -o yaml >slurm-config-templates.yaml
   ncn-m001# yq r slurm-config-templates.yaml 'data."slurm.conf"' >slurm.conf
   [Edit slurm.conf]
   ncn-m001# yq w -i slurm-config-templates.yaml 'data."slurm.conf"' "$(cat slurm.conf)"
   ncn-m001# kubectl apply -f slurm-config-templates.yaml
   ```

4. Obtain the currently running Slurm configuration job name, which is formatted `slurm-config-#`. In this example, the job name is `slurm-config-4`.

   ```
   ncn-m001:~ # kubectl get job -n services | grep slurm-config
   slurm-config-4                             1/1           74s        19h
   slurm-config-import-1.0.4                  1/1           96s        50d
   ```

5. Save the job description to `slurm-config.yaml`.

   ```
   ncn-m001# kubectl get job -n services -o yaml <JOB_NAME> &>slurm-config.yaml
   ```

6. Remove the “old” (currently running) Slurm configuration job.

   ```
   ncn-m001# kubectl delete -f slurm-config.yaml
   ```

7. Remove the auto-generated values from `slurm-config.yaml`.

   ```
   ncn-m001# yq d -i slurm-config.yaml spec.template.metadata
   ncn-m001# yq d -i slurm-config.yaml spec.selector
   ```

8. Recreate the Slurm configuration job.

   ```
   ncn-m001# kubectl apply -f slurm-config.yaml
   ```

9. Reconfigure `slurmctld`.

   ```
   ncn-m001# SLURMCTLD_POD=$(kubectl get pod -n user -lapp=slurmctld \
   -o jsonpath='{.items[0].metadata.name}')
   ncn-m001# kubectl exec -n user ${SLURMCTLD_POD} -c slurmctld -- scontrol reconfigure
   ```

10. If currently doing a full Slurm installation or upgrade:

    - Return to either *Prepare Computes and UANs for Bringup During Slurm Installation* or *Prepare Computes and UANs for Bringup During Slurm Upgrade*.

    Otherwise, this is a post-installation configuration change (i.e., compute and UAN nodes are booted):

    - Reboot UAN and compute nodes to restart Slurm. See the *Cray System Management User Guide* for complete details.

    ```
    ncn-m001# cray bos v1 session create --template-uuid <template> --operation reboot
    ```

    Successfully reaching this points indicates that the Slurm configuration is modified, running, and ready for use. If other HPE Cray Programming Environment components are being installed, refer to the table of contents of this guide. Otherwise, refer to the *HPE Cray EX System Software Getting Started Guide S-8000* for further options.

10.3.2 Update Slurm Ansible Configuration During or Post Installation

PREREQUISITES

- CPE is installed, i.e., the `install.sh` script has completed
- Slurm is the system's workload manager

OBJECTIVE

- Update the Ansible configuration, for example, the `slurm_rpms` or `slurm_conf_files` settings.
- Update Ansible content in the `slurm-config-management` repository.

PROCEDURE

1. Get the crayvcs user password used for `git clone` and push in the following steps.

   ```
   ncn-m001# kubectl get secret -n services vcs-user-credentials \
   --template={{.data.vcs_password}} | base64 --decode
   ```

2. Customize Slurm ansible content (requires crayvcs username and password).

   ```
   ncn-m001# git clone https://api-gw-service-nmn.local/vcs/cray/slurm-config-management.git
   ncn-m001# cd slurm-config-management
   ncn-m001# git merge origin/cray/slurm/<SLURMBLOB_VERSION>
   ```

   a. Create a file `group_vars/all/slurm.yaml` with the desired Ansible variable overrides.

      For example, the default definitions for `slurm_rpms` and `slurm_conf_files` are:

      ```
      slurm_rpms:
        - cray-rm-libpals
        - cray-rm-libpals-devel
        - slurm
        - slurm-devel
        - slurm-libpmi
        - slurm-slurmd
        - atp-slurm-plugin

      slurm_conf_files:
        - slurm.conf
        - plugstack.conf
        - cgroup.conf
      ```

      For example, to install the `slurm-torque` RPM, append `slurm-torque` to the `slurm_rpms` list. If a `gres.conf` file is added to the Slurm configuration template, append `gres.conf` to the `slurm_conf_files` list.

   b. Edit the `site.yaml` file. For example, to configure `hodagd`, add it to the list of compute roles. `hodagd` reports information about the Slingshot switch and the 200GB Slingshot NIC (also referred to as Slingshot 11) to vnid.

      ```
      # Set up Slurm on computes:
      - hosts: Compute
        vars:
          system_wlm: Slurm
        roles:
        - { role: keycloak_passwd, when: not cray_cfs_image|default(false)|bool }
        - { role: keycloak_group, when: not cray_cfs_image|default(false)|bool }
        - slurm_repos
        - atom
        - munged
        - slurm_node
        - hodagd
      ```

      Note: The `hodagd` role will only function on Slingshot 11 systems. It should not be configured on Slingshot 10 systems.

   c. Make additional changes and commits as desired.

   d. Push changes to VCS (username crayvcs).

```
ncn-m001# git push origin master
```

3. If currently doing a full Slurm installation or upgrade:

   - Return to either *Prepare Computes and UANs for Bringup During Slurm Installation* or *Prepare Computes and UANs for Bringup During Slurm Upgrade*.

   Otherwise this is a post-installation configuration change (i.e., compute and UAN nodes are booted):

   - Reboot UAN and compute nodes to restart Slurm, see the *Cray System Management User Guide*.

   ```
   ncn-m001# cray bos v1 session create --template-uuid <template> --operation reboot
   ```

   Successfully reaching this points indicates that the Slurm configuration is modified, running, and ready for use. If other HPE Cray Programming Environment components are being installed, refer to the table of contents of this guide. Refer to the *HPE Cray EX System Software Getting Started Guide S-8000* for further options.

10.3.3 Configure Slurm for Low Noise Mode

If some or all of the system compute nodes are configured with the COS (2.2.x or later) Low Noise Mode feature, HPE recommends configuring Slurm to avoid placing applications on CPU 0. This is done by adding CPUSpecList=0 to the node configuration line(s) for those nodes (as shown below) in either a new or existing Slurm template as described in *Add a New or Configure an Existing Slurm Template*.

Example:

```
NodeName=nid00000[1-4] NodeAddr=nid000001-nmn,nid000002-nmn,nid000003-nmn,nid000004-nmn Sockets=2
CoresPerSocket=4 ThreadsPerCore=1 RealMemory=40960 Feature=Intel_Xeon_Silver_4112 CPUSpecList=0
```

10.3.4 Configure Slurm for Systems with Slingshot Networks

The Slurm RPMs provided with CPE include a switch plugin which enables VNI allocation, network resource reservation, and traffic class configuration.

1. Set SwitchType=switch/hpe_slingshot in slurm.conf.

   The plugin's behavior may be configured with the SwitchParameters option. This option consists of a list of comma-separated parameters:

   - vnis=<min>-<max> - Range of VNIs to allocate for jobs and applications. The default value is 32768-65535.
   - tcs=<class1>[:<class2>]... - Set of traffic classes to configure for applications. Supported traffic classes are [DEDICATED_ACCESS], [LOW_LATENCY], [BULK_DATA] and [BEST_EFFORT].
   - single_node_vni - Allocate a VNI for single node job steps.
   - job_vni - Allocate an additional VNI for jobs, shared among all job steps.
   - def_<rsrc>=<val> - Per-CPU reserved allocation for this resource.
   - res_<rsrc>=<val> - Per-node reserved allocation for this resource. If set, overrides the per-CPU allocation.
   - max_<rsrc>=<val> - Maximum per-node application for this resource.

   The resources are:

   - txqs - Transmit command queues. The default is 3 per-CPU, maximum 1024 per-node.
   - tgqs - Target command queues. The default is 2 per-CPU, maximum 512 per-node.
   - eqs - Event queues. The default is 8 per-CPU, maximum 2048 per-node.
   - cts - Counters. The default is 2 per-CPU, maximum 2048 per-node.
   - tles - Trigger list entries. The default is 1 per-CPU, maximum 2048 per-node.
   - ptes - Portable table entries. The default is 8 per-CPU, maximum 2048 per-node.
   - les - List entries. The default is 134 per-CPU, maximum 65535 per-node.
   - acs - Addressing contexts. The default is 4 per-CPU, maximum 1024 per-node.

   For example, the following setting enables shared job VNIs and overrides the default resource reservation to 0 trigger list entries per CPU and 34 list entries per CPU:

   ```
   SwitchParameters=job_vni,def_tles=0,def_les=34
   ```

2. Refer to *Add a New or Configure an Existing Slurm Template* to complete the configuration.

10.4 Slurm Troubleshooting and Administrative Tasks

Helpful commands, troubleshooting tips, and Slurm administrative procedures.

10.4.1 Check Slurm Status

Check `slurmctld` status:

```
[user@uai ~]$ scontrol ping
```

Check `slurmdbd` status:

```
[user@uai ~]$ sacct
```

Check compute node status:

```
[user@uai ~]$ sinfo
```

For more detail:

```
[user@uai ~]$ sinfo --list-reasons
```

10.4.2 Check Slurm Version

Check the Slurm version installed:

```
[user@uai ~]$ rpm -qi slurm
```

10.4.3 Retrieve Slurm Logs

From a compute node:

```
nid000001# journalctl -u slurmd
```

Retrieve `slurmctld` logs:

```
ncn-m001# kubectl logs -n user --timestamps --tail=-1 -c slurmctld -lapp=slurmctld
```

Retrieve `slurmdbd` logs:

```
ncn-m001# kubectl logs -n user --timestamps --tail=-1 -c slurmdbd -lapp=slurmdbd
```

Retrieve accounting logs:

```
ncn-m001# kubectl logs -n user --timestamps --tail=-1 -lapp=slurmdb
```

10.4.4 Update Firmware to Resolve Slurm Issues

If Slurm is not able to use all nodes, it is likely due to an older version of firmware existing on the HPE Cray EX system. For information regarding updating firmware, see *Update Firmware with FAS*.

10.4.5 Downgrading Slurm Fails

OBJECTIVE

An error occurs when a site attempts to downgrade from Slurm on a system running HPE Cray EX Software 1.5.x (i.e., CSM 1.0.X and COS 2.1.X) to the version of Slurm that was installed on the system when it ran HPE Cray EX Software 1.4.x (i.e., CSM 0.8.2 and COS 2.0.X). The error is similar to: ***slurmctld: fatal: Can not recover assoc_usage state, incompatible version, got 9216 need >= 8448 <= 8960, start with '-i' to ignore this. Warning: using -i will lose the data that can't be recovered.***

PROCEDURE

To resolve this issue:

1. Edit the `slurmctld` deployment.

   ```
   ncn-m001# kubectl edit deployment -n user slurmctld
   ```

2. Add `args: ["-i"]` to the `slurmctld` container

3. Exit the editing session to redeploy the slurmctld container.

10.4.6 Slurm Pods Stuck in ContainerCreating State

If the slurmctld or slurmdbd Kubernetes pods are stuck in ContainerCreating state, this may be due to leftover macvlan IP address reservations. To confirm and resolve this issue, follow these steps:

1. Get the pod name and node.

   ```
   ncn-m001# kubectl get pod -n user -o wide
   ```

2. Check the pod events for an error message like:

   ```
   Warning  FailedCreatePodSandBox  58s (x37 over 9m5s)  kubelet, ncn-w001  (combined from similar
   events): Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network
   for sandbox "2d741057d8c40a49f0cae53db4ff5be6217127ef4c25a97166c6401101c990f0": Multus: Err in
   tearing down failed plugins: Multus: error in invoke Delegate add - "macvlan": failed to
      allocate for range 0: no IP addresses available in range set: 10.252.2.2-10.252.2.2
   ```

   ```
   ncn-m001# kubectl describe pod -n user <pod name>
   ```

3. If the above error message appears in the output, ssh to the node the pod is running on and remove the reservation files.

   - For the slurmctld pod:

     ```
     ncn-w001# rm /var/lib/cni/networks/macvlan-slurmctld-nmn-conf/*
     ```

   - For the slurmdbd pod:

     ```
     ncn-w001# rm /var/lib/cni/networks/macvlan-slurmdbd-nmn-conf/*
     ```

   If the error message is something else, refer to the *CSM Administration Guide* for more troubleshooting suggestions.

4. Check that the pod is running; it may take a few minutes to start.

   ```
   ncn-w001# kubectl get pod -n user -o wide
   ```

10.4.7 Slurm or PBS Pods Fail to Start

On systems running a version of Slurm or PBS Professional (PBS) prior to 22.04, a networking change within the upgrade from CSM 1.0 to CSM 1.2 can result in Slurm’s slurmctld and slurmdbd pods or PBS’s pbs pod not starting when migrated to a system running CSM-1.2 (i.e., HPE Cray (EX) Supercomputer 22.07 recipe for CSM).

See *Prevent WLM Pod Issue* to resolve this issue if the system is running HPE Cray (EX) Supercomputer 22.07 recipe for CSM and using a version of Slurm or PBS prior to 22.04.

10.4.8 Recover after UAN Image Corruption during Slurm Installation or Upgrade

This recovery procedure is only applicable for installations or upgrades **not** using sat bootprep to prepare compute and UAN nodes for bringup.

Follow these steps to create a new UAN image when the .post-bringup-install.sh script fails with ***UAN image ID to customize is empty. Please create new UAN image manually and run post-bringup-install.sh -r to resume***.

1. Determine the correct UAN image ID for use in the next step.

   ```
   ncn-m001# cray ims images list
   ```

2. Create a new UAN image.

   ```
   ncn-m001# cray cfs sessions delete "slurm-uan-<SLURMBLOB_VERSION>" || true
   ncn-m001# cray cfs sessions create \
   --name "slurm-uan-<SLURMBLOB_VERSION>" \
   --configuration-name <UAN_CFS_config> \
   --target-definition image \
   --target-group Application <IMAGE_ID>
   ```

3. After the previous command returns true, assign values for etag and path.

```
ncn-m001# BOS_ETAG=$(cray ims images describe $IMS_ID --format=json | jq -r .link.etag)
ncn-m001# BOS_PATH=$(cray ims images describe $IMS_ID --format=json | jq -r .link.path)
```

4. Restart the post installation/post upgrade script.

   ```
   ncn-m001# ./post-bringup-install.sh -r -c <COS_CFS_config> -s <COS_BOS_template> \
   -n <UAN_CFS_config> -u <UAN_BOS_template>
   ```

   - If this is an initial installation of Slurm and the `post-bringup-install.sh` script completes successfully, return to the installation procedure at *Validate the Content Post Slurm Installation*; otherwise, if `post-bringup-install.sh` failed, return to *Run the Slurm Bringup Script* and check **TIPS** for other possible suggestions.
   - If this is an upgrade installation of Slurm and the `post-bringup-install.sh` script completes successfully, return to the installation procedure at *Validate the Content Post Slurm Upgrade*; otherwise, if `post-bringup-install.sh` failed, return to *Run the Upgrade Slurm Bringup Script* and check **TIPS** for other possible suggestions.

10.4.9 Prevent WLM Pod Issue

On systems running a version of Slurm or PBS Professional (PBS) prior to 22.04, a networking change within the upgrade from CSM 1.0 to CSM 1.2 can result in Slurm's `slurmctld` and `slurmdbd` pods or PBS's `pbs` pod not starting when migrated to a system running CSM-1.2.

Solution: On systems running the HPE Cray (EX) Supercomputer 22.07 recipe for CSM and using a version of Slurm or PBS prior to 22.04, follow these steps to correct this issue. Note that the permanent fix for this issue is applied in Slurm or PBS versions 22.04 (or later).

- **Slurm -** Correct this issue as follows:

  1. Edit the `net-attach-def` configuration for `slurmctld`, change the value of `vlan002` to `bond0.nmn0` and save the changes.

     ```
     ncn-m001: kubectl edit net-attach-def -n user macvlan-slurmctld-nmn-conf
     ```

  2. Edit the `net-attach-def` configuration for `slurmdbd`, change the value of `vlan002` to `bond0.nmn0` and save the changes.

     ```
     ncn-m001: kubectl edit net-attach-def -n user macvlan-slurmdbd-nmn-conf
     ```

- **PBS -** Correct this issue as follows:

  Edit the `net-attach-def` configuration for `pbs`, change the value of `vlan002` to `bond0.nmn0` and save the changes.

  ```
  ncn-m001: kubectl edit net-attach-def -n user macvlan-pbs-nmn-conf
  ```

10.4.10 Back up Slurm Accounting Database

OBJECTIVE

The Slurm accounting database holds information about completed Slurm jobs. It is possible to recover from data loss or data corruption in the persistent volume using a backup. **TIP:** As this procedure is distruptive to Slurm operations, HPE recommends doing it only during upgrades or maintenance windows.

Beginning with CPE 22.06, the Slurm accounting database is automatically backed up on a schedule. By default, a backup is taken daily at 9:10PM in the system's configured time zone, and the last 3 backups are kept.

PROCEDURE

To change the schedule or number of retained backups, edit the `spec.backup.schedule` setting with:

```
ncn-m001# kubectl edit pxc -n user slurmdb
```

To perform an on-demand backup, follow these steps:

1. Create a `backup.yaml` file.

   ```
   apiVersion: pxc.percona.com/v1
   kind: PerconaXtraDBClusterBackup
   metadata:
     name: slurmdb-backup
   spec:
     pxcCluster: slurmdb
     storageName: backup
   ```

2. Start the backup.

   ```
   ncn-m001# kubectl apply -n user -f backup.yaml
   ```

3. Check the backup progress and results.

   ```
   ncn-m001# kubectl get pxc-backup -n user
   ```

10.4.11 Restore Slurm Accounting Database from Backup

PREREQUISITE

- A previously backed up Slurm accounting database exists

OBJECTIVE

In the event there is data loss or data corruption in the persistent volume used for the Slurm accounting database, which holds information about completed Slurm jobs, it may be possible to recover.

PROCEDURE

1. Get the name of the backup to be restored.

   ```
   ncn-m001# kubectl get pxc-backup -n user
   cron-slurmdb-backup-2022418000-372f8
   ```

2. Create a `restore.yaml` file with the following content.

   ```
   apiVersion: pxc.percona.com/v1
   kind: PerconaXtraDBClusterRestore
   metadata:
     name: slurmdb-restore
   spec:
     pxcCluster: slurmdb
     backupName: <backup name>
   ```

3. Start the restore:

   ```
   ncn-m001# kubectl apply -n user -f restore.yaml
   ```

4. Check the restore progress and results.

   ```
   ncn-m001# kubectl get pxc-restore -n user
   ```

To restore from a backup made before the transition to Percona XtraDB, run:

```
ncn-m001# kubectl apply -f kubernetes/slurmdb-restore.yaml
```

then check the results.

```
ncn-m001# kubectl get job -n user slurmdb-restore
```

10.4.12 Back up Slurm Spool Directory

PREREQUISITE

- Slurm is the system’s workload manager

OBJECTIVE

Beginning with CPE 22.06, the `install.sh` script automatically performs a Slurm spool directory backup and uploads it to the S3 “wlm” bucket `backups/slurm_spooldir-<SLURMBLOB_VERSION>.tar.gz` path.

Follow this procedure to create a manual backup of the spool directory.

PROCEDURE

1. Stop the `slurmctld` pod.

   ```
   ncn-m001# kubectl scale deployment -n user slurmctld --replicas=0
   ```

2. Apply the `kubernetes/slurm-backup.yaml` file to start the `slurm-backup` pod.

```
ncn-m001# kubectl apply -f kubernetes/slurm-backup.yaml
```

3. Copy the spool directory contents.

```
ncn-m001# kubectl exec -n user slurm-backup -- tar czf - -C /var/spool slurm >slurm_spooldir.tar.gz
```

4. Save the archive in S3.

```
ncn-m001# cray artifacts create wlm backups/slurm_spooldir.tar.gz ./slurm_spooldir.tar.gz
```

5. Delete the `slurm-backup` pod.

```
ncn-m001# kubectl delete -f kubernetes/slurm-backup.yaml.
```

6. Start the `slurmctld` pod.

```
ncn-m001# kubectl scale deployment -n user slurmctld --replicas=1
```

10.4.13 Restore Slurm Spool Directory from Backup

PREREQUISITE

- A previously backed up Slurm spool directory exists.

OBJECTIVE

The `install.sh` script automatically performs a Slurm spool directory backup and uploads it to the S3 “wlm” bucket `backups/slurm_spooldir-<S`
path. Additionally, sites can create a backup via the Back up Slurm Spool Directory procedure.

In the event that a Cray System Management (CSM) or Slurm upgrade causes Slurm to fail, it may be possible to restore from a backup.

PROCEDURE

1. Stop the `slurmctld` pod.

```
ncn-m001# kubectl scale deployment -n user slurmctld --replicas=0
```

2. Apply the file to start the `slurm-backup` pod.

```
ncn-m001# kubectl apply -f kubernetes/slurm-backup.yaml
```

3. Retrieve the backup from S3.

```
ncn-m001# cray artifacts get wlm backups/slurm_spooldir-<SLURMBLOB_VERSION>.tar.gz \
slurm_spooldir-<SLURMBLOB_VERSION>.tar.gz
```

4. Extract the archive.

```
ncn-m001# tar -xf slurm_spooldir.tar.gz
```

5. Copy the spool directory backup into place.

```
ncn-m001# kubectl cp slurm user/slurm-backup:/var/spool
```

6. Delete the `slurm-backup` pod.

```
ncn-m001# kubectl delete -f kubernetes/slurm-backup.yaml
```

7. Start the `slurmctld` pod.

```
ncn-m001# kubectl scale deployment -n user slurmctld --replicas=1
```

10.5 Install PBS Professional Workload Manager

PREREQUISITES

- The following system components must be installed:
  - CSM, which includes:
    - Loftsman
    - Helm
  - COS 2.1.X
  - UAN
- User home directories must be mounted on compute nodes

OBJECTIVE

Install PBS Pro as the system's workload manager.

IMPORTANT: Throughout this procedure, replace instances of:

- <PBSBLOB_VERSION>
- <spX> or <SPX>

with the values specified in ***Release Information***.

PROCEDURE

1. Start a typescript to capture the commands and output from this installation.

   ```
   ncn-m001# script -af product-pbs.$(date +%Y-%m-%d).txt
   ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
   ```

2. Copy PBS release tarball onto the system and unpackage the release.

   ```
   ncn-m001# tar -xf cpe-pbs-<CPE_VERSION>-sles15-<PBSBLOB_VERSION>.tar.gz
   ncn-m001# cd wlm-pbs-<PBSBLOB_VERSION>
   ```

10.5.1 Update Settings for PBS Installation

1. Get `customizations.yaml` from git or the `site-init` secret.

   a. If `customizations.yaml` is managed in an external Git repository, then clone a local working tree.

   ```
   ncn-m001# git clone <URL> /root/site-init
   ncn-m001# cd /root/site-init
   ```

   b. Otherwise, extract `customizations.yaml` from the `site-init` secret.

   ```
   ncn-m001# kubectl -n loftsman get secret site-init -o \
   jsonpath='{.data.customizations\.yaml}' | base64 -d - > customizations.yaml
   ```

2. Configure network settings.

   The `update-customizations.sh` script updates PBS network configuration in `customizations.yaml` to support the transition to HSN communication.

   TIP: the script is located in `/<TARFILE_DIR>/wlm-pbs-<PBSBLOB_VERSION>`, where `<TARFILE_DIR>` is the directory path where the release tarfile was downloaded. Ensure location in the correct working directory before proceeding.

   - To see a listing of default settings: `./update-customizations.sh`
     - The default settings assume a HSN network of 10.253.0.0/16 and must be overridden if a different HSN network is configured on the system.
   - To check the HSN network: `yq r customizations.yaml spec.network.high_speed`
   - To override the default settings, export environment variables matching the usage message:
     - For example, `export PBS_ADDR=10.253.123.4` to override the PBS IP address.

   ```
   ncn-m001# ./update-customizations.sh customizations.yaml
   ```

3. Update the `site-init` secret.

```
ncn-m001# kubectl delete secret -n loftsman site-init
ncn-m001# kubectl create secret -n loftsman generic site-init --from-file=customizations.yaml
```

4. If `customizations.yaml` is managed in an external Git repository, commit the changes.

```
ncn-m001# git add customizations.yaml
ncn-m001# git commit -m "Configure PBS"
ncn-m001# git push
```

10.5.2 Run the Installation Script

1. Run the `install.sh` script. Use the `-f` option to force a fresh install; use the `-r` option to continue execution after correcting any issues reported by the installation script.

   Option: to use a PBS RPM other than the provided distribution, copy it into the `rpms/cray-sles15-<spX>-cn/x86_64` directory. The RPM must be built for SLES 15 <SPX> using the `pbs.spec` file from the PBS distribution. Do this prior to running `install.sh`.

   ```
   ncn-m001# ./install.sh [-f] [-r]
   ```

2. Validate Kubernetes deployments; Check for a running pod.

   ```
   ncn-m001# kubectl get deployment -n user pbs
   NAME      READY   UP-TO-DATE   AVAILABLE   AGE
   pbs       1/1     1            1           0d0h
   ```

10.5.3 Apply Critical Workarounds Relevant to PBS Installation

WARNING: Critical issues were found in CPE 22.06 WLM installations after the release was distributed. To avoid these issues, complete the procedures in *Apply Critical Issue Workarounds* before proceeding further with this installation.

10.5.4 Configure UAIs for HSN Connectivity

If User Access Instances (UAIs) are used to launch jobs or applications with PBS or Slurm, the UAI network attachment definition must be updated to connect to the high speed network (HSN) rather than the node management network (NMN).

1. Edit the configuration.

   ```
   ncn-m001# kubectl edit net-attach-def -n user macvlan-uas-nmn-conf
   ```

2. Update the `master`, `subnet`, `rangeStart`, `rangeEnd`, and `routes` settings

   ```
   spec:
        config: '{ "cniVersion": "0.3.0", "type": "ipvlan", "master": "hsn0", "mode": "l2",
          "ipam": { "type": "host-local", "subnet": "10.253.0.0/16", "rangeStart": "10.253.124.10",
          "rangeEnd": "10.253.125.254",
          "routes": [{"dst": "10.106.0.0/17", "gw": "10.253.255.254"},
          { "dst": "10.92.100.0/24", "gw": "10.253.255.254" },
          { "dst": "10.103.3.0/25", "gw": "10.253.255.254" }] } }'
   ```

   TIP: this example assumes a HSN network `10.253.0.0/16`; the `subnet`, `rangeStart`, and `rangeEnd` values must be adjusted if a different network is configured. Only newly-created UAIs get the new network settings; existing UAIs are not affected.

10.5.5 Customize PBS

To customize PBS, see *Configure PBS During or Post Installation*.

10.5.6 Prepare Computes and UANs for Bringup During PBS Installation

PREREQUISITES

- COS 2.1.X (or later) is running
- System Admin Toolkit (SAT) version 2.2.16 (or later) is installed
  - Use `sat showrev` to determine which version of SAT is installed on the system.
- If these prerequisites cannot be met, see the *Run the PBS Bringup Script* to complete the PBS installation.

This procedure adds PBS software and configuration data to the `sat bootprep` input file. The `sat bootprep` command streamlines the process to build, configure, and boot COS compute and UAN images.

To include PBS software and configuration data in these operations, ensure that the `sat bootprep` input file includes content similar to that described in the following steps. **TIP:** the `sat bootprep` input file contains content for additional HPE Cray EX software products. The following examples focus on PBS entries only.

1. Configure COS – PBS Compute Node Content.

   Add the PBS layer to COS. This step can be performed in conjunction with steps for `cos-compute` image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` PBS layer COS configuration settings are as shown below (this is needed during the install or update of COS compute nodes). Replace `<pbs_version>` with the desired supported version of PBS.

   ```
   - name: pbs-master-<pbs_version>
     playbook: site.yml
     product:
       name: pbs
       version: <pbs_version>
       branch: master
   ```

2. Configure UAN – PBS UAN Content

   Add PBS layer to UAN. This step can be performed in conjunction with steps for the UAN image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` PBS layer UAN configuration settings are shown below (this is needed during install or update of UAN nodes). Replace `<pbs_version>` with the desired supported version of PBS.

   ```
   - name: pbs-master-<pbs_version>
     playbook: site.yml
     product:
       name: pbs
       version: <pbs_version>
       branch: master
   ```

3. The remaining steps utilize procedures in other publications:

   a. Run `sat bootprep`; refer to the ***SAT Bootprep*** section of the ***HPE Cray EX System Admin Toolkit (SAT) Guide (S-8031)***.

   b. Boot compute nodes; refer to ***HPE Cray Operating System Administration Guide: CSM on HPE Cray EX Systems (S-8025)***.

   c. Boot UAN nodes; refer to ***HPE Cray User Access Node (UAN) Software Administration Guide S-8033***.

   d. Refer to the ***HPE Cray EX System Software Getting Started Guide (S-8000)*** to determine the next steps for installing additional HPE Cray EX software products or beginning the operational activities required to complete CPE installation.

10.5.7 Run the PBS Bringup Script

This procedure is only required for installations that do not have the `sat bootprep` command, which is included in System Admin Toolkit (SAT) version 2.2.16, or later. **IMPORTANT:** HPE recommends using `sat bootprep` if it is available.

1. Determine the COS and UAN CFS configuration names.

   ```
   ncn-m001# cray cfs configurations list --format=json | jq -r .[].name
   ```

2. Determine the COS and UAN BOS template names.

   ```
   ncn-m001# cray bos sessiontemplate list --format=json | jq -r .[].name
   ```

3. Run the `post-bringup-install.sh` script, where the command options are as follows:

   - `-c` : COS configuration name
   - `-f` : Do fresh installation
   - `-h` : Help
   - `-l` : `filesystems.yml` path
   - `-n` : UAN configuration name

- `-p` : PBS license information
- `-r` : Resume installation from last failed checkpoint after the fix is applied
- `-s` : COS session template name
- `-u` : UAN session template name

```
ncn-m001# ./post-bringup-install.sh [-h] | [-f] | [-r] & [-c COS_CONFIG_NAME] \
[-s COS_SESSIONTEMPLATE_NAME] [-u UAN_SESSIONTEMPLATE_NAME] [-n UAN_NAME] \
[-l FILESYSTEMS_YML_PATH] [-p PBS_LICENSE_INFO]
```

Here is an example of a `filesystem.yaml` file:

```
filesystems:
  - src: 10.252.1.7:/var/lib/kubelet/fakelus
    mount_point: /lus
    fstype: nfs4
    opts: rw
    state: mounted
```

TIPS:

- Custom installation instructions require updating Ansible `.yml` files. These files must be updated with great caution. The syntax of Ansible files does not support using tabs for editing, only spaces. See https://docs.ansible.com/ for more information about Ansible syntax.

- If the error message ***UAN image ID to customize is empty Please create new UAN image manually and run ./post-bringup-install.sh -r to resume*** occurs, the UAN image is corrupted. Note the values of the UAN configuration name and `<PBSBLOB_VERSION>` and see ***Recover after UAN Image Corruption during PBS Installation*** to resolve this issue.

10.5.8 Validate the Content Post PBS Installation

1. Validate compute node content.

   a. Check that pbs is running.

   ```
   ncn-m001# ssh ncn-w001
   ncn-w001:~ # ssh nid000001
   nid000001# systemctl status pbs
   ```

   If pbs is not running, check the MOM logs in `/var/spool/pbs/mom_logs/<date>` for errors.

   b. Check that the compute nodes are in `free` state.

   ```
   nid000001:~ # pbsnodes -a
        Mom = nid000001
        ntype = PBS
        state = free
        pcpus = 256
        resources_available.arch = linux
        resources_available.host = nid000001
        resources_available.mem = 263580780kb
        resources_available.ncpus = 256
        resources_available.vnode = nid000001
        resources_assigned.accelerator_memory = 0kb
        resources_assigned.hbmem = 0kb
        resources_assigned.mem = 0kb
        resources_assigned.naccelerators = 0
        resources_assigned.ncpus = 0
        resources_assigned.vmem = 0kb
        resv_enable = True
        sharing = default_shared
        last_state_change_time = Wed Sep 22 09:58:06 2021
        last_used_time = Thu Sep 23 13:39:09 2021
   ...
   ```

```
nid00001# exit
ncn-w001:~ # exit
```

2. Validate UAN content by checking compute node PBS state. Note that the directory `/home/users/<USERNAME>` must exist on the compute nodes; if not, `qsub  -I` returns completed state immediately for non-root users.

   a. Look for `server_state` = `Active`, `scheduling` = `True`, and `license_count` > 0.

```
ncn-m001# ssh uan01
uan01# qstat -fB
Server: pbs-service-nmn
    server_state = Active
    server_host = pbs-host
    scheduling = True
    total_jobs = 0
    state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0 Begun:0
    acl_roots = root@*
    default_queue = workq
    log_events = 511
    mail_from = adm
    query_other_jobs = True
    resources_default.ncpus = 1
    default_chunk.ncpus = 1
    resources_assigned.ncpus = 0
    resources_assigned.nodect = 0
    scheduler_iteration = 600
    flatuid = True
    FLicenses = 100001
    resv_enable = True
    node_fail_requeue = 310
    max_array_size = 10000
    pbs_license_info = 6200@hostname.us.site.com:6200@hostname.us.site.com:
    6200@hostname.us.site.com
    pbs_license_min = 0
    pbs_license_max = 2147483647
    pbs_license_linger_time = 31536000
    license_count = Avail_Global:100000 Avail_Local:1 Used:0 High_Use:1
    pbs_version = 19.4.1.20191213044933
    eligible_time_enable = False
    max_concurrent_provision = 5
    max_job_sequence_id = 9999999
```

   b. Look for `state` = `free`.

```
nid000001:~ # pbsnodes -a
     Mom = nid000001
     ntype = PBS
     state = free
     pcpus = 256
     resources_available.arch = linux
     resources_available.host = nid000001
     resources_available.mem = 263580780kb
     resources_available.ncpus = 256
     resources_available.vnode = nid000001
     resources_assigned.accelerator_memory = 0kb
     resources_assigned.hbmem = 0kb
     resources_assigned.mem = 0kb
     resources_assigned.naccelerators = 0
     resources_assigned.ncpus = 0
     resources_assigned.vmem = 0kb
```

```
          resv_enable = True
          sharing = default_shared
          last_state_change_time = Wed Sep 22 09:58:06 2021
          last_used_time = Thu Sep 23 13:39:09 2021
     ...
```

c. Look for job ready.

```
uan01# qsub -I
qsub: waiting for job 7.pbs-service-nmn to start
qsub: job 7.pbs-service-nmn ready

uan01# exit
qsub: job 7.pbs-service-nmn completed
```

3. Finish the typescript file started at the beginning of this procedure.

```
ncn-m001# exit
```

Successfully reaching this point indicates that PBS Pro is running and ready for use. Next, proceed to ***Enable CPE in UAIs*** to ensure UAIs are running CPE with PBS Pro as the workload manager.

10.6 Upgrade PBS Professional Workload Manager

PREREQUISITES

- The following system components must be installed:
  - CSM, which includes:
    - Loftsman
    - Helm
  - COS 2.1.X
  - UAN
- The PBS home directory is backed up, see ***Backup PBS Home Directory***

OBJECTIVE

On a system running PBS, upgrade to a newer version.

IMPORTANT: Throughout this procedure, replace instances of the following variables with the values specified in ***Release Information***:

- <PBSBLOB_VERSION>
- <spX> or <SPX>

PROCEDURE

1. Start a typescript to capture the commands and output from this installation.

```
ncn-m001# script -af product-pbs.$(date +%Y-%m-%d).txt
ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
```

2. Copy the release tarball onto the system and unpackage the release.

```
ncn-m001# tar -xf cpe-pbs-<CPE_VERSION>-sles15-<PBSBLOB_VERSION>.tar.gz
ncn-m001# cd wlm-pbs-<PBSBLOB_VERSION>
```

3. To be safe, scale down the pbs pod before upgrading PBS.

   a. Back up the PBS home database, see ***Backup PBS Home Directory***.

   b. Scale down the PBS pod to 0.

```
ncn-m001# kubectl scale deployment -n user pbs --replicas 0
```

4. Remove nodes configured for Node Management Network (NMN) traffic.

   a. Check for nodes configured for NMN traffic; look for `Mom = nidXXXXXX-nmn` in the output.

```
ncn-m001# pbsnodes -a
```

b. If there are nodes configured for NMN traffic, remove them.

Because the MOM setting cannot be changed after vnode creation, nodes configured to use NMN for PBS MOM traffic must be deleted and then recreated with the correct setting.

```
ncn-m001# PBS_POD=$(kubectl get pod -n user -l app=pbs -o jsonpath='{.items[0].metadata.name}')
ncn-m001# kubectl exec -it -n user $PBS_POD -- qmgr
qmgr: delete node nidXXXXXX
```

The nodes are added back to PBS during installation.

10.6.1 Configure Network Settings to support HSN communication

IMPORTANT: this procedure is only necessary if upgrading from a CPE PBS 22.03 or earlier release to a CPE PBS 22.04 or later release. Otherwise, proceed to *Run the Upgrade PBS Installation Script*.

1. Get customizations.yaml from git or the site-init secret.

   a. If customizations.yaml is managed in an external Git repository, then clone a local working tree.

   ```
   ncn-m001# git clone <URL> /root/site-init
   ncn-m001# cd /root/site-init
   ```

   b. Otherwise, extract customizations.yaml from the site-init secret.

   ```
   ncn-m001# kubectl -n loftsman get secret site-init -o \
   jsonpath='{.data.customizations\.yaml}' | base64 -d - > customizations.yaml
   ```

2. Configure network settings; run update-customizations.sh.

   The update-customizations.sh script updates PBS network configuration in customizations.yaml to support the transition to HSN communication.

   TIP: the script is located in /<TARFILE_DIR>/wlm-pbs-<PBSBLOB_VERSION>, where <TARFILE_DIR> is the directory path where the release tarfile was downloaded. Ensure location in the correct working directory before proceeding.

   - To see a listing of default settings: ./update-customizations.sh
     - The default settings assume a HSN network of 10.253.0.0/16 and must be overridden if a different HSN network is configured on the system.
   - To check the HSN network: yq r customizations.yaml spec.network.high_speed
   - To override default settings, export environment variables matching the usage message:
     - For example: export PBS_ADDR=10.253.123.2 to override the PBS IP address.

   ```
   ncn-m001# yq customizations.yaml
   ```

3. Update the site-init secret.

   ```
   ncn-m001# kubectl delete secret -n loftsman site-init
   ncn-m001# kubectl create secret -n loftsman generic site-init --from-file=customizations.yaml
   ```

4. If customizations.yaml is managed in an external Git repository, commit the changes.

   ```
   ncn-m001# git add customizations.yaml
   ncn-m001# git commit -m "Configure PBS"
   ncn-m001# git push
   ```

10.6.2 Run the Upgrade PBS Installation Script

1. Run the installation script.

   Option: to use a PBS RPM other than the provided distribution, copy it into the rpms/cray-sles15-<spX>-cn/x86_64 directory. The RPM must be built for SLES 15 <SPX> using the pbs.spec file from the PBS distribution. Do this prior to running install.sh.

   a. The installation script is located in the wlm-pbs-<PBSBLOB_VERSION> subdirectory of the directory location where the release tarfile was originally downloaded. If

   ```
   ncn-m001# ./install.sh [-f] [-r]
   ```

2. Validate Kubernetes deployments. Check for running pods.

```
ncn-m001# kubectl get deployment -n user pbs
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
pbs       1/1     1            1           0d0h
```

10.6.3 Apply Critical Workarounds Relevant to PBS Upgrade

WARNING: Critical issues were found in CPE 22.04, 22.05, and 22.06 WLM installations after the releases were distributed. To avoid these issues, complete the procedures in *Apply Critical Issue Workarounds* before proceeding further with this upgrade.

10.6.4 Configure UAIs for HSN Connectivity

If User Access Instances (UAIs) are used to launch jobs or applications with PBS or Slurm, the UAI network attachment definition must be updated to connect to the high speed network (HSN) rather than the node management network (NMN).

1. Edit the configuration.

   ```
   ncn-m001# kubectl edit net-attach-def -n user macvlan-uas-nmn-conf
   ```

2. Update the `master`, `subnet`, `rangeStart`, `rangeEnd`, and `routes` settings

   ```
   spec:
        config: '{ "cniVersion": "0.3.0", "type": "ipvlan", "master": "hsn0", "mode": "l2",
          "ipam": { "type": "host-local", "subnet": "10.253.0.0/16", "rangeStart": "10.253.124.10",
          "rangeEnd": "10.253.125.254",
          "routes": [{"dst": "10.106.0.0/17", "gw": "10.253.255.254"},
          { "dst": "10.92.100.0/24", "gw": "10.253.255.254" },
          { "dst": "10.103.3.0/25", "gw": "10.253.255.254" }] } }'
   ```

   TIP: this example assumes a HSN network `10.253.0.0/16`; the `subnet`, `rangeStart`, and `rangeEnd` values must be adjusted if a different network is configured. Only newly-created UAIs get the new network settings; existing UAIs are not affected.

10.6.5 Customize PBS

To configure PBS following installation, see *Configure PBS During or Post Installation*.

10.6.6 Prepare Computes and UANs for Bringup During PBS Upgrade

PREREQUISITES

- COS 2.1.X (or later) is running
- System Admin Toolkit (SAT) version 2.2.16 (or later) is installed
  - Use `sat showrev` to determine which version of SAT is installed on the system.
- If these prerequisites cannot be met, see the *Run the Upgrade PBS Bringup Script* to complete the PBS installation.

This procedure adds PBS software and configuration data to the `sat bootprep` input file. The `sat bootprep` command streamlines the process to build, configure, and boot COS compute and UAN images.

To include PBS software and configuration data in these operations, ensure that the `sat bootprep` input file includes content similar to that described in the following steps. **TIP:** the `sat bootprep` input file contains content for additional HPE Cray EX software products. The following examples focus on PBS entries only.

1. Configure COS – PBS Compute Node Content.

   Add PBS layer to UAN. This step can be performed in conjunction with steps for the UAN image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` PBS layer COS configuration settings are as shown below (this is needed during the install or update of COS compute nodes). Replace `<pbs_version>` with the desired supported version of PBS.

   ```
     - name: pbs-master-<pbs_version>
       playbook: site.yml
       product:
         name: pbs
         version: <pbs_version>
         branch: master
   ```

2. Configure UAN – PBS UAN Content

   Add PBS layer to UAN. This step can be performed in conjunction with steps for the UAN image (see the ***SAT Bootprep Details*** section of ***HPE Cray Operating System Installation Guide: CSM on HPE Cray EX Systems (S-8025)***).

   The `sat bootprep` PBS layer UAN configuration settings are shown below (this is needed during install or update of UAN nodes). Replace `<pbs_version>` with the desired supported version of PBS.

   ```
   - name: pbs-master-<pbs_version>
     playbook: site.yml
     product:
       name: pbs
       version: <pbs_version>
       branch: master
   ```

3. The remaining steps utilize procedures in other publications:

   a. Run `sat bootprep`; refer to the ***SAT Bootprep*** section of the ***HPE Cray EX System Admin Toolkit (SAT) Guide (S-8031)***.

   b. Boot compute nodes; refer to ***HPE Cray Operating System Administration Guide: CSM on HPE Cray EX Systems (S-8025)***.

   c. Boot UAN nodes; refer to ***HPE Cray User Access Node (UAN) Software Administration Guide S-8033***.

   d. Refer to the ***HPE Cray EX System Software Getting Started Guide (S-8000)*** to determine the next steps for installing additional HPE Cray EX software products or beginning the operational activities required to complete CPE installation.

10.6.7 Run the Upgrade PBS Bringup Script

This procedure is only required for installations that do not have the `sat bootprep` command, which is included in System Admin Toolkit (SAT) version 2.2.16, or later. **IMPORTANT:** HPE recommends using `sat bootprep` if it is available.

1. Determine the COS and UAN CFS configuration names.

   ```
   ncn-m001# cray cfs configurations list --format=json | jq -r .[].name
   ```

2. Determine the COS and UAN BOS template names.

   ```
   ncn-m001# cray bos sessiontemplate list --format=json | jq -r .[].name
   ```

3. Run the `post-bringup-install.sh` script, where the command options are as follows:

   - `-c` : COS configuration name
   - `-f` : Do fresh installation
   - `-h` : Help
   - `-l` : `filesystems.yml` path
   - `-n` : UAN configuration name
   - `-p` : PBS license information
   - `-r` : Resume installation from last failed checkpoint after the fix is applied
   - `-s` : COS session template name
   - `-u` : UAN session template name

   ```
   ncn-m001# ./post-bringup-install.sh [-h] | [-f] | [-r] & [-c COS_CONFIG_NAME] \
   [-s COS_SESSIONTEMPLATE_NAME] [-u UAN_SESSIONTEMPLATE_NAME] [-n UAN_NAME] \
   [-l FILESYSTEMS_YML_PATH] [-p PBS_LICENSE_INFO]
   ```

   Here is an example of a `filesystem.yaml` file:

   ```
   filesystems:
     - src: 10.252.1.7:/var/lib/kubelet/fakelus
       mount_point: /lus
       fstype: nfs4
       opts: rw
       state: mounted
   ```

   TIPS:

- Custom installation instructions require updating Ansible .yml files. These files must be updated with great caution. The syntax of Ansible files does not support using tabs for editing, only spaces. See https://docs.ansible.com/ for more information about Ansible syntax.
- If the error message ***UAN image ID to customize is empty Please create new UAN image manually and run ./post-bringup-install.sh -r to resume*** occurs, the UAN image is corrupted. Note the values of the UAN configuration name and <PBSBLOB_VERSION> and see ***Recover after UAN Image Corruption during PBS Installation*** to resolve this issue.

10.6.8 Validate the Content Post PBS Upgrade

1. Validate compute node content.

   a. Check that pbs is running.

   ```
   ncn-m001# ssh ncn-w001
   ncn-w001:~ # ssh nid000001
   nid000001# systemctl status pbs
   ```

   If pbs is not running, check the MOM logs in /var/spool/pbs/mom_logs/<date> for errors.

   b. Check that the compute nodes are in free state.

```
nid000001:~ # pbsnodes -a
     Mom = nid000001
     ntype = PBS
     state = free
     pcpus = 256
     resources_available.arch = linux
     resources_available.host = nid000001
     resources_available.mem = 263580780kb
     resources_available.ncpus = 256
     resources_available.vnode = nid000001
     resources_assigned.accelerator_memory = 0kb
     resources_assigned.hbmem = 0kb
     resources_assigned.mem = 0kb
     resources_assigned.naccelerators = 0
     resources_assigned.ncpus = 0
     resources_assigned.vmem = 0kb
     resv_enable = True
     sharing = default_shared
     last_state_change_time = Wed Sep 22 09:58:06 2021
     last_used_time = Thu Sep 23 13:39:09 2021
...
nid00001# exit
ncn-w001:~ # exit
```

2. Validate UAN content by checking compute node PBS state. Note that the directory /home/users/<USERNAME> must exist on the compute nodes; if not, qsub -I returns completed state immediately for non-root users.

   a. Look for server_state = Active, scheduling = True, and license_count > 0.

   ```
   ncn-m001# ssh uan01
   uan01# qstat -fB
   Server: pbs-service-nmn
       server_state = Active
       server_host = pbs-host
       scheduling = True
       total_jobs = 0
       state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0 Begun:0
       acl_roots = root@*
       default_queue = workq
       log_events = 511
   ```

```
        mail_from = adm
        query_other_jobs = True
        resources_default.ncpus = 1
        default_chunk.ncpus = 1
        resources_assigned.ncpus = 0
        resources_assigned.nodect = 0
        scheduler_iteration = 600
        flatuid = True
        FLicenses = 100001
        resv_enable = True
        node_fail_requeue = 310
        max_array_size = 10000
        pbs_license_info = 6200@hostname.us.site.com:6200@hostname.us.site.com:
        6200@hostname.us.site.com
        pbs_license_min = 0
        pbs_license_max = 2147483647
        pbs_license_linger_time = 31536000
        license_count = Avail_Global:100000 Avail_Local:1 Used:0 High_Use:1
        pbs_version = 19.4.1.20191213044933
        eligible_time_enable = False
        max_concurrent_provision = 5
        max_job_sequence_id = 9999999
```

b. Look for `state` = `free`.

```
nid000001:~ # pbsnodes -a
     Mom = nid000001
     ntype = PBS
     state = free
     pcpus = 256
     resources_available.arch = linux
     resources_available.host = nid000001
     resources_available.mem = 263580780kb
     resources_available.ncpus = 256
     resources_available.vnode = nid000001
     resources_assigned.accelerator_memory = 0kb
     resources_assigned.hbmem = 0kb
     resources_assigned.mem = 0kb
     resources_assigned.naccelerators = 0
     resources_assigned.ncpus = 0
     resources_assigned.vmem = 0kb
     resv_enable = True
     sharing = default_shared
     last_state_change_time = Wed Sep 22 09:58:06 2021
     last_used_time = Thu Sep 23 13:39:09 2021
...
```

c. Look for job ready.

```
uan01# qsub -I
qsub: waiting for job 7.pbs-service-nmn to start
qsub: job 7.pbs-service-nmn ready

uan01# exit
qsub: job 7.pbs-service-nmn completed
```

3. Close the typescript file started at the beginning of this procedure.

```
ncn-m001# exit
```

Successfully reaching this points indicates that PBS is upgraded, running, and ready for use. Next, proceed to ***Enable CPE in UAIs*** to ensure CPE is using the latest PBS.

10.7 Configure PBS During or Post Installation

PREREQUISITES

- CPE is installed, i.e., the `install.sh` script has completed
- Compute nodes are deployed and booted
- The system's workload manager is PBS

OBJECTIVE

These procedures are optional based on site needs and system circumstances.

10.7.1 Modify PBS Configuration

```
ncn-m001# PBS_POD=$(kubectl get pod -n user -l app=pbs -o jsonpath='{.items[0].metadata.name}')
ncn-m001# kubectl exec -it -n user $PBS_POD -- qmgr
```

Run qmgr commands as described in PBS documentation

10.7.2 Update PBS Ansible Configuration During or Post Installation

OBJECTIVE

Update the Ansible configuration. The following example configures `hodagd`, which reports information about the local HPE Slingshot NICs (Slingshot 11) to `vnid`.

PROCEDURE

1. Get the crayvcs user password used for `git clone` and push in the following steps.

   ```
   ncn-m001# kubectl get secret -n services vcs-user-credentials \
   --template={{.data.vcs_password}} | base64 --decode
   ```

2. Customize PBS ansible content (requires crayvcs username and password).

   ```
   ncn-m001# git clone https://api-gw-service-nmn.local/vcs/cray/pbs-config-management.git
   ncn-m001# cd pbs-config-management
   ncn-m001# git merge origin/cray/pbs/<PBSBLOB_VERSION>
   ```

   a. Edit the `site.yml` file. Add `hodagd` to the list of compute roles.

      ```
      # Set up PBS on computes
      - hosts: Compute
        vars:
          system_wlm: PBS
        roles:
        - { role: keycloak_passwd, when: not cray_cfs_image|default(false)|bool }
        - { role: keycloak_group, when: not cray_cfs_image|default(false)|bool }
        - PBS_repos
        - PALS
        - atom
        - munged
        - PBS_node
        - hodagd
      ```

   b. Make additional changes and commits as desired.

   c. Push changes to VCS (username crayvcs).

      ```
      ncn-m001# git push origin master
      ```

3. If currently doing a full PBS installation or upgrade:

   - Return to either *Prepare Computes and UANs for Bringup During PBS Installation* or *Prepare Computes and UANs for Bringup During PBS Upgrade*.

Otherwise this is a post-installation configuration change (i.e., compute and UAN nodes are booted):

- Reboot UAN and compute nodes to restart Slurm, see the *Cray System Management User Guide*.

```
ncn-m001# cray bos v1 session create --template-uuid <template> --operation reboot
```

Successfully reaching this points indicates that the PBS configuration is modified, running, and ready for use. If other HPE Cray Programming Environment components are being installed, refer to the table of contents of this guide. Refer to the *HPE Cray EX System Software Getting Started Guide S-8000* for further options.

10.7.3 Configure PBS GPU Scheduling

For systems with compute nodes that have GPUs, use this procedure to configure PBS GPU scheduling.

```
ncn-m001# PBS_POD=$(kubectl get pod -n user -l app=pbs -o jsonpath='{.items[0].metadata.name}')
ncn-m001# kubectl exec -it -n user $PBS_POD -- /bin/bash
#Edit /var/spool/pbs/sched_priv/sched_config
#Add "ngpus" to the "resources" list
#For example: resources: "ncpus, mem, arch, host, vnode, aoe, eoe, ngpus"
ncn-m001# /etc/init.d/pbs restart
```

10.7.4 Enable PBS to Use Low Noise Mode

OBJECTIVE

This feature is only available on systems running COS 2.2.x or later.

Some application workloads show improved performance when compute node operating system tasks (sources of “OS noise”) are migrated to one or more system CPUs that are excluded from application use. Enabling PBS Pro to use Low Noise Mode ensures there is at least one node not available for application use.

For example purposes, this procedure isolates CPU 0, thereby exluding it for application use.

PROCEDURE

1. Create PBSPro cgroups hooks if not already created.

   ```
   ncn-m001# PBS_POD=$(kubectl get pod -n user -l app=pbs -o jsonpath='{.items[0].metadata.name}')
   ncn-m001# kubectl exec -it -n user $PBS_POD -- bash
   ```

   a. Export pbs_cgroups.

      ```
      pbs-hosts# qmgr -c "export hook pbs_cgroups application/x-config default" >pbs_cgroups.json
      ```

   b. Edit `pbs_cgroups.json` and set `exclude_cpus: [0]` to set system hyper-threads to only run on core 0.

   c. Import `pbs_cgroups.json`.

      ```
      pbs-host# qmgr -c "import hook pbs_cgroups application/x-config default pbs_cgroups.json"
      ```

   d. Enable PBS cgroups.

      ```
      pbs-host# qmgr -c "set hook pbs_cgroups enabled=True"
      pbs-host# exit
      ```

2. Upgdate PALSD configuration in `/etc/sysconfig/palsd`.

   a. Get the crayvcs user password used for `git clone` and push in the following steps.

      ```
      ncn-m001# kubectl get secret -n services vcs-user-credentials \
      --template={{.data.vcs_password}} | base64 --decode
      ```

   b. Customize PBS Ansible content (requires crayvcs username and password).

      ```
      ncn-m001# git clone https://api-gw-service-nmn.local/vcs/cray/pbs-config-management.git
      ncn-m001# cd pbs-config-management
      ncn-m001# git merge origin/cray/pbs/<PBSBLOB_VERSION>
      ```

c. The default PALS sysconfig settings are in `roles/palsd/defaults/main.yml`.

   Create a file `group_vars/all/pbs.yaml` with updated variable settings. For example, to set `PALSD_CONFIG_PERAPP=1` in `/etc/sysconfig/palsd`, create `group_vars/all/pbs.yaml` file with this content:

   ```
   palsd_config_perapp: 1
   ```

d. Make additional changes and commits as desired.

e. Push changes to VCS (username crayvcs).

   ```
   ncn-m001# git push origin master
   ```

3. Restart PBS.

   ```
   ncn-m001# systemctl restart pbs
   ```

4. Verify the setting is in effect by checking that CPU 0 is **not** in the `Cpus_allowed_list`. If it is, the `exclude_cpus` setting was not applied correctly. This example assumes a compute node with two CPUs.

   ```
   uan01# qsub -I -lselect=1
   uan01# module load cray-pals
   uan01# mpiexec grep Cpus_allowed /proc/self/status
   Cpus_allowed: 000000ff
   Cpus_allowed_list: 1
   ```

10.8 PBS Troubleshooting and Administrative Tasks

10.8.1 Check PBS Server or Scheduling Logs

The PBS server and scheduling logs are located in `/var/spool/pbs/server_logs/` and `/var/spool/pbs/sched_logs/`, respectively.

```
ncn-m001# PBS_POD=$(kubectl get pod -n user -l app=pbs -o jsonpath=‘{.items[0].metadata.name}’)
ncn-m001# kubectl exec -it -n user $PBS_POD - /bin/bash
```

10.8.2 PBS Pod Stuck in ContainerCreating State

If the `pbs` Kubernetes pod is stuck in `ContainerCreating` state, this may be due to leftover macvlan IP address reservations. To confirm and resolve this issue, follow these steps:

1. Get the pod name and node.

   ```
   ncn-m001# kubectl get pod -n user -o wide
   ```

2. Check the pod events for an error message like:

   ```
   Warning  FailedCreatePodSandBox  58s (x37 over 9m5s)  kubelet, ncn-w001  (combined from similar
   events): Failed to create pod sandbox: rpc error: code = Unknown desc = failed to setup network
   for sandbox "2d741057d8c40a49f0cae53db4ff5be6217127ef4c25a97166c6401101c990f0": Multus: Err in
   tearing down failed plugins: Multus: error in invoke Delegate add - "macvlan": failed to
   allocate for range 0: no IP addresses available in range set: 10.252.2.2-10.252.2.2
   ```

   ```
   ncn-m001# kubectl describe pod -n user <pod name>
   ```

3. If the above error message appears in the output, `ssh` to the node the pod is running on, and remove the reservation files:

   - For the pbs pod:

     ```
     ncn-w001# rm /var/lib/cni/networks/macvlan-pbs-nmn-conf/*
     ```

   If the error message is something else, refer to the *CSM Administration Guide* for more troubleshooting suggestions.

4. Check that the pod is running; it may take a few minutes to start.

   ```
   ncn-w001# kubectl get pod -n user -o wide
   ```

10.8.3 Recover After UAN Image Corruption During PBS Installation

This recovery procedure is only applicable for installations or upgrades **not** using `sat bootprep` to prepare compute and UAN nodes for bringup.

Follow these step to create a new UAN image when the `.post-bringup-install.sh` script fails with ***UAN image ID to customize is empty Please create new UAN image manually and run $0 -r to resume***.

1. Determine the correct UAN image ID for use in the next step.

   ```
   ncn-m001# cray ims images list
   ```

2. Create a new UAN image.

   ```
   ncn-m001# cray cfs sessions delete "pbs-uan-<PBSBLOB_VERSION>" || true
   ncn-m001# cray cfs sessions create \
   --name "pbs-uan-<PBSBLOB_VERSION>" \
   --configuration-name <UAN_CFS_config> \
   --target-definition image \
   --target-group Application <IMAGE_ID>
   ```

3. After the previous command returns `true`, assign values for `etag` and `path`.

   ```
   ncn-m001# BOS_ETAG=$(cray ims images describe $IMS_ID --format=json | jq -r .link.etag)
   ncn-m001# BOS_PATH=$(cray ims images describe $IMS_ID --format=json | jq -r .link.path)
   ```

4. Restart the post installation script.

   ```
   ncn-m001# ./post-bringup-install.sh -r -c <COS_CFS_config> -s <COS_BOS_template> \
   -n <UAN_CFS_config> -u <UAN_BOS_template>
   ```

   - If this is an initial installation of PBS and the `post-bringup-install.sh` script completes successfully, return to the installation procedure at *Validate the Content Post PBS Installation*; otherwise return to *Run the PBS Bringup Script* and check **TIPS** for other possible suggestions.
   - If this is an update installation of PBS and the `post-bringup-install.sh` script completes successfully, return to the installation procedure at *Validate the Content Post PBS Upgrade*; otherwise return to *Run the Bringup Script for PBS Upgrade* and check **TIPS** for other possible suggestions.

10.8.4 Backup PBS Home Directory

PREREQUISITES

- PBS Pro is the system's workload manager

OBJECTIVE

Prior to upgrading Cray System Management (CSM) or PBS, it is good practice to backup the PBS home directory. As this procedure is distruptive to PBS operations, HPE recommends doing it only during upgrades or maintenance windows.

PROCEDURE

1. Stop the PBS pod.

   ```
   ncn-m001# kubectl scale deployment -n user pbs  --replicas=0
   ```

2. Create a `pbs-backup.yaml` file with the following contents:

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: pbs-backup
     namespace: user
   spec:
     containers:
     - name: pbs-backup
       image: dtr.dev.cray.com/baseos/busybox:1.31.1
       command: ["/bin/sleep", "infinity"]
   ```

```
    volumeMounts:
    - name: pbs-data
      mountPath: /var/spool/pbs
  volumes:
  - name: pbs-data
    persistentVolumeClaim:
      claimName: pbs-data
```

3. Apply the file to start the pbs-backup pod.

   ```
   ncn-m001# kubectl apply -f pbs-backup.yaml
   ```

4. Copy the home directory contents.

   ```
   ncn-m001# kubectl exec -n user pbs-backup -- tar czf - -C /var/spool pbs >pbs_home.tar.gz
   ```

5. Save the archive in S3.

   ```
   ncn-m001# cray artifacts create wlm backups/pbs_home.tar.gz ./pbs_home.tar.gz
   ```

6. Delete the pbs-backup pod.

   ```
   ncn-m001# kubectl delete -f pbs-backup.yaml
   ```

7. Start the pbs pod.

   ```
   ncn-m001# kubectl scale deployment -n user pbs --replicas=1
   ```

10.8.5 Restore PBS Home Directory from Backup

PREREQUISITE

- A previously backed up PBS home directory is archived in S3, (via the *Backup PBS Home Directory* procedure)

OBJECTIVE

In the event that a Cray System Management (CSM) or PBS upgrade causes PBS to fail, it may be possible to restore from a backup.

PROCEDURE

1. Stop the pbs pod.

   ```
   ncn-m001# kubectl scale deployment -n user pbs --replicas=0
   ```

2. Create a pbs-backup.yaml file with the following contents:

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: pbs-backup
     namespace: user
   spec:
     containers:
     - name: pbs-backup
       image: dtr.dev.cray.com/baseos/busybox:1.31.1
       command: ["/bin/sleep", "infinity"]
       volumeMounts:
       - name: pbs-data
         mountPath: /var/spool/pbs
     volumes:
     - name: pbs-data
       persistentVolumeClaim:
         claimName: pbs-data
   ```

3. Apply the file to start the pbs-backup pod.

   ```
   ncn-m001# kubectl apply -f pbs-backup.yaml
   ```

4. Retrieve the backup from S3.

   ```
   ncn-m001# cray artifacts get wlm backups/pbs_home.tar.gz pbs_home.tar.gz
   ```

5. Extract the archive.

   ```
   ncn-m001# tar -xf pbs_home.tar.gz
   ```

6. Copy the backup into place.

   ```
   ncn-m001# kubectl cp pbs user/pbs-backup:/var/spool
   ```

7. Delete the pbs-backup pod.

   ```
   ncn-m001# kubectl delete -f pbs-backup.yaml
   ```

8. Start the pbs pod.

   ```
   ncn-m001# kubectl scale deployment -n user pbs --replicas=1
   ```

10.9 Apply Critical Issue Workarounds

OBJECTIVE: The following workarounds must be applied on systems running CPE 22.04, 22.05, or 22.06 with a WLM configured. The WLM installation and upgrade procedures indicate the point at which these procedures are appropriate; however, they can also be performed on systems running CPE 22.04 or later without doing an upgrade.

After completing these procedures, *Return to the Correct Installation or Upgrade Process* provides links back to the process in progress.

10.9.1 Prevent UAI Network Security Vulnerability

DESCRIPTION:

User Access Instances (UAIs) have access to previously forbidden areas of the node management network (NMN) after configuring them for High Speed Network (HSN) access during Workload Manager (WLM) installation. This problem exists in WLM configurations provided by CPE 22.04, 22.05, and 22.06.

ROOT CAUSE:

Beginning with the CPE 22.04 release, WLM pods moved from the NMN to the HSN. As part of this transition, instructions included in the CPE installation guide remove routes from the UAI network attachment definition. Those routes are used to drop traffic destined for portions of the node management network.

SOLUTION:

To correct this problem before a fix is released, sites using a WLM configuration provided by CPE 22.04, 22.05, or 22.06 must follow this procedure.

PROCEDURE:

1. Choose an unused HSN IP address as a black hole gateway (e.g., `10.253.255.254`).

   TIP: Use `/opt/cray/csm/scripts/networking/DNS/dns_records.py -p` to display a list of currently used addresses.

2. Obtain the `customizations.yaml` file.

   ```
   ncn-m001# kubectl -n loftsman get secret site-init -o \
   jsonpath='{.data.customizations\.yaml}' | base64 -d - > customizations.yaml
   ```

3. Edit the UAI route configuration in `customizations.yaml`.

   a. Change the `gw` values in `spec.wlm.macvlansetup.routes` to the HSN black hole gateway.

   b. Ensure `spec.kubernetes.services.cray-uas-mgr.uasConfig.uai_macvlan_routes` is set to:

      ```
      '{{ wlm.macvlansetup.routes }}'
      ```

4. Edit `net-attach-def`.

   ```
   ncn-m001# kubectl edit net-attach-def -n user macvlan-uas-nmn-conf
   ```

   a. Add the routes from the previous step to the `routes` list, replacing the `gw` setting with the chosen HSN black hole gateway IP address. For example:

```
    routes: [{"dst": "10.106.0.0/17", "gw": "10.253.255.254"},
    { "dst": "10.92.100.0/24", "gw": "10.253.255.254" },
    { "dst": "10.103.3.0/25", "gw": "10.253.255.254" }]
```

5. Restart UAI pods.

```
ncn-m001# kubectl delete pod -n user -l uas=managed
```

6. Validate that UAIs no longer have access to the blocked subnets by attempting to `ping` IP addresses in those subnets from a UAI pod.

7. Save changes to `customization.yaml`.

```
ncn-m001# kubectl delete secret -n loftsman site-init
ncn-m001# kubectl create secret -n loftsman generic site-init --from-file=customizations.yaml
```

10.9.2 Prevent Duplicate Network Packet Transmission

DESCRIPTION:

Network traffic generated by WLM k8 pods on Cray System Management (CSM) based systems causes the transmission of duplicate network packets on Slingshot fabrics. This problem exists in WLM configurations for CPE 22.04, 22.05, and 22.06.

ROOT CAUSE:

Beginning with the CPE 22.04 release, WLM k8 pods switched from Node Management Network (NMN) interfaces to Slingshot HSN interfaces. WLM k8 pod network interfaces are configured using macvlan. The use of macvlan on Slingshot HSN interface causes the WLM k8 containers to use a separate physical NIC MAC address instead of the expected Algorithmic MAC Address (AMA). Not using the node NIC AMA causes an undesirable broadcast of network packets.

SOLUTION:

Switching WLM and UAI k8 pods to use ipvlan to configure network interfaces resolves the problem by allowing the container to use the NIC assigned AMA. This configuration change will be included in the CPE 22.08 release. Customers using a WLM configuration provided by CPE 22.04, 22.05, or 22.06 must follow the procedure below to reconfigure use of ipvlan instead of macvlan to avoid the duplicate network packet transmission problem.

IMPORTANT: This procedure requires the reboot of WLM and UAI pods, resulting in a brief outage during restart operations.

PROCEDURE:

1. Edit `net-attach-def`.

```
ncn-m001# kubectl edit net-attach-def -n user
```

   a. Change all instances of `macvlan` to `ipvlan` and `bridge` to `l2`.

2. Restart WLM pods.

   - For Slurm:

```
ncn-m001# kubectl rollout restart -n user \
deployment/slurmctld deployment/slurmctld-backup \
deployment/slurmdbd deployment/slurmdbd-backup
```

   - For PBS:

```
ncn-m001# kubectl rollout restart -n user deployment/pbs
```

3. Restart UAI pods.

```
ncn-m001# kubectl delete pod -n user -l uas=managed
```

4. Ensure the MAC address in the WLM pod(s) matches the `hsn0` interface on the host NCN (e.g., `02:00:00:00:00:0d`).

```
ncn-m001# kubectl exec -n user <pod> - ip addr show net1{}
```

10.9.3 Return to the Correct Installation or Upgrade Process

Proceed based on which WLM system is used on the system.

- For systems running Slurm:
  - If installing Slurm for the first time, continue on after *Apply Critical Workarounds Relevant to Slurm Installation*.
  - If upgrading Slurm, continue on after *Apply Critical Workarounds Relevant to Slurm Upgrade*.
- For systems running PBS:
  - If installing PBS for the first time, continue on after *Apply Critical Workarounds Relevant to PBS Installation*.
  - If upgrading PBS, continue on after *Apply Critical Workarounds Relevant to PBS Upgrade*.