Hewlett Packard Enterprise

HPE Cray EX Series System Administration with HPE Performance Cluster Manager

Lab exercise customize images for user access

Contents

Customizing

This lab demonstrates how to work with images to add or remove software packages and modify configurations, create custom images, check images into version control, retrieve previous versions, and add software to running nodes.

Work with repositories

1. Review the cm repo man pages as needed throughout the lab exercise.

COMMAND BASE	MANPAGE COMMAND	DESCRIPTION
cm repo add	man cm-repo-add	Adds a repository
cm repo del	man cm-repo-del	Deletes a repository
cm repo group add	man cm-repo-group-add	Adds repositories to or create a repository group
cm repo group del	man cm-repo-group-del	Removes repositories from or delete a repository group
cm repo group show	man cm-repo-group-show	Shows repository groups or details on specific groups
cm repo reexport	man cm-repo-reexport	Re-exports all repositories with yume
cm repo refresh	man cm-repo-refresh	Refreshes repository metadata
cm repo select	man cm-repo-select	Marks a repository as selected
cm repo show	man cm-repo-show	Shows available repositories
cm repo unselect	man cm-repo-unselect	Marks a repository as not selected

1. Check whether the node that your lab group has been assigned is using an NFS rootfs option:

   ```
   cm node show -O -n <node> | cut -b 1-60
   ```

2. If your node is using an NFS rootfs option, set the rootfs to disk or tmpfs:

   ```
   cm node set -n <node> --rootfs <disk|tmpfs>
   ```

2. List the repository top level directories:

   ```
   ls /opt/clmgr/repos/
   ```

3. List cluster manager and related HPE software directories:

   ```
   ls /opt/clmgr/repos/cm
   ```

4. List contents of the distribution repository directory:

   ```
   ls /opt/clmgr/repos/distro/
   ```

5. Add the HPE AIOps repository:

   cm repo add /iso/Q9V62-11133.iso

   Observe the ISO mount, the copy operation start, and the rpmlist updates.

6. List the repositories; the Cluster-Manager-AIOps-1.6-sles15sp3-x86_64 repository is listed but not selected (no * character appears in the first column):

   cm repo show

7. Select the HPE AIOps repository:

   cm repo select Cluster-Manager-AIOps-1.6-sles15sp2-x86_64

8. List the directories under /opt/clmgr/repos/cm.

   ls /opt/clmgr/repos/cm

   The HPE AIOps repository appears under the /opt/clmgr/repos/cm directory.

9. List the repositories:

   cm repo show

10. Unselect the HPE AIOps repository:

    cm repo unselect Cluster-Manager-AIOps-1.6-sles15sp2-x86_64

11. List the repositories:

    cm repo show

    The * character no longer appears to the left of the HPE AIOps repository name, so RPMs in the repository are not available to cluster commands such as cinstallman.

12. Select the HPE AIOps repository:

    cm repo select Cluster-Manager-AIOps-1.6-sles15sp2-x86_64

Customize an image using RPM lists

1. On the admin node, create the repository for custom packages not in the OS distribution or HPE distribution.

   mkdir /opt/clmgr/repos/mypkgs

2. Prepare RPMs that correspond with your distribution for the mypkgs directory.

   cp /iso/mypkgs/* /opt/clmgr/repos/mypkgs

3. Verify that the RPMs copied successfully:

   ls -R /opt/clmgr/repos/mypkgs

4. Update the cluster software repository metadata with:

   cm repo add /opt/clmgr/repos/mypkgs --custom mypkgs

   cm repo show

   cm repo select mypkgs

   cm repo show

5. Ensure that an asterisk * character appears in front of the mypkgs repository.

Create a compute node image

3. On the admin node, make a copy of the example compute node image rpm list and edit the copy (replace <my-code> with your initials or a short code that uniquely identifies you).

   ```
   cd /opt/clmgr/image/rpmlists/generated

   ls

   cp generated-sles15sp3.rpmlist /class/<my-code>/<my-code>.rpmlist

   ls -l /class/<my-code>/<my-code>.rpmlist
   ```

   Note: Ensure that you do not use the `generated-admin-<distro>.rpmlist;` it contains packages for the admin node rather than the packages required for a compute (non-ICE, flat, service) node.

4. Edit the <my-code>.rpmlist file to add the following line at the end of the /class/<my-code>/<my-code>.rpmlist file:

   ```
   kernel-docs
   ```

   Run the cm image create command to create a new image. This command runs approximately 10 minutes, produces lots of output, and wraps onto a second line; the second option, -l for list is a lowercase letter l, not the digit 1 (replace <my-code> with your initials or a short code that uniquely identifies you—use the same entry that you used earlier).

   ```
   cm image create -i <my-code> -l /class/<my-code>/<my-code>.rpmlist
   ```

5. While cm image create is running, type the following command in another window on the admin node:

   ```
   tail -f /var/log/cinstallman
   ```

   If there are problems, compare differences between the generated and <my-code> rpmlists:

   ```
   diff generated-sles15sp3.rpmlist /class/<my-code>/<my-code>.rpmlist
   ```

   Note: The differences include the packages that you added to the rpmlist, any required dependencies, and default packages for the HPE AIOps repository.

6. Review the image version history:

   ```
   cm image revision history -i <my-code>
   ```

7. When you create an image by using the software repositories and an rpmlist, you can make site-specific changes such as copying custom files and scripts into the image you created then commit to revision control. For this lab step, use the echo command to create a file in the <my-code> image:

   ```
   echo "custom file for <my-code>" > /opt/clmgr/image/images/<my-code>/root/my-custom.txt
   ```

8. Confirm that you created the file:

   ```
   cat /opt/clmgr/image/images/<my-code>/root/my-custom.txt
   ```

9. Show the cluster domain:

   cadmin --show-cluster-domain

10. Edit the file /etc/idmapd.conf to set the cluster domain.
11. Copy the file /etc/idmapd.conf into the image:

    cp /etc/idmapd.conf /opt/clmgr/image/images/<my-code>/etc

12. List the <my-code> image details, including kernels:

    cm image show -d -i <my-code>

13. Change to root's home directory.

    cd

14. Commit the image into version control:

    cm image revision commit -i <my-code> -m "<image>v2 custom file, idmapd.conf"

15. Show image version history:

    cm image revision history -i <my-code>

16. Open terminal sessions to the admin node as needed.
17. Connect to the console of a compute node (replace # with the number associated with the node name):

    console <node>

18. Arrange these sessions so that you can monitor node provisioning:
19. Provision the compute nodes with the <my-code> image:

    cm node provision -n "<node>" -i <my-code>

    IMPORTANT: Monitor node provisioning. If one or more nodes stops in grub or fails to install a new image, power the node(s) off, reissue the ilorest command for one time PXE boot, and power on the node. Repeat as necessary.

    cm power off -t node <node>
    grep <node> /etc/hosts
    ilorest bootorder --onetimeboot=pxe -u ADMIN -p ADMIN --url 172.24.0.<#> --commit
    cm power on -t node <node>

    Other options to force the PXE boot include setting a next boot option with efibootmgr on a booted node or triggering Esc@ during BIOS initialization.

    On the admin node, confirm that the nodes completed imaging successfully by checking for your custom file and the change that you made to the idmapd.conf file:

    ssh <node> cat /root/my-custom.txt

    ssh <node> head -6 /etc/idmapd.conf

    ssh <node> cat /etc/opt/sgi/vcs-log-entry

    Look for your cm image revision commit log entry.

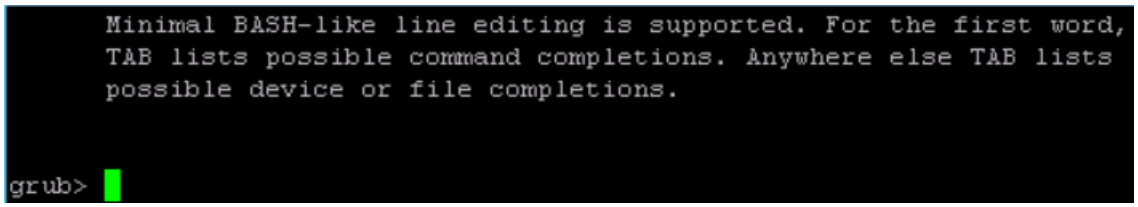
```
ssh <node> cat /proc/cmdline
```

Look for the value of IMAGE.

20. Troubleshoot nodes that failed provisioning by connecting to the node's console:
    a. Review node console output

    If the node has booted to operating system, run the efibootmgr command to check the current boot setting—did the node PXE boot? If not, configure the node to PXE boot with efibootmgr, ilorest, or by triggering Esc @ as the node moves through BIOS.

    If a node stops in grub, change the dhcp_bootfile attribute to ipxe-direct with cm node set:



    b. On the admin node, issue a power reset to the node (replace <node> with the name of the node; do not include the < and > characters).

```
cm power reset -t node <node>
```

    Monitor the node boot. If necessary, issue the cm power reset command a second or third time.

21. After the node completes imaging (installation), detach from the console command:

```
Ctrl+e c .
```

Capture an image from a node

1. In a terminal session connected with the admin node, login to<node>.

```
ssh  root@<node>
```

2. Enable and start a service:

```
systemctl -l status sysstat

systemctl enable sysstat

systemctl start sysstat

systemctl -l status sysstat
```

One of the /etc/opt/sgi/conf.d scripts disables sysstat when enabled; you will confirm that this service does not stay enabled.

The script, /etc/opt/sgi/conf.d/80-service-distro-services, contains these comments:

systat makes files in /var/log/sa that in some cases are 20M every several
minutes. 20M * 1408 nodes is 28.16 gigabytes for writable NFS support. You
may enable this by changing the below line to Enable (but the changes may be
lost in package upgrades), excluding this conf.d script, or making a new
conf.d script that runs after this one to override this setting. The changes

would be made in the image itself. The latter method is preferred.
initDisableServiceIfExists sysstat

3. Create a file:

```
echo "for image capture" > /root/image-capture
cat /root/image-capture
```

4. Exit from<node>:

```
exit
```

5. Review which repositories are selected to ensure that the selected repositories correspond with (support) the image you are capturing:

```
cm repo show
```

6. Capture an image from a running node.

```
cm image capture -i <my-code>-two -n <node>
```

7. Show the new image.

```
cm image show
```

8. Commit the captured image into the version control system:

```
cm image revision commit -i <my-code>-two -m "<image-name>v# image capture"
```

6. Provision <node> with the captured image.

```
cm node provision -i <my-code>-two -n <node>
```

IMPORTANT: Monitor node provisioning. If one or more nodes stops in grub or fails to install a new image, power the node(s) off, reissue the ilorest command for one time PXE boot, and power on the node. Repeat as necessary.

cm power off -t node <node>
grep <node> /etc/hosts
ilorest bootorder --onetimeboot=pxe -u ADMIN -p ADMIN --url 172.24.0.<#> --commit
cm power on -t node <node>

Other options to force the PXE boot include setting a next boot option with efibootmgr on a booted node or triggering Esc@ during BIOS initialization.

7. Monitor node imaging.
8. After <node> completes boot of the operating system, confirm that <node> contains the changes

```
ssh <node>
systemctl -l status sysstat
```

One of the /etc/opt/sgi/conf.d scripts disables sysstat when enabled; here, confirm that this service did not stay enabled.

```
cat /root/image-capture

cat /etc/opt/sgi/vcs-log-entry

exit
```

9. List the image associated with each node:

```
cm node show -I | cut -c 1-60
```

Add RPMs to, remove RPMs from, and version control compute images

1. In a terminal session connected to the admin node, remove a package from an image.

```
cm image zypper -i <my-code>-two remove xfsprogs
```

2. Place the modified image under version control:

```
cm image revision commit -i <my-code>-two -m "<image>v2 removed rpms"
```

3. Add RPMs to the image:

```
cm image zypper -i <my-code>-two install openldap2
```

4. Place the modified image under version control, then work with additional version control operations.

```
cm image revision commit -i <my-code>-two -m "<image>v3 installed
openldap2"
```

5. Review image revision history.

```
cm image revision history -i <my-code>-two
```

6. Specify showing image history between 2 revisions.

```
cm image revision history -i <my-code>-two --rev 2..3
```

7. The next command will compare differences between two images at specified revisions; review this excerpt from the rsync man page --itemize-change options that describes the 11 character string used to show file differences:

```
The general format is like the string YXcstpoguax, where Y is replaced by
the type of update being done, X is replaced by the file-type, and the other
letters represent attributes that may be output if they are being modified.

The update types that replace the Y are as follows:

o  A < means that a file is being transferred to the remote host (sent).

o  A > means that a file is being transferred to the local host (received).

o  A c means that a local change/creation is occurring for the item (such as
the creation of a directory or the changing of a symlink, etc.).

o  A h means that the item is a hard link to another item (requires --hard-
links).
```

o A . means that the item is not being updated (though it might have attributes that are being modified).

o A * means that the rest of the itemized-output area contains a message (e.g. "deleting").

The file-types that replace the X are: f for a file, a d for a directory, an L for a symlink, a D for a device, and a S for a special file (e.g. named sockets and fifos).

The other letters in the string above are the actual letters that will be output if the associated attribute for the item is being updated or a "." for no change. Three exceptions to this are: (1) a newly created item replaces each letter with a "+", (2) an identical item replaces the dots with spaces, and (3) an unknown attribute replaces each letter with a "?" (this can happen when talking to an older rsync).

The attribute that is associated with each letter is as follows:

o A c means either that a regular file has a different checksum (requires --checksum) or that a symlink, device, or special file has a changed value.
Note that if you are sending files to an rsync prior to 3.0.1, this change flag will be present only for checksum-differing regular files.

o A s means the size of a regular file is different and will be updated by the file transfer.

o A t means the modification time is different and is being updated to the sender's value (requires --times). An alternate value of T means that the modification time will be set to the transfer time, which happens when a file/symlink/device is updated without --times and when a symlink is changed and the receiver can't set its time. (Note: when using an rsync 3.0.0 client, you might see the s flag combined with t instead of the proper T flag

o A p means the permissions are different and are being updated to the sender's value (requires --perms).
o An o means the owner is different and is being updated to the sender's value (requires --owner and super-user privileges).
o A g means the group is different and is being updated to the sender's value (requires --group and the authority to set the group).
o The u slot is reserved for future use.
o The a means that the ACL information changed.
o The x means that the extended attribute information changed.

10. List the differences between the sles15sp3 and <my-code>-two images.

    cm image revision diff -i <my-code>-two --rev 3 -s sles15sp3

11. Review the revision differences between an image and a running node.

    cm image revision diff -i <my-code>-two --rev 3 -n<node>

12. Compare what changed between two revisions (the command wraps to a second line):

    cm image revision diff -i <my-code> --rev 2 -s <my-code> --source-rev 1

13. Show image revision history.

    cm image revision history -i <my-code>-two

14. List cm image revision revert help.

    cm image revision revert -h

15. Revert an image to an earlier revision.

    cm image revision revert -i <my-code>-two --rev 2

16. Show image revision history.

    cm image revision history -i <my-code>-two

17. Revert an image to a newer revision.

    cm image revision revert -i <my-code>-two --rev 3

18. Show image revision delete help.

    cm image revision delete -h

19. Delete stored revisions for an image.

    cm image revision delete -i <my-code>-two

20. Confirm that the deleted revisions no longer appear in image revision history.

    cm image revision history -i <my-code>-two

Add and remove RPMs from running compute nodes

1. On the admin node, ensure that the repositories that support software installation and remove operations are selected.

   cm repo show | grep ^*

2. Check for the installed status of 2 RPMs.

   ssh <node> rpm -q xfsprogs openldap2

3. Remove a package from a running node.

   cm node zypper -n<node> remove xfsprogs

4. Install the RPM on the running node:

   cm node zypper -n<node> install openldap2

5. Confirm that xfs is not installed and openlap-clients is installed:

   ```
   ssh <node> rpm -q xfsprogs openldap2
   ```

6. On the admin node, ensure that the repositories that support software installation and remove operations are selected.

   ```
   cm repo show | grep ^*
   ```

7. Ensuring that the pertinent repositories are selected or specified on the command line is crucial to installation, update, and removal operations. Unselect the operating system distribution repository to observe that the generated rpmlists are removed.

   ```
   cm repo unselect SLE-15-SP3-Full-x86_64
   ```

8. Confirm that the operating system distribution repository is not selected.
9. Confirm that the xfsprogs package is not installed on<node>:
10. Without the operating distribution selected or provided on the command line, issue a command to install software and observe the failure message.

    ```
    cm node zypper -n <node> install xfsprogs
    ```

11. Generated rpmlists change as you select and unselect repositories. Select the operating system distribution repository to observe that the generated rpmlists are recreated.

    ```
    cm repo select SLE-15-SP3-Full-x86_64
    cm repo show
    ```

Refresh an image with new packages

1. Edit the /class/<my-code>/<my-code>.rpmlist file to add the following line at the end of <my-code>.rpmlist file:

   ```
   munge
   ```

2. Refresh the image (the -l option is the lowercase letter l rather than the digit 1):

   ```
   cm image refresh -i <my-code> -l /class/<my-code>/<my-code>.rpmlist
   ```

3. Commit image to revision control (the command below wraps to a second line—enter it on one line).

   ```
   cm image revision commit -i <my-code> -m "installed munge"
   ```

4. Confirm that munge is not installed on the running node:

   ```
   ssh <node> rpm -q munge
   ```

5. Test refreshed image on node.

   ```
   cm node provision -i <my-code> -n <node>
   ```

6. After you confirm that the modified image works, provision the other compute nodes with the image:

```
cm node refresh rpms -n <node> -l /class/<my-code>/<my-code>.rpmlist
```

7. Confirm that all nodes have the RPMs installed:

```
pdsh -g compute rpm -q munge | dshbak -c
```

Update the kernel

1. Copy update kernel RPM directory from server (guest account password: cmdefault ; the directory contains 2 RPMs; the command wraps to a second line—do not enter the \ character):

   ```
   scp -pr student@192.168.235.97:/iso/sle15sp3/kernel-update \
   /opt/clmgr/repos
   ```

   Example output:

   ```
   admin# scp -pr student@192.168.235.97:/iso/sle15sp3/kernel-update /opt/clmgr/repos
   student@192.168.235.97's password: cmdefault
   kernel-default-5.3.18-59.24.1.x86_64.rpm              100%   64MB  73.9MB/s   00:00
   kernel-default-base-5.3.18-59.24.1.18.12.1.x86_64.rpm 100%   30MB  96.9MB/s   00:00
   ```

2. Confirm that the files are present.

   ```
   ls -l /opt/clmgr/repos/kernel-update
   ```

   Example output:

   ```
   # ls -l /opt/clmgr/repos/kernel-update
   total 96824
   -rw-r--r-- 1 root root 67452152 Dec 22 07:31 kernel-default-5.3.18-
   59.24.1.x86_64.rpm
   -rw-r--r-- 1 root root 31693732 Dec 22 07:31 kernel-default-base-5.3.18-
   59.24.1.18.12.1.x86_64.rpm
   ```

3. Add the kernel-update repository; the command wraps to a second line—do not enter the \ character:

   ```
   cm repo add /opt/clmgr/repos/kernel-update --custom \
   kernel-5.3.18-24.61.1
   ```

4. Commit the current working image to revision control before you perform the update operation; the command wraps to a second line—do not enter the \ character).

   ```
   cm image revision commit -i <my-code> -m "<my-code>v4 before \
   kernel update 5.3.18-24.61.1"
   ```

5. Show the added repository in the list of repositories; notice that no asterisk * character appears to the left of the added repository.

   ```
   cm repo show
   ```

6. Select the kernel update repository:

   ```
   cm repo select kernel-5.3.18-24.61.1
   ```

7. Show the list of repositories; the asterisk * character appears to the left of the selected kernel-187 repository.

   ```
   cm repo show
   ```

8. Update the image with the update kernel:

   ```
   cm image zypper -i <my-code> update kernel-default
   ```

9. Commit the updated image to revision control; the command wraps to a second line—do not enter the \ character).

   ```
   cm image revision commit -i <my-code> -m "<my-code> v5 kernel \
   updated to 5.3.18-24.61.1"
   ```

10. Show the image details that include the installed kernels:

    ```
    cm image show -di <my-code>
    ```

11. Test the new kernel: install one node with the kernel:

    ```
    cm node provision -n<node> -i <my-code> -k <version-default>
    ```

12. Confirm the node booted the update kernel:

    ```
    ssh <node> uname -r
    ```

13. Print the vcs-log-entry file for the node:

    ```
    ssh <node> cat /etc/opt/sgi/vcs-log-entry
    ```

    Since the image was not committed to revision history after the kernel update change, the vcs-log-entry shows the commit message from before the update; the provision operation uses the image in the working directory—in this case with the update kernel.

Repository groups

1. Review manpages:

   ```
   man cm-repo-group-show
   ```

   ```
   man cm-repo-group-add
   ```

   ```
   man cm-repo-group-del
   ```

2. Create your own repository group.

Post install and conf.d scripts

Custom configuration changes applied manually may have to be reapplied multiple times. Some tasks can be automated through local scripts on the admin node for repeated use on one or more cluster nodes. Create post installation (post install) scripts for compute (non-ICE, flat, service) and lead nodes. Create per-host-customization scripts for ICE compute nodes.

Note: You can customize a cluster without using scripts by using the methods in the previous module.

Post-installation customization scripts

Build customization scripts to see their impact when you image the lead and compute node:

1. Log in to the admin node.

   cd /class/<my-code>

2. On the admin node, copy scripts:

   cp /iso/scripts/* /class/<my-code>

3. Change the name of the scripts:

   mv 97compute.create-myfile 97<node>.create-<my-code>

   mv 98all.motd 98all.motd-<my-code>

4. Review the 97<node>.create-myfile script that applies to all compute nodes with names that start with base "n". The script creates the file /tmp/myfile.

   cat 97<node>.create-<my-code>

5. Review the 98all.motd-mod script that applies to all compute nodes and all leader nodes. The 98all.motd-mod script appends a time and date stamp to the file /etc/motd.

   cat 98all.motd-<my-code>

6. Edit the echo statement in the 98all.motd-<my-code> file to include your initials.

7. Review the post-install README:

   less /opt/clmgr/image/scripts/post-install/README

8. On the admin node, place the 97<node>.create-<my-code> and 98all.motd-<my-code> customization scripts in the post-install directory:

   cd /class/<my-code>

   cp 97* 98* /opt/clmgr/image/scripts/post-install/

   cd (return to root's home directory)

9. Coordinate with your lab partners to reinstall the compute node.

   cm node provision -n <node>

   Ensure that the node PXE boots and provisions.

10. When the node completes installation, verify the changes that the scripts made.

    ssh <node> cat /etc/motd

    ssh <node> cat /tmp/myfile

11. On the admin node, grep the console logs for the entries:

    grep --text -A2 97<node> /var/log/consoles/<node>

    grep --text -A2 98all /var/log/consoles/<node>

Conf.d scripts

1. Investigate the conf.d files on the admin node.

   ls -l /etc/opt/sgi/conf.d

2. Review the contents of the exclude file.

   cat /etc/opt/sgi/conf.d/exclude

3. Review one of the conf.d scripts.

cat /opt/clmgr/image/images/<my-code>/etc/opt/sgi/conf.d/80-limits-core-files

4. List conf.d files within an image (replace <image> with the name of an image on the admin node.

   ls -l /opt/clmgr/image/images/<my-code>/etc/opt/sgi/conf.d/

5. Log in to one of the compute nodes (root account password: cmdefault).

   ssh <node>

6. Browse the conf.d files on the running node.

   ls -l /etc/opt/sgi/conf.d

7. Log out of the compute node.

   exit

8. Skim the “Configuration manager framework” section in the HPE Performance Cluster Manager Administration Guide.

Preserving custom configuration changes

To prevent the update-configs command from overwriting configuration files you want to preserve, you can protect those files by adding the desired files or directories to file /etc/opt/sgi/conf.d/exclude-update-configs on the node itself or in an image on the admin node. The format of the file is one filename or directory name per line. Commented lines are ignored.