

1 BareMetalPrep

This guide uses diskimage-builder utility to build and configure OS images for sms node as well compute node. Preparing images is an optional part of overall recipe. If user has predefined images then environment variable `chpc_create_new_image` must be reset and path to images must be provided using environment variable `chpc_image_deploy_kernel`, `chpc_image_deploy_ramdisk`, `chpc_image_user`, and `chpc_image_sms`. In this example cloud images are build on controller node `[ctrlr]#` " Once images are build, they are stored at standard openHPC Path.

```
[ctrlr]# CHPC_CLOUD_IMAGE_PATH=/opt/ohpc/admin/images/cloud/
```

1.1 Install and Setup disimage-builder

Images can be built on any supported OS.

In this example we will install and build images on controller node, user can do same on a system independent of their production cluster.

Install diskimage-builder and its dependencies from base OS distro

```
[ctrlr]# yum y install diskimage-builder PyYAML
```

Install grub dependency

```
[ctrlr]# yum y install parted
```

Diskimage-builder installed from base distro does not have group install feature. So add a patch (Note: this probably will be included into RPM and will be part of that rpm installation)

```
[ctrlr]# yum y install <DIB patch>
```

Setup common environment for diskimage-builder

diskimage-builder, or dib, uses environment variables and elements to customize the images. For debugging purpose, we will create default user `chpc` with a password `intel8086`, with sudo privilege. These variables are used by element `devuser`.

```
[ctrlr]# export DIB_DEV_USER_USERNAME=chpc
```

```
[ctrlr]# export DIB_DEV_USER_PASSWORD=intel8086
```

```
[ctrlr]# export DIB_DEV_USER_PWDLESS_SUDO=1
```

Now add path to custom elements which are not part of base diskimage-builder. OpenHPC provides few HPC elements. [Note: This also can be part of openHPC provided rpm for dib. In that case remove this step]

```
[ctrlr]# export ELEMENTS_PATH="$(realpath ../../dib/hpc/elements)"
```

Add path to HPC specific files [note: same as earlier, this too can be part of rpm package]

```
[ctrlr]# export DIB_HPC_FILE_PATH="$(realpath ../../dib/hpc/hpc-files/)"
```

HPC elements are common for OpenHPC and Intel HPC Orchestrator, environment variable DIB_HPC_BASE tell dib which one to pick. For OpenHPC set environment variable

```
[ctrlr]# export DIB_HPC_BASE=ohpc
```

Make sure open hpc packages is installed. ohpc_pkg is one of the input setup earlier in this document.

```
[ctrlr]# yum -y install ${ohpc_pkg}
```

Export same to DIB.

```
[ctrlr]# export DIB_HPC_OHPC_PKG=${ohpc_pkg}
```

Create list of HPC elements needed to build HPC images, by starting hpc-env-base. This element will setup basic hpc environment to build hpc images.

```
[ctrlr]# DIB_HPC_ELEMENTS="hpc-env-base"
```

Preparing ironic deploy images

Ironic uses deploy images (aka kernel) to bootstrap the provisioning of user images.

Unset any previous environment flag

```
[ctrlr]# unset DIB_YUM_REPO_CONF
```

Git is used by some of the elements in diskimage-builder.

```
[ctrlr]# yum -y install git
```

Create deploy images using disk-image-create cli. This will download base centos image from distro, install ironic-agent on it and create kernel and initramfs images.

```
[ctrlr]# disk-image-create ironic-agent centos7 -o icloud-hpc-deploy-c7
```

Move deploy images to ohpc standard path.

```
[ctrlr]# chpc_image_deploy_kernel="$( realpath icloud-hpc-deploy-c7.kernel)"
```

```
[ctrlr]# chpc_image_deploy_ramdisk="$( realpath icloud-hpc-deploy-c7.initramfs)"
```

#Store Images file

```
[ctrlr]# mkdir -p $CHPC_CLOUD_IMAGE_PATH/
```

```
[ctrlr]# sudo mv -f $chpc_image_deploy_kernel $CHPC_CLOUD_IMAGE_PATH/
```

```
[ctrlr]# chpc_image_deploy_kernel=$CHPC_CLOUD_IMAGE_PATH/$(basename  
$chpc_image_deploy_kernel)
```

```
[ctrlr]# sudo mv -f $chpc_image_deploy_ramdisk $CHPC_CLOUD_IMAGE_PATH/
```

```
[ctrlr]# chpc_image_deploy_ramdisk=$CHPC_CLOUD_IMAGE_PATH/$(basename  
$chpc_image_deploy_ramdisk)
```

Preparing user images for bare metal instances

For HPC as a Service we will building 2 user images (1 for sms node and 1 for compute node) and 2 deploy images. User images we build here will be customized with OpenHPC using HPC specific elements.

Preparing user Image for head node OS

To build head node (aka sms) images, we need to install server packages of HPC components. This is accomplished by setting image type to sms. Default image type in hpc elements is compute.

```
[ctrlr]# export DIB_HPC_IMAGE_TYPE=sms
```

Now enable SLURM resource manager for head node.

```
[ctrlr]# DIB_HPC_ELEMENTS+=" hpc-slurm"
```

Add optional OpenHPC Components

```
[ctrlr]# if [[ ${enable_mrsh} -eq 1 ]];then
```

```
[ctrlr]# DIB_HPC_ELEMENTS+=" hpc-mrsh"
```

```
[ctrlr]# fi
```

We will also setup HPC development environment on HPC head node.

Enable gnu compiler

```
[ctrlr]# export DIB_HPC_COMPILER="gnu"
```

Enable openmpi & mvapich2

```
[ctrlr]# export DIB_HPC_MPI="openmpi mvapich2"
```

Enable performance tools

```
[ctrlr]# export DIB_HPC_PERF_TOOLS="perf-tools"
```

Enable 3rd party libraries serial-libs, parallel-libs, io-libs, python-libs and runtimes

```
[ctrlr]# export DIB_HPC_3RD_LIBS="serial-libs parallel-libs io-libs  
python-libs runtimes"
```

Add hpc development environment element to list of elements

```
[ctrlr]# DIB_HPC_ELEMENTS+=" hpc-dev-env"
```

Now create a sms image with element local-config, dhcp-all-interfaces, devuser, selinux-permissive and all hpc specific elements. Element local-config copies your local environment into image, which is the local users, their password and permissions. Element devuser will create new user specified by environment variable DIB_DEV_USER_USERNAME.

```
[ctrlr]# disk-image-create centos7 vm local-config dhcp-all-interfaces
devuser selinux-permissive $DIB_HPC_ELEMENTS -o icloud-hpc-cent7-sms
```

It will take a while to build an image. Once image is built copy it to standard openHPC path.

```
[ctrlr]# chpc_image_sms="$(realpath icloud-hpc-cent7.qcow2)"

[ctrlr]# mkdir -p $CHPC_CLOUD_IMAGE_PATH

[ctrlr]# mv -f $chpc_image_sms $CHPC_CLOUD_IMAGE_PATH

[ctrlr]# chpc_image_sms=$CHPC_CLOUD_IMAGE_PATH/$(basename $chpc_image_sms)
```

Preparing user image for compute node OS

To build compute node images, we need to install client packages of HPC components. This is accomplished by setting image type to compute. Default image type in hpc elements is compute.

```
[ctrlr]# export DIB_HPC_IMAGE_TYPE=compute
```

Now enable SLURM resource manager for compute node.

```
[ctrlr]# DIB_HPC_ELEMENTS+=" hpc-slurm"
```

Add optional OpenHPC Components

```
[ctrlr]# if [[ ${enable_mrsh} -eq 1 ]];then

[ctrlr]# DIB_HPC_ELEMENTS+=" hpc-mrsh"

[ctrlr]# fi
```

Now create a compute node image with element local-config, dhcp-all-interfaces, devuser, selinux-permissive and all hpc specific elements. Element local-config copies your local environment into image, which is the local users, their password and permissions. Element devuser will create new user specified by environment variable DIB_DEV_USER_USERNAME.

```
[ctrlr]# disk-image-create centos7 vm local-config dhcp-all-interfaces
devuser selinux-permissive $DIB_HPC_ELEMENTS -o icloud-hpc-cent7-sms
```

It will take a while to build an image. Once image is built copy it to standard OpenHPC path.

```
[ctrlr]# chpc_image_sms="$( realpath icloud-hpc-cent7.qcow2) "
[ctrlr]# mkdir -p $CHPC_CLOUD_IMAGE_PATH
[ctrlr]# mv -f $chpc_image_user$CHPC_CLOUD_IMAGE_PATH
[ctrlr]# chpc_image_user=$CHPC_CLOUD_IMAGE_PATH/$(basename $chpc_image_sms)
```

1.2 Introduction to diskimage-builder

It is a utility to build and configure OS images for sms node as well compute node. It uses prebuild minimum OS images from base distro, which it further customizes as per user request. Diskimage-builder is a framework which uses many elements (similar to plug-ins) to customize the image. Base distribution of diskimage-builder comes with pre-defined elements. This recipe uses additional HPC elements which were built to customize images based on OpenHPC components.

HPC Elements to build OpenHPC Images

HPC as a service uses 4 HPC specific elements in addition to pre-packaged elements comes with diskimage-builder. Hpc-dev-env: This is mainly used to create sms images to create HPC development environment. It creates hpc development environment by installing following OpenHPC components within image: ohpc-autotools, valgrind-ohpc, easybuild-ohpc, spack-ohpc, R_base-ohpc mpi and compiler for chosen MPI & compiler via environment variable, \$DIB_HPC_COMPILER, DIB_HPC_MPI Performance Tools lmod-default with their 3rd party libraries. Hpc-env-base: Hpc-mrsh Hpc-slurm Editing HPC Elements