HP Cloud OS for Moonshot Building Images

Version 1

HP Cloud OS for Moonshot: Building Images

You might have a need to create customized images suitable to your environment; Disk Image Builder is a tool that can help you do just that. This document shows you how to use Disk Image Builder to create images for the HP Cloud OS for Moonshot.

- Supported operating systems
- About Disk Image Builder
- Creating a disk image
- Using your disk image with OpenStack
- Creating an image for a specific workload

Supported operating systems

The following operating systems are supported for image building:

- Red Hat Enterprise Linux version 6.4 and above
- Ubuntu version 12.04 LTS and above

About Disk Image Builder

Disk Image Builder is the tool that can be used for building and customizing images for Nova-Baremetal deployment. It helps in building disk images and PXE images for use with HP Cloud OS for Moonshot. The disk images are a thin version of images created compared to other disk image building tools.

Disk Image Builder creates images based on different elements defined in its element directory. These scripts are bundled in a specific pattern which provides the core functionality of creating cloud images. It provides elements to create a basic Virtual Machine of either Ubuntu or Fedora. We can also create our own Virtual Machine images by including/writing new elements and putting them under the elements directory. One such example is a LAMP image which provides a cloud image of LAMP server to use in OpenStack.

Disk Image Builder creates images of format QCOW2, by default. This image format can be added to Glance and used with OpenStack using the `glance-create` command.

Prerequisites

Before you attempt to build an image, make sure you have:

- 4 GB physical RAM.
- Enough space in /tmp to hold two uncompressed cloud images. One uncompressed image is up to 800 MB in size.
- (Optional) Proxy settings for Internet connectivity.
- qemu-utils and git-core package, VLAN, busybox, open-iscsi, python-lxml, python-libvirt, libvirt, qemu-system installed on the server.

Creating a disk image

To create a disk image using the Disk Image Builder tool, follow the steps below:

1. As a root user, clone the git repository of a Disk Image Builder to the server:

   ```
   git clone https://github.com/openstack/diskimage-builder.git
   ```

   This creates a directory structure of /root/diskimage-builder, complete with its binary files, library file, elements, etc.

2. Edit your **~/.profile** file to add the directory structure /root/diskimage-builder/bin to the path in your shell:

   ```
   PATH="$HOME/bin:$PATH:/root/diskimage-builder/bin"
   ```

3. Create a disk image in one of the following ways:

   - **Ubuntu image**

     To create a basic Ubuntu VM image with the associated kernel and ramdisk image, use the following command:

     ```
     disk-image-create -o base -a amd64 vm base ubuntu baremetal
     ```

     This creates a disk image file named base.qcow2

   - **PXE image**

     To create a PXE image, use the following command:

     ```
     ramdisk-image-create -a amd64 – pxe deploy
     ```

     This creates two files: pxe.initramfs and pxe.kernel

   -

RHEL image

To create a RHEL VM image with associated kernel and ramdisk images, use the following command:

```
disk-image-create –o <image_prefix> –a amd64 –u base rhel baremetal
```

This creates three files: <image_prefix>.qcow2, <image_prefix>.vmlinux, and <image_prefix>.kernel

Using your disk image with OpenStack

Now that you have created your image, you can add it to Glance and manage it with OpenStack.

1. Copy the image file base.qcow2, and associated kernel and ramdisk images, to the OpenStack controller node using the following commands:

   ```
   KERNEL_ID=`glance image-create --name="demo_image-kernel" --public --disk-fo
   INITRD_ID=`glance image-create --name="demo_image-ramdisk" --public --disk-f
   ```

2. Use the `glance-create` command to add the image to the Glance repository:

   ```
   glance image-create --name="demo_image" --disk-format=qcow2 --property kerne
   ```

3. Verify that your image was added:

   ```
   glance image-list
   ```

4. From the UI or command line, create a key-pair:

   ```
   nova keypair-add demokey > demokey.pem
   ```

5. Create a VM using `demo_image` and `demokey`:

   ```
   nova boot --image demo_image flavor m1.medium -key_name demokey.pem demo_vm1
   ```

6. Once the VM is booted and gets an IP assigned, SSH to it using the key as follows:

   ```
   ssh -i demokey.pem <IP_of_demo_vm1>
   ```

You can now access and manage your VM image from the HP Cloud OS for Moonshot Administration Dashboard. For detailed instructions, see the **Manage Images** topic of the Administration Dashboard help (PDF).

Creating an image for a specific workload

You might want to create an image for a specific workload, such as an Internet server. To do this, you must have an element with appropriate hooks written for that specific workload. The rest of this section shows you how to create Ubuntu and RHEL images for specific workloads, and their associated elements.

Note: For detailed instructions on how to write elements, refer to the OpenStack Disk Image Builder documentation.

Creating an Ubuntu image for a specific workload

To create an Ubuntu image for a specific workload, you first need to create the nginx element.

Creating a nginx element

1. From a command prompt, create a directory structure of **elements/nginx** using following command:

   ```
   mkdir –p ~/elements/nginx
   ```

   Note: The tilde (~) represents your home directory.

2. Change to this newly created directory:

   ```
   cd ~/elements/nginx
   ```

3. From here, create another directory named **install.d**, which will contain the installation hooks:

   ```
   mkdir install.d
   ```

4. In install.d, create a file named **15-nginx** and change its execute permissions:

   ```
   touch 15-nginxchmod a+x 15-nginx
   ```

5. Using an editor, such as vi, add the following lines to the 15-nginx file:

   ```
   !/bin/bash
   set -euxinstall-packages nginx
   ```

Creating an image using nginx with Ubuntu

To use your new nginx element with Ubuntu to create a VM image that is pre-installed with associated kernel and ramdisk images, use the following command:

```
disk-image-create -o base -a amd64 vm base baremetal nginx
```

This creates a VM image file named base.qcow2. You can now upload and work with this image in HP Cloud OS for Moonshot. For detailed instructions, see the **Manage Images** topic of the Administration Dashboard help (PDF).

Creating a RHEL image for a specific workload

To create a RHEL image for a specific workload, you first need to create the mongodb element.

Creating a mongodb element

1. From a command prompt, create the directory structure **elements/mongodb** using following command:

   ```
   mkdir –p ~/elements/mongodb
   ```

   Note: The tilde (~) represents the path to your home directory.

2. Change to this newly created directory:

   ```
   cd ~/elements/mongodb
   ```

3. From here, create another directory named **pre-install.d**, which will contain the pre-installation hooks to setup yum repositories:

   ```
   mkdir pre-install.d
   ```

4. In pre-install.d, create a file named **10-mongodb** and change its execute permissions:

   ```
   touch 10-mongodbchmod a+x 10-mongodb
   ```

5. Using an editor, such as vi, add the following lines to the 10-mongodb file:

   ```
   !/bin/bash
   set -eux
   echo "[mongodb]" > /etc/yum.repos.d/mongodb.repo
   echo "name=mongodb_repo" >> /etc/yum.repos.d/mongodb.repo
   echo "baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64/" >>
   echo "enabled=1" >> /etc/yum.repos.d/mongodb.repo
   echo "gpgcheck=0" >> /etc/yum.repos.d/mongodb.repo
   ```

6.

While still in ~/elements/mongodb, create another directory named **install.d**, which will contain the installation hooks:

```
mkdir install.d
```

7. In install.d, create a file named **10-mongodb** and change its execute permissions:

   ```
   touch 10-mongodbchmod a+x 10-mongodb
   ```

8. Using your editor, add the following lines to the 10-mongodb file:

   ```
   !/bin/bash
   set -euxinstall-packages mongo-10gen mongo-10gen-server
   ```

9. While still in ~/elements/mongodb, create another directory named **finalise.d**, which will contain the finalise hooks:

   ```
   mkdir finalise.d
   ```

10. In finalise.d, create a file named **10-mongodb** and change its execute permissions:

    ```
    touch 10-mongodbchmod a+x 10-mongodb
    ```

11. Using an editor, add the following lines to the 10-mongodb file:

    ```
    !/bin/bash
    set -euxrm -Rf /etc/yum.repos.d/mongodb.repo
    ```

Creating an image using mongodb with RHEL

To use your new mongodb element with RHEL to create an image that is pre-installed with associated kernel and ramdisk images, use the following command:

```
disk-image-create –o <image_prefix> –a amd64–u base rhel baremetal mongodb
```

This creates a disk image file named base.qcow2. You can now upload and work with this image in HP Cloud OS for Moonshot. For detailed instructions, see the **Manage Images** topic of the Administration Dashboard help (PDF).