Beta 1: Installing HP Helion OpenStack on Baremetal

Contents

Beta 1: Installing HP Helion OpenStack® on Baremetal

Known Restrictions

- Ensure the *minimum hardware requirements* are met.
- All disks on the system(s) will be wiped: /dev/sda will be used for the OS but all other disks will be wiped.
- The deployer node must run on HP Linux for HP Helion OpenStack. It is part of the ISO you will download from the *Helion Downloads* page. There is only one download; it includes both the OS and the Beta0 installer in one ISO.
- Three NIC are tested, one 1G used for PXE and two bonded 10G for everything else. All machines can net boot from PXE and use the deployer as a DHCP server.
- All machines of a single type should be the same, that is, all computes, all VSAs, and so on.
- The deployer node must be a dedicated node in Beta0.
- The machine hosting the deployer and all baremetal systems must be connected to a management network. Nodes on this management network must be able to reach the iLO subsystem of each baremetal system to enable host reboots as part of the install process. The HP Helion OpenStack architecture requires that the IPMI network is a separate network and that a route exists from the management network to the IPMI network for iLO access.

Before you Start

Prepare your baremetal hardware, as follows, on all nodes:

- Setup the iLO Advanced license in the iLO configuration.
- Switch from UEFI to Legacy BIOS.
- Ensure that the network to be used for PXE installation has PXE enabled.
- Ensure that the other networks have PXE disabled.
- Insert the CD ROM in the Virtual Media drive on the iLO.

Set up the Deployer

1. Create LUN(s), if required.
2. Download the HP Helion OpenStack Deployer ISO from the *Helion Downloads* page after signing up and being approved for the program.
3. Boot your deployer from the ISO.
4. Enter "install" to start installation.
5. Select the language.
6. Select the location.
7. Select the keyboard layout.
8. Select the primary network interface, if prompted:
   - Assign IP address, netmask
9. Create new account:
   - Enter a username.
   - Enter a password.
   - Enter time zone if prompted to do so.
   - Synchronize the time on all nodes manually. NTP will be installed later.

At the end of this section you should have a deployer node set up with hLinux on it.

Configure and Run the Deployer

1. On the deployer node, enter the following command to create the SSH keypair if one is not already present:

   ```
   ssh-keygen -t rsa
   ```
2. Add ~/.ssh/id_rsa.pub to ~/.ssh/authorized_keys file:

   ```
   cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
   ```
3. Confirm that ssh localhost works without a password and that you can get from external sources, both with and without sudo.
4. Mount the install media at /media/cdrom, for example,

   ```
   sudo mount /media/cdrom
   ```
5. Unpack the following tarball:

   ```
   tar zxvf /media/cdrom/hos-2.0.0/hlm-deployer-2.0.0-20150805T115313Z.tgz
   ```
6. Run the following included script:

   ```
   ~/hlm-deployer/hlm-init-2.0.0-1.bash
   ```

At the end of this section you should have a local directory structure, as described below:

```
helion/                         Top level directory
helion/examples/                Directory contains the config input files of
 the example clouds
helion/my_cloud/definition/     Directory contains the config input files
helion/my_cloud/config/         Directory contains .j2 files which are
 symlinks to the /hlm/ansible directory
helion/hlm/                     Directory contains files used by the
 installer
```

Configure your Environment

1. Setup your configuration files, as follows:

   a. See the sample set of configuration files in the ~/helion/examples/one-region-poc-with-vsa directory. The accompanying README.md file explains the contents of each of the configuration files.

   b. Copy the example configuration files into the required setup directory and edit them as required:

      ```
      cp -r ~/helion/examples/one-region-poc-with-vsa/*
      ```

      Note: There have been changes to some of the files since Beta 0.

      The configuration files for editing can be found at the following location:

      ```
      ~/helion/my_cloud/definition/data
      ```

      - The baremetalConfig.yml file should specify the server information for your environment.
      - The servers.yml file contains the IP address information for the hardware.
      - The networks.yml file contains networking information.
      - The control_plane.yml file contains information about the services that will be installed.
      - The `/data/disks_controller.yml` contains the Swift information (swiftobj disk-group). This describes what disks on each of the servers are assigned to Swift and the rings uses those drives.
      - In the `net_interfaces.yml` file, replace all instances of “bond-mode: 1” to “bond-mode: active-backup”.

   c. Commit your configuration into git, as follows:

   ```
   git add –A
   git commit –m “My config”
   ```

2. Run the configuration processor, as follows:

   ```
   cd ~/helion/hlm/ansible
   ```

   ```
   ansible-playbook -i hosts/localhost config-processor-run.yml
   ```

   The CP output is placed on to the following two private branches of the git repo:

   - staging-ansible
   - staging-cp-persistent

3. a. Check the generated host files in `~/helion/my_cloud/stage/ansible/host_vars/` to ensure the correct IPs are included. You can review the ansible change by running the following command:

   ```
   git show staging-ansible
   ```

   b. Use the

   ```
   ansible-playbook -i hosts/localhost ready-deployment.yml
   ```

   file to create a deployment directory. Run your "verb_host" commands from the following directory:

   ```
    ~/scratch/ansible/next/hos/ansible
   ```

Deploy Cobbler

1. Run the following command:

   ```
   export ANSIBLE_HOST_KEY_CHECKING=False
   ```

2. Run the following playbook:

   ```
   ansible-playbook -i hosts/localhost cobbler-deploy.yml
   ```

Provision the Nodes

1. Run the following command, which will reimage all the nodes using PXE:

   ```
   ansible-playbook -i hosts/localhost bm-reimage.yml
   ```

2. Wait for the nodes to install and come back up.

Deploy the Cloud

1. Run the following command:

   ```
   ansible-playbook -i hosts/verb_hosts osconfig-run.yml
   ```

   a. Verify the network is working correctly. Ping each IP (excluding VSA-BLK and VIPs) from the /etc/hosts file from one of the controller nodes.

2. (**Optional**) To validate object storage cloud model, execute the following command on the deployer :

   ```
   ansible-playbook -i hosts/verb_hosts swift-ring-validate.yml
   ```

If any error is triggered when you validate the object stroage, refer to *Swift-Troubleshooting*

3. Modify the ./helion/hlm/ansible/hlm-deploy.yml to comment out the line containing horizon.
4. Run the following command:

```
ansible-playbook -i hosts/verb_hosts hlm-deploy.yml -e
 tuning_selector=medium
```

Install and Configure VSA (optional).

You can set up Virtual Storage Appliance (VSA) using the instructions on the following page: *Install and Configure VSA (optional)*.

Verification Steps

1. This step may be used to verify your cloud installation. Note that running the command below will download a cirros image from the internet, upload it to Glance, and create a Neutron external network.

```
ansible-playbook -i hosts/verb_hosts hlm-cloud-configure.yml
```

Note: You can optionally specify the external network CIDR here too. If you choose not to excercise this option or use a wrong value, the VMs will not be accessible over the network.

```
ansible-playbook -i hosts/verb_hosts hlm-cloud-configure.yml -e
 EXT_NET_CIDR=10.240.96.0/20
```

2. Run the following command, which will replace `/etc/hosts` on the deployer:

```
ansible-playbook -i hosts/localhost cloud-client-setup.yml
```

As the /etc/hosts no longer have entries for HLM, sudo commands may become a bit slower. To fix this issue, once this step is complete, add "hlm" after "127.0.0.1 localhost". The result will look like this:

```
...
# Localhost Information
127.0.0.1 localhost hlm
```

Swift- Troubleshooting

If the validation of Swift triggers an error, perform the following steps:

1. Log onto the node that has been designated as the Ring-Builder.

Note: You must run the swift-ring-validate playbook as mentioned in step **2** in *Deploy the Cloud* on page 5 otherwise the files needed to run the `swiftlm-ring-supervisor` will not be in place on the node.

2. Execute the following command:

```
sudo swiftlm-ring-supervisor --make-delta --report
```

The command will report any errors or problems with the input model. It also prints a summary of the ring create actions that are planned (i.e., that will occur) for the deploy phase of the process.