

ESX Installation

Contents

ESX Installation.....3

ESX Installation

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 use the HP Linux for HP Helion OpenStack ISO, which can be downloaded from the [Helion Downloads](#) page.
- 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, , and so on.
- The deployer node must be a dedicated node in Beta1.
- 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:

- Set up 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.

Set up the Deployer

1. Create LUN(s), if required.
2. Download the HP Linux for HP Helion OpenStack Deployer ISO from the [Helion Downloads](#) page.
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.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. Set up your configuration files, as follows:

- a. See the sample set of configuration files in the ~/helion/examples/one-region-poc-with-esx directory. The accompanying README.md file explains the contents of each of the configuration files.
- b. Copy the example configuration files in the required setup directory and edit them as required:

```
cp -r ~/helion/examples/one-region-poc-with-esx/* ~/helion/my_cloud/definition/
```

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 for controller nodes.
- The networks.yml file contains networking information.
- The control_plane.yml file contains information about the services that will be installed.
- In the net_interfaces.yml file, replace all instances of “bond-mode: 1” to “bond-mode: active-backup”.

2. Run the configuration processor, as follows:

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

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

Note that if the installer tells you to run `ansible-playbook -i hosts/localhost config-processor-run.yml`, don't. Run command as above (v2).

3. Check the generated host files in `~/helion/my_cloud/stage/ansible/host_vars/` to ensure the correct IPs are included. If they are not, run

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

and go to step 1.

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 set all nodes to PXE boot and power cycle them to start their OS install:

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

2. Wait for the nodes to install. They will power down at the end. You can make waiting easier with the following command:

```
ansible-playbook -i hosts/localhost cobbler-wait-for-shutdown.yml
```

This will complete once all machines are down.

3. Once all nodes are down, power up the recently-installed systems, using the following command:

```
ansible-playbook -i hosts/localhost cobbler-power-up.yml
```

4. You can make waiting easier with the command below, which will complete once all machines are up and the SSH daemon is responding:

```
ansible-playbook -i hosts/localhost cobbler-wait-for-ssh.yml
```

Deploy the Cloud

1. Run the following command:

```
ansible-playbook -i hosts/verb_hosts osconfig-runV2.yml
```

- a. Verify the network is working correctly by pinging each IP (excluding VSA-BLK and VIPs) from the deployer node. You can find the IP addresses in **generated_files/etc/hosts**. To do so, run:

```
stack@hlm:~/helion/hlm/ansible$ less generated_files/etc/hosts
```

2. Edit `roles/HZN-WEB/tasks/install.yml` and add the following after the 'install-package' section:

```
- name: HZN-WEB | install | TEMP fix horizon permissions
  sudo: yes
  command: "chown -R stack:stack {{ HORIZON_VENV_DIR }}/lib/python2.7/
site-packages"
```

the modified file should look as follows:

```
# Install pre-packaged Horizon venv from tarball
- name: HZN-WEB | install | Install Horizon
  install_package:
    name: horizon
    service: horizon
    state: present

- name: HZN-WEB | install | TEMP fix horizon permissions
  sudo: yes
  command: "chown -R stack:stack {{ HORIZON_VENV_DIR }}/lib/python2.7/
site-packages"
```

3. Run the following command:

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

Install and Configure ESX Compute and OVSvAPP on vCenter

The following sections describe the procedure to install and configure ESX compute and OVSvAPP on vCenter.

- Deploy a template
- Deploy a service
- Run the config processor
- Deploy a cloud

Deploy a template

Perform the following steps to deploy a template:

1. Import the `hlm-shell-vm.ova` in the vCenter using the vSphere client.
2. In the vSphere Client, click **File** and then click **Deploy OVF Template**.
3. Follow the instructions in the wizard to specify the data center, cluster, and node to install. Refer to the VMware vSphere documentation as needed.

Deploy a Service

Execute the following steps from the deployer to deploy a service using the EON CLI

1. Source `service.osrc`.
2. Add a vCenter to the eon database.

```
eon vcenter-add --name <vCenter Name> --ip-address <vCenter IP address> --
username <vCenter Username> --password<vCenter Password> --port<vCenter
Port>
```

where:

- vCenter Name - the name of the vCenter server where the service will be deployed.
- vCenter IP address - the IP address of the vCenter server where the service will be deployed.
- vCenter Username - the username for the vCenter administrator.
- vCenter Password - the password for the vCenter administrator.
- vCenter Port - the vCenter server port.

Sample Output:

```
+-----+-----+
| Property | Value |
+-----+-----+
| A |
```

```

| created_at | 2014-05-09T09:01:09.513550 |
| deleted    | False                      |
| deleted_at | None                       |
| id         | f7cb3811-0d33-44ec-a54a-a69cf7615f5e |
| ip_address | 10.1.192.98                |
| name       | Test_vcenter               |
| password   | <SANITIZED>                 |
| port       | 443                        |
| type       | vcenter                    |
| updated_at | 2014-05-09T09:53:33.500434 |
| username   | cdl\moonshotblr           |
+-----+-----+

```

3. Run the following command to update the network configuration:

```
eon get-network-info-template --filename < file name>
```

For example:

```
eon get-network-info-template --filename <net_conf.json>
```

4. Modify the json file as per your environment.

```
vi <file name>
```

5. Set the network information for a vCenter.

```
eon set-network-info --vcenter-id <vCenter ID> --config-json <abc.json>
```



Note: The vcenter ID is generated when you execute the above (step 3) command.

6. Execute the following command to view the list of clusters for the given vCenter.

```
eon cluster-list --vcenter-id <vCenter ID>
```

Sample Output

```

+-----+-----+-----+-----+-----+
+-----+
| MOID | Name | Datacenter | Import Status | Managed Status |
|      |      |            |               |                |
+-----+-----+-----+-----+-----+
+-----+
| domain-c26 | Cluster1 | COS | not_imported | False |
|            |          |     |               |      |
| domain-c28 | Cluster2 | COS | not_imported | False |
|            |          |     |               |      |
+-----+-----+-----+-----+-----+
+-----+

```

7. Import the cluster for the EON database under the given vCenter.

```
eon cluster-import --vcenter-id <vCenter ID> --cluster-name <Cluster Name>
--cluster-moid <Cluster Moid>
```

where:

- vCenter ID: ID of the vcenter containing the cluster.
- Cluster Name: the name of the cluster that needs to be imported.
- cluster Moid: Moid of the cluster that needs to be imported.

```

+-----+-----+-----+-----+-----+

```

Property	Value
id	7e7ed17d-de71-4207-af5f-3c5fca847d3d
name	Cluster1
resource_moid	domain-c26
resource_uuid	7e7ed17d-de71-4207-af5f-3c5fca847d3d
state	imported
vcenter_id	f7cb3811-0d33-44ec-a54a-a69cf7615f5e

One vCenter can have multiple clusters. But it allows you to import only one cluster at a time.

8. Activate the cluster for the selected vCenter.

```
eon cluster-activate --vcenter-id <vCenter ID> --cluster-moid <Cluster Moid>
```

Sample Output

required sample out for the activated cluster

Run the config processor

Execute the following commands:

```
cd ~/helion/hlm/ansible
ansible-playbook -i hosts/localhost config-processor-run.yml
```

Deploy a cloud

Execute the following command to deploy a cloud:

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

Verification Steps

Change this section as per esx

1. This step may be used to verify your cloud installation. Note that running the command below will download a cirros image format 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 exercise 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
```

3. 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
```