Deploying ESX Cloud

Contents

Deploying ESX Cloud

This page describes the procedure to deploy ESX cloud using input model.

Important:

Before you start your ESX cloud deployment ensure that you read the following instructions carefully.

Important Notes

- If you are looking for information about when to use the GUI installer and when to use the CLI, see the *Installation Overview*.
- We have put together a *Pre-Installation Checklist* that should help with the recommended pre-installation tasks.
- Ensure the *minimum hardware requirements* are met in your environment.
- There is no longer a requirement to use a dedicated deployer node. See the note in the *Release Notes* as well as the configuration section in the install steps below for details.
- There is no requirement to have a dedicated network for OS-install and system deployment. More information can be found on the *Supported Configuration* page.
- If you run into issues during installation, we have put together a list of *Troubleshooting Steps* you can reference.
- Read over the page about the *Helion OpenStack 2.0 Input Model* to learn about the configuration options for your cloud.
- Make sure **all disks on the system(s) are wiped** before you begin the install. (For Swift, refer to *Swift Requirements for Device Group Drives*)
- The `/dev/sda` disk on your systems will be used for the operating system.
- Both HP Linux for HP Helion OpenStack and HP Helion OpenStack are part of the ISO you will download from the *Helion Downloads* page. There is only one download; it includes both the operating system and the Helion OpenStack installer in one ISO.
- All machines of a given role should be the same configuration.
- 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

Important: We have put together a *Pre-Installation Checklist* that should help with the recommended pre-installation tasks.

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

- Set up the iLO Advanced license in the iLO configuration. Make sure the iLO user has admin privileges.
- HP Helion OpenStack 2.0 Beta 2 will detect and use the "BIOS" mode you have selected for each node; UEFI or legacy BIOS. UEFI support is new in Beta 2.
- Ensure that the network interface to be used for PXE installation has PXE enabled.
- Ensure that the other network interfaces have PXE disabled.
- Ensure that any logical drives (LUN) for the servers you will be using are created to meet the disk requirements outlined in the *Minimum Hardware Requirements*.

Prerequisite

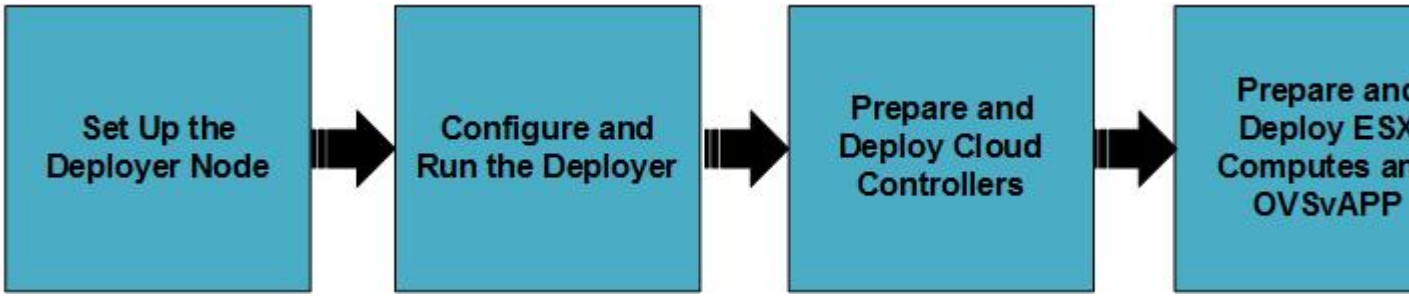
ESX/vCenter integration is not fully automatic, vCenter administrators are advised of the following responsibilities to ensure secure operation:

- The VMware administrator is responsible for administration of the vCenter servers and the ESX nodes using the VMware administration tools. These responsibilities include:

  - Installing and configuring vCenter Server
  - Installing and configuring ESX server and ESX cluster
  - Installing and configuring shared datastores
  - Establishing network connectivity between the ESX network and the HP Helion management network
- The VMware administration staff is responsible for the review of vCenter logs. These logs are not automatically included in Helion centralized logging.
- Logging levels for vCenter should be set appropriately to prevent logging of the password for the Helion message queue.
- The vCenter cluster and ESX Compute nodes must be appropriately backed up.
- Backup procedures for vCenter should ensure that the file containing the Helion configuration as part of Nova and Cinder volume services is backed up and the backups are protected appropriately.
- Since the file containing the Helion message queue password could appear in the swap area of a vCenter server, appropriate controls should be applied to the vCenter cluster to prevent discovery of the password via snooping of the swap area or memory dumps

Deploy ESX Cloud

At a high level, here are the steps to configure and deploy ESX cloud:



Procedure to Deploy ESX cloud

The following topics in this section explain how to deploy ESX cloud.

Set up the Deployer

You can use a dedicated deployer node or you can run these instructions on your first controller node. There is only one difference in the steps and that is notated in the *Configure Your Environment* section.

1. Download the HP Helion OpenStack Deployer ISO from the *Helion Downloads* page after signing up and being approved for the program.
2. Boot your deployer from the ISO. Insert the CD ROM in the Virtual Media drive on the iLO.
3. Enter "install" to start installation.

   Note: "install" is all lower case

4. Select the language. Note that only the English language selection is currently supported.
5. Select the location.
6. Select the keyboard layout.
7. Select the primary network interface, if prompted:
   - Assign IP address, netmask
8. Create new account:
   - Enter a username.
   - Enter a password.
   - Enter time zone if prompted to do so.
9. Ensure your deployer has a valid DNS nameserver in the `/etc/resolv.conf` file

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

Configure and Run the Deployer

Note: It's critical that you don't run as root. Run as the user you just created (or stack if you left the default of "stack"), but do not run as root.

1. Log in to the node you setup in the previous steps as the user you created during the setup phase, and mount the install media at /media/cdrom, for example,

```
sudo mount Helion-OpenStack-2.0.0-b.2-Beta2.iso /media/cdrom
```

2. Unpack the tarball that is in the `/media/cdrom/hos2.0.0/` directory:

```
tar zxvf /media/cdrom/hos-2.0.0/hos-2.0.0-b.2-20150920T130131Z.tgz
```

3. Run the following included script:

```
~/hos-2.0.0-b.2/hos-init.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 /hos/ansible directory
helion/hos/                      Directory contains files used by the
 installer
```

Prepare and Deploy Cloud Controllers

1. See a 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.
2. Copy the example configuration files into the required setup directory and edit them as required:

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

3. The configuration files for editing are available at `~/helion/my_cloud/definition/`. Refer to the ***Helion OpenStack 2.0 Input Model*** document for assistance with the configuration files.

Important: If you chose to use your first controller node as your deployer, ensure that your `baremetalConfig.yml` file contains the `is_deployer: true` notation in your controller options. If you are using a dedicated deployer node you can omit this. Here is an example snippet of a `baremetalConfig.yml` file where a user is using their first controller node as their deployer:

```
node_name: "ccn-0001"
role: "ROLE-CONTROLLER"
pxe_mac_addr: "b2:72:8d:ac:7c:6f"
pxe_interface: "eth2"
pxe_ip_addr: "192.168.10.3"
ilo_ip: "192.168.9.3"
ilo_user: "admin"
ilo_password: "password"
is_deployer: true
```

4. Modify the `~/helion/hos/ansible/hlm-deploy.yml` to uncomment the line containing `eon-deploy.yml`.
5. Modify the `neutron.conf.j2` at `~/helion/my_cloud/config/neutron/neutron.conf.j2`with the following values:

```
router_distributed = False
```

   And add `mechanism_drivers = ovsvapp` at the end of the `neutron.conf.j2`.
6. Modify `ml2_conf.ini.j2` at `~/helion/my_cloud/config/neutron/ml2_conf.ini.j2`with the following values:

```
[ml2]
mechanism_drivers = ovsvapp, openvswitch, l2population
[agent]
enable_distributed_routing = False
```

7. Commit your cloud deploy configuration to the *local git repo*, as follows:

```
cd ~/helion/hos/ansible
git add -A
git commit -m "My config"
```

Deploy Cobbler

1. Run the following command:

```
export ANSIBLE_HOST_KEY_CHECKING=False
```

2. Run the following playbook which confirms that there is iLo connectivity for each of your nodes so that they are accessible to be re-imaged in a later step:

```
cd ~/helion/hos/ansible
ansible-playbook -i hosts/localhost bm-power-status.yml
```

3. Run the following playbook to deploy Cobbler:

```
cd ~/helion/hos/ansible
ansible-playbook -i hosts/localhost cobbler-deploy.yml
```

4. The cobbler-deploy stage of the installation now prompts for a password. The entered value is used to set a user-defined password for subsequent node installs so that a user can gain console/terminal access to the nodes. For example, if a node maintenance is required or if the SSH keys are lost. The value is encrypted and placed in the kickstart for the node installs.

   The password prompts look like this:

```
Enter the password that will be used to access provisioned nodes:
confirm Enter the password that will be used to access provisioned nodes:
```

   Alternatively, you can also specify this password in the ansible playbook command like this:

```
ansible-playbook -i hosts/localhost cobbler-deploy.yml -e
 hlmuser_password="<password>"
```

Provision the Nodes

1. Run the following command, which will reimage all the nodes using PXE. Note: change directories to /helion/hos/ansible if not already there:

```
cd ~/helion/hos/ansible
ansible-playbook -i hosts/localhost bm-reimage.yml
```

2. The script will wait for the nodes to install and come back up.

Run the Configuration Processor

1. Run the configuration processor, as follows:

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

The configuration processor output is placed onto the following two private branches of the git repo:

- staging-ansible
- staging-cp-persistent

Note: The configuration processor may generate the following warnings which can be **safely ignored**:

```
################################################################################
# The configuration processor completed with warnings.
#   ans-host-vars-2.0         WRN: Unknown service 'vsa'as consumer
 of device-group vsa-datain disk-model DISK_SET_VSA.
#   ans-host-vars-2.0         WRN: Unknown service 'vsa'as consumer
 of device-group vsa-cachein disk-model DISK_SET_VSA.
################################################################################
```

Note: See *Using Git for Configuration Management* for more information.

2. Check your configuration:

   a. You can review the ansible change by running the following command:

```
git show staging-ansible
```

Deploy the Cloud

1. Use the playbook below to create a deployment directory:

```
cd ~/helion/hos/ansible
ansible-playbook -i hosts/localhost ready-deployment.yml
```

2. Run your "verb_host" commands using the following steps:

```
cd ~/scratch/ansible/next/hos/ansible
ansible-playbook -i hosts/verb_hosts site.yml
```

Note: The step above runs osconfig to configure the cloud and hlm-deploy, to deploy the cloud. Therefore, this step may run for quite some time while all the nodes are configured (perhaps 45 minutes or more, depending on the number of nodes, to a degree).

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

Prepare and Deploy ESX Computes and OVSvAPPs

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

- *Deploy Helion Linux Shell VM Template*
- *Preparation for ESX Cloud Deployment*

Deploy Helion Linux Shell VM Template

The first step in deploying the ESX compute proxy and OVSvAPPs is to create a VM template that will make it easier to deploy the ESX compute proxy for each Cluster and OVSvAPPs on each ESX server.

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.

Preparation for ESX Cloud Deployment

This section describes the procedures to prepare and deploy the ESX computes and OVSvAPPs for deployment.

1. Login to the deployer node.
2. Source `service.osrc`.
3. *Register a vCenter Server*
4. *Register ESX Cloud Network Configuration*
5. *Import Clusters*
6. *Activate Clusters*
7. *Modify the Volume Configuration File*
8. *Commit your Cloud Definition*
9. *Deploy ESX Compute Proxy and OVSvApps*

Manage vCenters and Clusters

The following section describes the detailed procedure on managing the vCenters and clusters.

Register a vCenter Server

vCenter provides centralized management of virtual host and virtual machines from a single console.

1. Add a vCenter using EON python client.

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

where:

- vCenter Name - the identical name of the vCenter server.
- vCenter IP address - the IP address of the vCenter server.
- vCenter Username - the admin privilege username for the vCenter.
- vCenter Password - the password for the above username.
- vCenter Port - the vCenter server port. By default it is 443.

Important: Please do not change the port unless you are sure about the vCenter Port.

Sample Output:

```
# eon vcenter-add --name vc01 --ip-address 10.1.200.41 --username
 administrator@vsphere.local --password password --port 443
+------------+--------------------------------------+
| Property   | Value                                |
+------------+--------------------------------------+
| created_at | 2015-08-20T12:08:09.000000           |
| deleted    | False                                |
| deleted_at | None                                 |
```

```
| id          | BC9DED4E-1639-481D-B190-2B54A2BF5674 |
| ip_address | 10.1.200.41                           |
| name        | vc01                                  |
| password    | <SANITIZED>                           |
| port        | 443                                   |
| type        | vcenter                               |
| updated_at | 2015-08-20T12:08:09.000000             |
| username    | administrator@vsphere.local           |
+------------+---------------------------------------+
```

Show vCenter

1. Show vCenter using EON python client.

```
# eon vcenter-show <vCenter ID>
```

Sample Output:

```
# eon vcenter-show <BC9DED4E-1639-481D-B190-2B54A2BF5674 >
+------------+---------------------------------------+
| Property   | Value                                 |
+------------+---------------------------------------+
| created_at | 2015-08-20T12:08:09.000000            |
| datacenters| DC1                                   |
| deleted    | False                                 |
| deleted_at | None                                  |
| id          | BC9DED4E-1639-481D-B190-2B54A2BF5674 |
| ip_address | 10.1.200.41                           |
| name        | vc01                                  |
| password    | <SANITIZED>                           |
| port        | 443                                   |
| type        | vcenter                               |
| updated_at | 2015-08-20T12:08:09.000000             |
| username    | administrator@vsphere.local           |
+------------+---------------------------------------+
```

Register ESX Cloud Network Configuration

This involve getting a sample network information template. Fill the details of the template and use that template to register cloud network configuration for the vCenter.

1. Execute the following command to get the network information template:

```
# eon get-network-info-template --filename <NETWORK_CONF_FILENAME>
```

For example:

```
# eon get-network-info-template --filename net_conf.json
```

Sample file of `net_conf.json` is shown below:

```
{
    "network": {
        # Deployer Network details
        # This network should be reachable from the Deployer node
        "deployer_network": {
            #Deployer Portgroup Name.
            "deployer_pg_name": "hlm-Deployer-PG",

            #VLAN id for Deployer Portgroup
            "deployer_vlan": "33",
```

```
            #Enable DHCP for Deployer N/W
            "enable_deployer_dhcp": "no",

            #CIDR and gateway for deployer network only when static
            "deployer_cidr": "10.20.18.0/23",
            "deployer_gateway_ip": "10.20.18.1",

            #Deployer Node's PXE IP Address
            "deployer_node_ip": "10.20.16.2"
        },

        #Management Network details
        "management_network": {
            #Mgmt DVS name.
            "mgmt_dvs_name": "hlm-Mgmt",

            #Physical NIC name for Mgmt DVS
            "mgmt_nic_name": "vmnic3",

            #Mgmt Portgroup Name.
            "mgmt_pg_name": "hlm-Mgmt-PG",

            #Interface order: Example eth1
            "mgmt_interface_order": "eth1",

            "active_nics": "",

            "load_balancing": "1",

            "network_failover_detection": "1",

            "notify_switches": "yes"
        },

        "data_network": {
            #Tenant network type
            "tenant_network_type": "vlan",

            "data_dvs_name": "hlm-Data",

            "data_nic_name": "vmnic2, vmnic1",

            #Data Portgroup Name.
            "data_pg_name": "hlm-Data-PG",

            #Interface order: Example eth2
            "data_interface_order": "eth2",

            #If more than one mgmt_nic_name are specified then NIC teaming
will be enabled by default
            #Active uplink NICs for NIC teaming(If this field is empty,
first data_nic_name will be "Active" and the rest will be in "Standby")
            "active_nics": "vmnic2",

            #Load Balancing. Please choose the corresponding number
            #    1 -> Route based on the originating virtual port
            #    2 -> Route based on IP hash
            #    3 -> Route based on source MAC hash
            #    4 -> Route based on physical NIC load
            #    5 -> Use explicit failover order
            "load_balancing": "1",

            #Network Failover Detection. Please choose the corresponding
number
```

```
            #    1 -> Link Status
            #    2 -> Beacon Probing
            "network_failover_detection": "1",

            #Notify Switches(yes/no)
            "notify_switches": "yes"
        },

        "hpvcn_trunk_network": {
            #Trunk DVS name
            "trunk_dvs_name": "hlm-Trunk",

            #Trunk Portgroup Name.
            "trunk_pg_name": "hlm-Trunk-PG",

            #Interface order: Example eth3
            "trunk_interface_order": "eth3"
        },

        #VLAN Range for Data & Trunk port group. Please provide the range
 separated by a hyphen(vlan-vlan).
        #Multiple vlan or vlan ranges has to be a comma separated
 value(*OPTIONAL)
        "vlan_range": "1-4094"
    },

    "template": {
        #Provide the template/appliance name that will be used for cloning
 Computeproxy and OVSvApp VMs
        "template_name": "hlm-template"
    },

    "vmconfig": {
        #Number of CPUs for OVSvApp/Computeproxy VM
        "cpu": "4",

        #Amount of RAM in MB
        "memory_in_mb": "4096",

        #SSH public key content for OVSvAPP/Computeproxy password less
 login.
        "ssh_key": "<deployer-ssh-pub-key-contents>"
    },
        # Do you want to skip inactive or maintenance mode hosts ?
    "skip_inactive_hosts": "yes",
        # Provide new host mo ids when you are adding a new host in an
 activated cluster.
    "new_host_mo_ids": ""
}
```

2. Modify the template (json file) as per your environment.

```
vi <NETWORK_CONF_FILENAME>
```

For example:

```
vi net_conf.json
```

3. Use the template to register Cloud Network Configuration. This sets the network information for a vCenter which is used to deploy and configure compute proxy and OVSvAPP VMs during the cluster activation.

```
# eon set-network-info --vcenter-id <vCenter ID> --datacenter-name
 <datacenter name> --config-json <NETWORK_CONF_FILENAME>
```

For example:

```
# eon set-network-info --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674
 --datacenter DC1 --config-json net_conf.json
```

Note: The vcenter ID is generated when you execute the above (**step 2**) command.

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

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

Sample Output

```
# eon cluster-list --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674
+------------+----------+------------+----------------+
| MOID       | Name     | Datacenter | Import Status |
+------------+----------+------------+----------------+
| domain-c21 | Cluster1 | DC1        | not_imported  |
+------------+----------+------------+----------------+
```

Import Cluster

You can use one or more ESX clusters for ESX Cloud Deployment. When a Import Cluster is invoked, required ESX Compute Proxy and OVSvApp nodes are deployed.

1. 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.

Sample Output

```
# eon cluster-import --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674 --
cluster-name Cluster1 --cluster-moid domain-c21
+--------------+-----------+
| Property     | Value     |
+--------------+-----------+
| cpu_free     | 83071.73  |
| cpu_total    | 83072     |
| cpu_used     | 0.27      |
| datacenter   | DC1       |
| disk_free    | 1022.79   |
| disk_total   | 1023.75   |
| errors       | []        |
| memory_free  | 496.82    |
| memory_total | 511.76    |
| memory_used  | 14.94     |
| name         | Cluster1  |
| state        | importing |
| switches     | []        |
```

```
+--------------+-----------+
```

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

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

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

Sample Output

```
# eon cluster-list --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674
+------------+----------+------------+---------------+
| MOID       | Name     | Datacenter | Import Status |
+------------+----------+------------+---------------+
| domain-c22 | Cluster2 | DC1        | imported      |
+------------+----------+------------+---------------+
```

Activate Clusters

Note: You can activate the cluster only after the import status of the cluster is changed to **imported**.

When you execute the active cluster command, the `server.yml` of the input model is updated with IP Addresses of compute proxy and OVSvApp.

1. Activate the cluster for the selected vCenter.

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

Sample Output

```
# eon cluster-activate --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674
 --cluster-moid domain-c22
+---------------
+-------------------------------------------------------------------------------------------
+
| Property      | Value
                                                                                            
                                                                                            
                                                                                            
                                                                                            
                                                                                            
                                                                                            
                                                                                            
                      |
+---------------
+-------------------------------------------------------------------------------------------
+
| node_info     | {u'computeproxy': {u'pxe-mac-addr':
 u'00:50:56:b6:ce:1b', u'pxe-ip-addr': u'172.170.2.4', u'name':
 u'COMPUTEPROXY_Cluster1', u'cluster-moid': u'domain-c21'},
 u'network_driver': {u'cluster_dvs_mapping': u'DC1/host/Cluster1:hlm-
Trunk', u'Cluster1': [{u'host-moid': u'host-29', u'pxe-ip-addr':
 u'172.170.2.3', u'esx_hostname': u'10.1.200.33', u'ovsvapp_node':
 u'ovsvapp-10-1-200-33', u'pxe-mac-addr': u'00:50:56:b6:5e:9a'}, {u'host-
moid': u'host-25', u'pxe-ip-addr': u'172.170.2.2', u'esx_hostname':
 u'10.1.200.66', u'ovsvapp_node': u'ovsvapp-10-1-200-66', u'pxe-mac-addr':
 u'00:50:56:b6:56:e6'}]}} |
| resource_moid | domain-c21
```

```
                        |
| resource_name | Cluster1




                        |
| state         | activated




                        |
+---------------
+-------------------------------------------------------------------------------
+
```

Modify the Volume Configuration File

Once the cluster is activated you must configure the volume.

Perform the following steps to modify the volume configuration files:

1. Change the directory. The `cinder.conf.j2` is present in following directories :

   ```
   cd /home/stack/helion/hos/ansible/roles/_CND-CMN/templates
   ```

   OR

   ```
   cd /home/stack/helion/my_cloud/config/cinder
   ```

   It is recommended to modify the `cinder.conf.j2` present in `/home/stack/helion/my_cloud/config/cinder`
2. Modify the `cinder.conf.j2` as follows:

   ```
   # Start of section for VMDK block storage
   #
   # If you have configured VMDK Block storage for cinder you must
   # uncomment this section, and replace all strings in angle brackets
   # with the correct values for vCenter you have configured. You
   # must also add the section name to the list of values in the
   # 'enabled_backends' variable above. You must provide unique section
   # each time you configure a new backend.
   #
   #[<unique-section-name>]
   #vmware_api_retry_count = 10
   #vmware_tmp_dir = /tmp
   #vmware_image_transfer_timeout_secs = 7200
   #vmware_task_poll_interval = 0.5
   #vmware_max_objects_retrieval = 100
   #vmware_volume_folder = cinder-volumes
   #volume_driver = cinder.volume.drivers.vmware.vmdk.VMwareVcVmdkDriver
   #vmware_host_ip = <ip_address_of_vcenter>
   #vmware_host_username = <vcenter_username>
   ```

```
#vmware_host_password = <password>
#
#volume_backend_name = <vmdk-backend-name>
#
# End of section for VMDK block storage
```

Commit your Cloud Definition

1. Add the cloud deployment definition to the git :

```
cd /home/stack/helion/hos/ansible;
git add -A;
git commit -m 'Adding ESX Configurations';
```

2. Prepare your environment for deployment:

```
ansible-playbook -i hosts/localhost config-processor-run.yml;
ansible-playbook -i hosts/localhost ready-deployment.yml;
cd /home/stack/scratch/ansible/next/hos/ansible;
```

Deploy ESX Compute Proxy and OVSvApps

Execute the following command to deploy a esx compute and OVSvApps:

```
ansible-playbook -i hosts/verb_hosts guard-deployment.yml
ansible-playbook -i hosts/verb_hosts osconfig-run.yml --limit '*esx-
ovsvapp:*esx-compute'
ansible-playbook -i hosts/verb_hosts hlm-deploy.yml --limit NOV-ESX:NEU-
OVSVAPP
```

Note: The variable **esx-ovsvapp** and **esx-compute** must be taken from the **name** key in the `resource-nodes` section in the `/data/control_plane.yml` file (`/home/stack/helion/my_cloud/definition/data/control_plane.yml`).