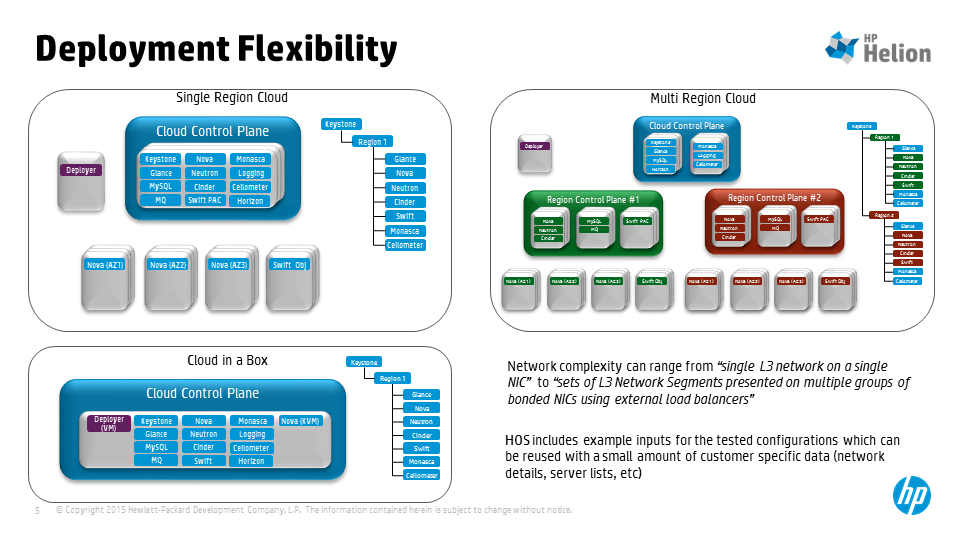
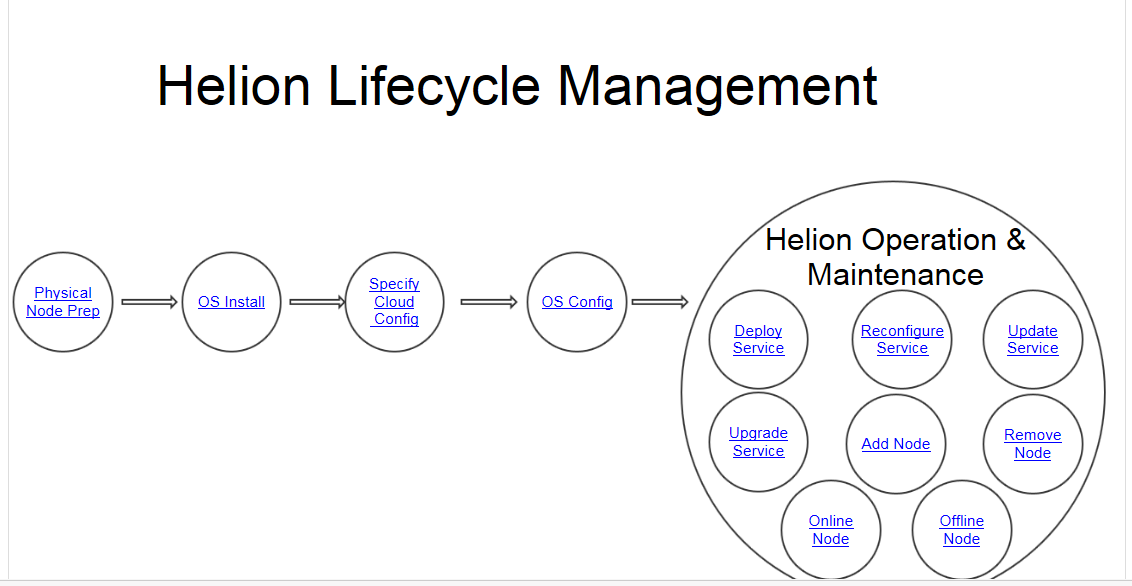
**Installing HP Helion OpenStack 2.0**

In HP Helion OpenStack 2.0 there is greater flexibility than there was previously with regard to the size and type of cloud you will build. Because of the number and variety of configurations possible with respect to size, topology, and configuration, there are a number that have been tested and known to work well. As all possible configurations cannot be tested, we will outline those that have been shown to work well in the **Supported Configurations** document. You may wish to read that document before continuing with your cloud planning. In addition, the [**Support Matrix**](http://docs.hpcloud.com/helion/openstack/1.1/support-matrix/)documentation outlines the hardware that is supported.



**5 models to come as RAs? Per Frank O’Neill’s meeting**

The deployer is a standalone package.



**Planning your Cloud**

Before you begin planning you will need to make decisions regarding hardware, networking, data storage, and more. You will have to decide what your configuration will look like before you begin installing.

The considerations we will cover here include:

* Hardware choices
* Capacity
* Redundancy
* Size of your control plane based on your requirements
* Data storage
  + Block storage. (Cinder)
  + Object storage. (Swift)
* Node types. You will need to identify the types of nodes you will have in your cloud.
  + For a simpler cloud implementation, you may decide to use a single type of node.
  + For larger, more complex implementations, you may want to rely on more node types wherein your compute nodes are different from block storage nodes, object storage nodes are yet another type, control plane nodes may be of a variety of types, etc. How is this decided?
* Network size, separation, zones, etc. How do you plan for this?
* Data center preparation. This is all about networking
  + Physical network: L1-L3 layers. Boxes wired into their switches, which networks are presented to which network interfaces
* Enabling virtualization will there still be choices re KVM, ESX, etc.
* Create SSH keys and certificates
* Virtual networking options VXLAN or provider VLAN

**Physical node preparation**

With version 2.0 of HP Helion OpenStack, you will install Helion using prewritten Ansible scripts that come in the Helion tarball. One of the first things you need to do is ensure that the firmware for storage, BIOS, and NIC cards is up to date. (Still to come. Probably generic wording about having latest firmware but to be decided.)

HLM will perhaps validate that

* Disc setup is right
* Nic setup is right
* Your config matches these and will work?

Customer has to be sure that

* Firmware is up to date
* Servers have connection
* Local disks are available
* BIOS setup ok

Need to provide for physical nodes for doing auto install of nodes’ hlinux OS

* Mac addresses
* Ipmi username pw and ip addresses

Deploy node will be on physical or on a VM. Install media: dvd or thumb drive. The deplyoyer node will remain to house HLM, although it does not have to remain up. However it will be used later for management. Maybe Ops console will run on it. So keep HLM installed on that node at least. LAN running management node. Not HA, etc. its data s/b backed up somewhere. Will be used for updating, adding hrdware, etc. may become a node type on its own.

1. Boot from the install media
2. Automated install follows
3. Interaction will be ip address for deploy node and creds for ansible user
4. User needs to know network addresses are avaible to him
5. Assign network address to deployer
6. Generate or collect keys

One NIC will be installed via HLM

The purpose of this phase is prepare a node for OS installation. At the end of this phase the node should

* Have its primary NIC connected to the customer's DC network
* Have its BIOS boot options set appropriately, e.g. set to PXE boot if using PXE for the OS installation phase
* System firmware at correct revision
* Local disk controller RAID setup is complete
* Any boot time kernel parameters required for the node are set on the boot options

The initial version of HLM will not provide any automated tooling for this phase.

The OS underlying all nodes for HP Helion OpenStack 2.0 is HP Linux for HP Helion OpenStack. You have the option of allowing the installer to install the OS for you, or you may install HP Linux for HP Helion OpenStack on all your nodes using your own tools and OS images.

**OS Installation**

This phase installs the host operating system onto the  Helion nodes, (virtual or bare metal).

One node will host the deployer as a vm or physical. Dvd or stick is inserted to that to start install.

After that nearly no interaction with user

Node remains deployer in future. Not HA etc. not needed to run cloud but to do future upgrades etc.

More than just deployement (may get renamed hlm node type)

**Inputs:**

* All nodes  with correct BIOS settings
* iLO username and password for each node these are only needed for HLM default auto OS install used for powering on and off the nodes because HLM will do this using iLO
* Connected to primary network
* Nodes set to PXE boot is required
* List of MAC addresses on primary interface, if using HLM OS Install tooling these are only for HLM default
* CIDR range for primary interfaces, if using HLM  OS Install tooling

**Outputs:**

* All nodes booted from boot disk with target OS installed
* Node is reachable via SSH on primary network with a sudo enabled user configured
* IP address set on primary interface and enumerated

**Tooling:**

Customers can use site specific tools to perform this step, e.g.  TripleO, MaaS, Holly/Blacksmith (HP Public Cloud) , Cobbler etc.

The initial version of HLM will provide a Cobbler based utility as part of the HLM-Deploy tooling to install hLinux on the target nodes.

**If you will be installing HP Linux for HP Helion OpenStack** on each of your nodes using your own images and tools, you will need to:

* You will need to know a list of IP addresses (How? In the YAML files? Can use one or more files. Why would you use several files?) but NOT Provide to HLM?
* You will need to know MAC addresses but NOT Provide to HLM?
* Configure, attach and partition local storage
* Assign IP addresses to each node?

**However,** **if you plan to have HP Helion OpenStack install HP Linux for HP Helion OpenStack on each of your nodes, you will need to provide to HLM**

* MAC addresses
* IP ranges and even multiple IP ranges (for different networks?) , for example. (How? YAML file? It can be one file or per rack or however customer wants.)
* You will also need to configure, attach and partition local storage
* Cobbler playbook will read your input and will run, installing the hLinux OS on each node you have identified previously.
* iLO credentials

**After hLinux is installed**

This is the point where the customer provides the detailed specification of their cloud. The specification

* Region organization (if deploying multi-region)
* Deployment topology - how services map to nodes, what services run on specif nodes, (e.g. specific nodes for Nova Compute,  Swift Object servers), control plane organization
* Network mapping: assigning different Helion traffic flows to separate VLANs
* Service configuration values, e.g. Storage types and back-ends for Cinder, Neutron tenant netowrks, plugins
* OS configuration: e.g. conntrack settings,

Have to start VMs yourself via the CLI?

What are they? What are the inputs, if any? Are they all used?

On the deployer, run script from deployer exploding packages - ansible and all services included

Ssh access keys will be needed and installed on deployer

**HLM OS install:**

HLM will power cycle all the nodes, shutting them down and then network booting them, setting up boot order and installing same packages for each node. All nodes can be ssh’d to from deployer node.

Start next script. This is where HLM takes over if you have installed yourself or HLM did it.

**Customer step: Checks such as SSH access etc would be performed here. How?**

**Check easier if you have done default**

**Now each node has basic config, and OS config will run, mostly in the background without needing input**

**Installing HP Helion OpenStack**

One your hardware and networks are set up and HP Linux for HP Helion OpenStack is installed everywhere, one NIC configured

These instructions cover a baremetal install scenario in which you are running with two networks: a separate network for management and a separate external network; both on a single network interface card (NIC), which is essentially the configuration HP Helion OpenStack 1.1 installations have used.

Playbook initiation is still going to be separate via a number of scripts so that user can step in at various points and make changes. So does this mean you need to start each discrete script on your own?

Tim Reddin’s steps

1. Set up deployer
   1. Create a hLinux VM – choose your own hypervisor (or a Baremetal host hLinux); Installed from ISO
      1. Note we are still evaluating options here so final system may have less options, e.g. bare metal only
   2. Copy bundle , e.g. tar file (HLM + HOS) to deployer system
   3. Un-tar
   4. Run script to setup deployer
   5. At this point the deployer has everything it needs (apart from user input) to deploy the cloud
2. Starting point for deploy is either:
   1. A set of pre-installed hLinux nodes
   2. A set of PXE bootable hardware nodes. In this case the deployer will create a basic hLinux OS on all nodes
3. Install hLinux
   1. Optional, can be done using customer tooling
   2. Inputs: are these ths inputs and outputs regardless of how hLinux is installed?
      1. CIDR (IP address range) used to assing initial IP addresses to all nodes
      2. List of nodes describing (MAC address, Node role)
   3. Outputs: A list of servers specifying
      1. IP address
      2. Server role
4. User edits input files to describe his cloud
   1. Optional GUI to guide this process
   2. This is the detailed part of the presentation above. It will need  System Design expertise/consultation for real systems
   3. We will provide sample templates for demo and MVP configurations
   4. User input is still needed for sample systems (e.g. IP address ranges)
5. Generate HLM Data model. This step takes a set of customer specifications ( from 4 above) and generates a data model used by HLM to consistently deploy the cloud. Inputs consist of:
   1. A cloud topology description, specifying regions, control plans and resource nodes. Includes assignment of OpenStack Services to server roles
   2. Rack description : what servers are in the rack, their role, , IP address,
   3. Network group: Allows different network separations to be provide
   4. NIC Mappings – how NICs on different server types are to be configured, e.g. bonded
   5. Disk layout – how customers want to assign disk to different services, e.g. Sift rings, Cinder back-ends
6. Deploy the cloud:
   1. This step takes as a starting point the set of nodes that have been configured to run hLinux and the data model produced at #5 above
   2. It runs a hierarchical series of ansible playbooks. The ansible scripts are named below if they will be separate and remain the same
   3. The first phase of this is responsible for configuring the operating system setup on all of the nodes to comply with the configuration settings. This includes:
      1. Operating system parameters
      2. Network setup including NIC bonding
      3. Logical volume setup, file system creation and mounting
   4. The next phase installs, configure and runs  the various Openstack services
      1. These are installed on the nodes that match the server roles specified by the customer
      2. Services are installed and configured in a controlled fashion so that core dependencies are met, i.e. the system is built up in a controlled fashion

**Verify the cloud deployment**

1. Run various commands such as nova-list to see running services
2. Create router and network key-pair and ?
3. Create floating IP and assign to proper port
4. Ping the VM running nova? Using floating IP address (?)

**Gavin’s steps. Not sure if the scrpits will need to be run individually by the person installing. Not clear.**

1. Download HP Helion OpenStack 2.0 installer
2. Create Ansible directory in root
3. cd to that directory and run the Ansible playbook, passing in any necessary config files
   1. Apache is installed
   2. Necessary packages and tarballs are downloaded and run
4. Run configuration processor (automatic?)
5. In Ansible directory, run Ansible Cobbler deploy playbook
6. Provision nodes and do PXE boot running cobbler-provision ansible playbook in ansible directory

**Once install has completed (hour or more?)**

1. In ansible directory, run Ansible playbook for cobbler-poweron to power up all the nodes
2. You may ssh into you server to check that it is running

Now nodes are up, have OSs installed

1. In ansible directory, run Ansible playbook for osconfig (10 minutes)
2. In ansible directory, run Ansible playbook for hlm-deploy (check naming)
   1. Rabbit, MySQL, Swift, Keystone, Nova, Glance, Neutron, Cinder, are installed
3. In ansible directory, run Ansible playbook for hlm-cloud-configure to upload images to Glance
4. Optional In ansible directory, run Ansible playbook for cloud-client-setup

**Later: what parts to modify**

If you start with a basic cloud can you make changes such as the below? Does HLM handle them?

NIC bonding/different configuration

More zones

More hardware

Storage modifications

NIC mapping PCI bus and NIC name

Different topology- region definitions. New roles added etc.

Playbooks to discover additional/added hardware?

GUI layer coming for user input?