Red Hat Hyperconverged Infrastructure for Virtualization 1.7

Automating RHHI for Virtualization deployment

Use Ansible to deploy your hyperconverged solution without manual intervention
Red Hat Hyperconverged Infrastructure for Virtualization 1.7 Automating RHHI for Virtualization deployment

Use Ansible to deploy your hyperconverged solution without manual intervention

Laura Bailey
lbailey@redhat.com
Abstract

Read this for information about using Ansible to deploy Red Hat Hyperconverged Infrastructure for Virtualization without needing to watch and tend to the deployment process.
# Table of Contents

CHAPTER 1. ANSIBLE BASED DEPLOYMENT WORKFLOW ........................................... 3

CHAPTER 2. SUPPORT REQUIREMENTS ................................................................. 4
  2.1. OPERATING SYSTEM .............................................................................. 4
  2.2. PHYSICAL MACHINES .......................................................................... 4
  2.3. VIRTUAL MACHINES ............................................................................ 4
  2.4. HOSTED ENGINE VIRTUAL MACHINE .................................................. 5
  2.5. NETWORKING ....................................................................................... 5
  2.6. STORAGE ............................................................................................... 6
    2.6.1. Disks .............................................................................................. 7
    2.6.2. RAID .............................................................................................. 7
    2.6.3. JBOD .............................................................................................. 8
    2.6.4. Logical volumes .............................................................................. 8
    2.6.5. Red Hat Gluster Storage volumes .................................................. 8
    2.6.6. Volume types .................................................................................. 8
  2.7. VIRTUAL DATA OPTIMIZER (VDO) ...................................................... 9
  2.8. SCALING ............................................................................................... 9
  2.9. EXISTING RED HAT GLUSTER STORAGE CONFIGURATIONS ............ 9
  2.10. DISASTER RECOVERY ........................................................................ 9
    2.10.1. Prerequisites for geo-replication ................................................... 10
    2.10.2. Prerequisites for failover and failback configuration ....................... 10
  2.11. ADDITIONAL REQUIREMENTS FOR SINGLE NODE DEPLOYMENTS .... 10
  2.12. INSTALL HOST PHYSICAL MACHINES .............................................. 11
    2.12.1. Installing Red Hat Virtualization Host ........................................... 11
    2.12.2. Enabling software repositories ...................................................... 13

CHAPTER 3. CONFIGURE PUBLIC KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD .......... 14
  3.1. GENERATING SSH KEY PAIRS WITHOUT A PASSWORD ..................... 14
  3.2. COPYING SSH KEYS ........................................................................... 15

CHAPTER 4. SETTING DEPLOYMENT VARIABLES ........................................... 16

CHAPTER 5. EXECUTING THE DEPLOYMENT PLAYBOOK ................................... 21

CHAPTER 6. VERIFY YOUR DEPLOYMENT ....................................................... 22
CHAPTER 1. ANSIBLE BASED DEPLOYMENT WORKFLOW

You can use Ansible to deploy Red Hat Hyperconverged Infrastructure for Virtualization without needing to watch and tend to the deployment process.

The workflow for deploying RHHI for Virtualization using Ansible is as follows.

1. Verify that your planned deployment meets the requirements: **Support requirements**

2. Install the physical machines that will act as hyperconverged hosts: **Installing host physical machines**

3. Configure key-based SSH authentication without a password to allow automatic host configuration: **Configuring public key based SSH authentication without a password**

4. Edit the variable file with details of your environment: **Setting deployment variables**

5. Execute the Ansible playbook to deploy RHHI for Virtualization: **Executing the deployment playbook**

6. **Verify your deployment.**
CHAPTER 2. SUPPORT REQUIREMENTS

Review this section to ensure that your planned deployment meets the requirements for support by Red Hat.

2.1. OPERATING SYSTEM

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) uses Red Hat Virtualization Host 4.3 as a base for all other configuration. Red Hat Enterprise Linux hosts are not supported.

See Requirements in the Red Hat Virtualization Planning and Prerequisites Guide for details on requirements of Red Hat Virtualization.

2.2. PHYSICAL MACHINES

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) requires at least 3 physical machines. Scaling to 6, 9, or 12 physical machines is also supported; see Scaling for more detailed requirements.

Each physical machine must have the following capabilities:

- at least 2 NICs (Network Interface Controllers) per physical machine, for separation of data and management traffic (see Section 2.5, “Networking” for details)
- for small deployments:
  - at least 12 cores
  - at least 64GB RAM
  - at most 48TB storage
- for medium deployments:
  - at least 12 cores
  - at least 128GB RAM
  - at most 64TB storage
- for large deployments:
  - at least 16 cores
  - at least 256GB RAM
  - at most 80TB storage

2.3. VIRTUAL MACHINES

The number of virtual machines that you are able to run on your hyperconverged deployment depends greatly on what those virtual machines do, and what load they are under. Test your workload’s CPU, memory, and throughput requirements and provision your hyperconverged environment accordingly.
See Virtualization limits for Red Hat Virtualization for information about maximum numbers of virtual machines and virtual CPUs, and use the RHHI for Virtualization Sizing Tool for assistance planning your deployment.

2.4. HOSTED ENGINE VIRTUAL MACHINE

The Hosted Engine virtual machine requires at least the following:

- 1 dual core CPU (1 quad core or multiple dual core CPUs recommended)
- 4GB RAM that is not shared with other processes (16GB recommended)
- 25GB of local, writable disk space (50GB recommended)
- 1 NIC with at least 1Gbps bandwidth

For more information, see Requirements in the Red Hat Virtualization 4.3 Planning and Prerequisites Guide.

2.5. NETWORKING

Fully-qualified domain names that are forward and reverse resolvable by DNS are required for all hyperconverged hosts and for the Hosted Engine virtual machine.

If external DNS is not available, for example in an isolated environment, ensure that the /etc/hosts file on each node contains the front and back end addresses of all hosts and the Hosted Engine node.

IPv6 is supported as a Technology Preview in IPv6-only environments (including DNS and gateway addresses). Environments with both IPv4 and IPv6 addresses are not supported.

**IMPORTANT**

Technology Preview features are provided with a limited support scope, as detailed on the Customer Portal: Technology Preview Features Support Scope.

Red Hat recommends usage of separate networks: a **front-end management network** for virtual machine traffic and a **back-end storage network** for gluster traffic and virtual machine migration.

**Red Hat recommends each node to have two Ethernet ports, one for each network**. This ensures optimal performance. For high availability, place each network on a separate network switch. For improved fault tolerance, provide a separate power supply for each switch.

**Front-end management network**

- Used by Red Hat Virtualization and virtual machines.
- Requires at least one 1Gbps Ethernet connection.
- IP addresses assigned to this network must be on the same subnet as each other, and on a different subnet to the back-end storage network.
- IP addresses on this network can be selected by the administrator.

**Back-end storage network**
• Used by storage and migration traffic between hyperconverged nodes.
• Requires at least one 10Gbps Ethernet connection.
• Requires maximum latency of 5 milliseconds between peers.

Network fencing devices that use Intelligent Platform Management Interfaces (IPMI) require a separate network.

If you want to use DHCP network configuration for the Hosted Engine virtual machine, then you must have a DHCP server configured prior to configuring Red Hat Hyperconverged Infrastructure for Virtualization.

**If you want to configure disaster recovery** by using geo-replication to store copies of data:

• Configure a reliable time source.

• Do not use IPv6 addresses.

---

**WARNING**

Bug 1688239 currently prevents IPv6 based geo-replication from working correctly. Do not use IPv6 addresses if you require disaster recovery functionality using geo-replication.

---

**Before you begin the deployment process** determine the following details:

• IP address for a gateway to the hyperconverged host. This address must respond to ping requests.

• IP address of the front-end management network.

• Fully-qualified domain name (FQDN) for the Hosted Engine virtual machine.

• MAC address that resolves to the static FQDN and IP address of the Hosted Engine.

**2.6. STORAGE**

A hyperconverged host stores configuration, logs and kernel dumps, and uses its storage as swap space. This section lists the minimum directory sizes for hyperconverged hosts. Red Hat recommends using the default allocations, which use more storage space than these minimums.

• `/ (root)` - 6GB

• `/home` - 1GB

• `/tmp` - 1GB

• `/boot` - 1GB
- /var - 15GB
- /var/crash - 10GB
- /var/log - 8GB

**IMPORTANT**

Red Hat recommends increasing the size of /var/log to at least 15GB to provide sufficient space for the additional logging requirements of Red Hat Gluster Storage.

Follow the instructions in Growing a logical volume using the Web Console to increase the size of this partition after installing the operating system.

- /var/log/audit - 2GB
- swap - 1GB (see Recommended swap size for details)

Anaconda reserves 20% of the thin pool size within the volume group for future metadata expansion. This is to prevent an out-of-the-box configuration from running out of space under normal usage conditions. Overprovisioning of thin pools during installation is also not supported.

- Minimum Total - 55GB

### 2.6.1. Disks

Red Hat recommends Solid State Disks (SSDs) for best performance. If you use Hard Drive Disks (HDDs), you should also configure a smaller, faster SSD as an LVM cache volume.

Do not host the bricks of a Gluster volume across disks that have different block sizes. Ensure that you verify the block size of any VDO devices used to host bricks before creating a volume, as the default block size for a VDO device changed from 512 bytes in version 1.6 to 4 KB in version 1.7. Check the block size (in bytes) of a disk by running the following command:

```bash
# blockdev --getss <disk_path>
```

### 2.6.2. RAID

RAID5 and RAID6 configurations are supported. However, RAID configuration limits depend on the technology in use.

- SAS/SATA 7k disks are supported with RAID6 (at most 10+2)
- SAS 10k and 15k disks are supported with the following:
  - RAID5 (at most 7+)
  - RAID6 (at most 10+2)

RAID cards must use flash backed write cache.

Red Hat further recommends providing at least one hot spare drive local to each server.
2.6.3. JBOD
As of Red Hat Hyperconverged Infrastructure for Virtualization 1.6, JBOD configurations are fully supported and no longer require architecture review.

2.6.4. Logical volumes
The logical volumes that comprise the **engine** gluster volume must be thick provisioned. This protects the Hosted Engine from out of space conditions, disruptive volume configuration changes, I/O overhead, and migration activity.

The logical volumes that comprise the **vmstore** and optional **data** gluster volumes must be thin provisioned. This allows greater flexibility in underlying volume configuration. If your thin provisioned volumes are on Hard Drive Disks (HDDs), configure a smaller, faster Solid State Disk (SSD) as an lvmcache for improved performance.

2.6.5. Red Hat Gluster Storage volumes
Red Hat Hyperconverged Infrastructure for Virtualization is expected to have 3–4 Red Hat Gluster Storage volumes.

- 1 **engine** volume for the Hosted Engine
- 1 **vmstore** volume for virtual machine operating system disk images
- 1 **data** volume for other virtual machine disk images
- 1 **shared_storage** volume for geo-replication metadata

Separate **vmstore** and **data** volumes are recommended to minimize backup storage requirements. Storing virtual machine data separate from operating system images means that only the **data** volume needs to be backed up when storage space is at a premium, since operating system images on the **vmstore** volume can be more easily rebuilt.

A Red Hat Hyperconverged Infrastructure for Virtualization deployment can contain at most 1 geo-replicated volume.

2.6.6. Volume types
Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) supports only the following volume types at deployment time:

- **Replicated volumes** (3 copies of the same data on 3 bricks, across 3 nodes). These volumes can be expanded into distributed-replicated volumes after deployment.
- **Arbitrated replicated volumes** (2 full copies of the same data on 2 bricks and 1 arbiter brick that contains metadata, across three nodes). These volumes can be expanded into arbitrated distributed-replicated volumes after deployment.
- **Distributed volumes** (1 copy of the data, no replication to other bricks).

Note that arbiter bricks store only file names, structure, and metadata. This means that a three-way arbitrated replicated volume requires about 75% of the storage space that a three-way replicated volume would require to achieve the same level of consistency. However, because the arbiter brick
stores only metadata, a three-way arbitrated replicated volume only provides the availability of a two-way replicated volume.

For more information on laying out arbitrated replicated volumes, see Creating multiple arbitrated replicated volumes across fewer total nodes in the Red Hat Gluster Storage Administration Guide.

2.7. VIRTUAL DATA OPTIMIZER (VDO)

A Virtual Data Optimizer (VDO) layer is supported as of Red Hat Hyperconverged Infrastructure for Virtualization 1.6.

VDO support is limited to new deployments only; do not attempt to add a VDO layer to an existing deployment.

Be aware that the default block size for a VDO device changed from 512 bytes in version 1.6 to 4 KB in version 1.7. Do not host the bricks of a Gluster volume across disks that have different block sizes.

2.8. SCALING

Initial deployments of Red Hat Hyperconverged Infrastructure for Virtualization are either 1 node or 3 nodes.

1 node deployments cannot be scaled.

3 node deployments can be scaled to 6, 9, or 12 nodes using one of the following methods:

1. Add new hyperconverged nodes to the cluster, in sets of three, up to the maximum of 12 hyperconverged nodes.

2. Create new Gluster volumes using new disks on new or existing nodes.

3. Expand existing Gluster volumes to span 6, 9, or 12 nodes using new disks on new or existing nodes.

Using the web console, you cannot create a volume that spans more than 3 nodes at creation time; you must create a 3-node volume first and then expand it across more nodes as necessary.

2.9. EXISTING RED HAT GLUSTER STORAGE CONFIGURATIONS

Red Hat Hyperconverged Infrastructure for Virtualization is supported only when deployed as specified in this document. Existing Red Hat Gluster Storage configurations cannot be used in a hyperconverged configuration. If you want to use an existing Red Hat Gluster Storage configuration, refer to the traditional configuration documented in Configuring Red Hat Virtualization with Red Hat Gluster Storage.

2.10. DISASTER RECOVERY

Red Hat strongly recommends configuring a disaster recovery solution. For details on configuring geo-replication as a disaster recovery solution, see Maintaining Red Hat Hyperconverged Infrastructure for Virtualization: https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure_for_virtualization/1.7/html/maintaining_red_hat_hyperconverged_backup-recovery.
WARNING

Bug 1688239 currently prevents IPv6 based geo-replication from working correctly. Do not use IPv6 addresses if you require disaster recovery functionality using geo-replication.

2.10.1. Prerequisites for geo-replication

Be aware of the following requirements and limitations when configuring geo-replication:

One geo-replicated volume only

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) supports only one geo-replicated volume. Red Hat recommends backing up the volume that stores the data of your virtual machines, as this usually contains the most valuable data.

Two different managers required

The source and destination volumes for geo-replication must be managed by different instances of Red Hat Virtualization Manager.

2.10.2. Prerequisites for failover and failback configuration

Versions must match between environments

Ensure that the primary and secondary environments have the same version of Red Hat Virtualization Manager, with identical data center compatibility versions, cluster compatibility versions, and PostgreSQL versions.

No virtual machine disks in the hosted engine storage domain

The storage domain used by the hosted engine virtual machine is not failed over, so any virtual machine disks in this storage domain will be lost.

Execute Ansible playbooks manually from a separate master node

Generate and execute Ansible playbooks manually from a separate machine that acts as an Ansible master node.

2.11. ADDITIONAL REQUIREMENTS FOR SINGLE NODE DEPLOYMENTS

Red Hat Hyperconverged Infrastructure for Virtualization is supported for deployment on a single node provided that all Support Requirements are met, with the following additions and exceptions.

A single node deployment requires a physical machine with:

- 1 Network Interface Controller
- at least 12 cores
- at least 64GB RAM
- at most 48TB storage

Single node deployments cannot be scaled, and are not highly available.
2.12. INSTALL HOST PHYSICAL MACHINES

Your physical machines need an operating system and access to the appropriate software repositories in order to be used as hyperconverged hosts.

1. Install Red Hat Virtualization Host on each physical machine.

2. Enable the Red Hat Virtualization Host software repository on each physical machine.

2.12.1. Installing Red Hat Virtualization Host

Red Hat Virtualization Host is a minimal operating system designed for setting up a physical machine that acts as a hypervisor in Red Hat Virtualization, or a hyperconverged host in Red Hat Hyperconverged Infrastructure.

Prerequisites

- Ensure that your physical machine meets the requirements outlined in Physical machines.

Procedure

1. Download the Red Hat Virtualization Host ISO image from the Customer Portal:
   b. Click Downloads in the menu bar.
   c. Click Red Hat Virtualization. Scroll up and click Download Latest to access the product download page.
   d. Go to Hypervisor Image for RHV 4.3 and click Download Now.
   e. Create a bootable media device. See Making Media in the Red Hat Enterprise Linux Installation Guide for more information.

2. Start the machine on which you are installing Red Hat Virtualization Host, and boot from the prepared installation media.

3. From the boot menu, select Install RHVH 4.3 and press Enter.

   **NOTE**

   You can also press the Tab key to edit the kernel parameters. Kernel parameters must be separated by a space, and you can boot the system using the specified kernel parameters by pressing the Enter key. Press the Esc key to clear any changes to the kernel parameters and return to the boot menu.

4. Select a language, and click Continue.

5. Select a time zone from the Date & Time screen and click Done.
IMPORTANT

Red Hat recommends using Coordinated Universal Time (UTC) on all hosts. This helps ensure that data collection and connectivity are not impacted by variation in local time, such as during daylight savings time.

6. Select a keyboard layout from the Keyboard screen and click Done.

7. Specify the installation location from the Installation Destination screen.

IMPORTANT

- Red Hat strongly recommends using the Automatically configure partitioning option.

- All disks are selected by default, so deselect disks that you do not want to use as installation locations.

- At-rest encryption is not supported. Do not enable encryption.

- Red Hat recommends increasing the size of /var/log to at least 15GB to provide sufficient space for the additional logging requirements of Red Hat Gluster Storage. Follow the instructions in Growing a logical volume using the Web Console to increase the size of this partition after installing the operating system.

Click Done.

8. Select the Ethernet network from the Network & Host Name screen.
   a. Click Configure… → General and select the Automatically connect to this network when it is available check box.


10. Click Begin Installation.

11. Set a root password and, optionally, create an additional user while Red Hat Virtualization Host installs.

   WARNING
   
   Red Hat strongly recommends not creating untrusted users on Red Hat Virtualization Host, as this can lead to exploitation of local security vulnerabilities.

12. Click Reboot to complete the installation.
NOTE

When Red Hat Virtualization Host restarts, `nodectl check` performs a health check on the host and displays the result when you log in on the command line. The message `node status: OK` or `node status: DEGRADED` indicates the health status. Run `nodectl check` to get more information. The service is enabled by default.

2.12.2. Enabling software repositories

1. Log in to the Web Console.
   Use the management FQDN and port 9090, for example, `https://server1.example.com:9090/`.

2. Navigate to Subscriptions, click Register System, and enter your Customer Portal user name and password.
   The Red Hat Virtualization Host subscription is automatically attached to the system.

3. Click Terminal.

4. Enable the Red Hat Virtualization Host 7 repository to allow later updates to the Red Hat Virtualization Host:

   ```bash
   # subscription-manager repos --enable=rhel-7-server-rhvh-4-rpms
   ```
CHAPTER 3. CONFIGURE PUBLIC KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD

Configure public key based SSH authentication without a password for the root user on the first hyperconverged host to all hosts, including itself. Do this for all storage and management interfaces, and for both IP addresses and FQDNs.

3.1. GENERATING SSH KEY PAIRS WITHOUT A PASSWORD

Generating a public/private key pair lets you use key-based SSH authentication. Generating a key pair that does not use a password makes it simpler to use Ansible to automate deployment and configuration processes.

Procedure

1. Log in to the first hyperconverged host as the root user.
2. Generate an SSH key that does not use a password.
   a. Start the key generation process.

   ```bash
   # ssh-keygen -t rsa
   Generating public/private rsa key pair.
   ```

   b. Enter a location for the key.
   The default location, shown in parentheses, is used if no other input is provided.

   ```bash
   Enter file in which to save the key (/home/username/.ssh/id_rsa): <location>/<keyname>
   ```

   c. Specify and confirm an empty passphrase by pressing Enter twice.

   ```bash
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

   The private key is saved in <location>/<keyname>. The public key is saved in <location>/<keyname>.pub.

   ```bash
   Your identification has been saved in <location>/<keyname>.
   Your public key has been saved in <location>/<keyname>.pub.
   The key fingerprint is SHA256:8BhZageKrLXM99z5f/AM9aPo/KAUd8ZZFPcPFWqK6+Mroot@server1.example.com
   The key's randomart image is:
   +---[ECDSA 256]---+
   | . . + = |
   | + . * . 0...
   | = . * . + .|
   | . + . So o * ..|
   | . o . + = .|
   | 0 00 ..= .|
   | 000...+ |
   | .E+++oo |
   +---[SHA256]-----+
WARNING

Your identification in this output is your private key. Never share your private key. Possession of your private key allows someone else to impersonate you on any system that has your public key.

3.2. COPYING SSH KEYS

To access a host using your private key, that host needs a copy of your public key.

Prerequisites

- Generate a public/private key pair.

Procedure

1. Log in to the first host as the root user.

2. Copy your public key to each host that you want to access, including the host on which you execute the command, using both the front-end and the back-end FQDNs.

   ```
   # ssh-copy-id -i <location>/<keyname>.pub <user>@<hostname>
   ```

   Enter the password for `<user>@<hostname>` when prompted.

WARNING

Make sure that you use the file that ends in `.pub`. Never share your private key. Possession of your private key allows someone else to impersonate you on any system that has your public key.

For example, if you are logged in as the root user on server1.example.com, you would run the following commands:

```bash
# ssh-copy-id -i <location>/<keyname>.pub root@server1front.example.com
# ssh-copy-id -i <location>/<keyname>.pub root@server2front.example.com
# ssh-copy-id -i <location>/<keyname>.pub root@server3front.example.com
# ssh-copy-id -i <location>/<keyname>.pub root@server1back.example.com
# ssh-copy-id -i <location>/<keyname>.pub root@server2back.example.com
# ssh-copy-id -i <location>/<keyname>.pub root@server3back.example.com
```
CHAPTER 4. SETTING DEPLOYMENT VARIABLES

The sample inventory file (/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/gluster_inventory.yml) and the additional variables file (/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/he_gluster_vars.json) needs to be updated as per the requirement.

With these two files as input, Red Hat Hyperconverged Infrastructure for Virtualization deployment can be automated.

Following section describes the variables in gluster_inventory.yml file:

1. Change the directory to:

   ```
   # cd /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment
   ```

2. Make a backup copy of the inventory and vars files.

   ```
   # cp gluster_inventory.yml gluster_inventory.yml.bkp
   # cp he_gluster_vars.json he_gluster_vars.json.bkp
   ```

3. Edit the inventory file.
   Make the following updates to the gluster_inventory.yml file.
   
   a. Add host back-end network FQDNs to the inventory file
      
         • On line 4, replace host1 with the back-end network FQDN of the first host.

         **Lines 3-4 of gluster_inventory.yml**

            ```
            # Host1
            servera.example.com:
            ```

         • On line 71, replace host2 with the back-end network FQDN of the second host.

         **Lines 70-71 of gluster_inventory.yml**

            ```
            #Host2
            serverb.example.com:
            ```

         • On line 138, replace host3 with the back-end network FQDN of the third host.

         **Lines 137-138 of gluster_inventory.yml**

            ```
            #Host3
            serverc.example.com:
            ```

         • On line 235 and 236, replace host2 and host3 with the front-end network FQDN of the second and third host respectively.

         **Lines 235-238 of gluster_inventory.yml**

            ```
            gluster:
            ```

Red Hat Hyperconverged Infrastructure for Virtualization 1.7 Automating RHHI for Virtualization deployment
hosts:
serverb.example.com:
serverc.example.com:

b. If you want to use VDO for deduplication and compression

i. Uncomment the **Dedupe & Compression config** and **With Dedupe & Compression** sections by removing the `#` symbol from the beginning of the following lines.

```
NOTE
Set VDO option emulate512=off, as VDO 4K Native volumes are now supported with RHHI-V from RHV 4.3.8
```

```
#gluster_infra_vdo:
  #- { name: 'vdo_sdb1', device: '/dev/sdb1', logicalsize: '3000G', emulate512: 'off',
      slabsize: '32G',
      #blockmapcachesize: '128M', readcachesize: '20M', readcache: 'enabled',
      writepolicy: 'auto' }
  #- { name: 'vdo_sdb2', device: '/dev/sdb2', logicalsize: '3000G', emulate512: 'off',
      slabsize: '32G',
      #blockmapcachesize: '128M', readcachesize: '20M', readcache: 'enabled',
      writepolicy: 'auto' }
```

```
NOTE
If you have non-VDO disk, add it with vgname 'vg_sdX' and pvname as /dev/sdX.
```

```
#gluster_infra_volume_groups:
  #- vgname: vg_sdb1
    #pvname: /dev/mapper/vdo_sdb1
  #- vgname: vg_sdb2
    #pvname: /dev/mapper/vdo_sdb2
```

The following table shows the VDO variables and their meaning.

**Table 4.1. VDO Values and their meaning**

<table>
<thead>
<tr>
<th>VDO Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>VDO volume name to be used.</td>
</tr>
<tr>
<td>logicalsize</td>
<td>Logical size of the VDO volume. This value is 10 times the size of the physical disk.</td>
</tr>
<tr>
<td>device</td>
<td>Disk name on which VDO volume to created.</td>
</tr>
<tr>
<td>VDO Variable</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>emulate512</td>
<td>Value of on makes the VDO device as 512e storage volume. NOTE: From RHV 4.3.8, 4KN devices are supported and this option should be off.</td>
</tr>
<tr>
<td>slabsize</td>
<td>VDO slab size. If VDO logical size &gt;= 1000G then slabsize is 32G else slabsize is 2G.</td>
</tr>
</tbody>
</table>

The following VDO variable values are as per Red Hat recommendation and are treated as constants:

- blockmapcachesize - 128M
- readcache - enabled
- readcachesize - 20M
- writepolicy - auto

ii. Comment out the **Without Dedupe & Compression** section by adding a `#` to the beginning of each line.

```bash
# Without Dedupe & Compression
gluster_infra_volume_groups:
  - vgname: vg_sdb1
    pvname: /dev/sdb1
  - vgname: vg_sdb2
    pvname: /dev/sdb2
```

iii. Define the mount point for the bricks.
Under **gluster_infra_mount_devices** create an entry corresponding to the number of bricks.

The following example shows two entries: **engine** and **data** mounted under `/gluster_bricks/engine` and `/gluster_bricks/data` respectively.

`lvname` corresponds to the name of the Logical Volume and `vgname` corresponds to Volume Group in which the Logical Volume (LV) is created.

```bash
gluster_infra_mount_devices:
  - path: /gluster_bricks/engine
    lvname: gluster_lv_engine
    vgname: vg_sdb1
  - path: /gluster_bricks/data
    lvname: gluster_lv_data
    vgname: vg_sdb2
```

iv. Define the thinpools
Red Hat recommends placing the **engine** brick in thickly provisioned Logical Volume (LV) of size 100GB.
gluster_infra_thick_lvs:
- vgname: vg_sdb1
  lvname: gluster_lv_engine
  size: 100G

Rest all the bricks are recommended to be a thinly provisioned Logical Volume (thin-LV).

The number of thinpools information should be defined under **gluster_infra_thinpools**.

```
gluster_infra_thinpools:
- {vgname: 'vg_sdb2', thinpoolname: 'thinpool_vg_sdb2', thinpoolsize: '500G',
  poolmetadatasize: '16G'}
```

**thinpoolsize**

Sum of sizes of all Logical Volumes (LV) to be created on that Volume group (VG). In case of VDO enabled, **thinpoolsize** is ten times the sum of sizes of all Logical Volumes to be created on that Volume Group.

Recommended values for **poolmetadatasize** is 16GB and that should be considered exclusive of **thinpoolsize**.

v. Define LVM cache.

LVM cache is considered for improved performance. The following example under **gluster_infra_cache_vars** enables LVM cache.

```
gluster_infra_cache_vars:
- vgname: gluster_vg_sdb
  cachedisk: /dev/sdb,/dev/sde
  cachelvname: cachelv_thinpool_sdb
  cachethinpoolname: gluster_thinpool_sdb
  cachelvsize: '250G'
  cachemode: writethrough
```

Variables used in the above example and their meaning.

**vgname**

Volume Group (VG) with the slow HDD device that needs caching.

**cachedisk**

Values of slow HDD and fast SSD separated by comma.

In the above example, /dev/sdb is the slow HDD and /dev/sde is fast SSD.

**cachelvname**

LV cache name.

**cachethinpoolname**

Thinpool to which the fast SSD is to be attached.

**cachelvsize**

Size of cache data LV. Make sure to leave 0.1% of total size of SSD for cache metadata logical volume.

**cachemode**
4. Edit the hosted engine variables file.
Update the following values in the **he_gluster_vars.json** file.

**he_appliance_password**
The root password of the host machine.

**he_admin_password**
The password for the root account of the Administration Portal.

**he_domain_type**
glusterfs - There is no need to change this value.

**he_fqdn**
The fully qualified domain name for the Hosted Engine virtual machine.

**he_vm_mac_addr**
A valid MAC address for the Hosted Engine virtual machine.

**he_default_gateway**
The IP address of the default gateway server.

**he_mgmt_network**
The name of the management network. The default value is **ovirtmgmt**.

**he_host_name**
The short name of this host.

**he_storage_domain_name**
HostedEngine

**he_storage_domain_path**
/engine

**he_storage_domain_addr**
The IP address of this host on the storage network.

**he_mount_options**
backup-volfile-servers=<host2-ip-address>:<host3-ip-address>, with the appropriate IP addresses inserted in place of <host2-ip-address> and <host3-ip-address>.

**he_bridge_if**
The name of the interface to be used as the bridge.

**he_enable_hc_gluster_service**
true

If you are using an isolated environment, add the **he_network_test** variable and specify a value of **ping** to ensure that your hosts can be found.
CHAPTER 5. EXECUTING THE DEPLOYMENT PLAYBOOK

1. Change into the playbooks directory on the first node.

   ```sh
   # cd /etc/ansible/roles/gluster_ansible/playbooks/hc-ansible-deployment
   ```

2. Run the following command as the root user to start the deployment process.

   ```sh
   # ansible-playbook -i gluster_inventory.yml hc_deployment.yml --extra-vars='@he_gluster_vars.json'
   ```
CHAPTER 6. VERIFY YOUR DEPLOYMENT

After deployment is complete, verify that your deployment has completed successfully.


2. Log in using the administrative credentials added during hosted engine deployment. When login is successful, the Dashboard appears.
3. Verify that your cluster is available.

**Administration Console Dashboard - Clusters**

4. Verify that at least one host is available.
   If you provided additional host details during Hosted Engine deployment, 3 hosts are visible here, as shown.

**Administration Console Dashboard - Hosts**

a. Click **Compute → Hosts**.

b. Verify that all hosts are listed with a **Status of Up**.

**Administration Console - Hosts**
5. Verify that all storage domains are available.
   
a. Click **Storage → Domains**.

b. Verify that the **Active** icon is shown in the first column.

**Administration Console - Storage Domains**