# Red Hat Hyperconverged Infrastructure 1.1

## Deploying Red Hat Hyperconverged Infrastructure

Instructions for deploying Red Hat Hyperconverged Infrastructure

Last Updated: 2018-04-18

# Red Hat Hyperconverged Infrastructure 1.1 Deploying Red Hat Hyperconverged Infrastructure

Instructions for deploying Red Hat Hyperconverged Infrastructure

Laura Bailey
lbailey@redhat.com

## Legal Notice

## Abstract

This document outlines how to deploy Red Hat Hyperconverged Infrastructure (RHHI) across three physical machines, using Red Hat Gluster Storage 3.3.1 and Red Hat Virtualization 4.1.8. This creates a discrete cluster for use in remote office branch office (ROBO) environments, where a remote office synchronizes data to a central data center on a regular basis, but can remain fully functional if connectivity to the central data center is lost.

# Table of Contents

# CHAPTER 1. KNOWN ISSUES

This section documents unexpected behavior known to affect Red Hat Hyperconverged Infrastructure (RHHI).

**BZ#1395087 - Virtual machines pause indefinitely when network unavailable**

When the network for Gluster and migration traffic becomes unavailable, virtual machines performing I/O become inaccessible, and cannot complete migration to another node until the hypervisor reboots. This is expected behavior for current fencing and migration methods. There is currently no workaround for this issue.

**BZ#1401969 - Arbiter brick becomes heal source**

When data bricks in an arbiter volume are taken offline and brought back online one at a time, the arbiter brick is incorrectly identified as the source of correct data when healing the other bricks. This results in virtual machines being paused, because arbiter bricks contain only metadata. There is currently no workaround for this issue.

**BZ#1425767 - Sanity check script does not fail**

The sanity check script sometimes returns zero (success) even when disks do not exist, or are not empty. Since the sanity check appears to succeed, gdeploy attempts to create physical volumes and fails. To work around this issue, ensure that the `disk` value in the gdeploy configuration file is correct, and that the disk has no partitions or labels, and retry deployment.

**BZ#1432326 - Associating a network with a host makes network out of sync**

When the Gluster network is associated with a Red Hat Gluster Storage node’s network interface, the Gluster network enters an out of sync state. To work around this issue, click the **Management** tab that corresponds to the node and click **Refresh Capabilities**.

**BZ#1434105 - Live Storage Migration failure**

Live migration from a Gluster-based storage domain fails when I/O operations are still in progress during migration. There is currently no workaround for this issue.

# PART I. PLAN

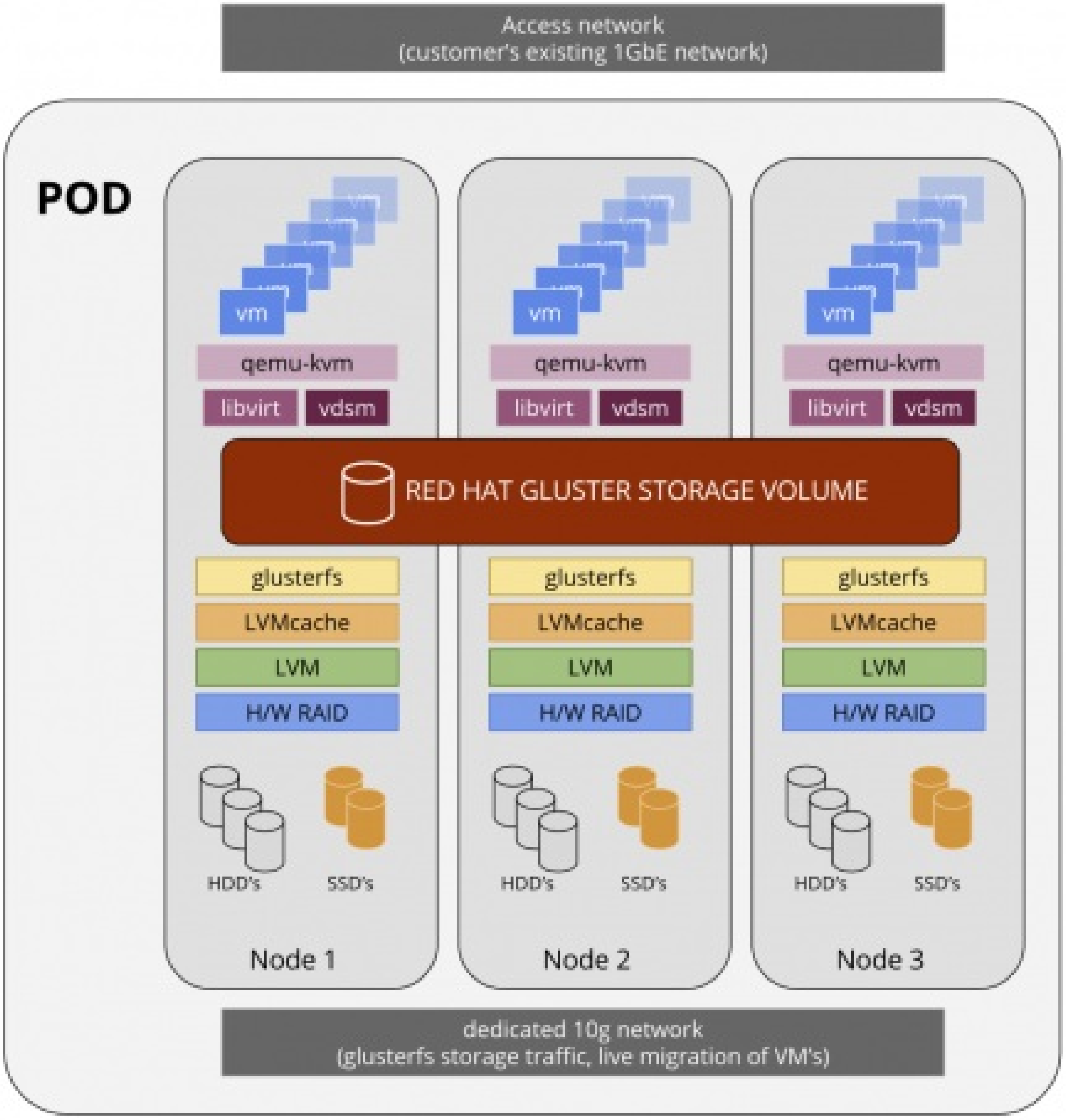
# CHAPTER 2. ARCHITECTURE

Red Hat Hyperconverged Infrastructure (RHHI) combines compute, storage, networking, and management capabilities in one deployment.

RHHI is deployed across three physical machines to create a discrete cluster or *pod* using Red Hat Gluster Storage 3.3.1 and Red Hat Virtualization 4.1.8.

The dominant use case for this deployment is in remote office branch office (ROBO) environments, where a remote office synchronizes data to a central data center on a regular basis, but does not require connectivity to the central data center to function.

The following diagram shows the basic architecture of a single cluster.


Access network
(customer's existing 1GbE network)
POD
vm
qemu-kvm
libvirt
vdsm
RED HAT GLUSTER STORAGE VOLUME
glusterfs
LVMcache
LVM
H/W RAID
HDD's
SSD's
Node 1
Node 2
Node 3
dedicated 10g network
(glusterfs storage traffic, live migration of VM's)

# CHAPTER 3. SUPPORT REQUIREMENTS

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

## 3.1. OPERATING SYSTEM

Red Hat Hyperconverged Infrastructure (RHHI) is supported only on Red Hat Virtualization Host 4.1. Use Red Hat Virtualization Host 4.1 as a base for all other configuration.

See the Red Hat Virtualization *Planning and Prerequisites Guide*for details on requirements of Red Hat Virtualization: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/planning_and_prerequisites_guide/requirements

## 3.2. PHYSICAL MACHINES

Red Hat Hyperconverged Infrastructure (RHHI) requires **at least 3 physical machines**. Scaling to 6 physical machines or 9 physical machines is also supported.

Each physical machine must have the following capabilities.

- at least 2 NICs per physical machine, for separation of data and management traffic (see Section 3.4, “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

## 3.3. VIRTUAL MACHINES

Each virtual machine can have at most 4 virtual CPUs and 2TB virtual disk space.

The supported number of virtual machines depends on their size and resource usage.

## 3.4. NETWORKING

**Each node requires 3 x 1 Gigabit Ethernet ports.** To enable high availability, these must be split across two network switches. Ensuring that switches have separate power supplies further improves fault tolerance.

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

**Client and management traffic in the cluster must be separated.** This means that Red Hat Hyperconverged Infrastructure requires two separate networks:

**A front-end management network**

This network is used by Red Hat Virtualization and virtual machines.

- This network should be capable of transmitting at Gigabit Ethernet speeds.
- IP addresses assigned to this network can be selected by the administrator, but must be on the same subnet as each other.
- IP addresses assigned to this network must not be in the same subnet as the back-end storage and migration network.

**A back-end storage network**

This network is used for storage and migration traffic between storage peers.

- Red Hat recommends a 10Gbps network for the back-end storage network.
- Red Hat Gluster Storage requires a maximum latency of 5 milliseconds between peers.

A separate network is required for network fencing devices using IPMI.

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.

If you want to use geo-replication to store copies of data for disaster recovery purposes, a reliable time source is required.

## 3.5. STORAGE

### 3.5.1. Disks

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

#### 3.5.1.1. RAID

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+1)
  - 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.

### 3.5.2. JBOD

JBOD configurations require architecture review. Contact your Red Hat representative for details.

### 3.5.3. 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.

### 3.5.4. Red Hat Gluster Storage Volumes

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

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

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

### 3.5.5. Volume Types

Red Hat Hyperconverged Infrastructure (RHHI) supports only **replicated** or **arbitrated replicated** volume types.

The arbitrated replicated volume type carries the following additional limitations:

- Arbitrated replicated volumes are supported on the first three nodes only.
- Arbitrated replicated volumes must use `replica 3 arbiter 1` configuration.
- Arbiter bricks do not store file data; they only store 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 further details, see the Red Hat Gluster Storage 3.3 *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html-single/administration_guide/#Creating_Arbitrated_Replicated_Volumes

## 3.6. SUPPORT LIMITATIONS

- One arbitrated replicated volume is supported as part of the initial deployment of Red Hat Hyperconverged Infrastructure (RHHI). Expanding this arbitrated replicated volume is not supported. Adding additional arbitrated replicated volumes is not supported.

- Arbitrated replicated volumes are currently supported only in a `replica 3 arbiter 1` configuration.

For further details about these volume types, see the Red Hat Gluster Storage *Administration Guide*:

- https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/sect-creating_replicated_volumes

- https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/creating_arbitrated_replicated_volu

## 3.7. 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*: https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/maintaining_red_hat_hyperconverged_infrastructu recovery-geo-rep.

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

- Red Hat Hyperconverged Infrastructure (RHHI) supports only one geo-replicated volume; Red Hat recommends backing up the volume that stores the data of your virtual machines.

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

# PART II. DEPLOY

# CHAPTER 4. DEPLOYMENT WORKFLOW

The workflow for deploying Red Hat Hyperconverged Infrastructure (RHHI) is as follows:

1. Verify that your planned deployment meets Requirements: Chapter 3, *Support Requirements*.

2. Install the physical machines that will act as virtualization hosts: Chapter 5, *Install Host Physical Machines*.

3. Configure key-based SSH authentication to enable automated configuration of the hosts: Chapter 6, *Configure Public Key based SSH Authentication*.

4. Configure Red Hat Gluster Storage on the physical hosts using the Cockpit UI: Chapter 7, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI*

5. Deploy the Hosted Engine using the Cockpit UI: Chapter 8, *Deploy the Hosted Engine using the Cockpit UI*.

6. Configure the Red Hat Gluster Storage nodes using the Red Hat Virtualization management UI: Chapter 9, *Configure Red Hat Gluster Storage as a Red Hat Virtualization storage domain*.

# CHAPTER 5. INSTALL HOST PHYSICAL MACHINES

Install Red Hat Virtualization Host 4.1 on your three physical machines. See the following section for details about installing a virtualization host: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/installation_guide/red_hat_virtualization_hosts.

# CHAPTER 6. CONFIGURE PUBLIC KEY BASED SSH AUTHENTICATION

From the first virtualization host, configure Public Key authentication based SSH for the root user to all three virtualization hosts.

Ensure that you use the hostnames or IP addresses associated with the back-end management network.

See the Red Hat Enterprise Linux 7 **Installation Guide** for more details: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/s1-ssh-configuration.html#s2-ssh-configuration-keypairs.

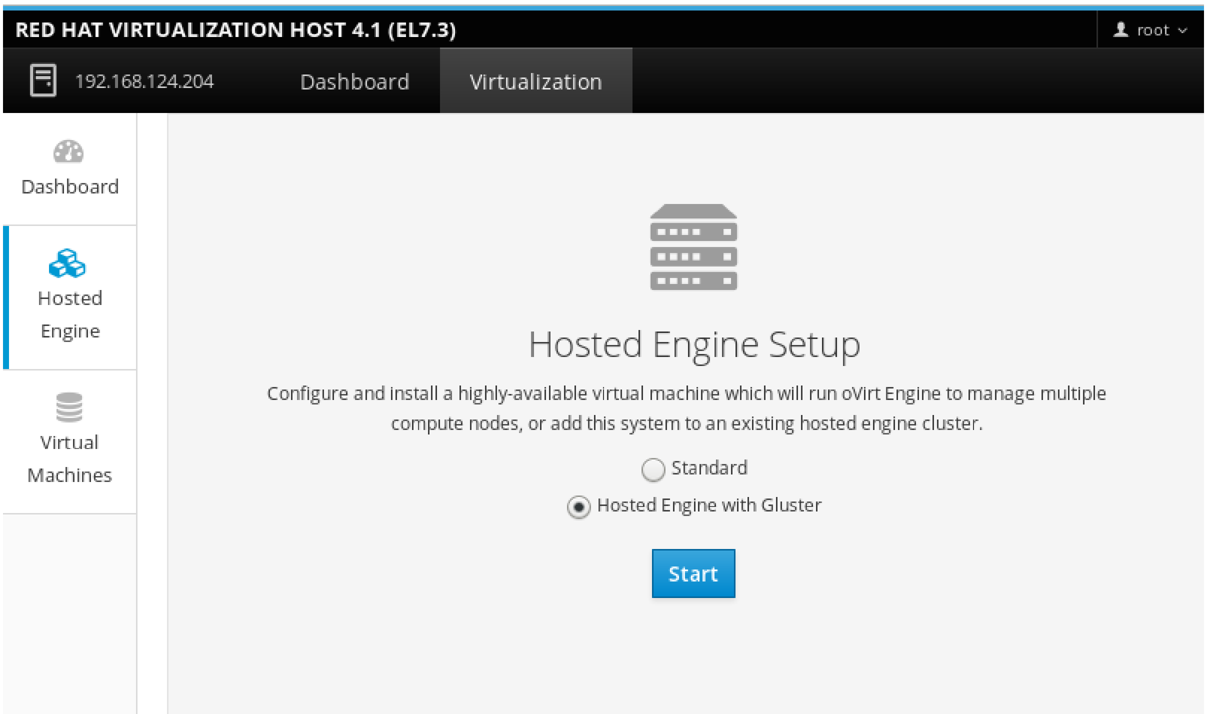
# CHAPTER 7. CONFIGURE RED HAT GLUSTER STORAGE FOR HOSTED ENGINE USING THE COCKPIT UI

### IMPORTANT

Ensure that disks specified as part of this deployment process do not have any partitions or labels.

1. Log into the Cockpit UI
   Browse to the Cockpit management interface of the first virtualization host, for example, **https://node1.example.com:9090/**, and log in with the credentials you created in Chapter 5, *Install Host Physical Machines*.

2. Start the deployment wizard
   Click Virtualization → Hosted Engine, select **Hosted Engine with Gluster** , and click Start. The deployment wizard appears.



Hosted Engine Setup

Configure and install a highly-available virtual machine which will run oVirt Engine to manage multiple compute nodes, or add this system to an existing hosted engine cluster.

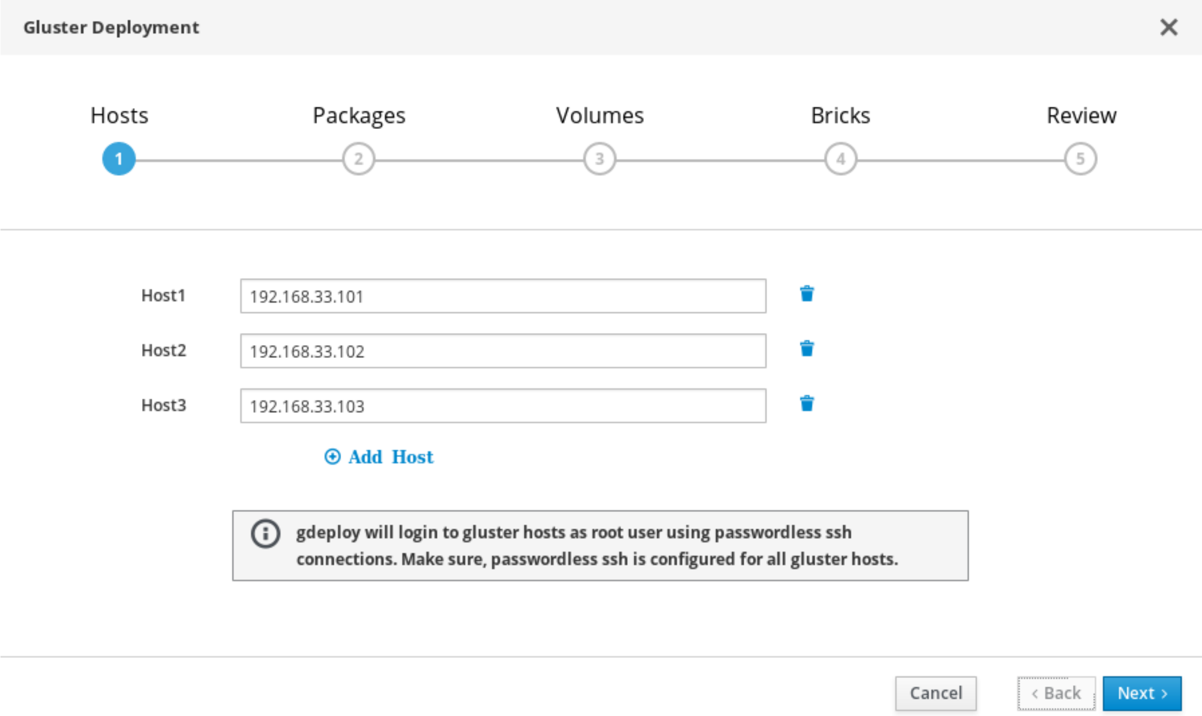
○ Standard

● Hosted Engine with Gluster

3. Deployment wizard: Specify storage hosts
   Specify the back-end gluster network addresses (not the management network addresses) of the three virtualization hosts. The virtualization host that can SSH using key pairs should be listed first, as it is the host that will run gdeploy and the hosted engine.

### NOTE

If you plan to create an arbitrated replicated volume, ensure that you specify the host with the arbiter brick as **Host3** on this screen.



Host1 192.168.33.101

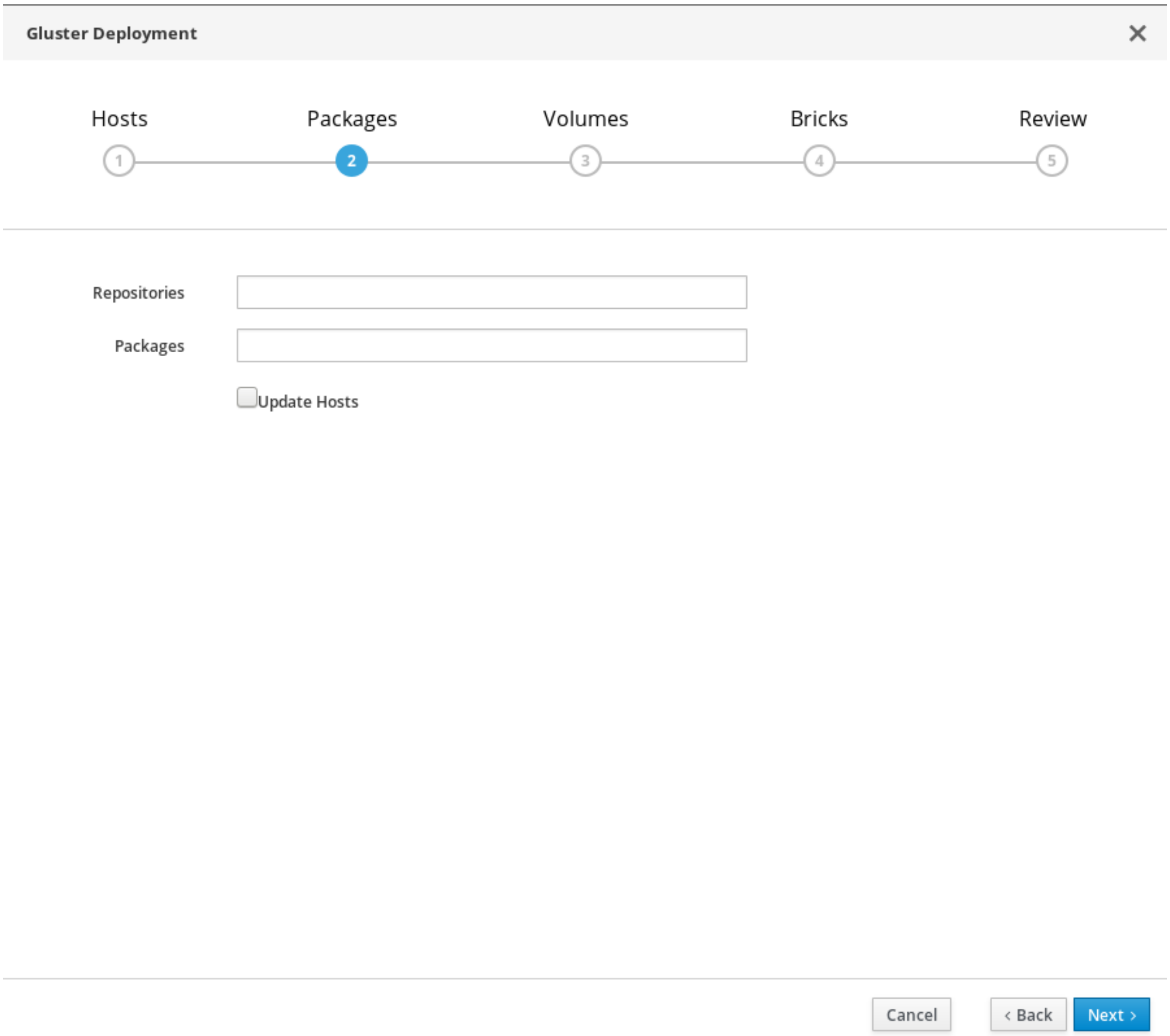
Host2 192.168.33.102

Host3 192.168.33.103

gdeploy will login to gluster hosts as root user using passwordless ssh connections. Make sure, passwordless ssh is configured for all gluster hosts.

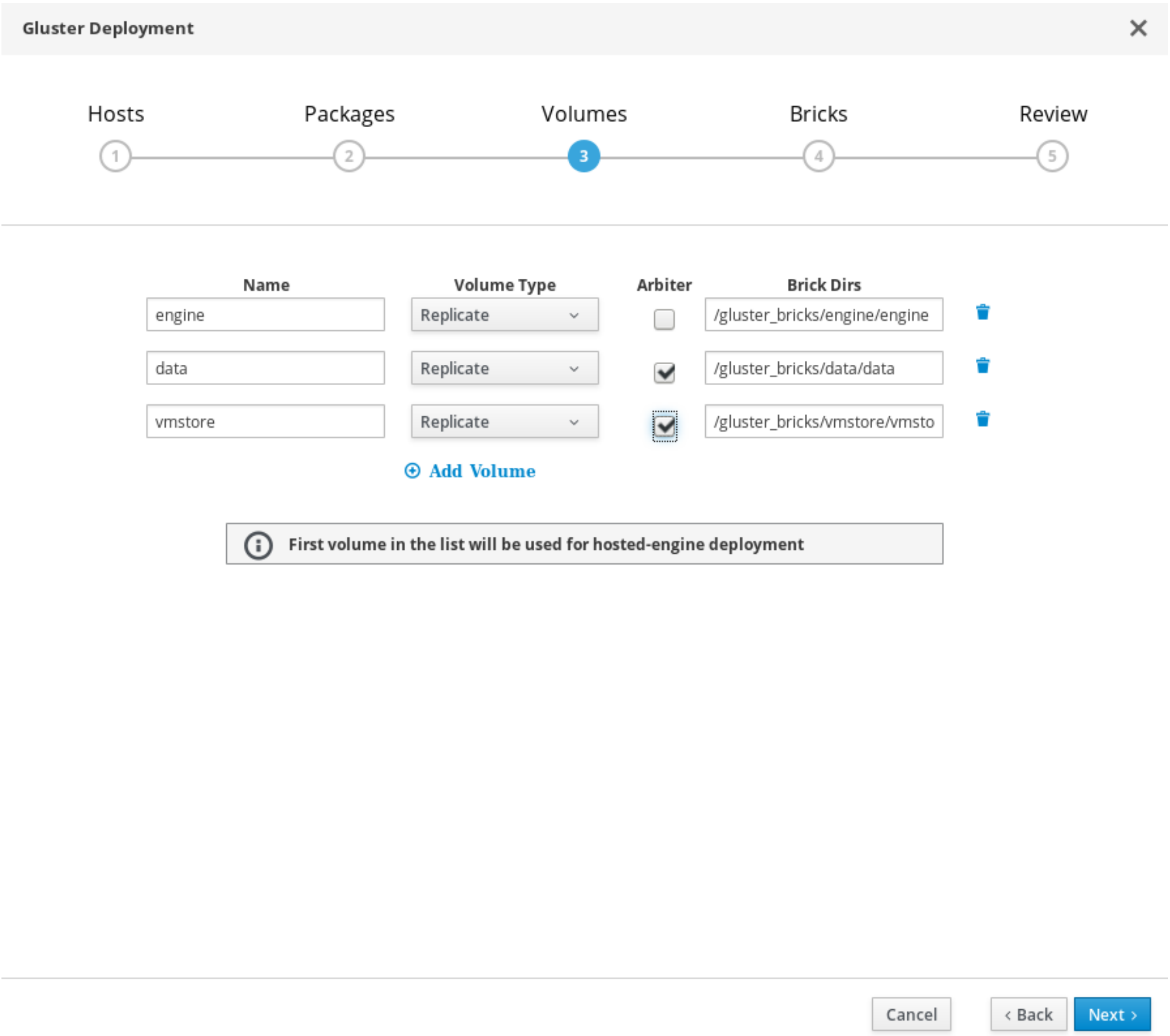
Click Next.

4. Deployment wizard: Specify packages
   There is no need to install packages. Delete any values from these fields and uncheck the boxes.



Click Next.

5. Deployment wizard: Specify volumes
   Specify the volumes to be created.



| Name | Volume Type | Arbiter | Brick Dirs |
|---|---|---|---|
| engine | Replicate | ☐ | /gluster_bricks/engine/engine |
| data | Replicate | ☑ | /gluster_bricks/data/data |
| vmstore | Replicate | ☑ | /gluster_bricks/vmstore/vmsto |

First volume in the list will be used for hosted-engine deployment

**Name**

Specify the name of the volume to be created.

**Volume Type**

Specify a **Replicate** volume type. Only replicated volumes are supported for this release.

**Arbiter**

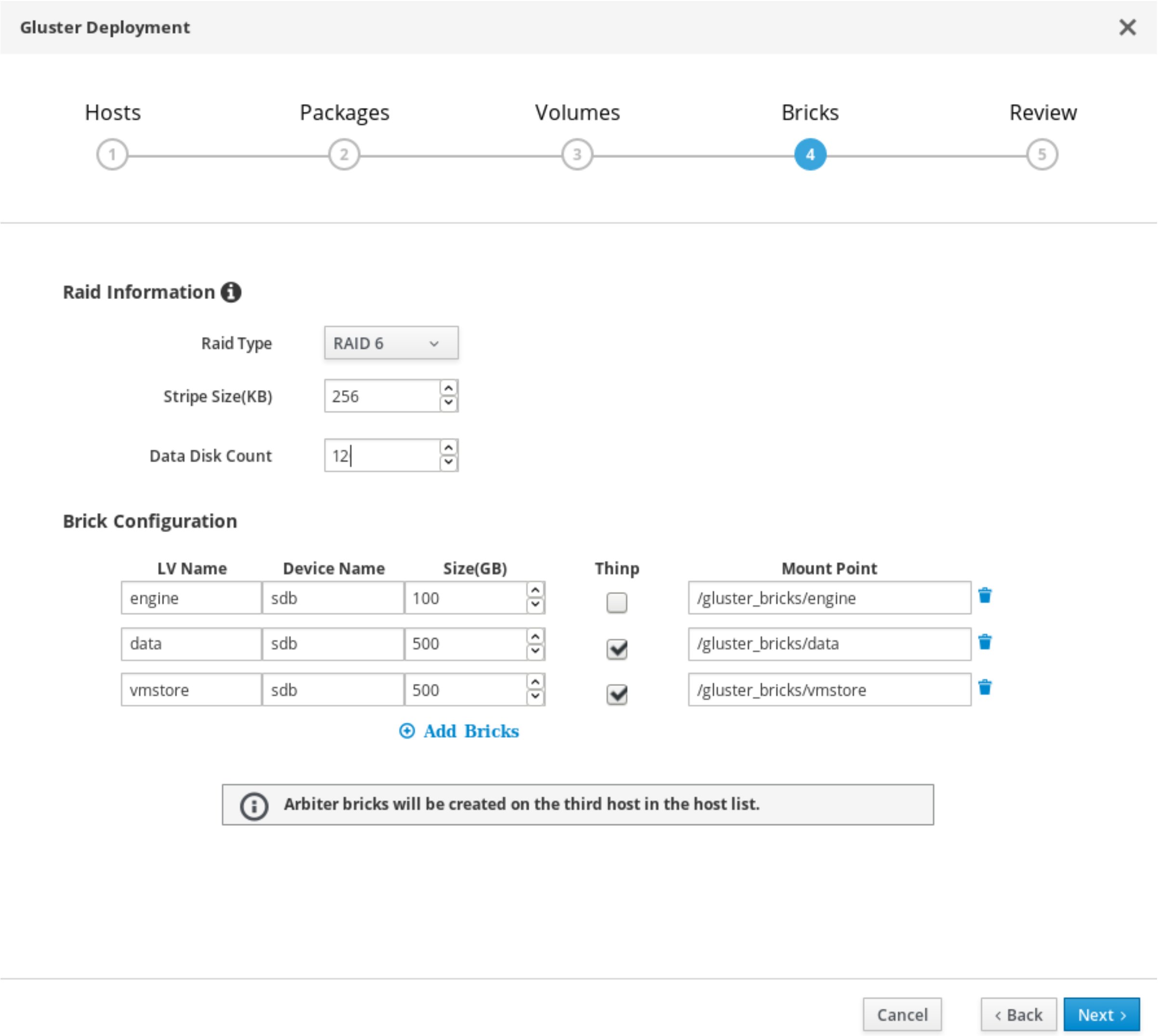
Specify whether to create the volume with an arbiter brick. If this box is checked, the third disk stores only metadata.

**Brick Dirs**

The directory that contains this volume’s bricks.

The default values are correct for most installations.

6. Deployment wizard: Specify bricks
   Specify the bricks to be created.



Raid Information

Raid Type: RAID 6

Stripe Size(KB): 256

Data Disk Count: 12

Brick Configuration

| LV Name | Device Name | Size(GB) | Thinp | Mount Point |
|---|---|---|---|---|
| engine | sdb | 100 | □ | /gluster_bricks/engine |
| data | sdb | 500 | ☑ | /gluster_bricks/data |
| vmstore | sdb | 500 | ☑ | /gluster_bricks/vmstore |

⊕ Add Bricks

Arbiter bricks will be created on the third host in the host list.

**LV Name**

Specify the name of the logical volume to be created.

**Device**

Specify the raw device you want to use. Red Hat recommends an unpartitioned device.

**Size**

Specify the size of the logical volume to create in GB. Do not enter units, only the number.

**Mount Point**

Specify the mount point for the logical volume. This should match the brick directory that you specified on the previous page of the wizard.

**Thinp**

Specify whether to provision the volume thinly or not. Note that thick provisioning is recommended for the **engine** volume.

**RAID**

Specify the RAID configuration to use. This should match the RAID configuration of your host. Supported values are **raid5**, **raid6**, and **jbod**.

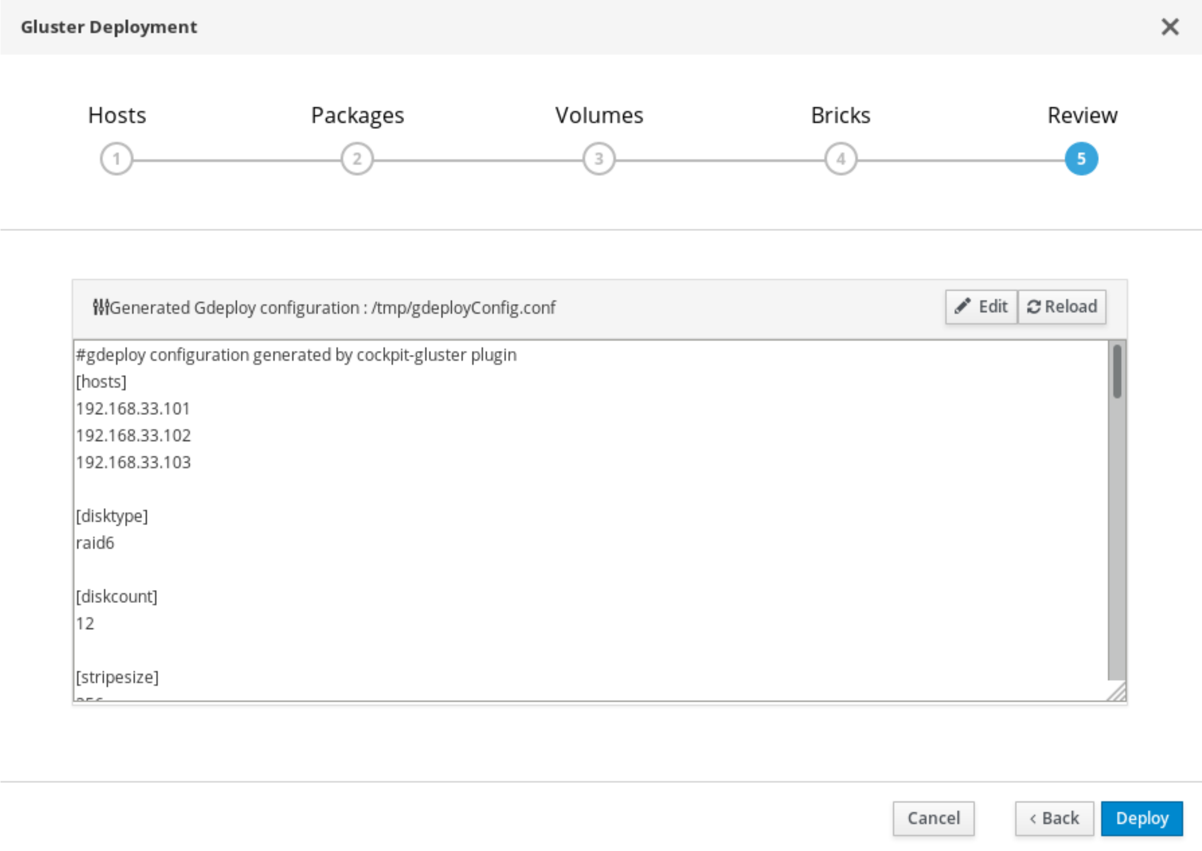
**Stripe Size**

Specify the RAID stripe size in KB. Do not enter units, only the number. This can be ignored for JBOD configurations.

**Disk Count**

Specify the number of data disks in a RAID volume. This can be ignored for JBOD configurations.

7. Deployment wizard: Review and edit configuration



```
#gdeploy configuration generated by cockpit-gluster plugin
[hosts]
192.168.33.101
192.168.33.102
192.168.33.103

[disktype]
raid6

[diskcount]
12

[stripesize]
```

a. Click **Edit** to begin editing the generated **gdeployConfig.conf** file.

b. (Optional) Configure Transport Layer Security (TLS/SSL)
This can be configured during or after deployment. If you want to configure TLS/SSL encryption as part of deployment, see one of the following sections:

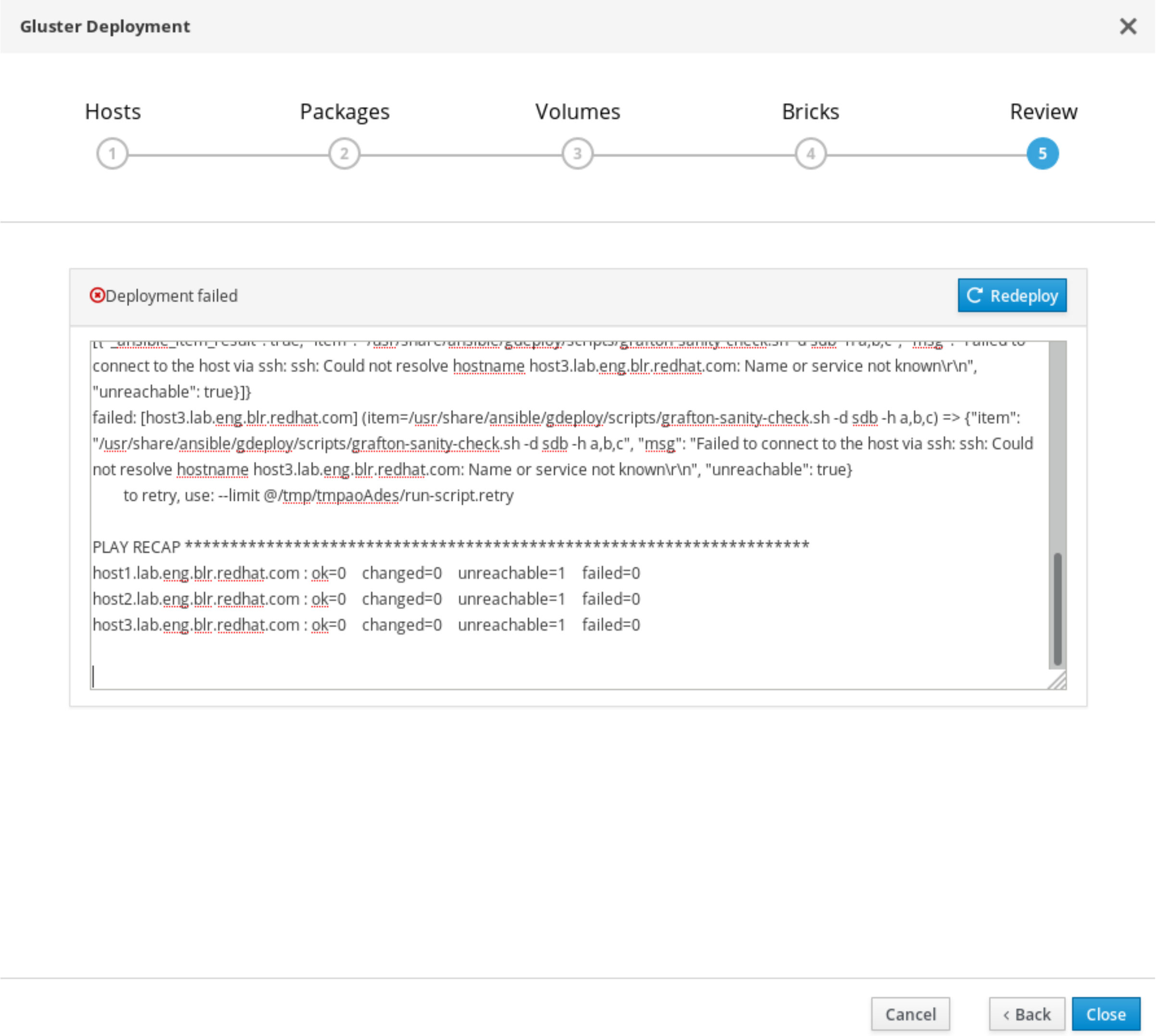
- Section A.1, “Configuring TLS/SSL during deployment using a Certificate Authority signed certificate”

- Section A.2, “Configuring TLS/SSL encryption during deployment using a self signed certificate”

c. Review the configuration file
When you are satisfied with the configuration details, click Save and then click Deploy.

8. Wait for deployment to complete
You can watch the deployment process in the text field as the gdeploy process runs using the generated configuration file.

9. In case of deployment failure



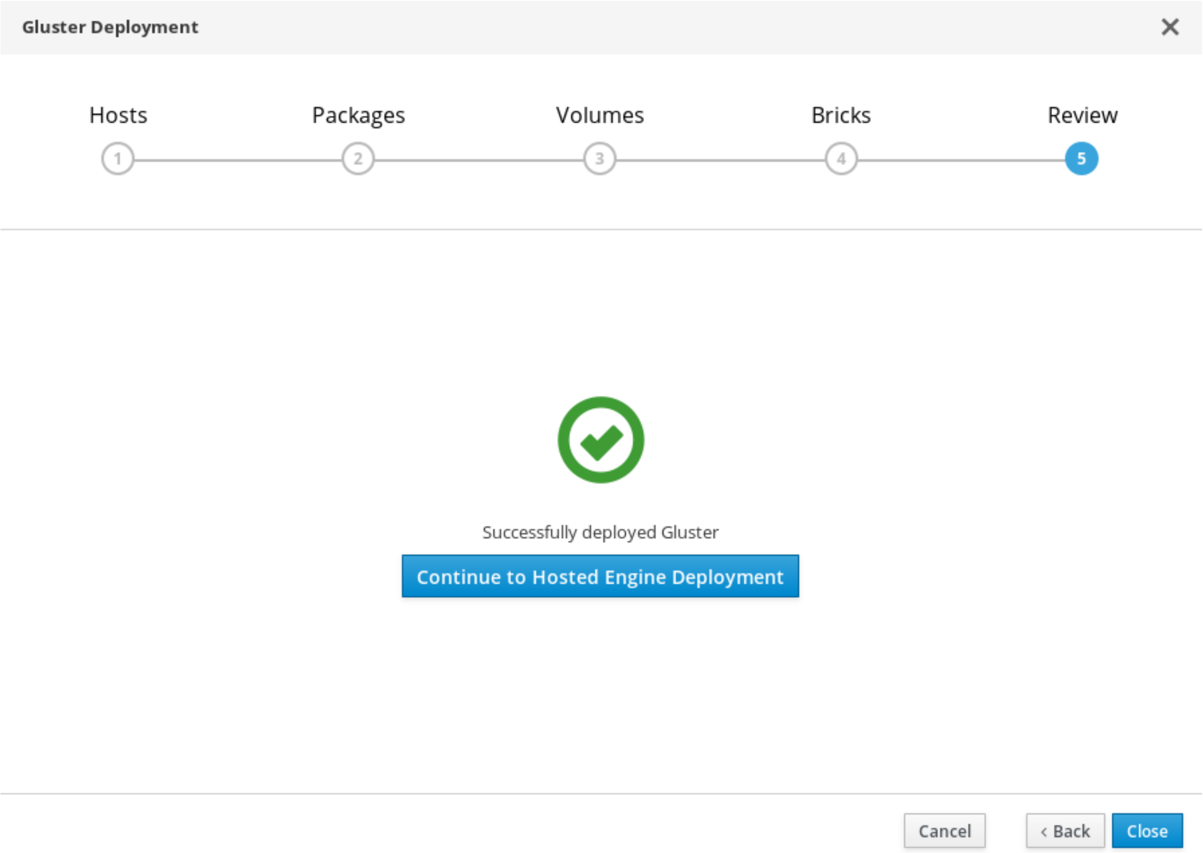
```
[{"_ansible_item_result": true, "item": "/usr/share/ansible/gdeploy/scripts/grafton-sanity-check.sh -d sdb -h a,b,c", "msg": "Failed to
connect to the host via ssh: ssh: Could not resolve hostname host3.lab.eng.blr.redhat.com: Name or service not known\r\n",
"unreachable": true}]}
failed: [host3.lab.eng.blr.redhat.com] (item=/usr/share/ansible/gdeploy/scripts/grafton-sanity-check.sh -d sdb -h a,b,c) => {"item":
"/usr/share/ansible/gdeploy/scripts/grafton-sanity-check.sh -d sdb -h a,b,c", "msg": "Failed to connect to the host via ssh: ssh: Could
not resolve hostname host3.lab.eng.blr.redhat.com: Name or service not known\r\n", "unreachable": true}
      to retry, use: --limit @/tmp/tmpaoAdes/run-script.retry

PLAY RECAP ********************************************************************************
host1.lab.eng.blr.redhat.com : ok=0    changed=0    unreachable=1    failed=0
host2.lab.eng.blr.redhat.com : ok=0    changed=0    unreachable=1    failed=0
host3.lab.eng.blr.redhat.com : ok=0    changed=0    unreachable=1    failed=0
```

If deployment fails, click the **Redeploy** button. This returns you to the *Review and edit configuration* tab so that you can correct any issues in the generated configuration file before reattempting deployment.

It may be necessary to clean up previous deployment attempts before you try again. Follow the steps in Chapter 14, *Cleaning up Automated Red Hat Gluster Storage Deployment Errors*to clean up previous attempts.

10. The deployment script completes and displays **Successfully deployed gluster.**



Successfully deployed Gluster

# CHAPTER 8. DEPLOY THE HOSTED ENGINE USING THE COCKPIT UI

This section shows you how to deploy the Hosted Engine using the Cockpit UI. Following this process results in Red Hat Virtualization Manager running as a virtual machine on the first physical machine in your deployment. It also configures a Default cluster comprised of the three physical machines, and enables Red Hat Gluster Storage functionality and the **virtual-host** *tuned* performance profile for each machine in the cluster.

1. Navigate to the Hosted Engine deployment wizard
   After completing Chapter 7, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI*, click **Continue to Hosted Engine Deployment** to go to the wizard.

2. Agree to the installation

   ```
   Continuing will configure this host for serving as hypervisor
   and create a VM where you have to install the engine afterwards.
   Are you sure you want to continue?
   ```

   Type *Yes* in the field provided. Click **Next** and wait for the environment to be set up.

3. Answer deployment questions when prompted
   Provide answers to the hosted engine deployment wizard prompts to install and configure the hosted engine. You can press **Ctrl+D** at any time to halt the process.

   **Table 8.1. Table Hosted Engine deployment wizard prompts**

| Prompt text | Action |
|---|---|
| Do you wish to install ovirt-engine-appliance rpm? | Enter **Yes**. |
| Do you want to configure this host and its cluster for gluster? | Enter **Yes**. |
| iptables was detected on your computer, do you wish setup to configure it? | Enter **No**. |
| Please indicate a pingable gateway IP address | Enter the IP address of the front-end gateway server. |
| Please indicate a nic to set ovirtmgmt bridge on | Enter the front-end management network address. |
| The following appliance have been found on your system:<br>[1] The RHV-M Appliance image (OVA) - 4.1.20170328.1.el7ev [2] Directly select an OVA file.<br>Please select an appliance | Enter the appropriate number for the hosted engine appliance (usually *1*). |

| Prompt text | Action |
|---|---|
| Would you like to use cloud-init to customize the appliance on first boot | Enter **Yes**. |
| Would you like to generate on-fly a cloud-init ISO image (of no-cloud type) or do you have an existing one? | Enter **Generate**. |
| Please provide the FQDN you would like to use for the engine appliance. Note: This will be the FQDN of the engine VM you are now going to launch, it should not point to the base host or to any other existing machine. Engine VM FQDN: | Enter the fully-qualified domain name that you want to use for your hosted engine (for example, **engine.example.com**). |
| Please provide the domain name you would like to use for the engine appliance. Engine VM domain: | Verify that the detected domain name is correct. |
| Automatically execute engine-setup on the engine appliance on first boot | Enter **Yes**. |
| Enter root password that will be used for the engine appliance | Enter the password to be used for remotely logging in to the hosted engine. |
| Confirm appliance root password | Re-enter the password to be used for remotely logging in to the hosted engine. |
| Enter ssh public key for the root user that will be used for the engine appliance | Leave this field empty. |
| Do you want to enable ssh access for the root user | Enter **Yes**. |
| Please specify the size of the VM disk in GB | Use the default value. |
| Please specify the memory size of the VM in MB | Use the default value. |
| The following CPU types are supported by this host:<br>Please specify the CPU type to be used by the VM | Use the default value. |
| Please specify the number of virtual CPUs for the VM | Use the default value. |
| You may specify a unicast MAC address for the VM or accept a randomly generated default | Enter the MAC address that resolves to the fully-qualified domain name specified for the hosted engine. |

| Prompt text | Action |
|---|---|
| How should the engine VM network be configured | Enter **DHCP**. |
| Add lines for the appliance itself and for this host to /etc/hosts on the engine VM? (Note: ensuring that this host could resolve the engine VM hostname is still up to you) | Enter **No**. |
| Enter engine admin password (for the RHV UI) | Enter the password that the Red Hat Virtualization Manager administrator will use. |
| Confirm engine admin password | Re-enter the password that the Red Hat Virtualization Manager administrator will use. |
| Please provide the name of the SMTP server through which we will send notifications | Use the default value. |
| Please provide the TCP port number of the SMTP server | Use the default value. |
| Please provide the email address from which notifications will be sent | Use the default value. |
| Please provide a comma-separated list of email addresses which will get notifications | Enter any email addresses that you want to receive notifications from the hosted engine. The default (root@localhost) is adequate. |

4. Confirm installation settings

```
Please confirm installation settings
```

Review the configuration values and verify that they are correct. When satisfied, enter *Yes* in the field provided and click **Next**.

**NOTE**

There is currently no way to edit configuration at this point in the process; if your configuration details are incorrect, you need to start this process again to correct them, or continue the process and correct them during deployment.

5. Wait for deployment to complete
This will take some time (about 30 minutes).

When deployment is complete, the following message is displayed:

```
Hosted Engine Setup successfully completed!
```

You can now log in to Red Hat Virtualization Manager to complete configuration.

If deployment did not complete, see Chapter 13, *Viewing Hosted Engine Deployment Errors*.

# CHAPTER 9. CONFIGURE RED HAT GLUSTER STORAGE AS A RED HAT VIRTUALIZATION STORAGE DOMAIN

## 9.1. CREATE THE LOGICAL NETWORK FOR GLUSTER TRAFFIC

1. Log in to the engine
   Browse to the engine user interface (for example, **http://engine.example.com/ovirt-engine**) and log in using the administrative credentials you configured in Chapter 8, *Deploy the Hosted Engine using the Cockpit UI*.

2. Create a logical network for gluster traffic

   a. Click the **Networks** tab and then click **New**. The **New Logical Network** wizard appears.

   b. On the **General** tab of the wizard, provide a **Name** for the new logical network, and uncheck the **VM Network** checkbox.

   c. On the **Cluster** tab of the wizard, uncheck the **Required** checkbox.

   d. Click **OK** to create the new logical network.

3. Enable the new logical network for gluster

   a. Click the **Networks** tab and select the new logical network.

   b. Click the **Clusters** sub-tab and then click **Manage Network**. The **Manage Network** dialogue appears.

   c. In the **Manage Network** dialogue, check the **Migration Network** and **Gluster Network** checkboxes.

   d. Click **OK** to save.

4. Attach the gluster network to the host

   a. Click the **Hosts** tab and select the host.

   b. Click the **Network Interfaces** subtab and then click **Setup Host Networks**.

   c. Drag and drop the newly created network to the correct interface.

   d. Ensure that the **Verify connectivity** checkbox is checked.

   e. Ensure that the **Save network configuration** checkbox is checked.

   f. Click **OK** to save.

5. Verify the health of the network
   Click the **Hosts** tab and select the host. Click the **Network Interfaces** subtab and check the state of the host’s network.

   If the network interface enters an "Out of sync" state or does not have an IPv4 Address, click the **Management** tab that corresponds to the host and click **Refresh Capabilities**.

## 9.2. CREATE THE MASTER STORAGE DOMAIN

1. Click the **Storage** tab and then click **New Domain**.
2. Select **GlusterFS** as the **Storage Type** and provide a **Name** for the domain.
3. Check the **Use managed gluster volume** option.
   A list of volumes available in the cluster appears.

   Select the **vmstore** volume and add the following to the **Mount Options**:

   ```
   backup-volfile-servers=server2:server3
   ```

4. Click **OK** to save.

The hosted engine storage domain is imported automatically after master storage domain creation.

**IMPORTANT**

Wait until the Hosted Engine virtual machine and its storage domain are available before continuing with other tasks; this ensures that the Hosted Engine tab is available for use.

## 9.3. ADD ADDITIONAL VIRTUALIZATION HOSTS TO THE HOSTED ENGINE

Follow these steps in Red Hat Virtualization Manager for each of the other hosts.

1. Add virtualization hosts to the host inventory
   a. Click the **Hosts** tab and then click **New** to open the *New Host* dialog.
   b. Provide a **Name**, **Address**, and **Password** for the new host.
   c. Under **Advanced Parameters**, uncheck the **Automatically configure host firewall** checkbox, as firewall rules are already configured by gdeploy.
   d. In the **Hosted Engine** tab of the *New Host* dialog, set the value of **Choose hosted engine deployment action** to **deploy**. This ensures that the hosted engine can run on the new host.
   e. Click **OK**.
2. Attach the gluster network to the host
   a. Click the **Hosts** tab and select the host.
   b. Click the **Network Interfaces** subtab and then click **Setup Host Networks**.
   c. Drag and drop the newly created network to the correct interface.
   d. Ensure that the **Verify connectivity** checkbox is checked.
   e. Ensure that the **Save network configuration** checkbox is checked.
   f. Click **OK** to save.

3. In the **General** subtab for this host, verify that the value of **Hosted Engine HA** is *Active*, with a positive integer as a score.

   **IMPORTANT**

   If **Score** is listed as **N/A**, you may have forgotten to select the **deploy** action for **Choose hosted engine deployment action**. Follow the steps in Reinstalling a virtualization host in *Maintaining Red Hat Hyperconverged Infrastructure*to reinstall the host with the **deploy** action.

4. Verify the health of the network
   Click the **Hosts** tab and select the host. Click the **Network Interfaces** subtab and check the state of the host's network.

   **IMPORTANT**

   If the network interface enters an "Out of sync" state or does not have an IPv4 Address, right-click on the host and click on **Management > Refresh Capabilities.**

See the Red Hat Virtualization 4.1 *Self-Hosted Engine Guide*for further details: https://access.redhat.com/documentation/en/red-hat-virtualization/4.1/paged/self-hosted-engine-guide/chapter-7-installing-additional-hosts-to-a-self-hosted-environment

# PART III. VERIFY

# CHAPTER 10. VERIFY YOUR DEPLOYMENT

When deployment is complete, use Red Hat Virtualization Manager to check the status of the deployment.

Create a virtual machine to verify that your deployment is operating as expected. For details, see the Red Hat Virtualization *Virtual Machine Management Guide*: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/virtual_machine_management_guide/.

# PART IV. NEXT STEPS

# CHAPTER 11. POST-DEPLOYMENT CONFIGURATION SUGGESTIONS

Depending on your requirements, you may want to perform some additional configuration on your newly deployed Red Hat Hyperconverged Infrastructure. This section contains suggested next steps for additional configuration.

Details on these processes are available in *Maintaining Red Hat Hyperconverged Infrastructure*.

## 11.1. CONFIGURE A LOGICAL VOLUME CACHE FOR IMPROVED PERFORMANCE

If your main storage devices are not Solid State Disks (SSDs), Red Hat recommends configuring a logical volume cache (lvmcache) to achieve the required performance for Red Hat Hyperconverged Infrastructure deployments.

See Configuring a logical volume cache for improved performance for details.

## 11.2. CONFIGURE FENCING FOR HIGH AVAILABILITY

Fencing allows a cluster to enforce performance and availability policies and react to unexpected host failures by automatically rebooting virtualization hosts.

See Configure High Availability using fencing policies for further information.

## 11.3. CONFIGURE GEO-REPLICATION FOR DISASTER RECOVERY

Geo-replication is used to synchronize data from one Red Hat Gluster Storage cluster to another. Synchronizing the local data volume from your discrete Red Hat Hyperconverged Infrastructure (RHHI) cluster to a central data center on a regular basis helps ensure you can restore your cluster to a working state after an outage.

See Configuring geo-replication for disaster recovery for details.

# PART V. TROUBLESHOOT

# CHAPTER 12. LOG FILE LOCATIONS

During the deployment process, progress information is displayed in the web browser. This information is also stored on the local file system so that the information logged can be archived or reviewed at a later date, for example, if the web browser stops responding or is closed before the information has been reviewed.

The log file for the Cockpit based deployment process (documented in Chapter 7, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI*) is stored in the **`~/.gdeploy/logs/gdeploy.log`** file, where `~` is the home directory of the administrative user logged in to the Cockpit UI. If you log in to the Cockpit UI as root, the log file is stored as **`/root/.gdeploy/logs/gdeploy.log`**.

The log files for the Hosted Engine setup portion of the deployment process (documented in Chapter 8, *Deploy the Hosted Engine using the Cockpit UI*) are stored in the **`/var/log/ovirt-hosted-engine-setup`** directory, with file names of the form **`ovirt-hosted-engine-setup-<date>.log`**.

# CHAPTER 13. VIEWING HOSTED ENGINE DEPLOYMENT ERRORS

If an error occurs during hosted engine deployment, deployment pauses, and the following message is displayed:

```
Please check Engine VM configuration
Make a selection from the options below:
(1) Continue setup - Engine VM configuration has been fixed
(2) Abort setup
```

Error details are then listed in red.

Contact Red Hat Support with the details of your error for assistance in correcting it.

# CHAPTER 14. CLEANING UP AUTOMATED RED HAT GLUSTER STORAGE DEPLOYMENT ERRORS

1. Create a volume_cleanup.conf file based on the volume_cleanup.conf file in Appendix B, *Example cleanup configuration files for gdeploy*.

2. Run gdeploy using the volume_cleanup.conf file.

   ```
   # gdeploy -c volume_cleanup.conf
   ```

3. Create a lv_cleanup.conf file based on the lv_cleanup.conf file in Appendix B, *Example cleanup configuration files for gdeploy*.

4. Run gdeploy using the lv_cleanup.conf file.

   ```
   # gdeploy -c lv_cleanup.conf
   ```

5. Check mount configurations on all hosts
   Check the /etc/fstab file on all hosts, and remove any lines that correspond to XFS mounts of automatically created bricks.

# PART VI. REFERENCE MATERIAL

# APPENDIX A. CONFIGURING ENCRYPTION DURING DEPLOYMENT

## A.1. CONFIGURING TLS/SSL DURING DEPLOYMENT USING A CERTIFICATE AUTHORITY SIGNED CERTIFICATE

### A.1.1. Before you begin

Ensure that you have appropriate certificates signed by a Certificate Authority before proceeding. Obtaining certificates is outside the scope of this document.

### A.1.2. Configuring TLS/SSL encryption using a CA-signed certificate

1. Ensure that the following files exist in the following locations on all nodes.

   **/etc/ssl/glusterfs.key**

   The node's private key.

   **/etc/ssl/glusterfs.pem**

   The certificate signed by the Certificate Authority, which becomes the node's certificate.

   **/etc/ssl/glusterfs.ca**

   The Certificate Authority's certificate. For self-signed configurations, this file contains the concatenated certificates of all nodes.

2. Enable management encryption.
   Create the **/var/lib/glusterd/secure-access** file on each node.

   ```
   # touch /var/lib/glusterd/secure-access
   ```

3. Configure encryption.
   Add the following lines to each volume listed in the configuration file generated as part of Chapter 7, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI*This creates and configures TLS/SSL based encryption between gluster volumes using CA-signed certificates as part of the deployment process.

   ```
   key=client.ssl,server.ssl,auth.ssl-allow
   value=on,on,"host1;host2;host3"
   ```

   Ensure that you save the generated file after editing.

## A.2. CONFIGURING TLS/SSL ENCRYPTION DURING DEPLOYMENT USING A SELF SIGNED CERTIFICATE

Add the following lines to the configuration file generated in Chapter 7, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI* to create and configure TLS/SSL based encryption between gluster volumes using self signed certificates as part of the deployment process. Certificates generated by gdeploy are valid for one year.

In the configuration for the first volume, add lines for the **enable_ssl** and **ssl_clients** parameters and their values:

```
[volume1]
enable_ssl=yes
ssl_clients=<Gluster_Network_IP1>,<Gluster_Network_IP2>,
<Gluster_Network_IP3>
```

In the configuration for subsequent volumes, add the following lines to define values for the **client.ssl**, **server.ssl**, and **auth.ssl-allow** parameters:

```
[volumeX]
key=client.ssl,server.ssl,auth.ssl-allow
value=on,on,"<Gluster_Network_IP1>;<Gluster_Network_IP2>;
<Gluster_Network_IP3>"
```

# APPENDIX B. EXAMPLE CLEANUP CONFIGURATION FILES FOR GDEPLOY

In the event that deployment fails, it is necessary to clean up the previous deployment attempts before retrying the deployment. The following two example files can be run with gdeploy to clean up previously failed deployment attempts so that deployment can be reattempted.

**volume_cleanup.conf**

```
[hosts]
<Gluster_Network_NodeA>
<Gluster_Network_NodeB>
<Gluster_Network_NodeC>

[volume1]
action=delete
volname=engine

[volume2]
action=delete
volname=vmstore

[volume3]
action=delete
volname=data

[peer]
action=detach
```

**lv_cleanup.conf**

```
[hosts]
<Gluster_Network_NodeA>
<Gluster_Network_NodeB>
<Gluster_Network_NodeC>

[backend-reset]
pvs=sdb,sdc
unmount=yes
```

# APPENDIX C. UNDERSTANDING THE GENERATED GDEPLOY CONFIGURATION FILE

Gdeploy automatically provisions one or more machines with Red Hat Gluster Storage based on a configuration file.

The Cockpit UI provides provides a wizard that allows users to generate a gdeploy configuration file that is suitable for performing the base-level deployment of Red Hat Hyperconverged Infrastructure.

This section explains the gdeploy configuration file that would be generated if the following configuration details were specified in the Cockpit UI:

- 3 hosts with IP addresses 192.168.0.101, 192.168.0.102, and 192.168.0.103
- No additional packages or repositories.
- Arbiter configuration for non-engine volumes.
- Three-way replication for the engine volume.
- 12 bricks that are configured with RAID 6 with a stripe size of 256 KB.

This results in a gdeploy configuration file with the following sections.

For further details on any of the sections defined here, see the Red Hat Gluster Storage *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-red_hat_storage_volumes#chap-Red_Hat_Storage_Volumes-gdeploy_configfile.

### [hosts] section

```
[hosts]
192.168.0.101
192.168.0.102
192.168.0.103
```

The **`[hosts]`** section defines the IP addresses of the three physical machines to be configured according to this configuration file.

### [script1] section

```
[script1]
action=execute
ignore_script_errors=no
file=/usr/share/ansible/gdeploy/scripts/grafton-sanity-check.sh -d sdb -h
192.168.0.101,192.168.0.102,192.168.0.103
```

The **`[script1]`** section specifies a script to run to verify that all hosts are configured correctly in order to allow gdeploy to run without error.

### Underlying storage configuration

```
[disktype]
raid6
```

```
[diskcount]
12

[stripesize]
256
```

The **[disktype]** section specifies the hardware configuration of the underlying storage for all hosts.

The **[diskcount]** section specifies the number of disks in RAID storage. This can be omitted for JBOD configurations.

The **[stripesize]** section specifies the RAID storage stripe size in kilobytes. This can be omitted for JBOD configurations.

### Enable and restart chronyd

```
[service1]
action=enable
service=chronyd

[service2]
action=restart
service=chronyd
```

These service sections enable and restart the network time service, chronyd, on all servers.

### Create physical volume on all hosts

```
[pv1]
action=create
devices=sdb
ignore_pv_errors=no
```

The **[pv1]** section creates a physical volume on the **sdb** device of all hosts.

### Create volume group on all hosts

```
[vg1]
action=create
vgname=gluster_vg_sdb
pvname=sdb
ignore_vg_errors=no
```

The **[vg1]** section creates a volume group in the previously created physical volume on all hosts.

### Create the logical volume thin pool

```
[lv1:{192.168.0.101,192.168.0.102}]
action=create
poolname=gluster_thinpool_sdb
ignore_lv_errors=no
vgname=gluster_vg_sdb
lvtype=thinpool
```

```
poolmetadatasize=16GB
size=1000GB

[lv2:192.168.0.103]
action=create
poolname=gluster_thinpool_sdb
ignore_lv_errors=no
vgname=gluster_vg_sdb
lvtype=thinpool
poolmetadatasize=16GB
size=20GB
```

The **[lv1:*]** section creates a 1000 GB thin pool on the first two hosts with a meta data pool size of 16 GB.

The **[lv2:*]** section creates a 20 GB thin pool on the third host with a meta data pool size of 16 GB. This is the logical volume used for the arbiter brick.

The **chunksize** variable is also available, but should be used with caution. **chunksize** defines the size of the chunks used for snapshots, cache pools, and thin pools. By default this is specified in kilobytes. For RAID 5 and 6 volumes, gdeploy calculates the default chunksize by multiplying the stripe size and the disk count.

> **WARNING**
>
> Red Hat recommends using at least the default chunksize. If the chunksize is too small and your volume runs out of space for metadata, the volume is unable to create data. Red Hat recommends monitoring your logical volumes to ensure that they are expanded or more storage created before metadata volumes become completely full.

**Create underlying engine storage**

```
[lv3:{192.168.0.101,192.168.0.102}]
action=create
lvname=gluster_lv_engine
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/engine
size=100GB
lvtype=thick

[lv4:192.168.0.103]
action=create
lvname=gluster_lv_engine
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/engine
size=10GB
lvtype=thick
```

The **[lv3:*]** section creates a 100 GB thick provisioned logical volume called **gluster_lv_engine** on the first two hosts. This volume is configured to mount on **/gluster_bricks/engine**.

The **[lv4:*]** section creates a 10 GB thick provisioned logical volume for the engine on the third host. This volume is configured to mount on **/gluster_bricks/engine**.

### Create underlying data and virtual machine boot disk storage

```
[lv5:{192.168.0.101,192.168.0.102}]
action=create
lvname=gluster_lv_data
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/data
lvtype=thinlv
poolname=gluster_thinpool_sdb
virtualsize=500GB

[lv6:192.168.0.103]
action=create
lvname=gluster_lv_data
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/data
lvtype=thinlv
poolname=gluster_thinpool_sdb
virtualsize=10GB

[lv7:{192.168.0.101,192.168.0.102}]
action=create
lvname=gluster_lv_vmstore
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/vmstore
lvtype=thinlv
poolname=gluster_thinpool_sdb
virtualsize=500GB

[lv8:192.168.0.103]
action=create
lvname=gluster_lv_vmstore
ignore_lv_errors=no
vgname=gluster_vg_sdb
mount=/gluster_bricks/vmstore
lvtype=thinlv
poolname=gluster_thinpool_sdb
virtualsize=10GB
```

The **[lv5:*]** and **[lv7:*]** sections create 500 GB logical volumes as bricks for the data and vmstore volumes on the first two hosts.

The **[lv6:*]** and **[lv8:*]** sections create 10 GB logical volumes as arbiter bricks for the data and vmstore volumes on the third host.

The data bricks are configured to mount on **`/gluster_bricks/data`**, and the vmstore bricks are configured to mount on **`/gluster_bricks/vmstore`**.

### Configure SELinux file system labels

```
[selinux]
yes
```

The **`[selinux]`** section specifies that the storage created should be configured with appropriate SELinux file system labels for Gluster storage.

### Start glusterd

```
[service3]
action=start
service=glusterd
slice_setup=yes
```

The **`[service3]`** section starts the glusterd service and configures a control group to ensure glusterd cannot consume all system resources; see the Red Hat Enterprise Linux *Resource Management Guide* for details: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Resource_Management_Guide/index.html.

### Configure the firewall

```
[firewalld]
action=add
ports=111/tcp,2049/tcp,54321/tcp,5900/tcp,5900-
6923/tcp,5666/tcp,16514/tcp,54322/tcp
services=glusterfs
```

The **`[firewalld]`** section opens the ports required to allow gluster traffic.

### Disable gluster hooks

```
[script2]
action=execute
file=/usr/share/ansible/gdeploy/scripts/disable-gluster-hooks.sh
```

The **`[script2]`** section disables gluster hooks that can interfere with the Hyperconverged Infrastructure.

### Create gluster volumes

```
[volume1]
action=create
volname=engine
transport=tcp
replica=yes
replica_count=3
key=group,storage.owner-uid,storage.owner-gid,network.ping-
timeout,performance.strict-o-direct,network.remote-dio,cluster.granular-
entry-heal,features.shard-block-size
```

```
value=virt,36,36,30,on,off,enable,64MB
brick_dirs=192.168.0.101:/gluster_bricks/engine/engine,192.168.0.102:/glus
ter_bricks/engine/engine,192.168.0.103:/gluster_bricks/engine/engine
ignore_volume_errors=no

[volume2]
action=create
volname=data
transport=tcp
replica=yes
replica_count=3
key=group,storage.owner-uid,storage.owner-gid,network.ping-
timeout,performance.strict-o-direct,network.remote-dio,cluster.granular-
entry-heal,features.shard-block-size
value=virt,36,36,30,on,off,enable,64MB
brick_dirs=192.168.0.101:/gluster_bricks/data/data,192.168.0.102:/gluster_
bricks/data/data,192.168.0.103:/gluster_bricks/data/data
ignore_volume_errors=no
arbiter_count=1

[volume3]
action=create
volname=vmstore
transport=tcp
replica=yes
replica_count=3
key=group,storage.owner-uid,storage.owner-gid,network.ping-
timeout,performance.strict-o-direct,network.remote-dio,cluster.granular-
entry-heal,features.shard-block-size
value=virt,36,36,30,on,off,enable,64MB
brick_dirs=192.168.0.101:/gluster_bricks/vmstore/vmstore,192.168.0.102:/gl
uster_bricks/vmstore/vmstore,192.168.0.103:/gluster_bricks/vmstore/vmstore
ignore_volume_errors=no
arbiter_count=1
```

The **[volume*]** sections configure Red Hat Gluster Storage volumes. The **[volume1]** section configures one three-way replicated volume, **engine**. The additional **[volume*]** sections configure two arbitrated replicated volumes: **data** and **vmstore**, which have one arbiter brick on the third host.

The **key** and **value** parameters are used to set the following options:

- **group=virt**
- **storage.owner-uid=36**
- **storage.owner-gid=36**
- **network.ping-timeout=30**
- **performance.strict-o-direct=on**
- **network.remote-dio=off**
- **cluster.granular-entry-heal=enable**
- **features.shard-block-size=64MB**