Red Hat Hyperconverged Infrastructure for Virtualization 1.5

Deploying Red Hat Hyperconverged Infrastructure for Virtualization

Instructions for deploying Red Hat Hyperconverged Infrastructure for Virtualization

Last Updated: 2019-02-20

Red Hat Hyperconverged Infrastructure for Virtualization 1.5 Deploying Red Hat Hyperconverged Infrastructure for Virtualization

Instructions for deploying Red Hat Hyperconverged Infrastructure for Virtualization

Laura Bailey
lbailey@redhat.com

Legal Notice

Abstract

This document outlines how to deploy Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) across three physical machines, using Red Hat Gluster Storage 3.4 and Red Hat Virtualization 4.2. 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

PART I. PLAN 4

CHAPTER 1. ARCHITECTURE 5
1.1. UNDERSTANDING VDO 5

CHAPTER 2. SUPPORT REQUIREMENTS 7
2.1. OPERATING SYSTEM 7
2.2. PHYSICAL MACHINES 7
2.3. HOSTED ENGINE VIRTUAL MACHINE 8
2.4. NETWORKING 8
2.5. STORAGE 9
2.5.1. Disks 9
2.5.2. RAID 10
2.5.3. JBOD 10
2.5.4. Logical volumes 10
2.5.5. Red Hat Gluster Storage volumes 10
2.5.6. Volume types 10
2.6. VIRTUAL DATA OPTIMIZER (VDO) 11
2.7. SCALING 11
2.8. EXISTING RED HAT GLUSTER STORAGE CONFIGURATIONS 11
2.9. DISASTER RECOVERY 11
2.9.1. Prerequisites for geo-replication 12
2.9.2. Prerequisites for failover and failback configuration 12
2.10. ADDITIONAL REQUIREMENTS FOR SINGLE NODE DEPLOYMENTS 12

PART II. DEPLOY 13

CHAPTER 3. DEPLOYMENT WORKFLOW 14

CHAPTER 4. INSTALL HOST PHYSICAL MACHINES 15

CHAPTER 5. CONFIGURE PUBLIC KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD 16

CHAPTER 6. CONFIGURE RED HAT GLUSTER STORAGE FOR HOSTED ENGINE USING THE COCKPIT UI . 17

CHAPTER 7. DEPLOY THE HOSTED ENGINE USING THE COCKPIT UI 23

CHAPTER 8. CONFIGURE RED HAT GLUSTER STORAGE AS A RED HAT VIRTUALIZATION STORAGE DOMAIN 34
8.1. CREATE THE LOGICAL NETWORK FOR GLUSTER TRAFFIC 34
8.2. ADD ADDITIONAL VIRTUALIZATION HOSTS TO THE HOSTED ENGINE 34

PART III. VERIFY 36

CHAPTER 9. VERIFY YOUR DEPLOYMENT 37

PART IV. NEXT STEPS 40

CHAPTER 10. POST-DEPLOYMENT CONFIGURATION SUGGESTIONS 41
10.1. CONFIGURE A LOGICAL VOLUME CACHE FOR IMPROVED PERFORMANCE 41
10.2. CONFIGURE FENCING FOR HIGH AVAILABILITY 41
10.3. CONFIGURE BACKUP AND RECOVERY OPTIONS 41

PART V. TROUBLESHOOT 42

CHAPTER 11. LOG FILE LOCATIONS 43

CHAPTER 12. VIEWING HOSTED ENGINE DEPLOYMENT ERRORS 44
12.1. FAILED TO PREPARE VIRTUAL MACHINE 44
12.2. FAILED TO DEPLOY HOSTED ENGINE 44

CHAPTER 13. CLEANING UP AUTOMATED RED HAT GLUSTER STORAGE DEPLOYMENT ERRORS . . 46

PART VI. REFERENCE MATERIAL 47

APPENDIX A. CONFIGURING ENCRYPTION DURING DEPLOYMENT 48
A.1. CONFIGURING TLS/SSL DURING DEPLOYMENT USING A CERTIFICATE AUTHORITY SIGNED CERTIFICATE 48
A.1.1. Prerequisites 48
A.1.2. Configuring TLS/SSL encryption using a CA-signed certificate 48
A.2. CONFIGURING TLS/SSL ENCRYPTION DURING DEPLOYMENT USING A SELF SIGNED CERTIFICATE 48

APPENDIX B. EXAMPLE CLEANUP CONFIGURATION FILES FOR GDEPLOY 50

APPENDIX C. UNDERSTANDING THE GENERATED GDEPLOY CONFIGURATION FILE 51

APPENDIX D. EXAMPLE GDEPLOY CONFIGURATION FILE FOR CONFIGURING COMPRESSION AND DEDUPLICATION 58

PART I. PLAN

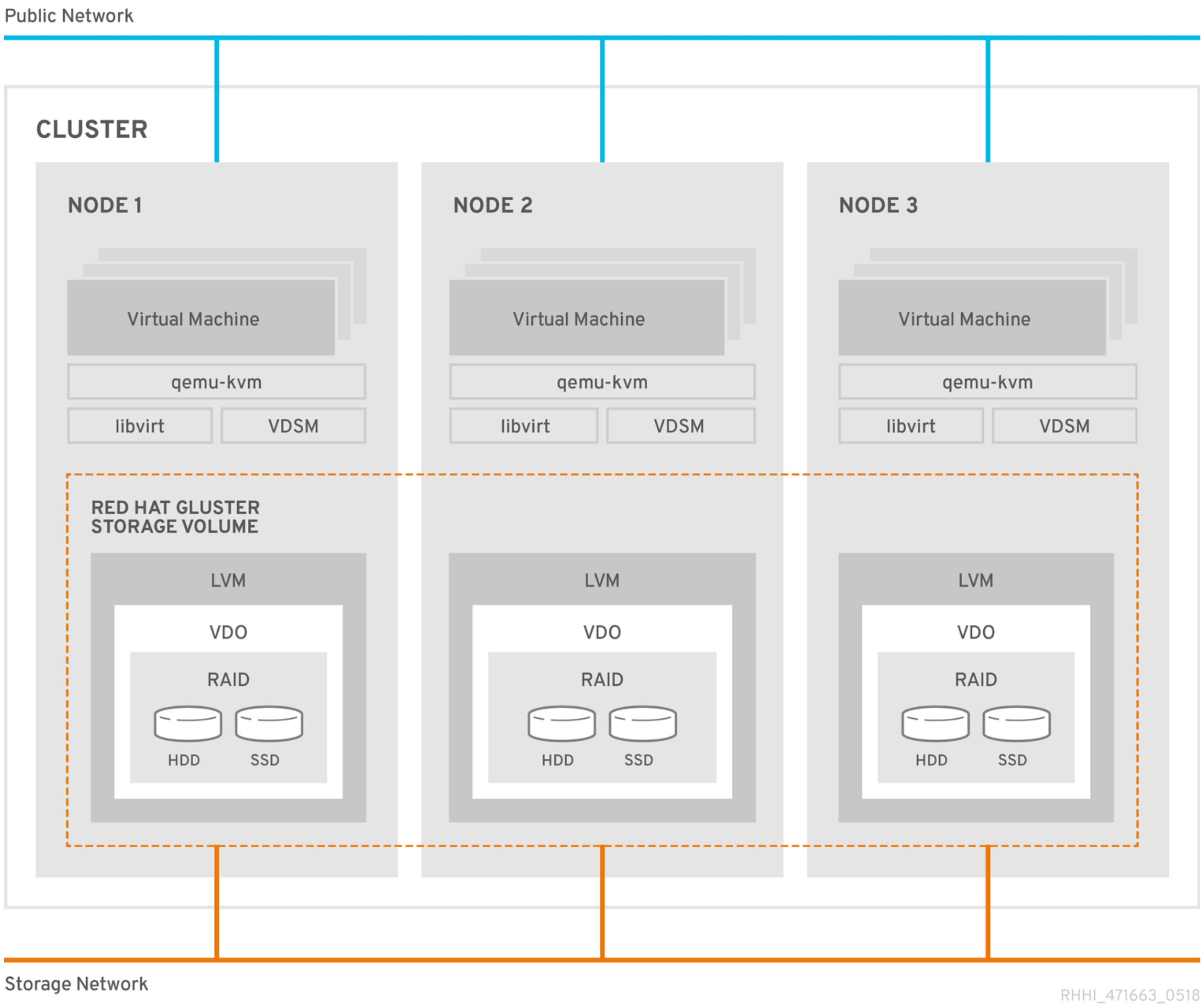
CHAPTER 1. ARCHITECTURE

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

RHHI for Virtualization is deployed across three physical machines to create a discrete cluster or *pod* using Red Hat Gluster Storage 3.4 and Red Hat Virtualization 4.2.

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.



1.1. UNDERSTANDING VDO

As of Red Hat Hyperconverged Infrastructure for Virtualization 1.5, you can configure a Virtual Data Optimizer (VDO) layer to provide data reduction and deduplication for your storage.

VDO is supported only when enabled on new installations at deployment time, and cannot be enabled on deployments upgraded from earlier versions of RHHI for Virtualization.

Additionally, thin provisioning is not currently compatible with VDO. These two technologies are not supported on the same device.

VDO performs following types of data reduction to reduce the space required by data.

Deduplication

Eliminates zero and duplicate data blocks. VDO finds duplicated data using the UDS (Universal Deduplication Service) Kernel Module. Instead of writing the duplicated data, VDO records it as a reference to the original block. The logical block address is mapped to the physical block address by VDO.

Compression

Reduces the size of the data by packing non-duplicate blocks together into fixed length (4 KB) blocks before writing to disk. This helps to speed up the performance for reading data from storage.

At best, data can be reduced to 15% of its original size.

Because reducing data has additional processing costs, enabling compression and deduplication reduces write performance. As a result, VDO is not recommended for performance sensitive workloads. Red Hat strongly recommends that you test and verify that your workload achieves the required level of performance with VDO enabled before deploying VDO in production.

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 4.2 as a base for all other configuration. The following table shows the the supported versions of each product to use for a supported RHHI for Virtualization deployment.

Table 2.1. Version compatibility

RHHI version	RHGS version	RHV version
1.0	3.2	4.1.0 to 4.1.7
1.1	3.3.1	4.1.8 to 4.2.0
1.5	3.4 Batch 1 Update	4.2.7 to current
1.5.1	3.4 Batch 2 Update	4.2.8 to current

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 2.4, “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

2.3. 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 4.2 *Planning and Prerequisites Guide*.

2.4. 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 that provides Red Hat Virtualization Manager.

Client storage traffic and management traffic in the cluster must use separate networks: a **front-end management network** and a **back-end storage network**.

Each node requires 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 use geo-replication to store copies of data for disaster recovery purposes, a reliable time source is required.

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

- IP address for a gateway to the virtualization 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.

2.5. 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`** - 22GB
- **`/var/log`** - 15GB
- **`/var/log/audit`** - 2GB
- **`swap`** - 1GB (for the recommended swap size, see https://access.redhat.com/solutions/15244)
- 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 - 52GB**

2.5.1. 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.

4K native devices are not supported with Red Hat Hyperconverged Infrastructure for Virtualization, as Red Hat Virtualization requires 512 byte emulation (512e) support.

2.5.2. 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+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.

2.5.3. JBOD

As of Red Hat Hyperconverged Infrastructure for Virtualization 1.5, JBOD configurations are fully supported and no longer require architecture review.

2.5.4. 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.

When VDO is not in use, 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.

Thin provisioning is not required for the **vmstore** and **data** volumes if VDO is being used on these volumes.

2.5.5. 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 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 for Virtualization deployment can contain at most 1 geo-replicated volume.

2.5.6. Volume types

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) supports only the following volume types:

- Replicated volumes (3 copies of the same data on 3 bricks, across 3 nodes).
- Arbitrated replicated volumes (2 full copies of the same data on 2 bricks and 1 arbiter brick that contains metadata. across three nodes).
- Distributed volumes (1 copy of the data, no replication to other bricks).

All replicated and arbitrated-replicated volumes must span exactly three nodes.

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*.

2.6. VIRTUAL DATA OPTIMIZER (VDO)

A Virtual Data Optimizer (VDO) layer is supported as of Red Hat Hyperconverged Infrastructure for Virtualization 1.5.

The following limitations apply to this support:

- VDO is supported only on new deployments.
- VDO is compatible only with thick provisioned volumes. VDO and thin provisioning are not supported on the same device.

2.7. 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 existing hyperconverged nodes.

You cannot create a volume that spans more than 3 nodes, or expand an existing volume so that it spans across more than 3 nodes at a time.

2.8. 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.

2.9. 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/1.5/html/maintaining_red_hat_hyperconvergec backup-recovery.

2.9.1. 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 is 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.

2.9.2. 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.

2.10. 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.

PART II. DEPLOY

CHAPTER 3. DEPLOYMENT WORKFLOW

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

1. Verify that your planned deployment meets support requirements: Chapter 2, *Support requirements*.

2. Install the physical machines that will act as virtualization hosts: Chapter 4, *Install Host Physical Machines*.

3. Configure key-based SSH authentication without a password to enable automated configuration of the hosts: Chapter 5, *Configure Public Key based SSH Authentication without a password*.

4. Configure Red Hat Gluster Storage on the physical hosts using the Cockpit UI: Chapter 6, *Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI*.

5. Deploy the Hosted Engine using the Cockpit UI: Chapter 7, *Deploy the Hosted Engine using the Cockpit UI*.

6. Configure the Red Hat Gluster Storage nodes using the Red Hat Virtualization management UI: Log in to Red Hat Virtualization Manager to complete configuration.

CHAPTER 4. INSTALL HOST PHYSICAL MACHINES

Install Red Hat Virtualization Host 4.2 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/4.2/html/installation_guide/red_hat_virtualization_hosts.

Ensure that you customize your installation to provide the following when you install each host:

- Increase the size of **`/var/log`** to 15GB to provide sufficient space for the additional logging requirements of Red Hat Gluster Storage.

CHAPTER 5. 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 virtualization host to all hosts, including itself. Do this for all storage and management interfaces, and for both IP addresses and FQDNs.

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 6. CONFIGURE RED HAT GLUSTER STORAGE FOR HOSTED ENGINE USING THE COCKPIT UI

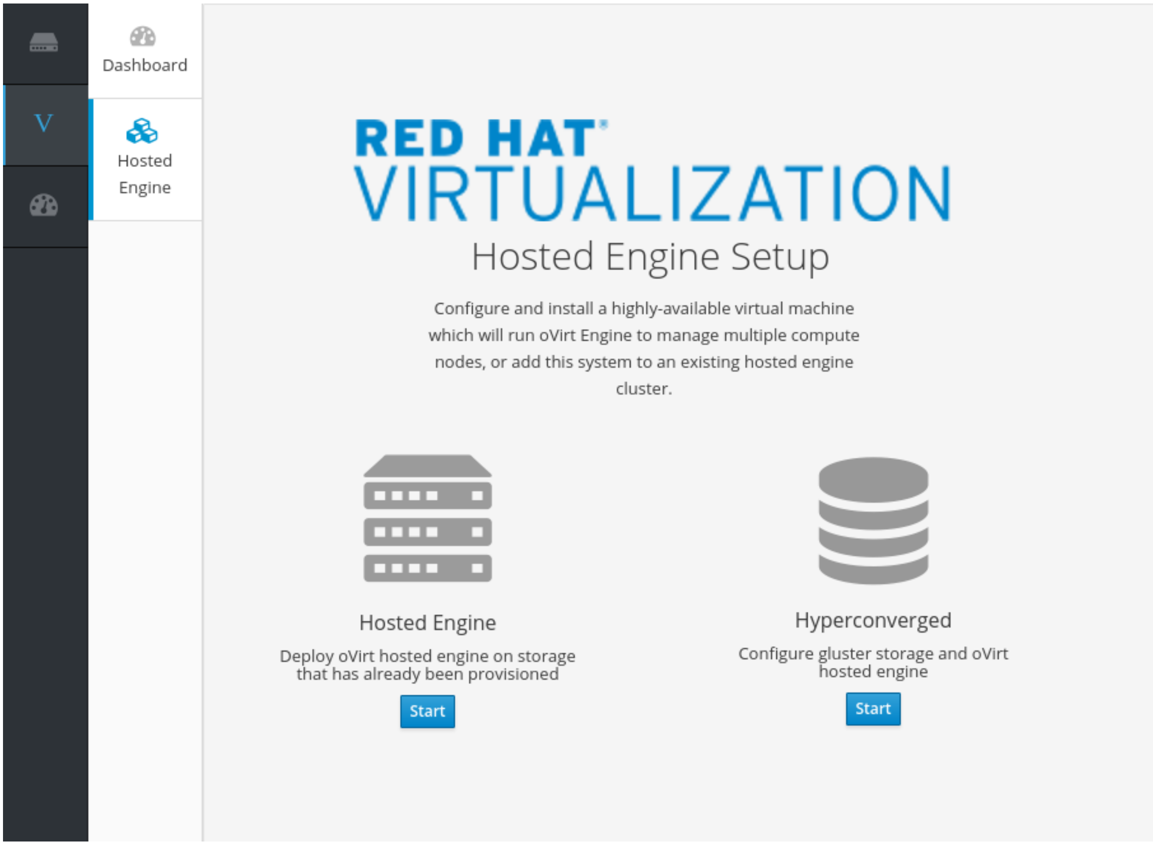
IMPORTANT

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 4, *Install Host Physical Machines*.

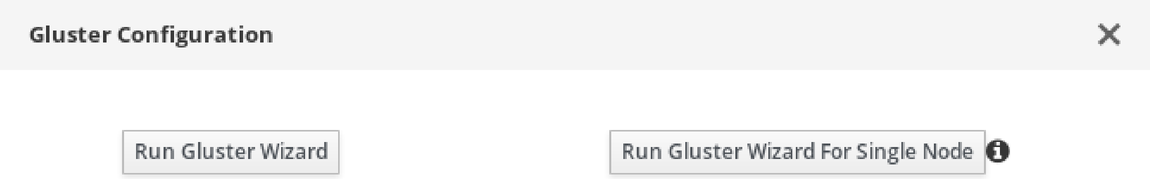
2. **Start the deployment wizard**

   a. Click **Virtualization** → **Hosted Engine** and click **Start** underneath **Hyperconverged**.


Dashboard
V
Hosted Engine
RED HAT VIRTUALIZATION
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.
Hosted Engine
Deploy oVirt hosted engine on storage that has already been provisioned
Start
Hyperconverged
Configure gluster storage and oVirt hosted engine
Start

   The *Gluster Configuration* window opens.

   b. Click the **Run Gluster Wizard** button.



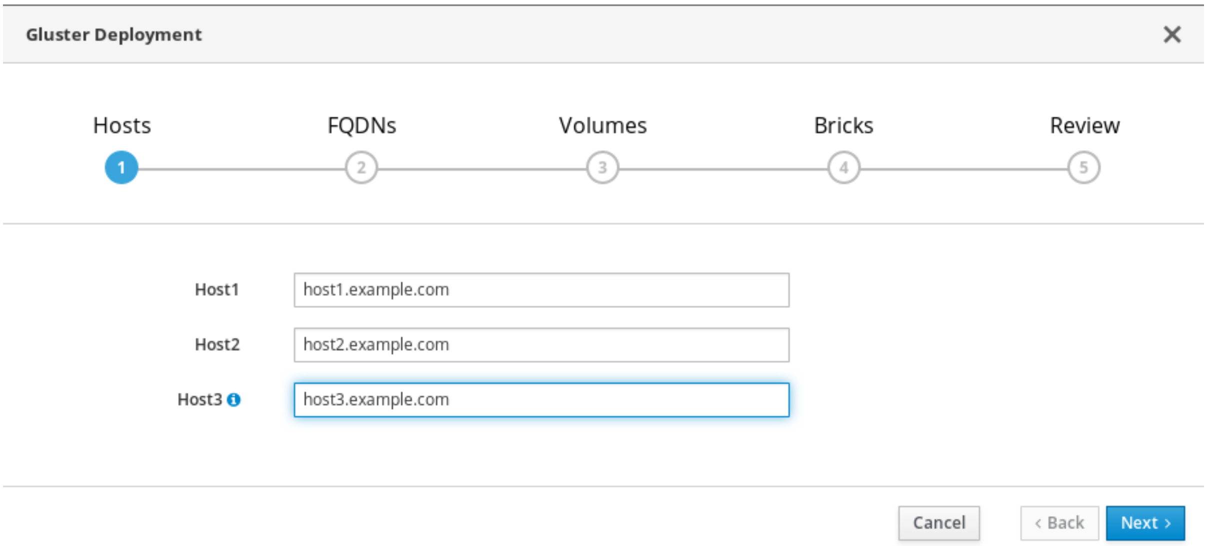
   The *Gluster Deployment* window opens in 3 node mode.

3. **Specify storage hosts**

Specify the back-end FQDNs on the storage network (not the management network) 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.



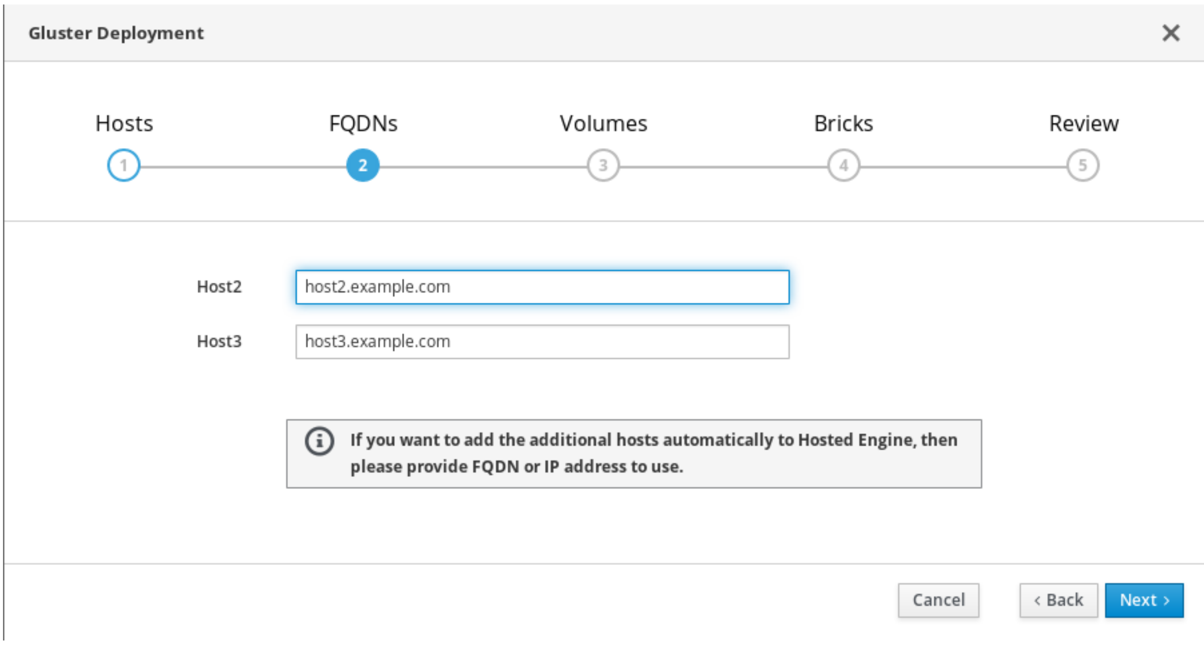
Host1 host1.example.com

Host2 host2.example.com

Host3 host3.example.com

Click Next.

4. **Specify additional hosts**
   For multi-node deployments, add the fully qualified domain names or IP addresses of the other two virtualization hosts to have them automatically added to Red Hat Virtualization Manager when deployment is complete.



Host2 host2.example.com

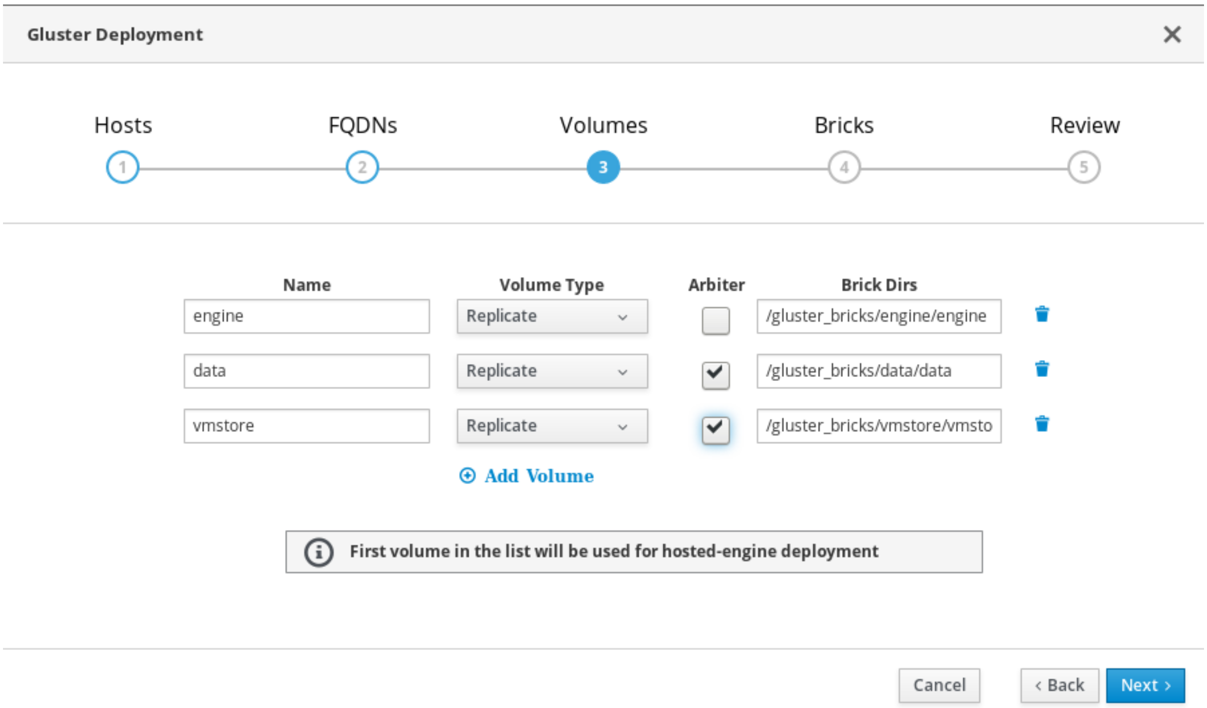
Host3 host3.example.com

If you want to add the additional hosts automatically to Hosted Engine, then please provide FQDN or IP address to use.

NOTE

If you do not add additional hosts now, you can also add them after deployment using Red Hat Virtualization Administration Portal, as described in Add additional virtualization hosts to the hosted engine.

5. **Specify volumes**
   Specify the volumes to create.



Gluster Deployment

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

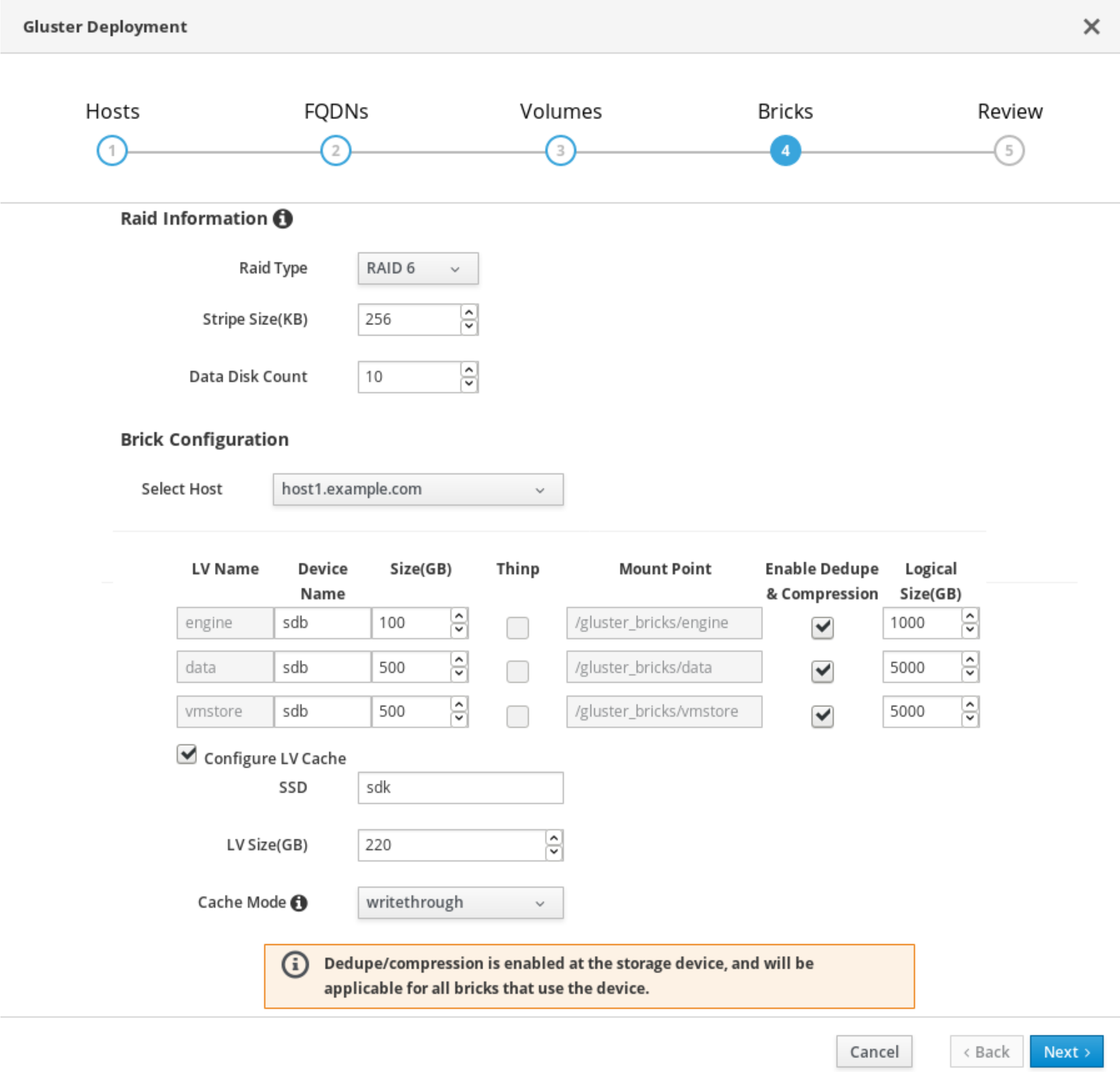
Specify whether to create the volume with an arbiter brick. If this box is checked, the third disk stores only metadata.

Brick Dirs

The directory that contains this volume’s bricks.

The default values are correct for most installations.

6. **Specify bricks**
   Enter details of the bricks to be created. Use the **Select host** drop-down menu to change the host being configured.



Raid Information

Raid Type: RAID 6

Stripe Size(KB): 256

Data Disk Count: 10

Brick Configuration

Select Host: host1.example.com

LV Name	Device Name	Size(GB)	Thinp	Mount Point	Enable Dedupe & Compression	Logical Size(GB)
engine	sdb	100	□	/gluster_bricks/engine	☑	1000
data	sdb	500	□	/gluster_bricks/data	☑	5000
vmstore	sdb	500	□	/gluster_bricks/vmstore	☑	5000

☑ Configure LV Cache

SSD: sdk

LV Size(GB): 220

Cache Mode: writethrough

Dedupe/compression is enabled at the storage device, and will be applicable for all bricks that use the device.

RAID

Specify the RAID configuration to use. This should match the RAID configuration of your host. Supported values are **raid5**, **raid6**, and **jbod**. Setting this option ensures that your storage is correctly tuned for your RAID configuration.

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.

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. This number should be the same for all bricks in a replicated set. Arbiter bricks can be smaller than other bricks in their replication set.

Mount Point

Specify the mount point for the logical volume. This should be inside 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. Do not use **Enable Dedupe & Compression** at the same time as this option.

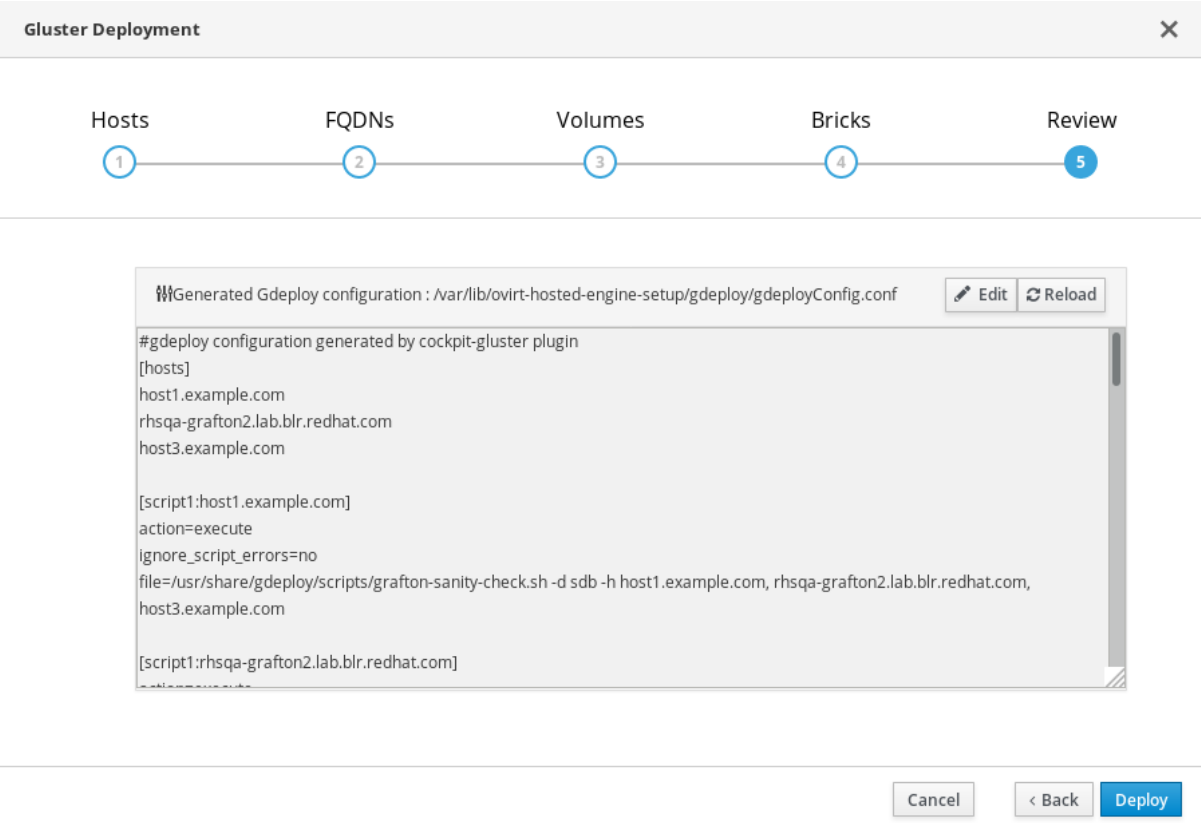
Enable Dedupe & Compression

Specify whether to provision the volume using VDO for compression and deduplication at deployment time. Do not use **Thinp** at the same time as this option.

Logical Size (GB)

Specify the logical size of the VDO volume. This can be up to 10 times the size of the physical volume, with an absolute maximum logical size of 4 PB.

7. **Review and edit configuration**


Gluster Deployment
×
Hosts
1
FQDNs
2
Volumes
3
Bricks
4
Review
5
Generated Gdeploy configuration : /var/lib/ovirt-hosted-engine-setup/gdeploy/gdeployConfig.conf
Edit
Reload
Cancel
‹ Back
Deploy

```
#gdeploy configuration generated by cockpit-gluster plugin
[hosts]
host1.example.com
rhsqa-grafton2.lab.blr.redhat.com
host3.example.com

[script1:host1.example.com]
action=execute
ignore_script_errors=no
file=/usr/share/gdeploy/scripts/grafton-sanity-check.sh -d sdb -h host1.example.com, rhsqa-grafton2.lab.blr.redhat.com,
host3.example.com

[script1:rhsqa-grafton2.lab.blr.redhat.com]
```

a. Click **Edit** to begin editing the generated deployment configuration 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:

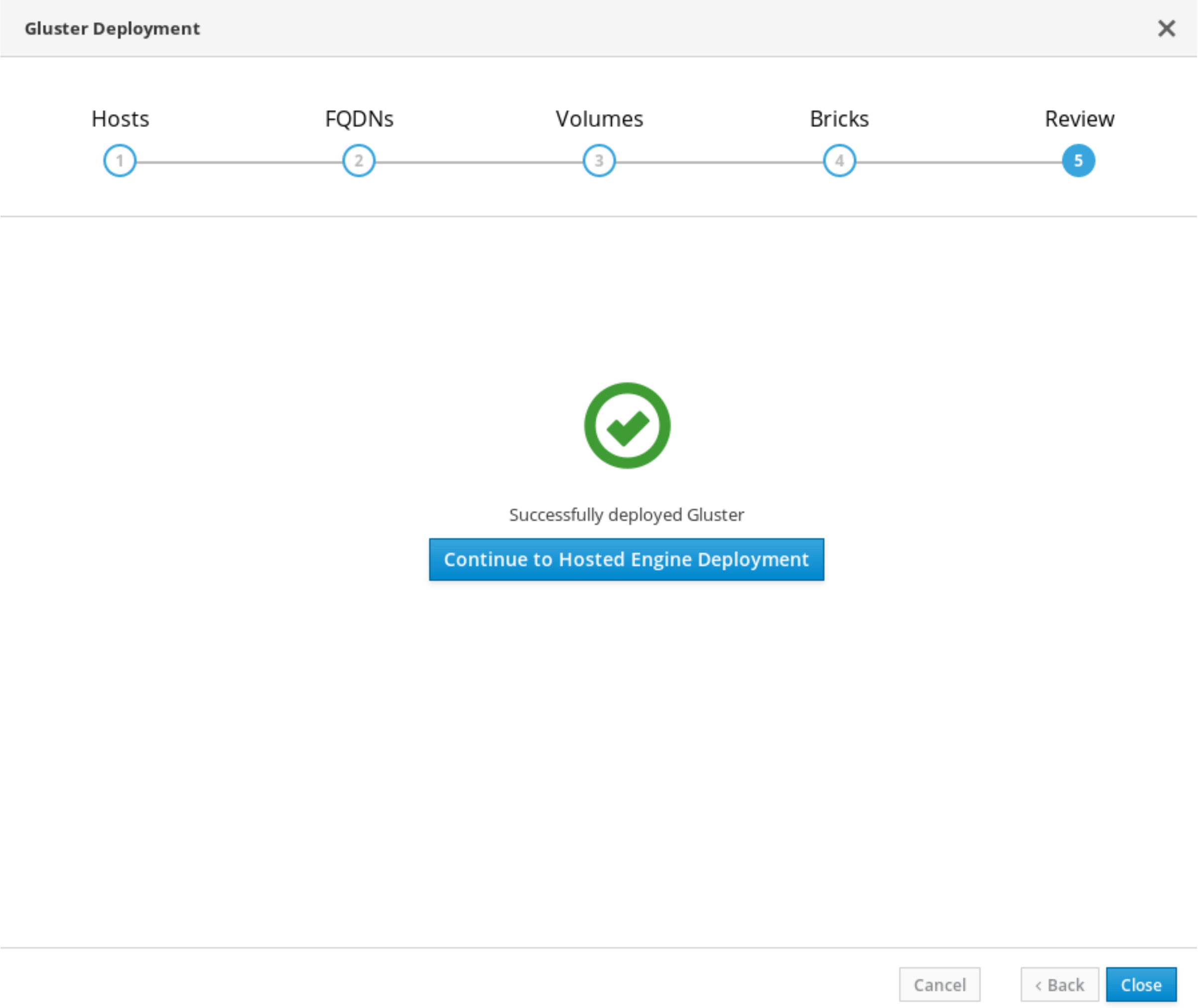
- Section A.1, “Configuring TLS/SSL during deployment using a Certificate Authority signed certificate”

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

c. **Review the configuration file**
If the configuration details are correct, 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.

   The window displays **Successfully deployed gluster** when complete.



Successfully deployed Gluster

Continue to Hosted Engine Deployment

   Click **Continue to Hosted Engine Deployment** and continue the deployment process with the instructions in Chapter 7, *Deploy the Hosted Engine using the Cockpit UI*.

IMPORTANT

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 13, *Cleaning up automated Red Hat Gluster Storage deployment errors* to clean up previous deployment attempts.

CHAPTER 7. DEPLOY THE HOSTED ENGINE USING THE COCKPIT UI

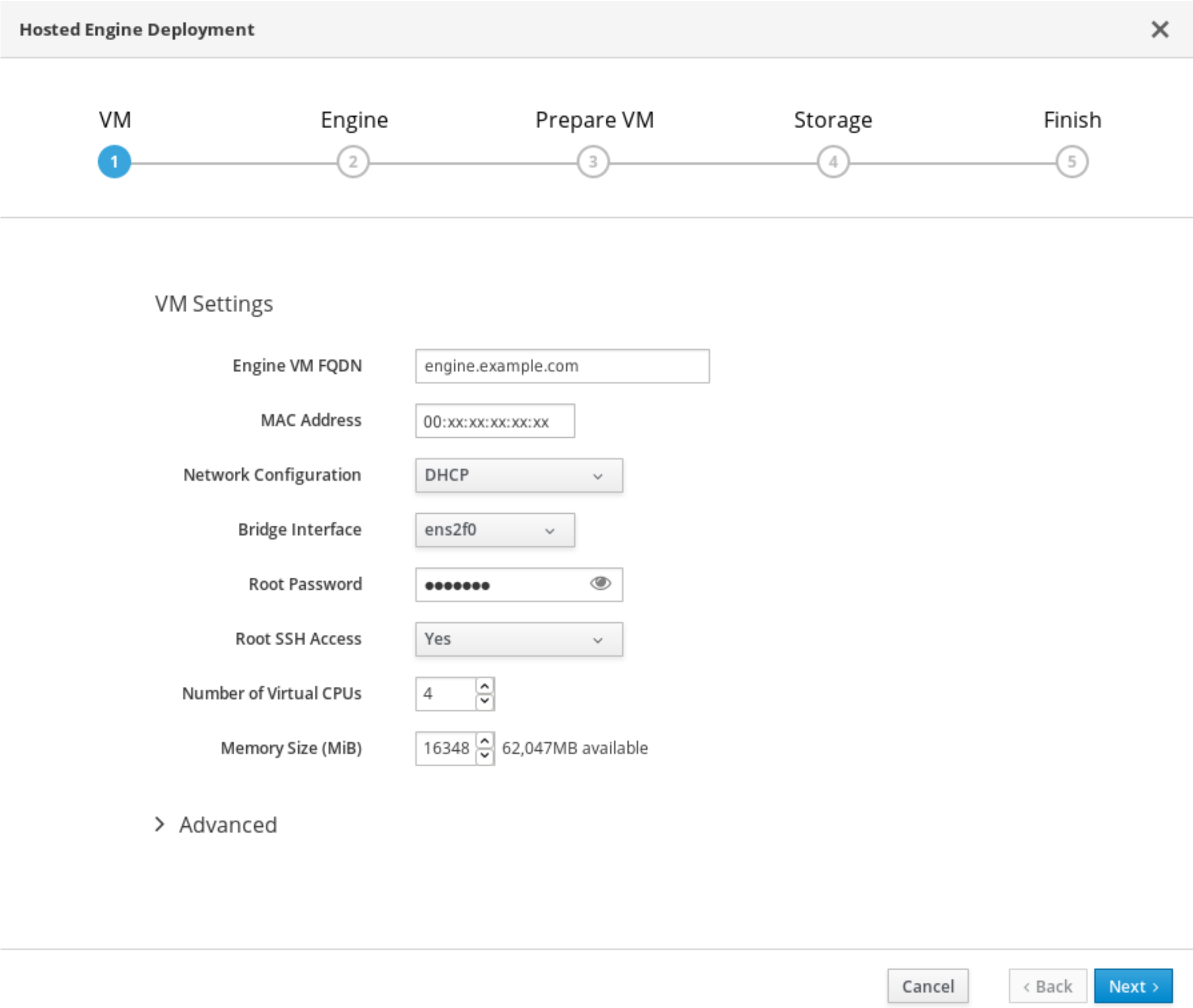
This section shows you how to deploy the Hosted Engine using the Cockpit UI. Following this process results in Red Hat Virtualization Manager running as a virtual machine on the first physical machine in your deployment. It also configures a Default cluster comprised of the three physical machines, and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

Prerequisites

- This procedure assumes that you have continued directly from the end of Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI.
- Gather the information you need for Hosted Engine deployment
  Have the following information ready before you start the deployment process.
  - IP address for a pingable gateway to the virtualization host
  - 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

Procedure

1. **Specify virtual machine details**



Hosted Engine Deployment

VM Settings

Engine VM FQDN: engine.example.com

MAC Address: 00:xx:xx:xx:xx:xx

Network Configuration: DHCP

Bridge Interface: ens2f0

Root Password: •••••••

Root SSH Access: Yes

Number of Virtual CPUs: 4

Memory Size (MiB): 16348 62,047MB available

a. Enter the following details:

Engine VM FQDN

The fully qualified domain name to be used for the Hosted Engine virtual machine.

MAC Address

The MAC address associated with the FQDN to be used for the Hosted Engine virtual machine.

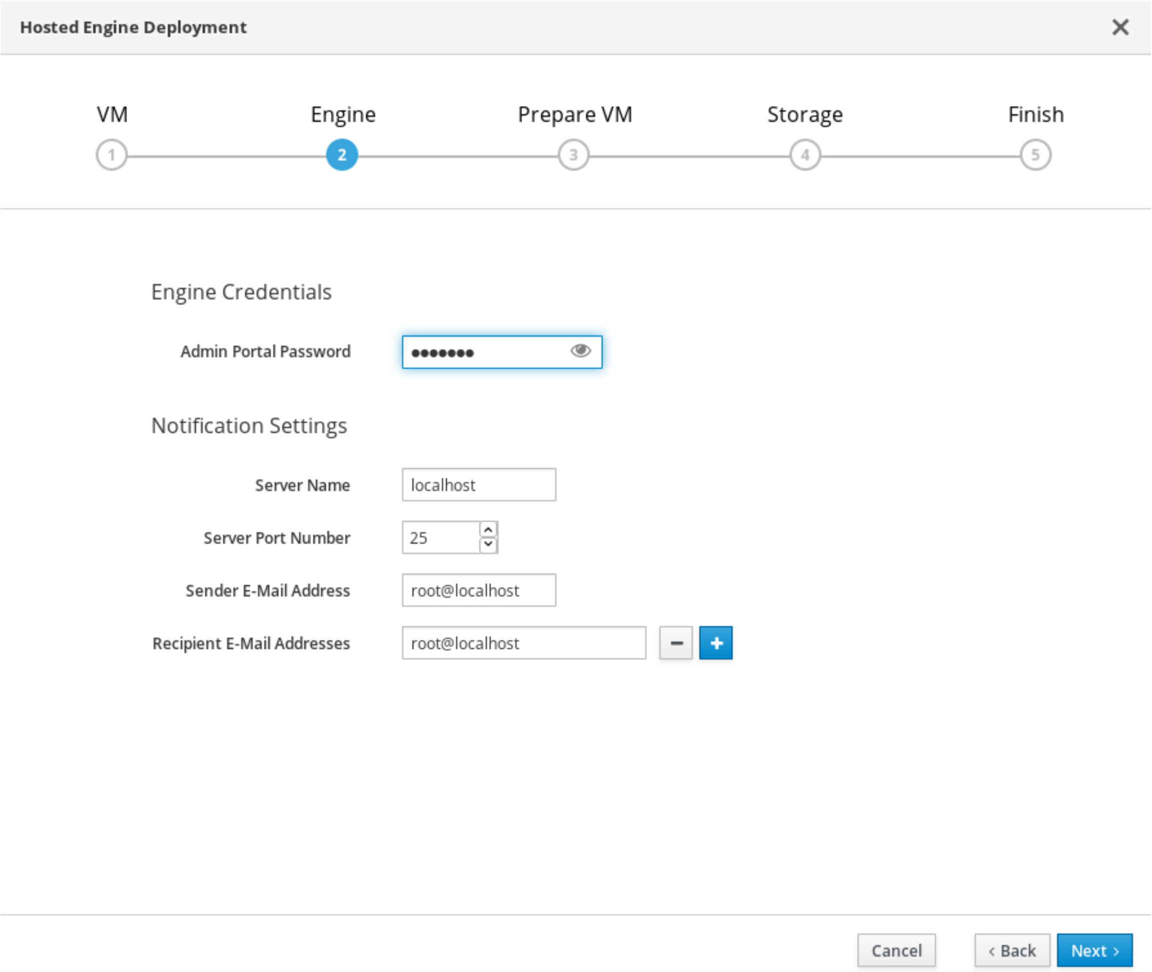
Root password

The root password to be used for the Hosted Engine virtual machine.

b. Click **Next**.

2. **Specify virtualization management details**

a. Enter the password to be used by the `admin` account in Red Hat Virtualization Manager. You can also specify notification behaviour here.



Hosted Engine Deployment

Engine Credentials

Admin Portal Password: •••••••

Notification Settings

Server Name: localhost

Server Port Number: 25

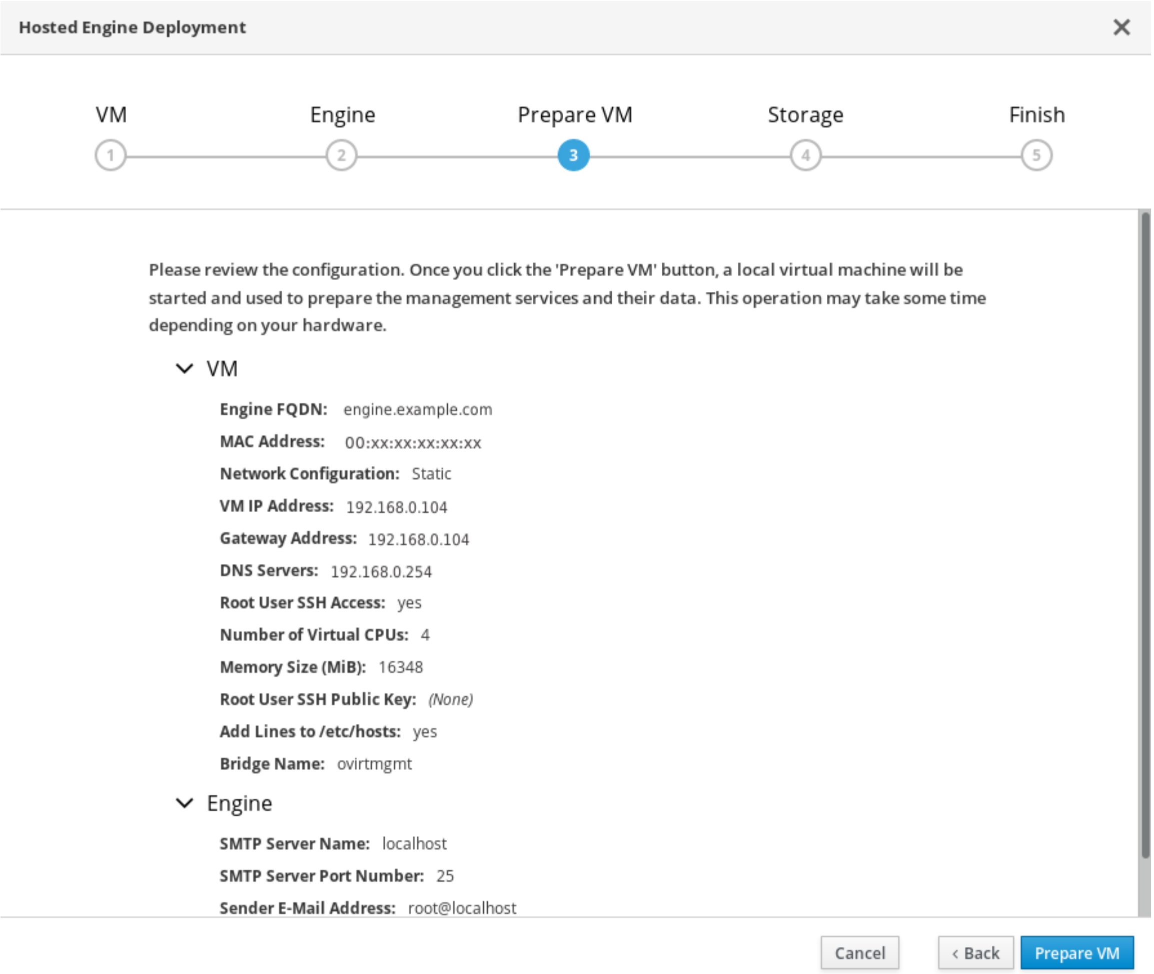
Sender E-Mail Address: root@localhost

Recipient E-Mail Addresses: root@localhost

b. Click **Next**.

3. **Review virtual machine configuration**

   a. Ensure that the details listed on this tab are correct. Click **Back** to correct any incorrect information.



Please review the configuration. Once you click the 'Prepare VM' button, a local virtual machine will be started and used to prepare the management services and their data. This operation may take some time depending on your hardware.

VM

Engine FQDN: engine.example.com

MAC Address: 00:xx:xx:xx:xx:xx

Network Configuration: Static

VM IP Address: 192.168.0.104

Gateway Address: 192.168.0.104

DNS Servers: 192.168.0.254

Root User SSH Access: yes

Number of Virtual CPUs: 4

Memory Size (MiB): 16348

Root User SSH Public Key: *(None)*

Add Lines to /etc/hosts: yes

Bridge Name: ovirtmgmt

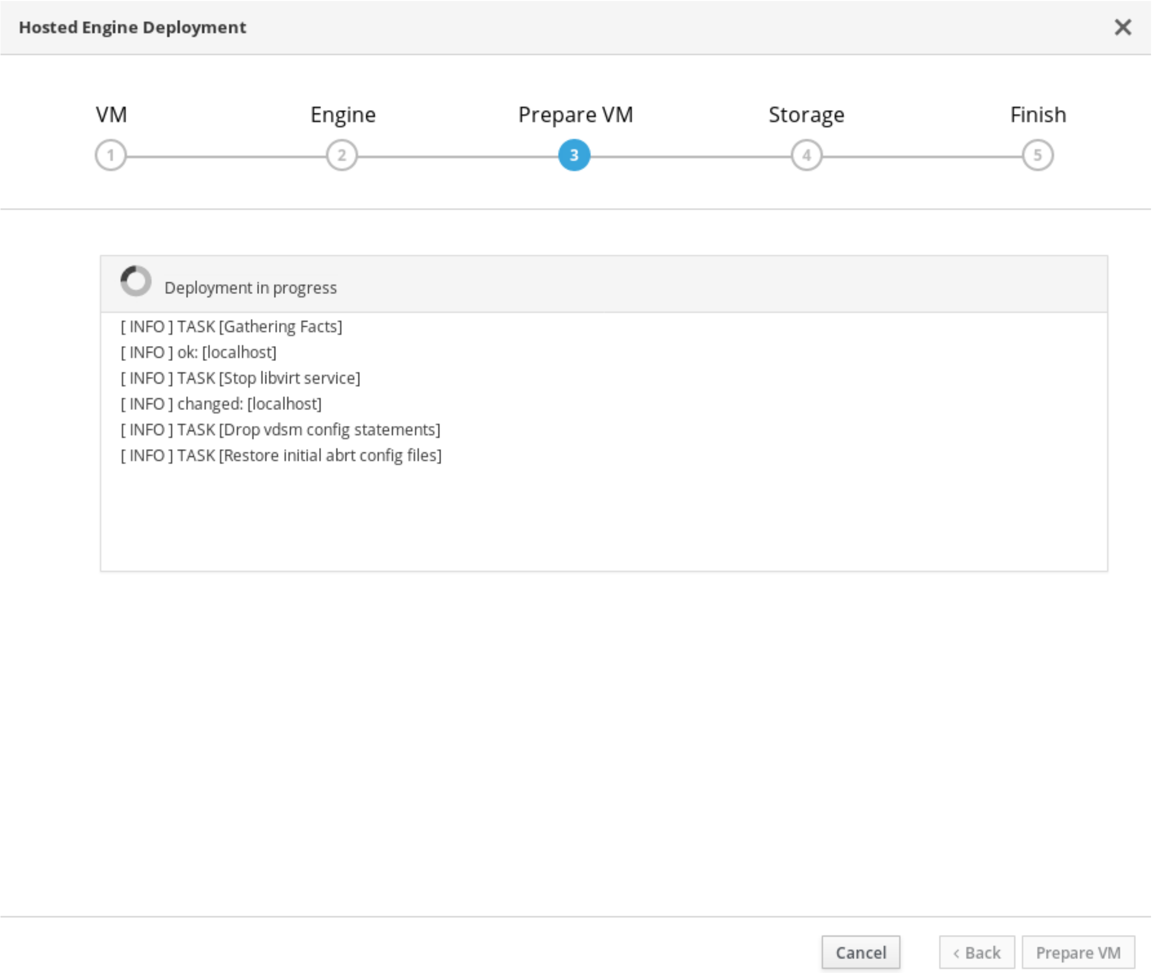
Engine

SMTP Server Name: localhost

SMTP Server Port Number: 25

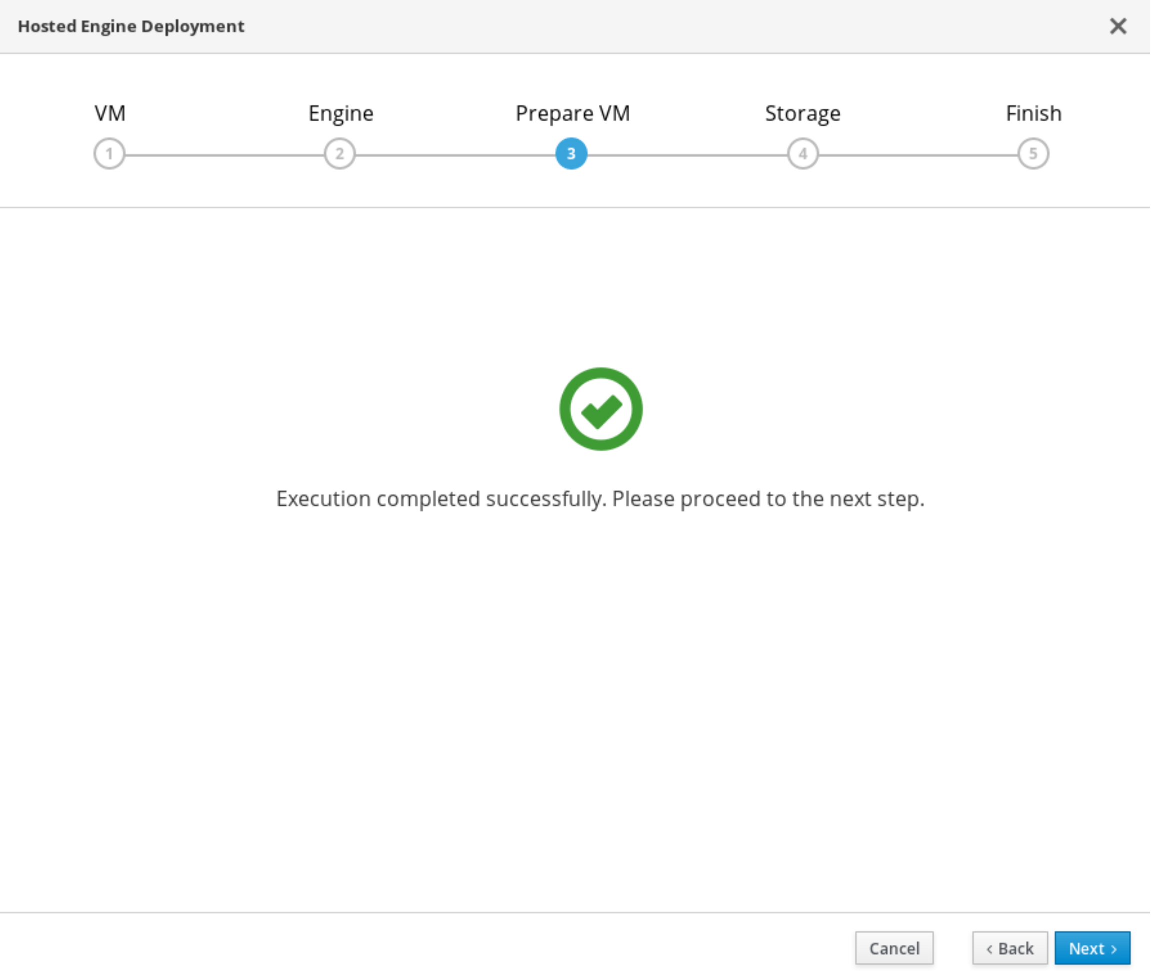
Sender E-Mail Address: root@localhost

b. Click **Prepare VM**.



```
[ INFO ] TASK [Gathering Facts]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Stop libvirt service]
[ INFO ] changed: [localhost]
[ INFO ] TASK [Drop vdsm config statements]
[ INFO ] TASK [Restore initial abrt config files]
```

c. Wait for virtual machine preparation to complete.



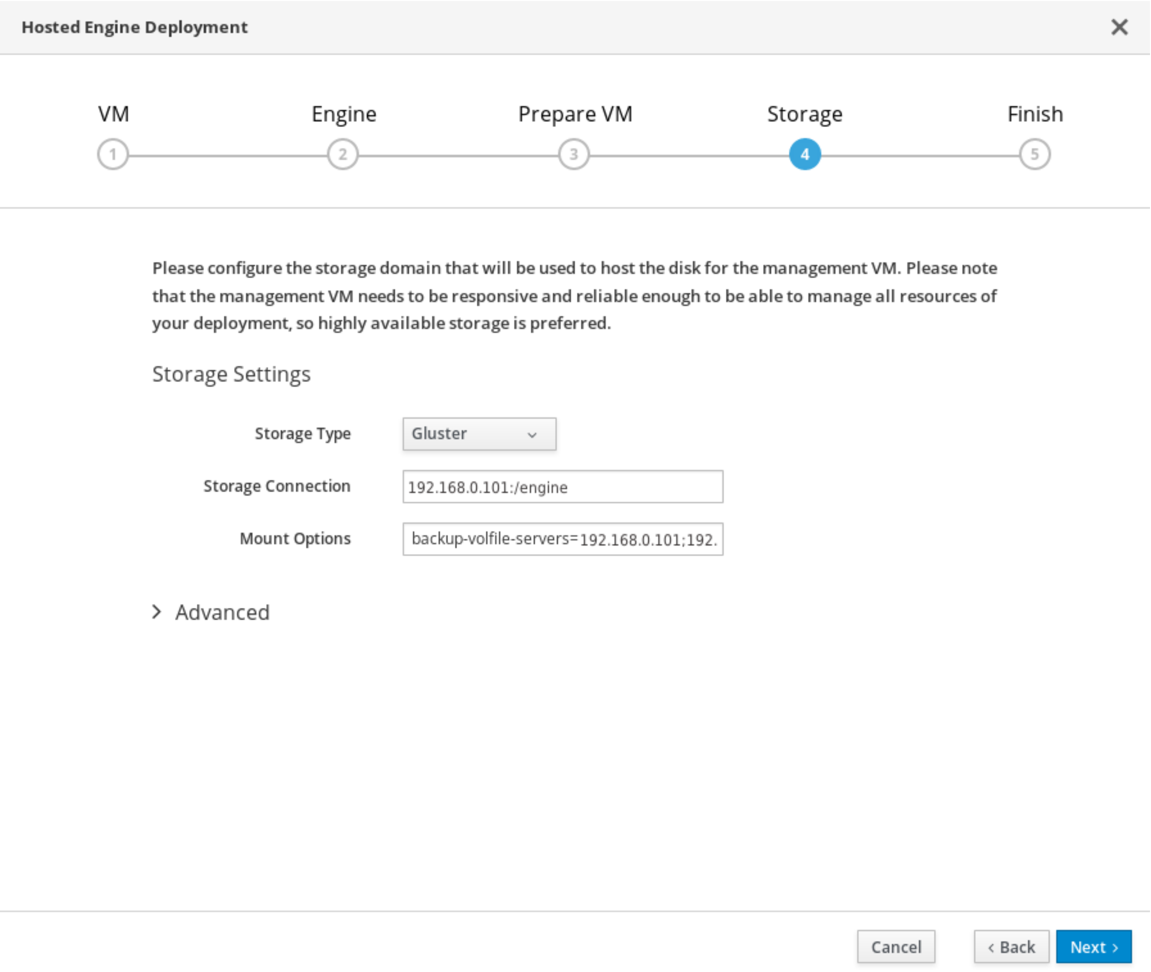
Execution completed successfully. Please proceed to the next step.

If preparation does not occur successfully, see Viewing Hosted Engine deployment errors.

d. Click **Next**.

4. **Specify storage for the Hosted Engine virtual machine**

    a. Specify the primary host and the location of the `engine` volume, and ensure that the `backup-volfile-servers` values listed in **Mount Options** are the IP addresses of the additional virtualization hosts.



Please configure the storage domain that will be used to host the disk for the management VM. Please note that the management VM needs to be responsive and reliable enough to be able to manage all resources of your deployment, so highly available storage is preferred.

Storage Settings

Storage Type: Gluster

Storage Connection: 192.168.0.101:/engine

Mount Options: backup-volfile-servers=192.168.0.101:192.

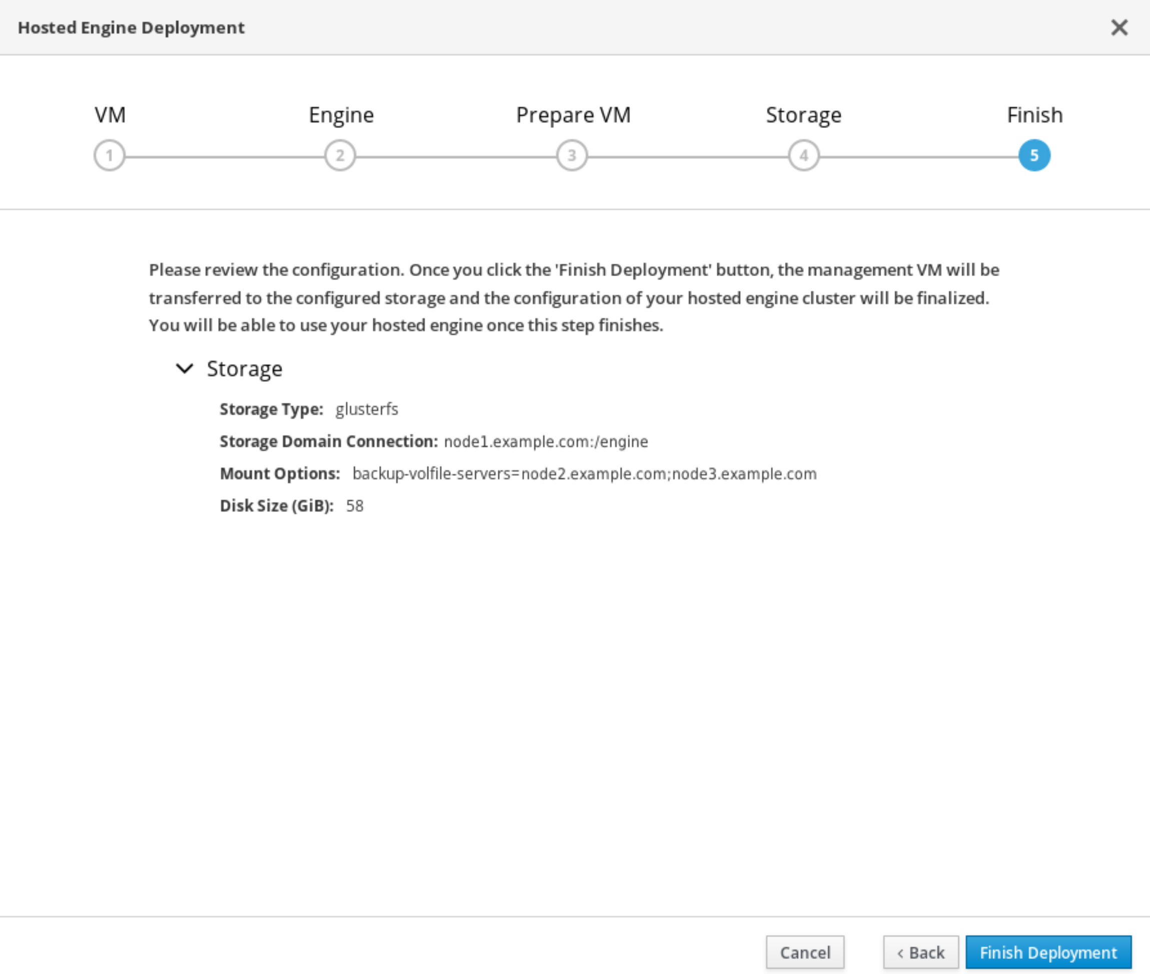
b. Click **Next**.

5. **Finalize Hosted Engine deployment**

a. Review your deployment details and verify that they are correct.

NOTE

The responses you provided during configuration are saved to an answer file to help you reinstall the hosted engine if necessary. The answer file is created at **`/etc/ovirt-hosted-engine/answers.conf`** by default. This file should not be modified manually without assistance from Red Hat Support.



Please review the configuration. Once you click the 'Finish Deployment' button, the management VM will be transferred to the configured storage and the configuration of your hosted engine cluster will be finalized. You will be able to use your hosted engine once this step finishes.

Storage

Storage Type: glusterfs

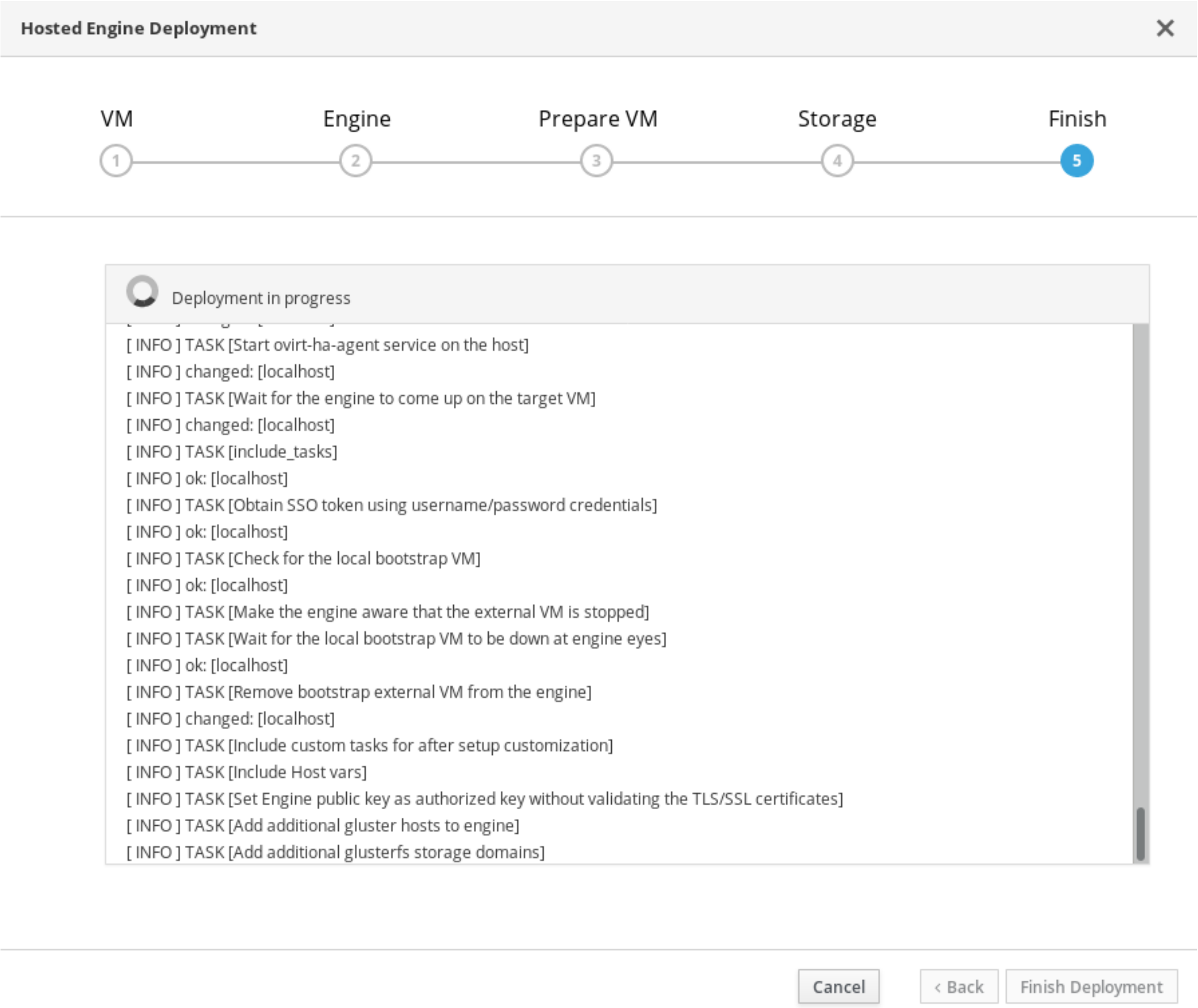
Storage Domain Connection: node1.example.com:/engine

Mount Options: backup-volfile-servers=node2.example.com;node3.example.com

Disk Size (GiB): 58

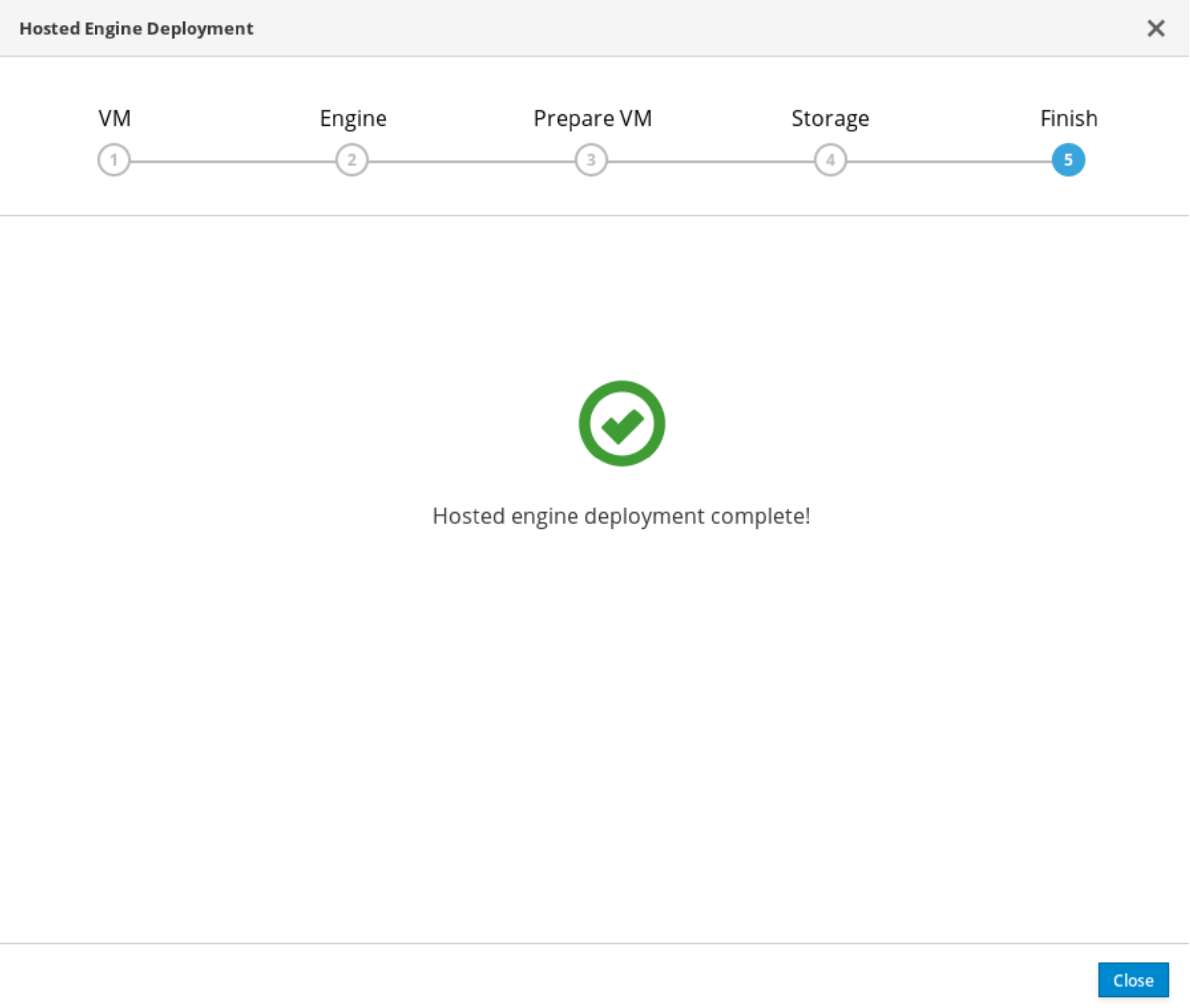
b. Click **Finish Deployment**.

6. **Wait for deployment to complete**
   This takes up to 30 minutes.



```
[ INFO ] TASK [Start ovirt-ha-agent service on the host]
[ INFO ] changed: [localhost]
[ INFO ] TASK [Wait for the engine to come up on the target VM]
[ INFO ] changed: [localhost]
[ INFO ] TASK [include_tasks]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Obtain SSO token using username/password credentials]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Check for the local bootstrap VM]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Make the engine aware that the external VM is stopped]
[ INFO ] TASK [Wait for the local bootstrap VM to be down at engine eyes]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Remove bootstrap external VM from the engine]
[ INFO ] changed: [localhost]
[ INFO ] TASK [Include custom tasks for after setup customization]
[ INFO ] TASK [Include Host vars]
[ INFO ] TASK [Set Engine public key as authorized key without validating the TLS/SSL certificates]
[ INFO ] TASK [Add additional gluster hosts to engine]
[ INFO ] TASK [Add additional glusterfs storage domains]
```

The window displays the following when complete.



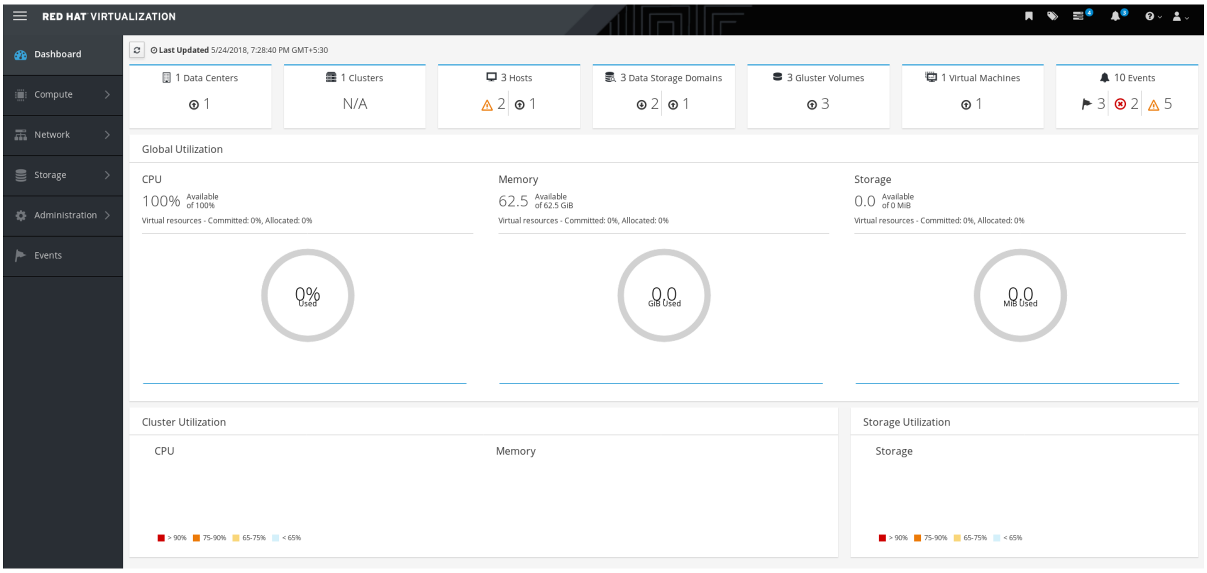
Hosted engine deployment complete!

IMPORTANT

If deployment does not complete successfully, see Viewing Hosted Engine deployment errors.

Click **Close**.

7. **Verify hosted engine deployment**
   Browse to the engine user interface (for example, http://engine.example.com/ovirt-engine) and verify that you can log in using the administrative credentials you configured earlier. Click **Dashboard** and look for your hosts, storage domains, and virtual machines.



Next steps

- Log in to Red Hat Virtualization Manager to complete configuration.

CHAPTER 8. CONFIGURE RED HAT GLUSTER STORAGE AS A RED HAT VIRTUALIZATION STORAGE DOMAIN

8.1. 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 7, *Deploy the Hosted Engine using the Cockpit UI*.

2. **Create a logical network for gluster traffic**

   a. Click **Network** → **Networks** 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 **Network** → **Networks** and select the new logical network.

   b. Click the **Clusters** subtab 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 **Compute** → **Hosts** and select the host.

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

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

   d. Ensure that the **Verify connectivity between Host and Engine** 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**
   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 **Management** → **Refresh Capabilities**.

8.2. ADD ADDITIONAL VIRTUALIZATION HOSTS TO THE HOSTED ENGINE

If you did not specify additional virtualization hosts as part of Configure Red Hat Gluster Storage for Hosted Engine using the Cockpit UI, follow these steps in Red Hat Virtualization Manager for each of the other virtualization hosts.

1. **Add virtualization hosts to the host inventory**
   a. Click **Compute** → **Hosts** and then click **New** to open the *New Host* window.
   b. Provide the **Name**, **Hostname**, and **Password** for the host that you want to manage.
   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 new host**
   a. Click the name of the newly added host to go to the host page.
   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 for Virtualization* to reinstall the host with the **deploy** action.

4. **Verify the health of the network**
   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 **Management** → **Refresh Capabilities**.

See the Red Hat Virtualization 4.2 *Self-Hosted Engine Guide* for further details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.2/html/self-hosted_engine_guide/chap-installing_additional_hosts_to_a_self-hosted_environment

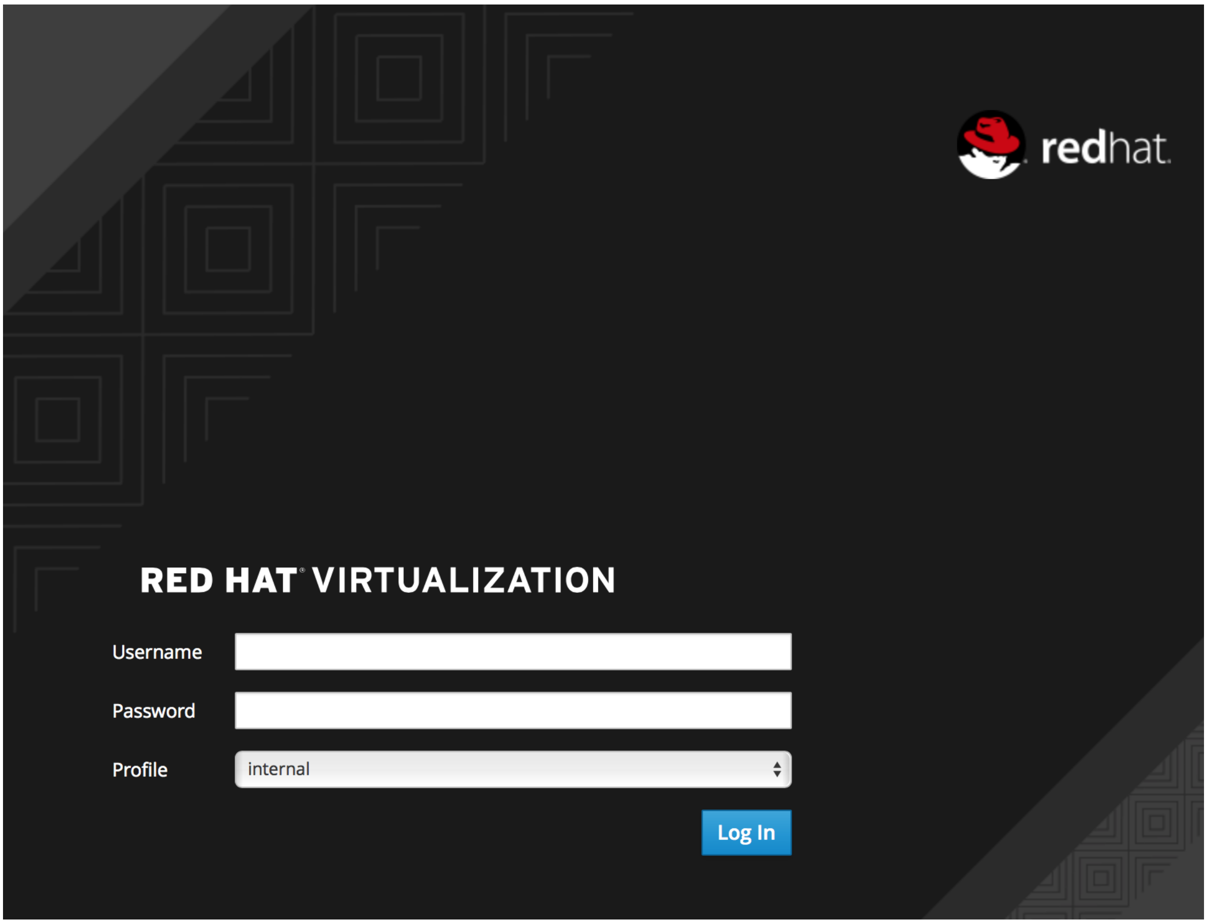
PART III. VERIFY

CHAPTER 9. VERIFY YOUR DEPLOYMENT

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

1. Browse to the engine user interface, for example, http://engine.example.com/ovirt-engine.

   Administration Console Login

   RED HAT® VIRTUALIZATION

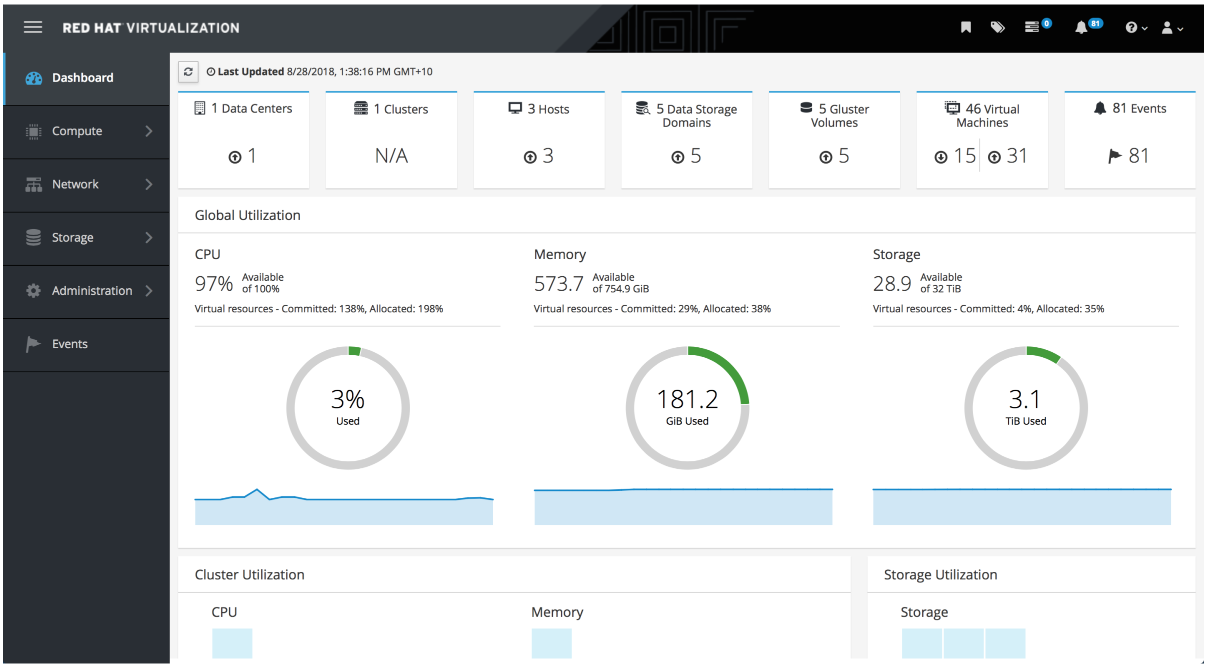
   Username

   Password

   Profile internal

2. Log in using the administrative credentials added during hosted engine deployment. When login is successful, the Dashboard appears.

   Administration Console Dashboard



3. Verify that your cluster is available.

   Administration Console Dashboard - Clusters

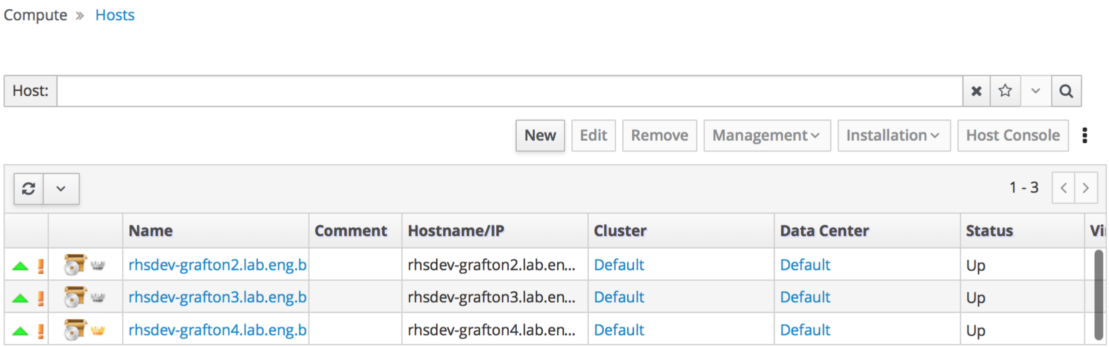
4. Verify that at least one host is available.
   If you provided additional host details during Hosted Engine deployment, 3 hosts are visible here, as shown.

   Administration Console Dashboard - Hosts

   a. Click **Compute** → **Hosts**.

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

      Administration Console - Hosts



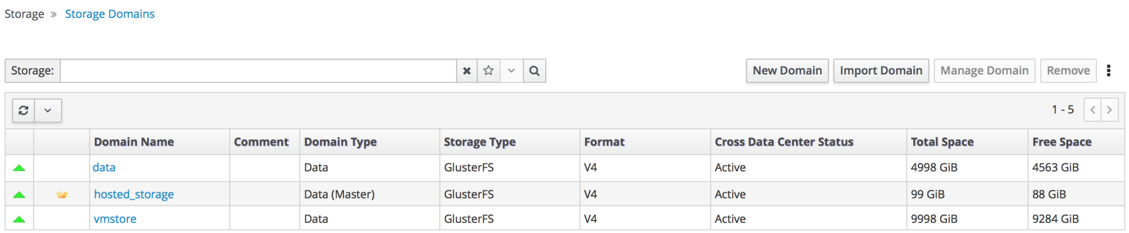
		Name	Comment	Hostname/IP	Cluster	Data Center	Status	Vi
▲ !		rhsdev-grafton2.lab.eng.b		rhsdev-grafton2.lab.en...	Default	Default	Up	
▲ !		rhsdev-grafton3.lab.eng.b		rhsdev-grafton3.lab.en...	Default	Default	Up	
▲ !		rhsdev-grafton4.lab.eng.b		rhsdev-grafton4.lab.en...	Default	Default	Up	

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



		Domain Name	Comment	Domain Type	Storage Type	Format	Cross Data Center Status	Total Space	Free Space
▲		data		Data	GlusterFS	V4	Active	4998 GiB	4563 GiB
▲		hosted_storage		Data (Master)	GlusterFS	V4	Active	99 GiB	88 GiB
▲		vmstore		Data	GlusterFS	V4	Active	9998 GiB	9284 GiB

PART IV. NEXT STEPS

CHAPTER 10. POST-DEPLOYMENT CONFIGURATION SUGGESTIONS

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

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

10.1. 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 for Virtualization deployments.

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

10.2. 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.

10.3. CONFIGURE BACKUP AND RECOVERY OPTIONS

Red Hat recommends configuring at least basic disaster recovery capabilities on all production deployments.

See Configuring backup and recovery options in *Maintaining Red Hat Hyperconverged Infrastructure for Virtualization* for more information.

PART V. TROUBLESHOOT

CHAPTER 11. 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 6, *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`**.

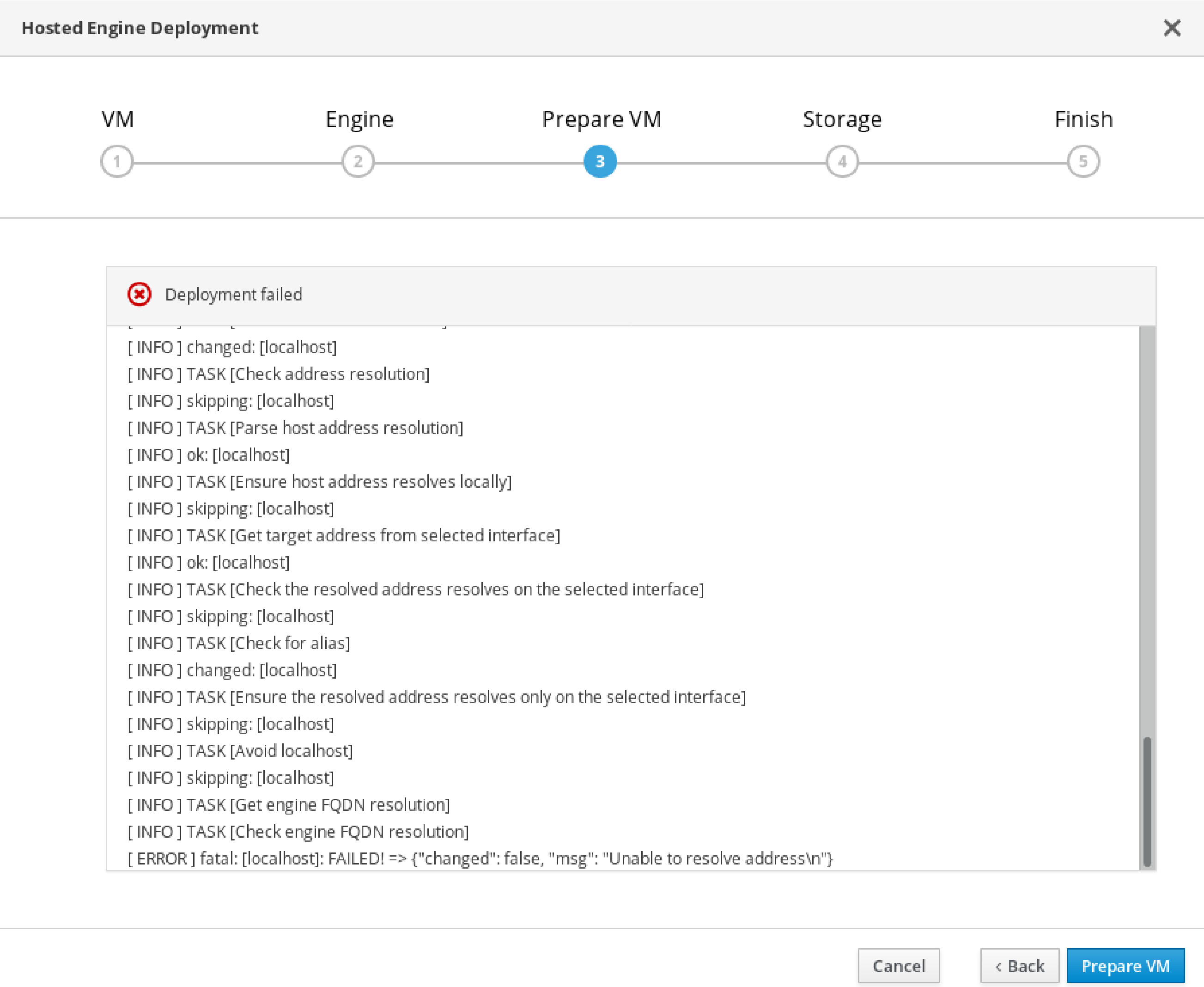
The log files for the Hosted Engine setup portion of the deployment process (documented in Chapter 7, *Deploy the Hosted Engine using the Cockpit UI*) are stored in the **`/var/log/ovirt-hosted-engine-setup`** directory, with file names of the form **`ovirt-hosted-engine-setup-<date>.log`**.

CHAPTER 12. VIEWING HOSTED ENGINE DEPLOYMENT ERRORS

12.1. FAILED TO PREPARE VIRTUAL MACHINE

If an error occurs while preparing the virtual machine, deployment pauses, and you see a screen similar to the following:

Preparing virtual machine failed



```
[ INFO ] changed: [localhost]
[ INFO ] TASK [Check address resolution]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Parse host address resolution]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Ensure host address resolves locally]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Get target address from selected interface]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Check the resolved address resolves on the selected interface]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Check for alias]
[ INFO ] changed: [localhost]
[ INFO ] TASK [Ensure the resolved address resolves only on the selected interface]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Avoid localhost]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Get engine FQDN resolution]
[ INFO ] TASK [Check engine FQDN resolution]
[ ERROR ] fatal: [localhost]: FAILED! => {"changed": false, "msg": "Unable to resolve address\n"}
```

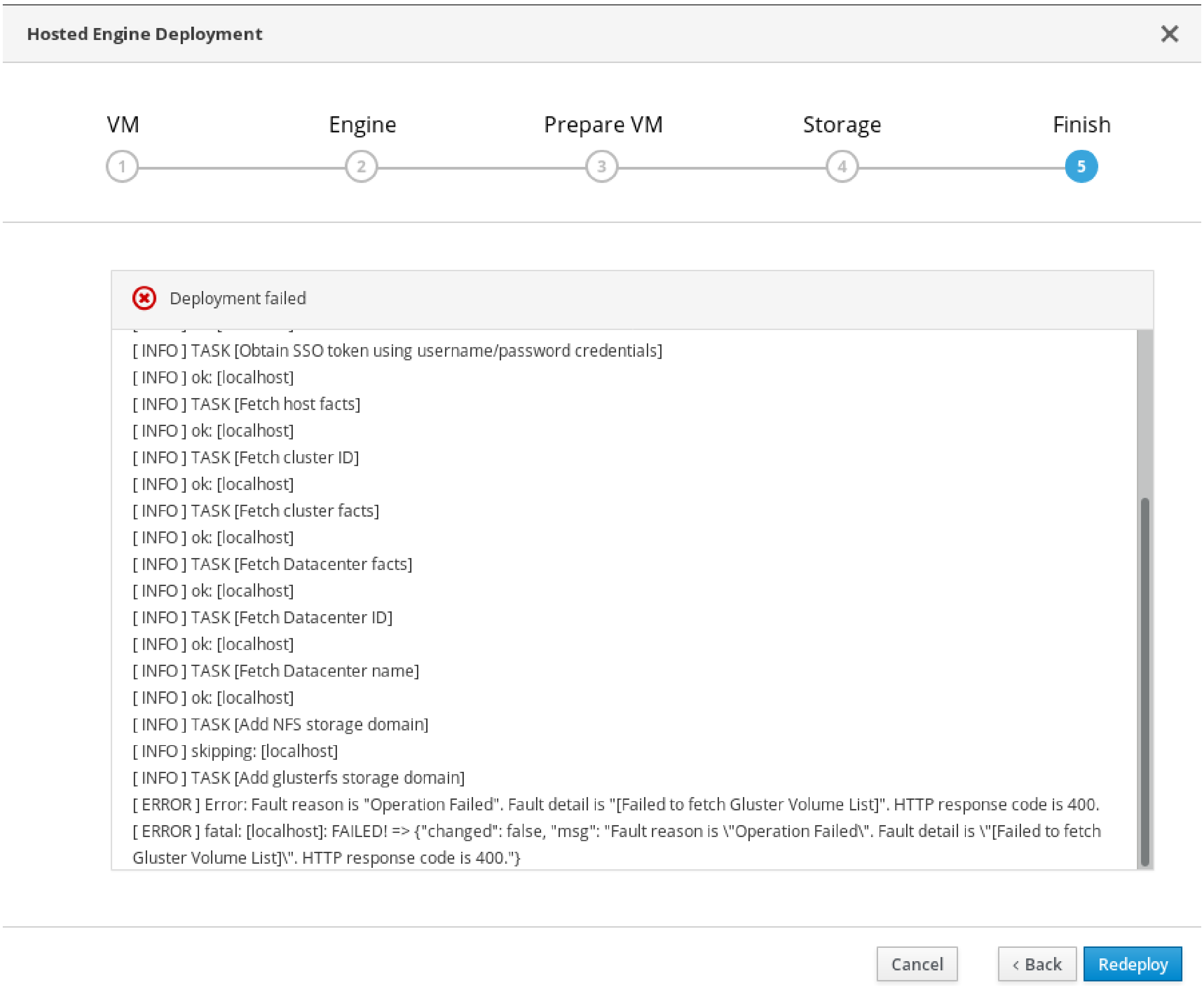
Review the output, and click **Back** to correct any entered values and try again.

Contact Red Hat Support with details of errors for assistance in correcting them.

12.2. FAILED TO DEPLOY HOSTED ENGINE

If an error occurs during hosted engine deployment, deployment pauses, and you see a screen similar to the following:

Hosted engine deployment failed



Deployment failed

```
[ INFO ] TASK [Obtain SSO token using username/password credentials]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch host facts]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch cluster ID]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch cluster facts]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch Datacenter facts]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch Datacenter ID]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Fetch Datacenter name]
[ INFO ] ok: [localhost]
[ INFO ] TASK [Add NFS storage domain]
[ INFO ] skipping: [localhost]
[ INFO ] TASK [Add glusterfs storage domain]
[ ERROR ] Error: Fault reason is "Operation Failed". Fault detail is "[Failed to fetch Gluster Volume List]". HTTP response code is 400.
[ ERROR ] fatal: [localhost]: FAILED! => {"changed": false, "msg": "Fault reason is \"Operation Failed\". Fault detail is \"[Failed to fetch
Gluster Volume List]\". HTTP response code is 400."}
```

Review the output for error information.

Click **Back** to correct any entered values and try again.

If deployment failed after the physical volume or volume group were created, you must also follow the steps in Chapter 13, *Cleaning up automated Red Hat Gluster Storage deployment errors* to return your system to a fresh state for the deployment process.

If you need help resolving errors, contact Red Hat Support with details.

CHAPTER 13. CLEANING UP AUTOMATED RED HAT GLUSTER STORAGE DEPLOYMENT ERRORS

If the deployment process fails after the physical volumes and volume groups are created, you need to undo that work to start the deployment from scratch. Follow this process to clean up a failed deployment so that you can try again.

Procedure

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. Prerequisites

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 6, *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 6, *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 for Virtualization.

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
- 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/3.4/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.

NOTE

If you enable dedeuplication and compression during deployment time, devices in [pv1] and pvname in [vg1] will be **`/dev/mapper/vdo_sdb`**. For more information on VDO configuration, see Appendix D, *Example gdeploy configuration file for configuring compression and deduplication*.

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`**

APPENDIX D. EXAMPLE GDEPLOY CONFIGURATION FILE FOR CONFIGURING COMPRESSION AND DEDUPLICATION

Virtual Data Optimizer (VDO) volumes are supported as of Red Hat Hyperconverged Infrastructure for Virtualization 1.5 when enabled at deployment time. VDO cannot be enabled on existing deployments.

Deploying Red Hat Hyperconverged Infrastructure for Virtualization 1.5 with a Virtual Data Optimizer volume reduces the actual disk space required for a workload, as it enables data compression and deduplication capabilities. This reduces capital and operating expenses.

The **`gdeployConfig.conf`** file is located at **`/var/lib/ovirt-hosted-engine-setup/gdeploy/gdeployConfig.conf`**. This configuration file is applied when **Enable Dedupe & Compression** is checked during deployment.

```
# VDO Configuration
[vdo1:@HOSTNAME@]
action=create
devices=sdb,sdd
names=vdo_sdb,vdo_sdd
logicalsize=164840G,2000G

# Logical size(G) is ten times of actual brick size. If logicalsize >=
1000G, then slabsize=32G.

blockmapcachesize=128M
readcache=enabled
readcachesize=20M
emulate512=on
writepolicy=sync
ignore_vdo_errors=no
slabsize=32G,32G


# Create physical volume on all hosts
[pv1]
action=create
devices=/dev/mapper/vdo_sdb
ignore_pv_errors=no


# Create volume group on all hosts
[vg1]
action=create
vgname=gluster_vg_sdb
pvname=/dev/mapper/vdo_sdb
ignore_vg_errors=no
```