



Red Hat Hyperconverged Infrastructure for Virtualization 1.8

Deploying Red Hat Hyperconverged Infrastructure for Virtualization on a single node

Create a hyperconverged configuration with a single server

Red Hat Hyperconverged Infrastructure for Virtualization 1.8 Deploying Red Hat Hyperconverged Infrastructure for Virtualization on a single node

Create a hyperconverged configuration with a single server

Laura Bailey

lbailey@redhat.com

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Read this for information about deploying a single self-contained Red Hat Hyperconverged Infrastructure for Virtualization server.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. WORKFLOW FOR DEPLOYING A SINGLE HYPERCONVERGED HOST	5
CHAPTER 2. ADDITIONAL REQUIREMENTS FOR SINGLE NODE DEPLOYMENTS	6
CHAPTER 3. INSTALLING OPERATING SYSTEMS	7
3.1. INSTALLING HYPERCONVERGED HOSTS	7
3.1.1. Installing a hyperconverged host with Red Hat Virtualization 4	7
3.1.1.1. Downloading the Red Hat Virtualization 4 operating system	7
3.1.1.2. Installing the Red Hat Virtualization 4 operating system on hyperconverged hosts	7
3.2. INSTALLING NETWORK-BOUND DISK ENCRYPTION KEY SERVERS	9
3.2.1. Installing an NBDE key server with Red Hat Enterprise Linux 8	9
3.2.1.1. Downloading the Red Hat Enterprise Linux 8 operating system	9
3.2.1.2. Installing the Red Hat Enterprise Linux 8 operating system on Network-Bound Disk Encryption key servers	9
3.2.2. Installing an NBDE key server with Red Hat Enterprise Linux 7	11
3.2.2.1. Downloading the Red Hat Enterprise Linux 7 operating system	11
3.2.2.2. Installing the Red Hat Enterprise Linux 7 operating system on Network-Bound Disk Encryption key servers	11
CHAPTER 4. INSTALL ADDITIONAL SOFTWARE	13
4.1. CONFIGURING SOFTWARE ACCESS	13
4.1.1. Configuring software repository access using the Web Console	13
4.2. INSTALLING SOFTWARE	14
4.2.1. Installing disk encryption software	14
CHAPTER 5. MODIFYING FIREWALL RULES	15
5.1. MODIFYING FIREWALL RULES FOR DISK ENCRYPTION	15
CHAPTER 6. CONFIGURE KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD	16
6.1. GENERATING SSH KEY PAIRS WITHOUT A PASSWORD	16
6.2. COPYING SSH KEYS	17
CHAPTER 7. CONFIGURE DISK ENCRYPTION	18
7.1. CONFIGURING NETWORK-BOUND DISK ENCRYPTION KEY SERVERS	18
7.2. CONFIGURING HYPERCONVERGED HOSTS AS NETWORK-BOUND DISK ENCRYPTION CLIENTS	18
7.2.1. Defining disk encryption configuration details	18
7.2.2. Executing the disk encryption configuration playbook	19
CHAPTER 8. CONFIGURING A SINGLE NODE RHVI FOR VIRTUALIZATION DEPLOYMENT	21
8.1. CONFIGURING RED HAT GLUSTER STORAGE ON A SINGLE NODE	21
8.2. DEPLOY THE HOSTED ENGINE ON A SINGLE NODE USING THE WEB CONSOLE	26
CHAPTER 9. VERIFY YOUR DEPLOYMENT	37
CHAPTER 10. POST-DEPLOYMENT CONFIGURATION SUGGESTIONS	39
10.1. CONFIGURE NOTIFICATIONS	39
10.2. (OPTIONAL)CONFIGURE HOST POWER MANAGEMENT	39
10.3. CONFIGURE BACKUP AND RECOVERY OPTIONS	39
CHAPTER 11. NEXT STEPS	40
11.1. ENABLING THE RED HAT VIRTUALIZATION MANAGER REPOSITORIES	40
PART I. TROUBLESHOOT	42

CHAPTER 12. LOG FILE LOCATIONS	43
CHAPTER 13. DEPLOYMENT ERRORS	44
13.1. ORDER OF CLEANUP OPERATIONS	44
13.2. FAILED TO DEPLOY STORAGE	44
13.2.1. Cleaning up Network-Bound Disk Encryption after a failed deployment	45
13.2.2. Error: VDO signature detected on device	45
13.2.3. Manually cleaning up a VDO device	46
13.3. FAILED TO PREPARE VIRTUAL MACHINE	46
13.4. FAILED TO DEPLOY HOSTED ENGINE	47
PART II. REFERENCE MATERIAL	50
APPENDIX A. WORKING WITH FILES ENCRYPTED USING ANSIBLE VAULT	51
A.1. ENCRYPTING FILES	51
A.2. EDITING ENCRYPTED FILES	51
A.3. REKEYING ENCRYPTED FILES TO A NEW PASSWORD	52
APPENDIX B. UNDERSTANDING THE LUKS_TANG_INVENTORY.YML FILE	53
B.1. CONFIGURATION PARAMETERS FOR DISK ENCRYPTION	53
B.2. EXAMPLE LUKS_TANG_INVENTORY.YML	55
APPENDIX C. GLOSSARY OF TERMS	58
C.1. VIRTUALIZATION TERMS	58
C.2. STORAGE TERMS	59
C.3. HYPERCONVERGED INFRASTRUCTURE TERMS	59

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. WORKFLOW FOR DEPLOYING A SINGLE HYPERCONVERGED HOST

1. Check requirements.

Verify that your planned deployment meets support requirements: [Requirements](#), and fill in the [installation checklist](#) so that you can refer to it during the deployment process.

2. Install operating systems.

- a. Install an operating system on each physical machine that will act as a hyperconverged host: [Installing hyperconverged hosts](#).
- b. (Optional) Install an operating system on each physical or virtual machine that will act as an Network-Bound Disk Encryption (NBDE) key server: [Installing NBDE key servers](#).

3. Modify firewall rules for additional software.

- a. (Optional) Modify firewall rules for disk encryption: [Section 5.1, “Modifying firewall rules for disk encryption”](#).

4. Configure authentication between hyperconverged hosts.

Configure key-based SSH authentication without a password to enable automated configuration of the hosts: [Configure key-based SSH without a password](#).

5. (Optional) Configure disk encryption.

- a. [Configure NBDE key servers](#).
- b. [Configure hyperconverged hosts as NBDE clients](#).

6. Configure the hyperconverged node.

Browse to the Web Console and [deploy a single hyperconverged node](#).

CHAPTER 2. 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

Single node deployments cannot be scaled and are not highly available. This deployment type is lower cost, but removes the option of availability.

CHAPTER 3. INSTALLING OPERATING SYSTEMS

3.1. INSTALLING HYPERCONVERGED HOSTS

The supported operating system for hyperconverged hosts is the latest version of Red Hat Virtualization 4.

3.1.1. Installing a hyperconverged host with Red Hat Virtualization 4

3.1.1.1. Downloading the Red Hat Virtualization 4 operating system

1. Navigate to the [Red Hat Customer Portal](#).
2. Click **Downloads** to get a list of product downloads.
3. Click **Red Hat Virtualization**.
4. Click **Download latest**
5. In the **Product Software** tab, click the **Download** button beside the latest Hypervisor Image, for example, **Hypervisor Image for RHV 4.4**.
6. When the file has downloaded, verify its SHA-256 checksum matches the one on the page.

```
$ sha256sum image.iso
```

7. Use the downloaded image to create an installation media device.
See [Creating installation media](#) in the Red Hat Enterprise Linux 8 documentation.

3.1.1.2. Installing the Red Hat Virtualization 4 operating system on hyperconverged hosts

Prerequisites

- Be aware that this operating system is only supported for hyperconverged hosts. Do not install an Network-Bound Disk Encryption (NBDE) key server with this operating system.
- Be aware of additional server requirements when enabling disk encryption on hyperconverged hosts. See [Disk encryption requirements](#) for details.

Procedure

1. Start the machine and boot from the prepared installation media.
2. From the boot menu, select **Install Red Hat Virtualization 4** and press **Enter**.
3. Select a language and click **Continue**.
4. Accept the default **Localization** options.
5. Click **Installation destination**.
 - a. Deselect any disks you do not want to use as installation locations, for example, any disks that will be used for storage domains.

**WARNING**

Disks with a check mark will be formatted and all their data will be lost. If you are reinstalling this host, ensure that disks with data that you want to retain do not show a check mark.

- b. Select the **Automatic partitioning** option.
- c. (Optional) If you want to use disk encryption, select **Encrypt my data** and specify a password.

**WARNING**

Remember this password, as your machine will not boot without it.

This password is used as the **rootpassphrase** for this host during Network-Bound Disk Encryption setup.

- d. Click **Done**.
6. Click **Network and Host Name**
 - a. Toggle the **Ethernet** switch to **ON**.
 - b. Select the network interface and click **Configure**
 - i. On the **General** tab, check the **Connect automatically with priority** checkbox.
 - ii. (Optional) To use IPv6 networking instead of IPv4, specify network details on the **IPv6 settings** tab.
For static network configurations, ensure that you provide the static IPv6 address, prefix, and gateway, as well as IPv6 DNS servers and additional search domains.

**IMPORTANT**

You must use either IPv4 or IPv6; mixed networks are not supported.

- iii. Click **Save**.
 - c. Click **Done**.
7. (Optional) Configure Security policy.
 8. Click **Begin installation**.
 - a. Set a root password.

**WARNING**

Red Hat recommends not creating additional users on hyperconverged hosts, as this can lead to exploitation of local security vulnerabilities.

- b. Click **Reboot** to complete installation.
9. Increase the size of the `/var/log` partition.
You need at least 15 GB of free space for Red Hat Gluster Storage logging requirements. Follow the instructions in [Growing a logical volume using the Web Console](#) to increase the size of this partition.

3.2. INSTALLING NETWORK-BOUND DISK ENCRYPTION KEY SERVERS

If you want to use Network-Bound Disk Encryption to encrypt the contents of your disks in Red Hat Hyperconverged Infrastructure for Virtualization, you need to install at least one key server.

The supported operating systems for Network-Bound Disk Encryption (NBDE) key servers are the latest versions of Red Hat Enterprise Linux 7 and 8.

3.2.1. Installing an NBDE key server with Red Hat Enterprise Linux 8

3.2.1.1. Downloading the Red Hat Enterprise Linux 8 operating system

1. Navigate to the [Red Hat Customer Portal](#).
2. Click **Downloads** to get a list of product downloads.
3. Click **Red Hat Enterprise Linux 8**
4. In the **Product Software** tab, click **Download** beside the latest binary DVD image, for example, **Red Hat Enterprise Linux 8.2 Binary DVD**.
5. When the file has downloaded, verify its SHA-256 checksum matches the one on the page.

```
$ sha256sum image.iso
```

6. Use the image to create an installation media device.
See [Creating installation media](#) in the Red Hat Enterprise Linux 8 documentation for details.

3.2.1.2. Installing the Red Hat Enterprise Linux 8 operating system on Network-Bound Disk Encryption key servers

Procedure

1. Start the machine and boot from the prepared installation media.
2. From the boot menu, select **Install Red Hat Enterprise Linux 8** and press **Enter**.

3. Select a language and click **Continue**.
4. Accept the default **Localization** and **Software** options.
5. Click **Installation destination**.
 - a. Select the disk that you want to install the operating system on.

**WARNING**

Disks with a check mark will be formatted and all their data will be lost. If you are reinstalling this host, ensure that disks with data that you want to retain do not show a check mark.

- b. (Optional) If you want to use disk encryption, select **Encrypt my data** and specify a password.

**WARNING**

Remember this password, as your machine will not boot without it.

- c. Click **Done**.
 6. Click **Network and Host Name**.
 - a. Toggle the **Ethernet** switch to **ON**.
 - b. Select the network interface and click **Configure**.
 - i. On the **General** tab, check the **Connect automatically with priority** checkbox.
 - ii. (Optional) To use IPv6 networking instead of IPv4, specify network details on the **IPv6 settings** tab.

For static network configurations, ensure that you provide the static IPv6 address, prefix, and gateway, as well as IPv6 DNS servers and additional search domains.
- iii. Click **Save**.
 - c. Click **Done**.
7. (Optional) Configure Security policy.

**IMPORTANT**

You must use either IPv4 or IPv6; mixed networks are not supported.

8. Click **Begin installation**.
 - a. Set a root password.
 - b. Click **Reboot** to complete installation.
9. From the **Initial Setup** window, accept the licensing agreement and register your system.

3.2.2. Installing an NBDE key server with Red Hat Enterprise Linux 7

3.2.2.1. Downloading the Red Hat Enterprise Linux 7 operating system

1. Navigate to the [Red Hat Customer Portal](#).
2. Click **Downloads** to get a list of product downloads.
3. Click **Versions 7 and below**.
4. In the **Product Software** tab, click **Download** beside the latest binary DVD image, for example, **Red Hat Enterprise Linux 7.8 Binary DVD**.
5. When the file has downloaded, verify its SHA-256 checksum matches the one on the page.

```
$ sha256sum image.iso
```

6. Use the image to create an installation media device.
See [Creating installation media](#) in the Red Hat Enterprise Linux 8 documentation for details.

3.2.2.2. Installing the Red Hat Enterprise Linux 7 operating system on Network-Bound Disk Encryption key servers

Prerequisites

- Be aware that this operating system is only supported for Network-Bound Disk Encryption (NBDE) key servers. Do not install a hyperconverged host with this operating system.

Procedure

1. Start the machine and boot from the prepared installation media.
2. From the boot menu, select **Install Red Hat Enterprise Linux 7** and press **Enter**.
3. Select a language and click **Continue**.
4. Click **Date & Time**.
 - a. Select a time zone.
 - b. Click **Done**.
5. Click **Keyboard**.
 - a. Select a keyboard layout.
 - b. Click **Done**.

6. Click **Installation destination**.
 - a. Deselect any disks you do not want to use as an installation location.
 - b. If you want to use disk encryption, select **Encrypt my data** and specify a password.

**WARNING**

Remember this password, as your machine will not boot without it.

- c. Click **Done**.
7. Click **Network and Host Name**.
 - a. Click **Configure... → General**.
 - b. Check the **Automatically connect to this network when it is available** check box.
 - c. Click **Done**.
8. Optionally, configure language support, security policy, and kdump.
9. Click **Begin installation**.
 - a. Set a root password.
 - b. Click **Reboot** to complete installation.
10. From the **Initial Setup** window, accept the licensing agreement and register your system.

CHAPTER 4. INSTALL ADDITIONAL SOFTWARE

You need to perform some additional configuration for access to software and updates.

- Ensure you have access to software updates: [Configure software repository access using the web console](#).
- If your hyperconverged hosts use disk encryption, [Install disk encryption software](#).

4.1. CONFIGURING SOFTWARE ACCESS

4.1.1. Configuring software repository access using the Web Console

Prerequisites

- This process is for hyperconverged hosts based on Red Hat Virtualization 4.

Procedure

1. On each hyperconverged host:

- Log in to the Web Console.
Use the management FQDN and port 9090, for example, **`https://server1.example.com:9090/`**.
- Click **Subscriptions**.
- Click **Register System**.
 - Enter your Customer Portal user name and password.
 - Click **Done**.
The Red Hat Virtualization Host subscription is automatically attached to the system.
- Enable the Red Hat Virtualization 4 repository to allow later updates to the Red Hat Virtualization Host:

```
# subscription-manager repos \
--enable=rhvh-4-for-rhel-8-x86_64-rpms
```

2. (Optional) If you use disk encryption, execute the following on each Network-Bound Disk Encryption (NBDE) key server:

- Log in to the NBDE key server.
- Register the NBDE key server with Red Hat.

```
# subscription-manager register --username=username --password=password
```

- Attach the subscription pool:

```
# subscription-manager attach --pool=pool_id
```

- Enable the repositories required for disk encryption software:

- i. For NBDE key servers based on Red Hat Enterprise Linux 8:

```
# subscription-manager repos \  
--enable="rhel-8-for-x86_64-baseos-rpms" \  
--enable="rhel-8-for-x86_64-appstream-rpms"
```

- ii. For NBDE key servers based on Red Hat Enterprise Linux 7:

```
# subscription-manager repos --enable="rhel-7-server-rpms"
```

4.2. INSTALLING SOFTWARE

4.2.1. Installing disk encryption software

The Network-Bound Disk Encryption key server requires an additional package to support disk encryption.

Prerequisites

- [Configuring software repository access using the web console](#) .

Procedure

1. On each Network-Bound Disk Encryption (NBDE) key server, install the server-side packages.

```
# yum install tang -y
```

CHAPTER 5. MODIFYING FIREWALL RULES

5.1. MODIFYING FIREWALL RULES FOR DISK ENCRYPTION

On Network-Bound Disk Encryption (NBDE) key servers, you need to open ports so that encryption keys can be served.

Procedure

1. On each NBDE key server:
 - a. Open ports required to serve encryption keys.



NOTE

The default port is **80/tcp**. To use a custom port, see [Deploying a tang server with SELinux in enforcing mode](#) in the Red Hat Enterprise Linux 8 documentation.

```
# firewall-cmd --add-port=80/tcp
# firewall-cmd --add-port=80/tcp --permanent
```

- b. Verify that the port appears in the output of the following command.

```
# firewall-cmd --list-ports | grep '80/tcp'
```

CHAPTER 6. CONFIGURE KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD

Configure key-based SSH authentication without a password for the root user from the host to itself.

6.1. GENERATING SSH KEY PAIRS WITHOUT A PASSWORD

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

Procedure

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

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

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

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

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

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

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

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

**WARNING**

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

6.2. COPYING SSH KEYS

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

Prerequisites

- Generate a public/private key pair with no password.

Procedure

1. Log in to the host as the root user.
2. Copy the public key to the same host:

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

Enter the password for **root@<hostname>** when prompted.

**WARNING**

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

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

```
# ssh-copy-id -i <location>/<keyname>.pub root@server1.example.com
```

CHAPTER 7. CONFIGURE DISK ENCRYPTION

7.1. CONFIGURING NETWORK-BOUND DISK ENCRYPTION KEY SERVERS

Prerequisites

- You must have installed a Network-Bound Disk Encryption key server ([Installing Network-Bound Disk Encryption key servers](#)).

Procedure

- Start and enable the tangd service:
Run the following command on each Network-Bound Disk Encryption (NBDE) key server.

```
# systemctl enable tangd.socket --now
```

- Verify that hyperconverged hosts have access to the key server.
 - Log in to a hyperconverged host.
 - Request a decryption key from the key server.

```
# curl key-server.example.com/adv
```

If you see output like the following, the key server is accessible and advertising keys correctly.

```
{"payload":"eyJrZXlzljpbeyJhbGciOiJFQ01SliwiY3J2ljoic01MjEiLCJrZXlfb3BzljpbImRlcmll
2ZUtleSjdLCJrdHkiOiJFQyIsIngiOiJBQ2ZjNVFwVmlhal9wNWcwUIE4VW52dmdNN1AyRT
Rqa21XUEpSM3VRUkFsVWp0eWlfZ0Y5WEV3WmU5TmhlIdHhDaG53OXhMSkphajRieVks
1ZVFGNGxhcXQ2liwieSI6IkFOMmhpcmNpU2tnWG5HV2VHeGN1Nzk3N3B3empCTzZjZ
Wt5TFJZdlh4SkNvb3BfNmdZdnR2bEpJUk4wS211Y1g3WHUwMINVWlpcTVVxU3EtdGwy
eEQ1SGcifSx7ImFsZyl6IkVTNTEyIiwia3J2ljoic01MjEiLCJrZXlfb3BzljpbInZicmlmeSjdLC
JrdHkiOiJFQyIsIngiOiJBQXlXeU8zTTFEWEdlS1PZ04tRFhHU29yNI9BcUIJdzQ5OHhRTz
dMam1kMnJ5bDN2WUFXTUVyR1I2MVhKdzdvbEhxdEdDQnhqV0I4RzZZV09vLWRpTUx
wliwieSI6IkFVWkNXUTAxd3lVMXIYR2R0SUMtOHJhVUVadWM5V3JyekFVbUIyQVF5VTR
sWDcxd1RUWTJEeDIMMzliQU9tVk5oRGstS2lQNFZfYUlsZDFqVI9zdHRuVGofV19","prot
ected":"eyJhbGciOiJFUzUxMlslmN0eSI6Imp3ay1zZXQranNvbiJ9","signature":"ARiMIYnGj
7-1C-
ZAQ_CKee676s_vYpi9J94WBibroou5MRsO6ZhRohqh_SCbW1jWWJr8btymTfQgBF_Rwz
VNCnlIAXt_D5KSu8UDc4LnKU-egjV-
02b61aiWB0udiEfYkF66krlajzA9y5j7qTdZpWsBObYVvuoJvIRo_jpzXJv0qEMi"}
```

7.2. CONFIGURING HYPERCONVERGED HOSTS AS NETWORK-BOUND DISK ENCRYPTION CLIENTS

7.2.1. Defining disk encryption configuration details

- Log in to the first hyperconverged host.

2. Change into the **hc-ansible-deployment** directory:

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

3. Make a copy of the **luks_tang_inventory.yml** file for future reference.

```
cp luks_tang_inventory.yml luks_tang_inventory.yml.backup
```

4. Define your configuration in the `luks_tang_inventory.yml` file.
Use the example **luks_tang_inventory.yml** file to define the details of disk encryption on each host. A complete outline of this file is available in [Understanding the luks_tang_inventory.yml file](#).

5. Encrypt the **luks_tang_inventory.yml** file and specify a password using **ansible-vault**.
The required variables in **luks_tang_inventory.yml** include password values, so it is important to encrypt the file to protect the password values.

```
# ansible-vault encrypt luks_tang_inventory.yml
```

Enter and confirm a new vault password when prompted.

7.2.2. Executing the disk encryption configuration playbook

Prerequisites

- Define configuration in the **luks_tang_inventory.yml** playbook: [Section 7.2.1, “Defining disk encryption configuration details”](#).
- Hyperconverged hosts must have encrypted boot disks.

Procedure

1. Log in to the first hyperconverged host.
2. Change into the `hc-ansible-deployment` directory.

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

3. Run the following command as the root user to start the configuration process.

```
# ansible-playbook -i luks_tang_inventory.yml tasks/luks_tang_setup.yml --tags=blacklistdevices,luksencrypt,bindtang --ask-vault-pass
```

Enter the vault password for this file when prompted to start disk encryption configuration.

Verify

- Reboot each host and verify that they are able to boot to a login prompt without requiring manual entry of the decryption passphrase.
- Note that the devices that use disk encryption have a path of `/dev/mapper/luks_sdX` when you continue with Red Hat Hyperconverged Infrastructure for Virtualization setup.

Troubleshooting

- The given boot device/**dev/sda2** is not encrypted.

```
TASK [Check if root device is encrypted]
fatal: [server1.example.com]: FAILED! => {"changed": false, "msg": "The given boot device
/dev/sda2 is not encrypted."}
```

Solution: Reinstall the hyperconverged hosts using the process outlined in [Section 3.1, “Installing hyperconverged hosts”](#), ensuring that you select **Encrypt my data** during the installation process and follow all directives related to disk encryption.

- The output has been hidden due to the fact that `no_log: true` was specified for this result.

```
TASK [gluster.infra/roles/backend_setup : Encrypt devices using key file]
failed: [host1.example.com] (item=None) => {"censored": "the output has been hidden due to
the fact that no_log: true was specified for this result", "changed": true}
```

This output has been censored in order to not expose a passphrase. If you see this output for the **Encrypt devices using key file** task, the device failed to encrypt. You may have provided the incorrect disk in the inventory file.

Solution: Clean up the deployment attempt using [Cleaning up Network-Bound Disk Encryption after a failed deployment](#). Then correct the disk names in the inventory file.

- Non-zero return code from Tang server

```
TASK [gluster.infra/roles/backend_setup : Download the advertisement from tang server for
IPv4] * failed: [host1.example.com] (item={url: http://tang-server.example.com}) =>
{"ansible_index_var": "index", "ansible_loop_var": "item", "changed": true, "cmd": "curl -sfg
'http://tang-server.example.com/adv\' -o /etc/adv0.jws", "delta": "0:02:08.703711", "end":
"2020-06-10 18:18:09.853701", "index": 0, "item": {"url": "http://tang-server.example.com"},
"msg": "non-zero return code*", "rc": 7, "start": "2020-06-10 18:16:01.149990", "stderr": "",
"stderr_lines": [], "stdout": "", "stdout_lines": []}
```

This error indicates that the server cannot access the **url** provided, either because the **FQDN** provided is incorrect or because it cannot be found from the host.

Solution: Correct the **url** value provided for the NBDE key server or ensure that the **url** value is accessible from the host. Then run the playbook again with the **bindtang** tag:

```
# ansible-playbook -i luks_tang_inventory.yml tasks/luks_tang_setup.yml --ask-vault-pass --
tags=bindtang
```

- For any other playbook failures, use the instructions in [Cleaning up Network-Bound Disk Encryption after a failed deployment](#) to clean up your deployment. Review the playbook and inventory files for incorrect values and test access to all servers before executing the configuration playbook again.

CHAPTER 8. CONFIGURING A SINGLE NODE RHHI FOR VIRTUALIZATION DEPLOYMENT

8.1. CONFIGURING RED HAT GLUSTER STORAGE ON A SINGLE NODE



IMPORTANT

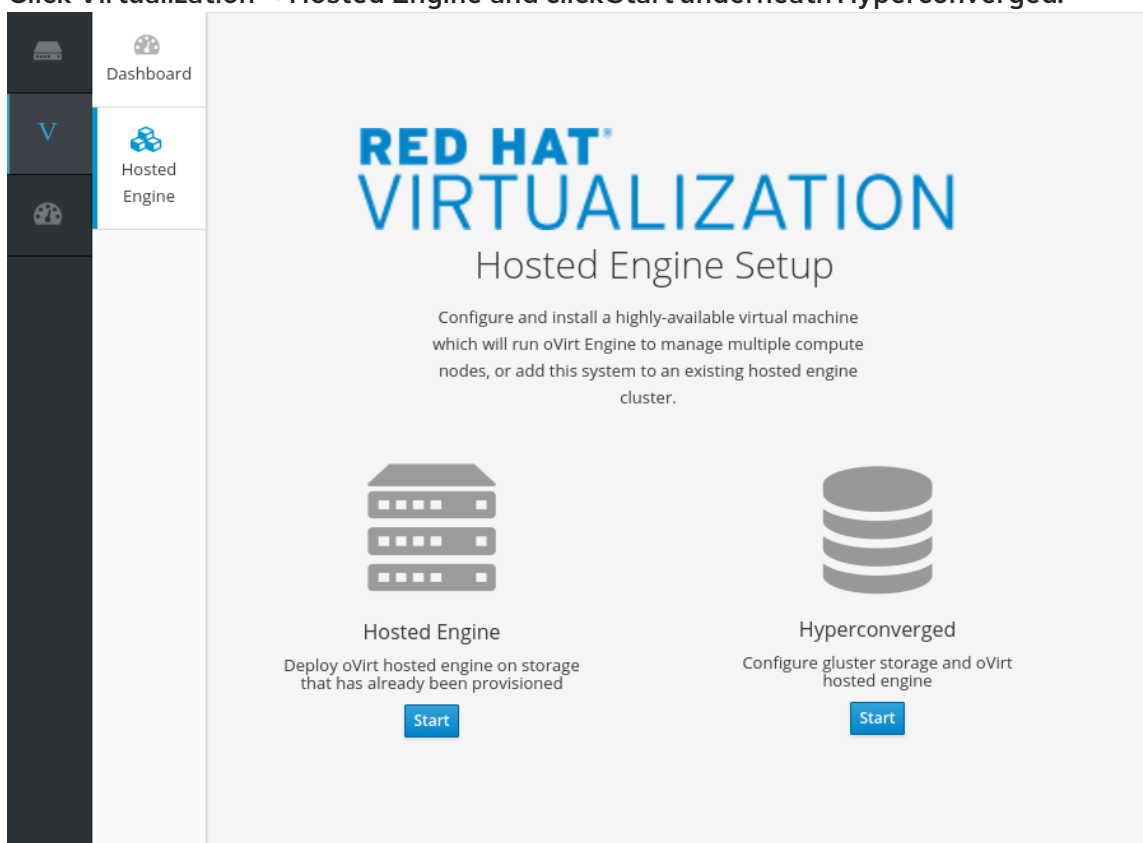
Ensure that disks specified as part of this deployment process do not have any partitions or labels.

1. Log into the Web Console

Browse to the Web Console management interface of the first hyperconverged host, for example, <https://node1.example.com:9090/>, and log in with the credentials you created in the previous section.

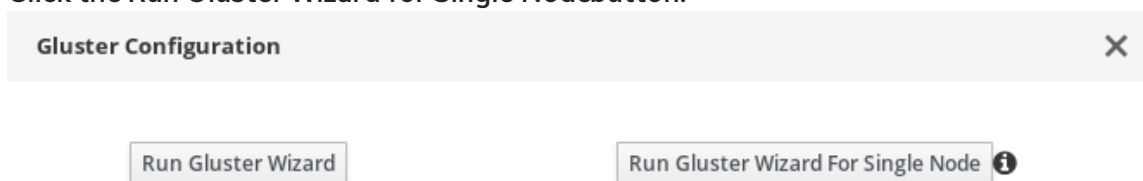
2. Start the deployment wizard

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



The *Gluster Configuration* window opens.

- b. Click the **Run Gluster Wizard for Single Node** button.



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

3. Specify host

If your hosts use IPv6 networking, check the **Select if hosts are using IPv6** checkbox. Your host must use FQDNs if you select this option; IPv6 addresses are not supported.

4. Specify volumes

Specify the volumes to create.

Name	Volume Type	Arbiter	Brick Dirs
engine	Distribute	<input type="checkbox"/>	/gluster_bricks/engine/engine
data	Distribute	<input type="checkbox"/>	/gluster_bricks/data/data
vmstore	Distribute	<input type="checkbox"/>	/gluster_bricks/vmstore/vmstc

Name

Specify the name of the volume to be created.

Volume Type

Only distributed volumes are supported for single node deployments.

Brick Dirs

The directory that contains this volume’s bricks. Use a brick path of the format **gluster_bricks/<volname>/<volname>**.

The default values are correct for most installations.

If you need more volumes, click **Add Volumes** to add another row and enter your extra volume details.

5. Specify bricks

Enter details of the bricks to be created.

Gluster Deployment
✕

Hosts
Volumes
Bricks
Review

1
2
3
4

Raid Information ⓘ

Raid Type

Stripe Size(KB)

Data Disk Count

Multipath Configuration ⓘ

Blacklist Gluster Devices

Brick Configuration

Select Host

LV Name	Device Name	LV Size(GB)	Enable Dedupe & Compression
engine	/dev/sdb	100	<input type="checkbox"/>
data	/dev/sdc	500	<input type="checkbox"/>
vmstore	/dev/sdd	500	<input type="checkbox"/>

Configure LV Cache

SSD

Thinpool device

LV Size(GB)

Cache Mode ⓘ

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

Cancel
< Back
Next >

RAID Type

Specify the RAID configuration of the 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. This can be ignored for jbod configurations.

Data Disk Count

Specify the number of data disks in your host's RAID volume. This can be ignored for jbod configurations.

Blacklist Gluster Devices

Prevents the disk that is specified as a Gluster brick from using a multipath device name. If you want to use a multipath device name, uncheck this checkbox and use the `/dev/mapper/<WWID>` format to specify your device in the `Device` field.

Select Host

This option is not valid for single node deployments.

LV Name

The name of the logical volume to be created. This is pre-filled with the name that you specified on the previous page of the wizard.

Device Name

Specify the raw device you want to use in the format `/dev/sdc`. Use `/dev/mapper/<WWID>` format for multipath devices. Use `/dev/mapper/luks_<name>` format for devices using Network-Bound Disk Encryption.

LV 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.

Enable Dedupe & Compression

Specify whether to provision the volume using VDO for compression and deduplication at deployment time. The logical size of the brick is expanded to 10 times the size of physical volume as part of VDO space savings.



NOTE

Ensure to enable **Dedupe & Compression** on all the bricks which are part of the volume.

Configure LV Cache

Optionally, check this checkbox to configure a small, fast SSD device as a logical volume cache for a larger, slower logical volume.

- Add the device path to the SSD field.
- Specify the Thinpool device to attach the cache device to.
- Add the size to the LV Size (GB) field.
- Set the Cache Mode used by the device.



WARNING

To avoid data loss when using write-back mode, Red Hat recommends using two separate SSD/NVMe devices. Configuring the two devices in a RAID-1 configuration (via software or hardware), significantly reduces the potential of data loss from lost writes.

For further information about lvmcache configuration, see [LVM cache logical volumes](#) in the Red Hat Enterprise Linux 8 documentation.

6. Review and edit configuration

Gluster Deployment
✕

Hosts Volumes Bricks Review

① ————— ② ————— ③ ————— ④

Generated Ansible inventory : /etc/ansible/hc_wizard_inventory.yml
✎ Edit ↻ Reload

```

hc_nodes:
hosts:
storage.example.com:
gluster_infra_volume_groups:
- vgname: gluster_vg_vdb
  pvname: /dev/vdb
- vgname: gluster_vg_vdc
  pvname: /dev/vdc
- vgname: gluster_vg_vdd
  pvname: /dev/vdd
gluster_infra_mount_devices:
- path: /gluster_bricks/engine
  lvname: gluster_lv_engine

```

Enable Debug Logging

Cancel < Back Deploy

- a. Click Edit to begin editing the generated deployment configuration file. Make any changes required and click Save.
 - b. Review the configuration file. If all configuration details are correct, click Deploy.
7. Wait for deployment to complete
You can watch the progress of the deployment in the text field.

The window displays Successfully deployed gluster when complete.

Gluster Deployment
✕


Hosts
Volumes
Bricks
Review

1

2

3

4



Successfully deployed Gluster

Continue to Hosted Engine Deployment

Cancel

< Back

Close

Click **Continue to Hosted Engine Deployment** and continue the deployment process with the instructions in [Section 8.2, "Deploy the Hosted Engine on a single node using the Web Console"](#).



IMPORTANT

If deployment fails, click **Clean up** to remove any potentially incorrect changes to the system.

When cleanup is complete, click **Redeploy**. 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.

8.2. DEPLOY THE HOSTED ENGINE ON A SINGLE NODE USING THE WEB CONSOLE

This section shows you how to deploy the Hosted Engine on a single node using the Web Console. Following this process results in Red Hat Virtualization Manager running in a virtual machine on your node, and managing that virtual machine. It also configures a Default cluster consisting only of that node, and enables Red Hat Gluster Storage functionality and the virtual-host *tuned* performance profile for the cluster of one.

Prerequisites

- The RHV-M Appliance is installed during the deployment process; however, if required, you can install it on the deployment host before starting the installation:

```
# yum install rhvm-appliance
```

Manually installing the Manager virtual machine is not supported.

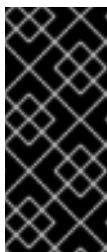
- [Configure Red Hat Gluster Storage on a single node](#)
- Gather the information you need for Hosted Engine deployment
Have the following information ready before you start the deployment process.
 - IP address for a pingable gateway to the hyperconverged host
 - IP address of the front-end management 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. Open the Hosted Engine Deployment wizard
If you continued directly from the end of [Configure Red Hat Gluster Storage on a single node](#), the wizard is already open.

Otherwise:

- a. Click Virtualization → Hosted Engine.
- b. Click Start underneath Hyperconverged.
- c. Click Use existing configuration.



IMPORTANT

If the previous deployment attempt failed, click Clean up instead of Use existing configuration to discard the previous attempt and start from scratch. If your deployment uses Network-Bound Disk Encryption, you must then follow the process in [Cleaning up Network-Bound Disk Encryption after a failed deployment](#).

2. Specify virtual machine details

Hosted Engine Deployment
✕

VM Engine Prepare VM Storage Finish

1 2 3 4 5

VM Settings

Engine VM FQDN

MAC Address

Network Configuration

Bridge Interface

Root Password

Root SSH Access

Number of Virtual CPUs

Memory Size (MiB) 62,047MB available

> Advanced

a. Enter the following details:

Engine VM FQDN

The fully qualified domain name to be used for the Hosted Engine virtual machine, for example, **engine.example.com**.

MAC Address

The MAC address associated with the Engine VM FQDN.



IMPORTANT

The pre-populated MAC address must be replaced.

Network Configuration

Choose either DHCP or Static from the Network Configuration drop-down list.

- If you choose DHCP, you must have a DHCP reservation for the Hosted Engine virtual machine so that its host name resolves to the address received from DHCP. Specify its MAC address in the MAC Address field.
- If you choose Static, enter the following details:
 - VM IP Address - The IP address must belong to the same subnet as the host. For example, if the host is in 10.1.1.0/24, the Hosted Engine virtual machine's IP must be in the same subnet range (10.1.1.1-254/24).

- Gateway Address
- DNS Servers

Bridge Interface

Select the Bridge Interface from the drop-down list.

Root password

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

Root SSH Access

Specify whether to allow Root SSH Access. The default value of Root SSH Access is set to Yes.

Number of Virtual CPUs

Enter the Number of Virtual CPUs for the virtual machine.

Memory Size (MiB)

Enter the Memory Size (MiB). The available memory is displayed next to the input field.



NOTE

Red Hat recommends to retain the values of Root SSH Access, Number of Virtual CPUs and Memory Size to default values.

- b. Optionally expand the Advanced fields.

Advanced

Root SSH Public Key

Edit Hosts File

Bridge Name

Gateway Address

Host FQDN

Apply OpenSCAP profile

Network Test

Cancel < Back Next >

Root SSH Public Key

Enter a Root SSH Public Key to use for root access to the Hosted Engine virtual machine.

Edit Hosts File

Select or clear the Edit Hosts File check box to specify whether to add entries for the Hosted Engine virtual machine and the base host to the virtual machine's /etc/hosts file. You must ensure that the host names are resolvable.

Bridge Name

Change the management Bridge Name, or accept the default ovirtmgmt.

Gateway Address

Enter the Gateway Address for the management bridge.

Host FQDN

Enter the Host FQDN of the first host to add to the Manager. This is the front-end FQDN of the base host you are running the deployment on.

Network Test

If you have a static network configuration or are using an isolated environment with addresses defined in `/etc/hosts`, set Network Test to Ping.

- c. Click Next. Your FQDNs are validated before the next screen appears.

3. Specify virtualization management details

- a. Enter the password to be used by the **admin** account in the Administration Portal. You can also specify an email address for notifications, the notifications can also be configured post deployment; see [Chapter 10, Post-deployment configuration suggestions](#).

Hosted Engine Deployment
✕

Engine Credentials

Admin Portal Password

Notification Settings

Server Name

Server Port Number

Sender E-Mail Address

Recipient E-Mail Addresses - +

Cancel < Back Next >

- b. Click Next.

4. Review virtual machine configuration

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

Hosted Engine Deployment ✕

VM Engine Prepare VM Storage Finish

① ② ③ ④ ⑤

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.
- c. Wait for virtual machine preparation to complete.

Hosted Engine Deployment ✕

VM Engine Prepare VM Storage Finish

① ——— ② ——— ③ ——— ④ ——— ⑤



Execution completed successfully. Please proceed to the next step.

If preparation does not occur successfully, see [Viewing Hosted Engine deployment errors](#).

- d. Click Next.
5. Specify storage for the Hosted Engine virtual machine
 - a. Specify the back-end address and location of the **engine** volume.

Hosted Engine Deployment
✕

VM
Engine
Prepare VM
Storage
Finish

1

2

3

4

5

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

Storage Connection

Mount Options

> Advanced

b. Click Next.

6. 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.

Hosted Engine Deployment ✕

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:** host1.example.com.:engine
- Mount Options:**
- Disk Size (GiB):** 58


- b. Click Finish Deployment.
7. Wait for deployment to complete
This can take some time, depending on your configuration details.

The window displays the following when complete.

Hosted Engine Deployment ✕

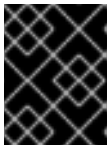
VM Engine Prepare VM Storage Finish

① ——— ② ——— ③ ——— ④ ——— ⑤



Hosted engine deployment complete!

Close



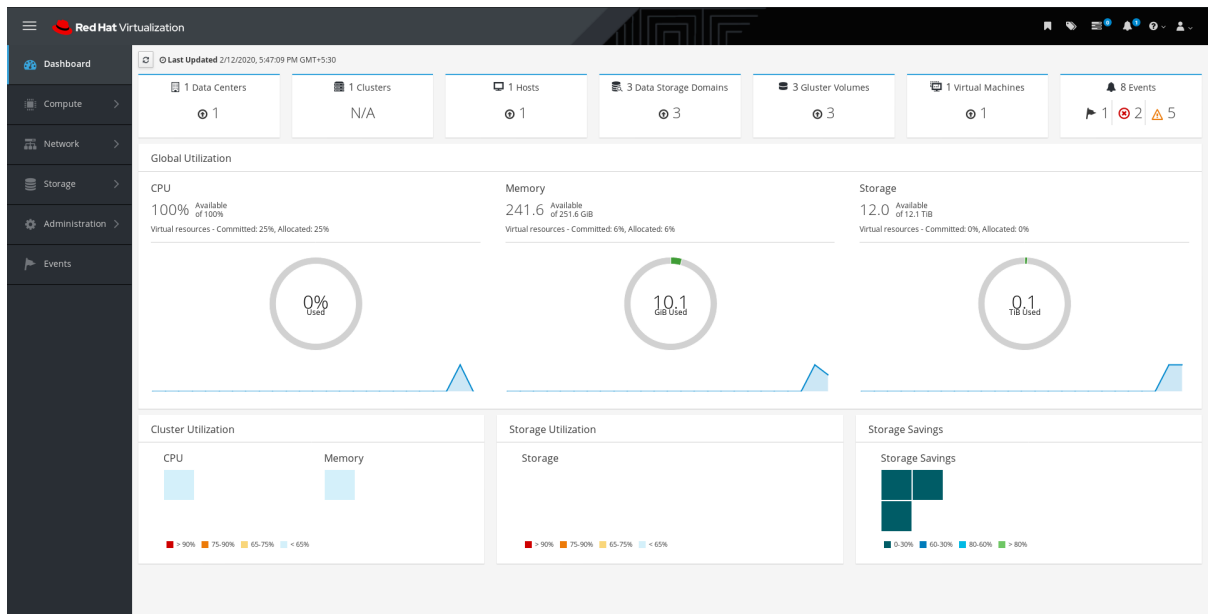
IMPORTANT

If deployment does not complete successfully, see [Viewing Hosted Engine deployment errors](#).

Click Close.

8. Verify hosted engine deployment

Browse to the Administration Portal (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

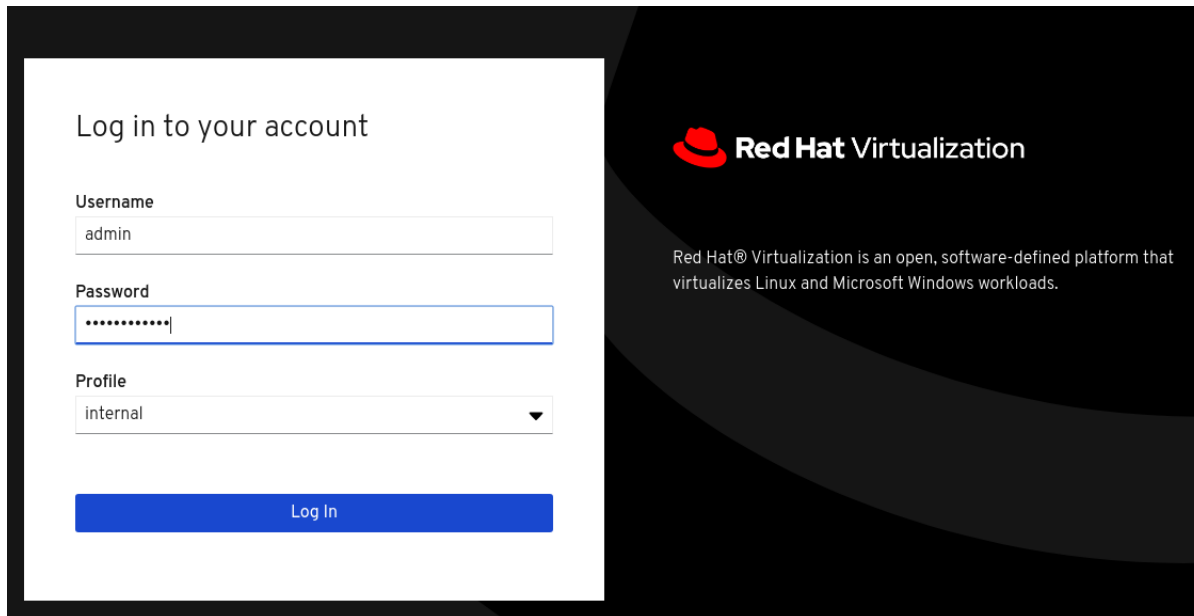
- [Log in to the Administration Portal to complete configuration](#)

CHAPTER 9. VERIFY YOUR DEPLOYMENT

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

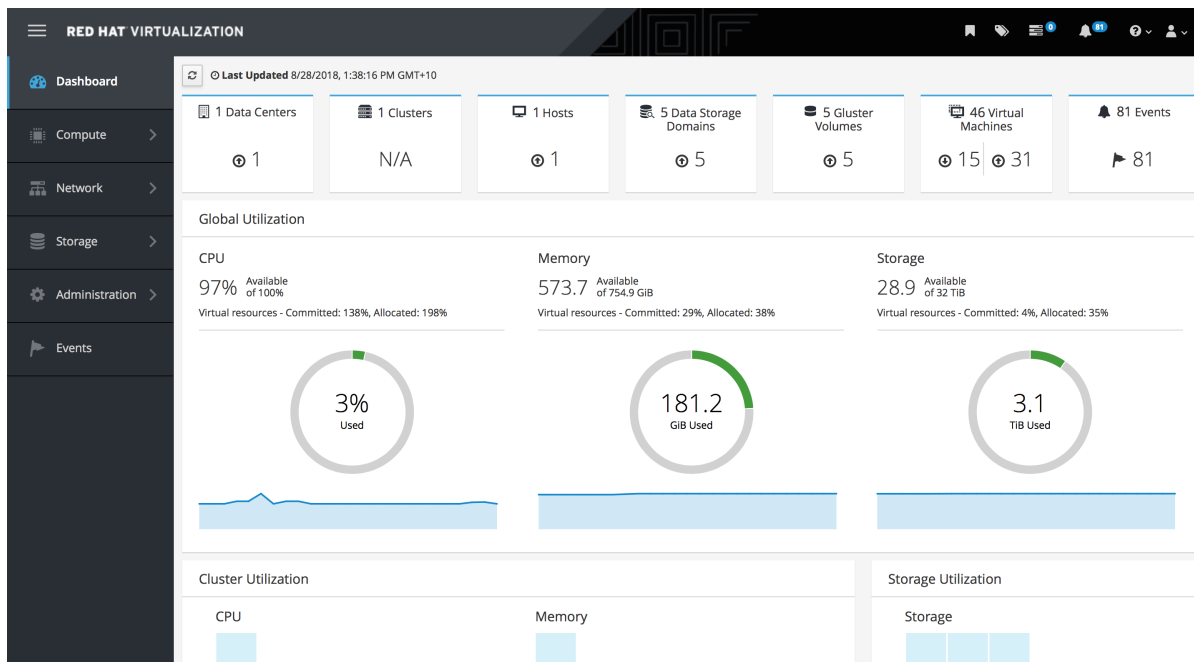
1. Browse to the Administration Portal, for example, <http://engine.example.com/ovirt-engine>.

Administration Console Login



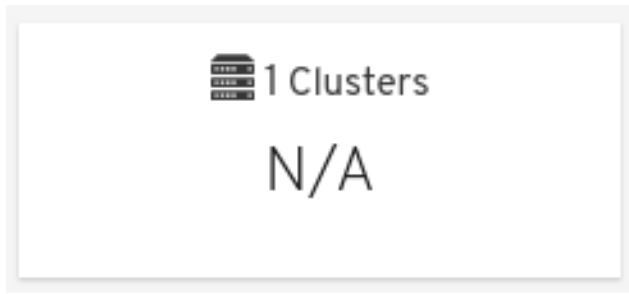
2. Log in using the administrative credentials added during hosted engine deployment. When login is successful, the Dashboard appears.

Administration Console Dashboard

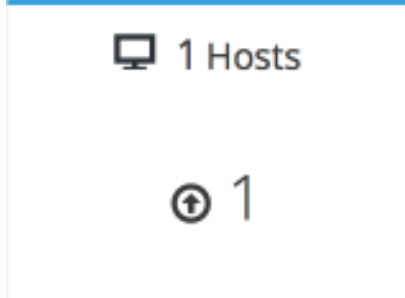


3. Verify that your cluster is available.

Administration Console Dashboard - Clusters



4. Verify that one host is available.



- a. Click Compute → Hosts.
- b. Verify that your host is listed with a Status of **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

Storage > Storage Domains

Storage:

Status	Domain Name	Comment	Domain Type	Storage Type	Format	Cross Data Center Status	Total Space	Free Space	Guaranteed Free Space	Description
▲	data		Data	GlusterFS	VS	Active	122878 GiB	120792 GiB	120792 GiB	
▲	hosted_storage		Data (Master)	GlusterFS	VS	Active	99 GiB	92 GiB	92 GiB	
▲	testvol		Data	GlusterFS	VS	Active	1023 GiB	1006 GiB	1006 GiB	

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 NOTIFICATIONS

See [Configuring Event Notifications in the Administration Portal](#) to configure email notifications.

10.2. (OPTIONAL) CONFIGURE HOST POWER MANAGEMENT

The Red Hat Virtualization Manager 4.4 is capable of rebooting hosts that have entered a non-operational or non-responsive state, as well as preparing to power off under-utilized hosts to save power. This functionality depends on a properly configured power management device.

See [Configuring Host Power Management Settings](#) 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.

CHAPTER 11. NEXT STEPS

11.1. ENABLING THE RED HAT VIRTUALIZATION MANAGER REPOSITORIES

Register the system with Red Hat Subscription Manager, attach the **Red Hat Virtualization Manager** subscription, and enable Manager repositories.

Procedure

1. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```



NOTE

If you are using an IPv6 network, use an IPv6 transition mechanism to access the Content Delivery Network and subscription manager.

2. Find the Red Hat Virtualization Manager subscription pool and record the pool ID:

```
# subscription-manager list --available
```

3. Use the pool ID to attach the subscription to the system:

```
# subscription-manager attach --pool=pool_id
```



NOTE

To view currently attached subscriptions:

```
# subscription-manager list --consumed
```

To list all enabled repositories:

```
# yum repolist
```

4. Configure the repositories:

```
# subscription-manager repos \
  --disable='*' \
  --enable=rhel-8-for-x86_64-baseos-rpms \
  --enable=rhel-8-for-x86_64-appstream-rpms \
  --enable=rhv-4.4-manager-for-rhel-8-x86_64-rpms \
  --enable=fast-datapath-for-rhel-8-x86_64-rpms \
  --enable=jb-eap-7.4-for-rhel-8-x86_64-rpms \
  --enable=openstack-16.2-cinderlib-for-rhel-8-x86_64-rpms \
  --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

5. Enable the pki-deps module.

```
# yum module -y enable pki-deps
```

6. Enable version 12 of the postgresql module.

```
# yum module -y enable postgresql:12
```

7. Synchronize installed packages to update them to the latest available versions.

```
# yum distro-sync
```

Additional resources

For information on modules and module streams, see the following sections in *Installing, managing, and removing user-space components*

- [Module streams](#)
- [Selecting a stream before installation of packages](#)
- [Resetting module streams](#)
- [Switching to a later stream](#)
- Learn to create and manage Red Hat Gluster Storage using the Administration Portal in [Managing Red Hat Gluster Storage using the RHV Administration Portal](#)
- Learn to create and manage virtual machines in the Red Hat Virtualization [Virtual Machine Management Guide](#).
- Review the [RHHI for Virtualization documentation](#) on the Red Hat Customer Portal.

PART I. 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 Web Console based deployment process (documented in [Section 8.1, “Configuring Red Hat Gluster Storage on a single node”](#)) is stored in the `/var/log/cockpit/ovirt-dashboard/gluster-deployment.log` file by default.

The log files for the Hosted Engine setup portion of the deployment process (documented in [Section 8.2, “Deploy the Hosted Engine on a single node using the Web Console”](#)) 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. DEPLOYMENT ERRORS

13.1. ORDER OF CLEANUP OPERATIONS

Depending on where deployment fails, you may need to perform a number of cleanup operations.

Always perform cleanup for tasks in reverse order to the order of the tasks themselves. For example, during deployment, we perform the following tasks in order:

1. Configure Network-Bound Disk Encryption using Ansible.
2. Configure Red Hat Gluster Storage using the Web Console.
3. Configure the Hosted Engine using the Web Console.

If deployment fails at step 2, perform cleanup for step 2. Then, if necessary, perform cleanup for step 1.

13.2. FAILED TO DEPLOY STORAGE

If an error occurs during [storage deployment](#), the deployment process halts and **Deployment failed** is displayed.

Deploying storage failed

Gluster Deployment
✕

Hosts
Additional Hosts
Volumes
Bricks
Review

1

2

3

4

5

⊗ Deployment failed

CleanUp

↺ Redeploy

```

fatal: [host2.example.com]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: Could not
resolve hostname host2.example.com: Name or service not known", "unreachable": true}

NO MORE HOSTS LEFT *****
NO MORE HOSTS LEFT *****

PLAY RECAP *****
host1.example.com      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
host2.example.com      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
host3.example.com      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0

Please check /var/log/cockpit/ovirt-dashboard/gluster-deployment.log for more informations.
                    
```

Cancel

< Back

Close

- Review the Web Console output for error information.

- Click Clean up to remove any potentially incorrect changes to the system. If your deployment uses Network-Bound Disk Encryption, you must then follow the process in [Cleaning up Network-Bound Disk Encryption after a failed deployment](#)
- Click Redeploy and correct any entered values that may have caused errors. If you need help resolving errors, contact Red Hat Support with details.
- Return to [storage deployment](#) to try again.

13.2.1. Cleaning up Network-Bound Disk Encryption after a failed deployment

If you are using Network-Bound Disk Encryption and deployment fails, you cannot just click the Cleanup button in order to try again. You must also run the `luks_device_cleanup.yml` playbook to complete the cleaning process before you start again.

Run this playbook as shown, providing the same `luks_tang_inventory.yml` file that you provided during setup.

```
# ansible-playbook -i luks_tang_inventory.yml /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/tasks/luks_device_cleanup.yml --ask-vault-pass
```

13.2.2. Error: VDO signature detected on device

During storage deployment, the **Create VDO with specified size** task may fail with the **VDO signature detected on device** error.

```
TASK [gluster.infra/roles/backend_setup : Create VDO with specified size]
task path: /etc/ansible/roles/gluster.infra/roles/backend_setup/tasks/vdo_create.yml:9
failed: [host1.example.com] (item={u'writepolicy': u'auto', u'name': u'vdo_sdb', u'readcachesize': u'20M', u'readcache': u'enabled', u'emulate512': u'off', u'logicalsize': u'11000G', u'device': u'/dev/sdb', u'slabsize': u'32G', u'blockmapcachesize': u'128M'}) => {"ansible_loop_var": "item", "changed": false, "err": "vdo: ERROR - vdo signature detected on /dev/sdb at offset 0; use --force to override\n", "item": {"blockmapcachesize": "128M", "device": "/dev/sdb", "emulate512": "off", "logicalsize": "11000G", "name": "vdo_sdb", "readcache": "enabled", "readcachesize": "20M", "slabsize": "32G", "writepolicy": "auto"}, "msg": "Creating VDO vdo_sdb failed.", "rc": 5}
```

This error occurs when the specified device is already a VDO device, or when the device was previously configured as a VDO device and was not cleaned up correctly.

- If you specified a VDO device accidentally, return to storage configuration and specify a different non-VDO device.
- If you specified a device that has been used as a VDO device previously:
 - a. Check the device type.

```
# blkid -p /dev/sdb
/dev/sdb: UUID="fee52367-c2ca-4fab-a6e9-58267895fe3f" TYPE="vdo" USAGE="other"
```

If you see `TYPE="vdo"` in the output, this device was not cleaned correctly.

- b. Follow the steps in [Manually cleaning up a VDO device](#) to use this device. Then return to [storage deployment](#) to try again.

Avoid this error by specifying clean devices, and by using the Clean up button in the storage deployment window to clean up any failed deployments.

13.2.3. Manually cleaning up a VDO device

Follow this process to manually clean up a VDO device that has caused a deployment failure.



WARNING

This is a destructive process. You will lose all data on the device that you clean up.

Procedure

- Clean the device using `wipefs`.

```
# wipefs -a /dev/sdX
```

Verify

- Confirm that the device does not have `TYPE="vdo"` set any more.

```
# blkid -p /dev/sdb  
/dev/sdb: UUID="fee52367-c2ca-4fab-a6e9-58267895fe3f" TYPE="vdo" USAGE="other"
```

Next steps

- Return to [storage deployment](#) to try again.

13.3. FAILED TO PREPARE VIRTUAL MACHINE

If an error occurs while preparing the virtual machine in [Hosted Engine deployment](#), deployment pauses, and you see a screen similar to the following:

Preparing virtual machine failed

Hosted Engine Deployment
✕

VM
Engine
Prepare VM
Storage
Finish

1

2

3

4

5

✕
Deployment 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"}

```

Cancel
< Back
Prepare VM

- Review the Web Console output for error information.
- Click Back and correct any entered values that may have caused errors. Ensure proper values for network configurations are provided in VM tab. If you need help resolving errors, contact Red Hat Support with details.
- Ensure that the **rhvm-appliance** package is available on the first hyperconverged host.

```
# yum install rhvm-appliance
```

- Return to [Hosted Engine deployment](#) to try again.
If you closed the deployment wizard while you resolved errors, you can select Use existing configuration when you retry the deployment process.

13.4. FAILED TO DEPLOY HOSTED ENGINE

If an error occurs during hosted engine deployment, deployment pauses and **Deployment failed** is displayed.

Hosted engine deployment failed

Hosted Engine Deployment
✕

VM Engine Prepare VM Storage Finish

① ② ③ ④ ⑤

✕
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."}

```

Cancel
< Back
Redeploy

1. Review the Web Console output for error information.
2. Remove the contents of the engine volume.
 - a. Mount the engine volume.

```
# mount -t glusterfs <server1>:/engine /mnt/test
```
 - b. Remove the contents of the volume.

```
# rm -rf /mnt/test/*
```
 - c. Unmount the engine volume.

```
# umount /mnt/test
```
3. Click Redeploy and correct any entered values that may have caused errors.
4. If the deployment fails after performing the above steps a, b and c. Perform these steps again and this time clean the Hosted Engine:

```
# ovirt-hosted-engine-cleanup
```

5. Return to [Hosted Engine deployment](#) to try again.

If you closed the deployment wizard while you resolved errors, you can select Use existing configuration when you retry the deployment process.

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

PART II. REFERENCE MATERIAL

APPENDIX A. WORKING WITH FILES ENCRYPTED USING ANSIBLE VAULT

Red Hat recommends encrypting the contents of deployment and management files that contain passwords and other sensitive information. Ansible Vault is one method of encrypting these files. More information about Ansible Vault is available in the [Ansible documentation](#).

A.1. ENCRYPTING FILES

You can create an encrypted file by using the **ansible-vault create** command, or encrypt an existing file by using the **ansible-vault encrypt** command.

When you create an encrypted file or encrypt an existing file, you are prompted to provide a password. This password is used to decrypt the file after encryption. You must provide this password whenever you work directly with information in this file or run a playbook that relies on the file's contents.

Creating an encrypted file

```
$ ansible-vault create variables.yml
New Vault password:
Confirm New Vault password:
```

The **ansible-vault create** command prompts for a password for the new file, then opens the new file in the default text editor (defined as **\$EDITOR** in your shell environment) so that you can populate the file before saving it.

If you have already created a file and you want to encrypt it, use the **ansible-vault encrypt** command.

Encrypting an existing file

```
$ ansible-vault encrypt existing-variables.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

A.2. EDITING ENCRYPTED FILES

You can edit an encrypted file using the **ansible-vault edit** command and providing the Vault password for that file.

Editing an encrypted file

```
$ ansible-vault edit variables.yml
New Vault password:
Confirm New Vault password:
```

The **ansible-vault edit** command prompts for a password for the file, then opens the file in the default text editor (defined as **\$EDITOR** in your shell environment) so that you can edit and save the file contents.

A.3. REKEYING ENCRYPTED FILES TO A NEW PASSWORD

You can change the password used to decrypt a file by using the **ansible-vault rekey** command.

```
$ ansible-vault rekey variables.yml  
Vault password:  
New Vault password:  
Confirm New Vault password:  
Rekey successful
```

The **ansible-vault rekey** command prompts for the current Vault password, and then prompts you to set and confirm a new Vault password.

APPENDIX B. UNDERSTANDING THE LUKS_TANG_INVENTORY.YML FILE

B.1. CONFIGURATION PARAMETERS FOR DISK ENCRYPTION

hc_nodes (required)

A list of hyperconverged hosts that uses the back-end FQDN of the host, and the configuration details of those hosts. Configuration that is specific to a host is defined under that host's back-end FQDN. Configuration that is common to all hosts is defined in the vars: section.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      [configuration specific to this host]
    host2backend.example.com:
    host3backend.example.com:
    host4backend.example.com:
    host5backend.example.com:
    host6backend.example.com:
  vars:
    [configuration common to all hosts]
```

blacklist_mpath_devices (optional)

By default, Red Hat Virtualization Host enables multipath configuration, which provides unique multipath names and worldwide identifiers for all disks, even when disks do not have underlying multipath configuration. Include this section if you do not have multipath configuration so that the multipath device names are not used for listed devices. Disks that are not listed here are assumed to have multipath configuration available, and require the path format `/dev/mapper/<WWID>` instead of `/dev/sdx` when defined in subsequent sections of the inventory file.

On a server with four devices (`sda`, `sdb`, `sdc` and `sdd`), the following configuration blacklists only two devices. The path format `/dev/mapper/<WWID>` is expected for devices not in this list.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      blacklist_mpath_devices:
        - sdb
        - sdc
```

gluster_infra_luks_devices (required)

A list of devices to encrypt and the encryption passphrase to use for each device.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      gluster_infra_luks_devices:
        - devicename: /dev/sdb
          passphrase: Str0ngPa55#
```

devicename

The name of the device in the format `/dev/sdx`.

passphrase

The password to use for this device when configuring encryption. After disk encryption with Network-Bound Disk Encryption (NBDE) is configured, a new random key is generated, providing greater security.

rootpassphrase (required)

The password that you used when you selected Encrypt my data during operating system installation on this host.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      rootpassphrase: h1-Str0ngPa55#
```

rootdevice (required)

The root device that was encrypted when you selected Encrypt my data during operating system installation on this host.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      rootdevice: /dev/sda2
```

networkinterface (required)

The network interface this host uses to reach the NBDE key server.

```
hc_nodes:
  hosts:
    host1backend.example.com:
      networkinterface: ens3s0f0
```

ip_version (required)

Whether to use IPv4 or IPv6 networking. Valid values are **IPv4** and **IPv6**. There is no default value. Mixed networks are not supported.

```
hc_nodes:
  vars:
    ip_version: IPv4
```

ip_config_method (required)

Whether to use DHCP or static networking. Valid values are **dhcp** and **static**. There is no default value.

```
hc_nodes:
  vars:
    ip_config_method: dhcp
```

The other valid value for this option is **static**, which requires the following additional parameters and is defined individually for each host:

```
hc_nodes:
```

```

hosts:
  host1backend.example.com:
    ip_config_method: static
    host_ip_addr: 192.168.1.101
    host_ip_prefix: 24
    host_net_gateway: 192.168.1.100
  host2backend.example.com:
    ip_config_method: static
    host_ip_addr: 192.168.1.102
    host_ip_prefix: 24
    host_net_gateway: 192.168.1.100
  host3backend.example.com:
    ip_config_method: static
    host_ip_addr: 192.168.1.102
    host_ip_prefix: 24
    host_net_gateway: 192.168.1.100

```

gluster_infra_tangservers

The address of your NBDE key server or servers, including **http://**. If your servers use a port other than the default (80), specify a port by appending **:_port_** to the end of the URL.

```

hc_nodes:
  vars:
    gluster_infra_tangservers:
      - url: http://key-server1.example.com
      - url: http://key-server2.example.com:80

```

B.2. EXAMPLE LUKS_TANG_INVENTORY.YML

Dynamically allocated IP addresses

```

hc_nodes:
  hosts:
    host1-backend.example.com:
      blacklist_mpath_devices:
        - sda
        - sdb
        - sdc
      gluster_infra_luks_devices:
        - devicename: /dev/sdb
          passphrase: dev-sdb-encrypt-passphrase
        - devicename: /dev/sdc
          passphrase: dev-sdc-encrypt-passphrase
      rootpassphrase: host1-root-passphrase
      rootdevice: /dev/sda2
      networkinterface: eth0
    host2-backend.example.com:
      blacklist_mpath_devices:
        - sda
        - sdb
        - sdc
      gluster_infra_luks_devices:
        - devicename: /dev/sdb

```

```

    passphrase: dev-sdb-encrypt-passphrase
  - devicename: /dev/sdc
    passphrase: dev-sdc-encrypt-passphrase
  rootpassphrase: host2-root-passphrase
  rootdevice: /dev/sda2
  networkinterface: eth0
host3-backend.example.com:
  blacklist_mpath_devices:
  - sda
  - sdb
  - sdc
  gluster_infra_luks_devices:
  - devicename: /dev/sdb
    passphrase: dev-sdb-encrypt-passphrase
  - devicename: /dev/sdc
    passphrase: dev-sdc-encrypt-passphrase
  rootpassphrase: host3-root-passphrase
  rootdevice: /dev/sda2
  networkinterface: eth0
vars:
  ip_version: IPv4
  ip_config_method: dhcp
  gluster_infra_tangservers:
  - url: http://key-server1.example.com:80
  - url: http://key-server2.example.com:80

```

Static IP addresses

```

hc_nodes:
  hosts:
    host1-backend.example.com:
    blacklist_mpath_devices:
    - sda
    - sdb
    - sdc
    gluster_infra_luks_devices:
    - devicename: /dev/sdb
      passphrase: dev-sdb-encrypt-passphrase
    - devicename: /dev/sdc
      passphrase: dev-sdc-encrypt-passphrase
    rootpassphrase: host1-root-passphrase
    rootdevice: /dev/sda2
    networkinterface: eth0
    host_ip_addr: host1-static-ip
    host_ip_prefix: network-prefix
    host_net_gateway: default-network-gateway
    host2-backend.example.com:
    blacklist_mpath_devices:
    - sda
    - sdb
    - sdc
    gluster_infra_luks_devices:
    - devicename: /dev/sdb
      passphrase: dev-sdb-encrypt-passphrase
    - devicename: /dev/sdc
      passphrase: dev-sdc-encrypt-passphrase

```

```
rootpassphrase: host2-root-passphrase
rootdevice: /dev/sda2
networkinterface: eth0
host_ip_addr: host1-static-ip
host_ip_prefix: network-prefix
host_net_gateway: default-network-gateway
host3-backend.example.com:
  blacklist_mpath_devices:
    - sda
    - sdb
    - sdc
  gluster_infra_luks_devices:
    - devicename: /dev/sdb
      passphrase: dev-sdb-encrypt-passphrase
    - devicename: /dev/sdc
      passphrase: dev-sdc-encrypt-passphrase
  rootpassphrase: host3-root-passphrase
  rootdevice: /dev/sda2
  networkinterface: eth0
  host_ip_addr: host1-static-ip
  host_ip_prefix: network-prefix
  host_net_gateway: default-network-gateway
vars:
  ip_version: IPv4
  ip_config_method: static
  gluster_infra_tangservers:
    - url: http://key-server1.example.com:80
    - url: http://key-server2.example.com:80
```

APPENDIX C. GLOSSARY OF TERMS

C.1. VIRTUALIZATION TERMS

Administration Portal

A web user interface provided by Red Hat Virtualization Manager, based on the oVirt engine web user interface. It allows administrators to manage and monitor cluster resources like networks, storage domains, and virtual machine templates.

Hosted Engine

The instance of Red Hat Virtualization Manager that manages RHHI for Virtualization.

Hosted Engine virtual machine

The virtual machine that acts as Red Hat Virtualization Manager. The Hosted Engine virtual machine runs on a virtualization host that is managed by the instance of Red Hat Virtualization Manager that is running on the Hosted Engine virtual machine.

Manager node

A virtualization host that runs Red Hat Virtualization Manager directly, rather than running it in a Hosted Engine virtual machine.

Red Hat Enterprise Linux host

A physical machine installed with Red Hat Enterprise Linux plus additional packages to provide the same capabilities as a Red Hat Virtualization host. This type of host is not supported for use with RHHI for Virtualization.

Red Hat Virtualization

An operating system and management interface for virtualizing resources, processes, and applications for Linux and Microsoft Windows workloads.

Red Hat Virtualization host

A physical machine installed with Red Hat Virtualization that provides the physical resources to support the virtualization of resources, processes, and applications for Linux and Microsoft Windows workloads. This is the only type of host supported with RHHI for Virtualization.

Red Hat Virtualization Manager

A server that runs the management and monitoring capabilities of Red Hat Virtualization.

Self-Hosted Engine node

A virtualization host that contains the Hosted Engine virtual machine. All hosts in a RHHI for Virtualization deployment are capable of becoming Self-Hosted Engine nodes, but there is only one Self-Hosted Engine node at a time.

storage domain

A named collection of images, templates, snapshots, and metadata. A storage domain can be comprised of block devices or file systems. Storage domains are attached to data centers in order to provide access to the collection of images, templates, and so on to hosts in the data center.

virtualization host

A physical machine with the ability to virtualize physical resources, processes, and applications for client access.

VM Portal

A web user interface provided by Red Hat Virtualization Manager. It allows users to manage and monitor virtual machines.

C.2. STORAGE TERMS

brick

An exported directory on a server in a trusted storage pool.

cache logical volume

A small, fast logical volume used to improve the performance of a large, slow logical volume.

geo-replication

One way asynchronous replication of data from a source Gluster volume to a target volume. Geo-replication works across local and wide area networks as well as the Internet. The target volume can be a Gluster volume in a different trusted storage pool, or another type of storage.

gluster volume

A logical group of bricks that can be configured to distribute, replicate, or disperse data according to workload requirements.

logical volume management (LVM)

A method of combining physical disks into larger virtual partitions. Physical volumes are placed in volume groups to form a pool of storage that can be divided into logical volumes as needed.

Red Hat Gluster Storage

An operating system based on Red Hat Enterprise Linux with additional packages that provide support for distributed, software-defined storage.

source volume

The Gluster volume that data is being copied from during geo-replication.

storage host

A physical machine that provides storage for client access.

target volume

The Gluster volume or other storage volume that data is being copied to during geo-replication.

thin provisioning

Provisioning storage such that only the space that is required is allocated at creation time, with further space being allocated dynamically according to need over time.

thick provisioning

Provisioning storage such that all space is allocated at creation time, regardless of whether that space is required immediately.

trusted storage pool

A group of Red Hat Gluster Storage servers that recognise each other as trusted peers.

C.3. HYPERCONVERGED INFRASTRUCTURE TERMS

Red Hat Hyperconverged Infrastructure (RHHI) for Virtualization

RHHI for Virtualization is a single product that provides both virtual compute and virtual storage resources. Red Hat Virtualization and Red Hat Gluster Storage are installed in a converged configuration, where the services of both products are available on each physical machine in a cluster.

hyperconverged host

A physical machine that provides physical storage, which is virtualized and consumed by virtualized processes and applications run on the same host. All hosts installed with RHHI for Virtualization are hyperconverged hosts.

Web Console

The web user interface for deploying, managing, and monitoring RHHI for Virtualization. The Web Console is provided by the Web Console service and plugins for Red Hat Virtualization Manager.