Red Hat Hyperconverged Infrastructure 1.1

Maintaining Red Hat Hyperconverged Infrastructure

Common maintenance tasks for Red Hat Hyperconverged Infrastructure

Last Updated: 2018-04-18

Red Hat Hyperconverged Infrastructure 1.1 Maintaining Red Hat Hyperconverged Infrastructure

Common maintenance tasks for Red Hat Hyperconverged Infrastructure

Laura Bailey
lbailey@redhat.com

Abstract

Red Hat Hyperconverged Infrastructure (RHHI) combines compute, storage, networking, and management capabilities into a single solution, simplifying deployment and reducing the cost of acquisition and maintenance. This document explains how to perform maintenance tasks specific to Red Hat Hyperconverged Infrastructure.

Table of Contents

PART I. CONFIGURATION TASKS

CHAPTER 1. ADD COMPUTE AND STORAGE RESOURCES

Red Hat Hyperconverged Infrastructure (RHHI) can be scaled in multiples of three nodes to a maximum of nine nodes.

1.1. SCALING RHHI DEPLOYMENTS

1.1.1. Before you begin

- Be aware that the only supported method of scaling Red Hat Hyperconverged Infrastructure (RHHI) is to create additional volumes that span the new nodes. Expanding the existing volumes to span across more nodes is not supported.

- Arbitrated replicated volumes are not supported for scaling.

- If your existing deployment uses certificates signed by a Certificate Authority for encryption, prepare the certificates that will be required for the new nodes.

1.1.2. Scaling RHHI by adding additional volumes on new nodes

1. Install the three physical machines
   Follow the instructions in *Deploying Red Hat Hyperconverged Infrastructure*:
   https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/deploying_red_hat_hyperconverged_infrastruct
   host-physical-machines.

   NOTE

   Only one arbitrated replicated volume is supported per deployment.

2. Configure key-based SSH authentication
   Follow the instructions in *Deploying Red Hat Hyperconverged Infrastructure* to configure key-based SSH authentication from one node to all nodes:
   https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/deploying_red_hat_hyperconverged_infrastruct
   configure-key-based-ssh-auth

3. Automatically configure new nodes

   a. Create an **add_nodes.conf** file based on the template provided in Section B.3, “Example gdeploy configuration file for scaling to additional nodes”.

   b. Run gdeploy using the **add_nodes.conf** file:

   ```
   # gdeploy -c add_nodes.conf
   ```

4. (Optional) If encryption is enabled

   a. 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.

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

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

c. Restart the glusterd service

```
# systemctl restart glusterd
```

d. Update the auth.ssl-allow parameter for all volumes
Use the following command on any existing node to obtain the existing settings:

```
# gluster volume get engine auth.ssl-allow
```

Set auth.ssl-allow to the old value with the new IP addresses appended.

```
# gluster volume set <vol_name> auth.ssl-allow "<old_hosts>;
<new_hosts>"
```

5. Disable multipath for each node’s storage devices

a. Add the following lines to the beginning of the **/etc/multipath.conf** file.

```
# VDSM REVISION 1.3
# VDSM PRIVATE
```

b. Add Red Hat Gluster Storage devices to the blacklist definition in the **/etc/multipath.conf** file.

```
blacklist {
    devnode "^sd[a-z]"
}
```

c. Restart multipathd

```
# systemctl restart multipathd
```

6. In Red Hat Virtualization Manager, add the new hosts to the existing cluster
For details on adding a host to a cluster, follow the instructions in *Adding a Host to the Red Hat Virtualization Manager* in the Red Hat Virtualization *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/administration_guide/sect-host_tasks.

Ensure that you perform the following configuration items:

- Select **Hosted Engine** and the **deploy** action.

- Uncheck **Automatically configure firewall**.
- Enable **Power management** settings.

7. Attach the gluster network to the new hosts

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

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

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

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

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

    f. Click **OK** to save.

    g. Verify the health of the network
    Click the **Hosts** tab and select the host.

    Click the **Network Interfaces** subtab and check the state of the host’s network

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

8. Create new bricks

    a. Click the **Hosts** tab.

    b. Select a host, and then select the **Storage Devices** subtab.

    c. Select a storage device from the list. Click **Create Brick**.

    d. In the *Create Brick* window, verify the **Raid Type** is correct and enter the following details. Note that these details must match the details of the underlying storage.

    - Brick name
    - Mount point
    - Number of physical disks in RAID volume

    e. Click **OK**.
    A new thin provisioned logical volume is created from the specified storage devices.

9. Create a new volume

    a. Click the **Volumes** tab.

    b. Click **New**. The *New volume* window opens.

    c. Specify values for the following fields:

    - Data Center
    - Volume Cluster
    - Name

d. Set **Type** to **Replicate**.

e. Click the **Add Bricks** button and select the bricks that comprise this volume.

f. Check the **Optimize for virt-store** checkbox.

g. Set the following volume options:

- Set **cluster.granular-entry-heal** to **on**.
- Set **network.remote-dio** to **off**
- Set **performance.strict-o-direct** to **on**

10. Start the new volume
In the **Volumes** tab, select the volume to start and click **Start**.

11. Create a new storage domain

a. Click the **Storage** tab and then click **New Domain**.

b. Provide a **Name** for the domain.

c. Set the **Domain function** to **Data**.

d. Set the **Storage Type** to **GlusterFS**.

e. Check the **Use managed gluster volume** option.
A list of volumes available in the cluster appears.

f. Click **OK**.

CHAPTER 2. CONFIGURE HIGH AVAILABILITY USING FENCING POLICIES

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

Several policies specific to Red Hat Gluster Storage must be enabled to ensure that fencing activities do not disrupt storage services in a Red Hat Hyperconverged (RHHI) Infrastructure deployment.

This requires enabling and configuring fencing at both the cluster level and at the host level. See the following sections for details.

2.1. CONFIGURING FENCING POLICIES IN THE CLUSTER

1. In Red Hat Virtualization Manager, click the **Clusters** tab.

2. Click **Edit**. The *Edit Cluster* window opens.

3. Click the **Fencing policy** tab.

4. Check the **Enable fencing** checkbox.

5. Check the checkboxes for at least the following fencing policies:

   - Skip fencing if gluster bricks are up

   - Skip fencing if gluster quorum not met

   See Appendix A, *Fencing Policies for Red Hat Gluster Storage* for details on the effects of these policies.

6. Click **OK** to save settings.

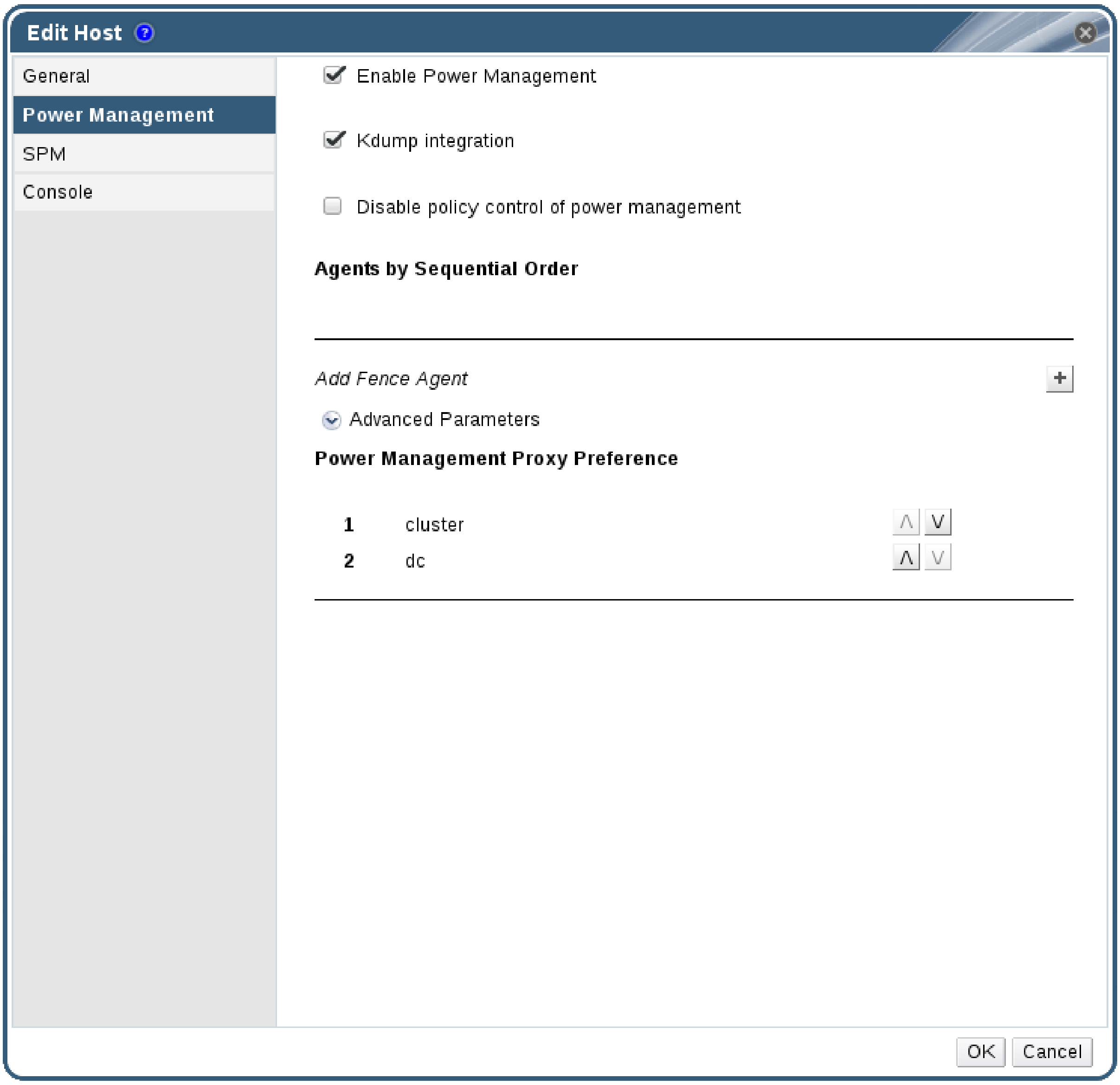
2.2. CONFIGURING FENCING PARAMETERS ON THE HOSTS

1. In Red Hat Virtualization Manager, click the **Hosts** tab.

2. Select the host to configure, and click **Edit** to open the **Edit Host** window.

3. Click the **Power Management** tab.

Figure 2.1. Power Management Settings



☑ Enable Power Management

☑ Kdump integration

□ Disable policy control of power management

1	cluster
2	dc

4. Check the **Enable Power Management** check box. This enables other fields on the tab.

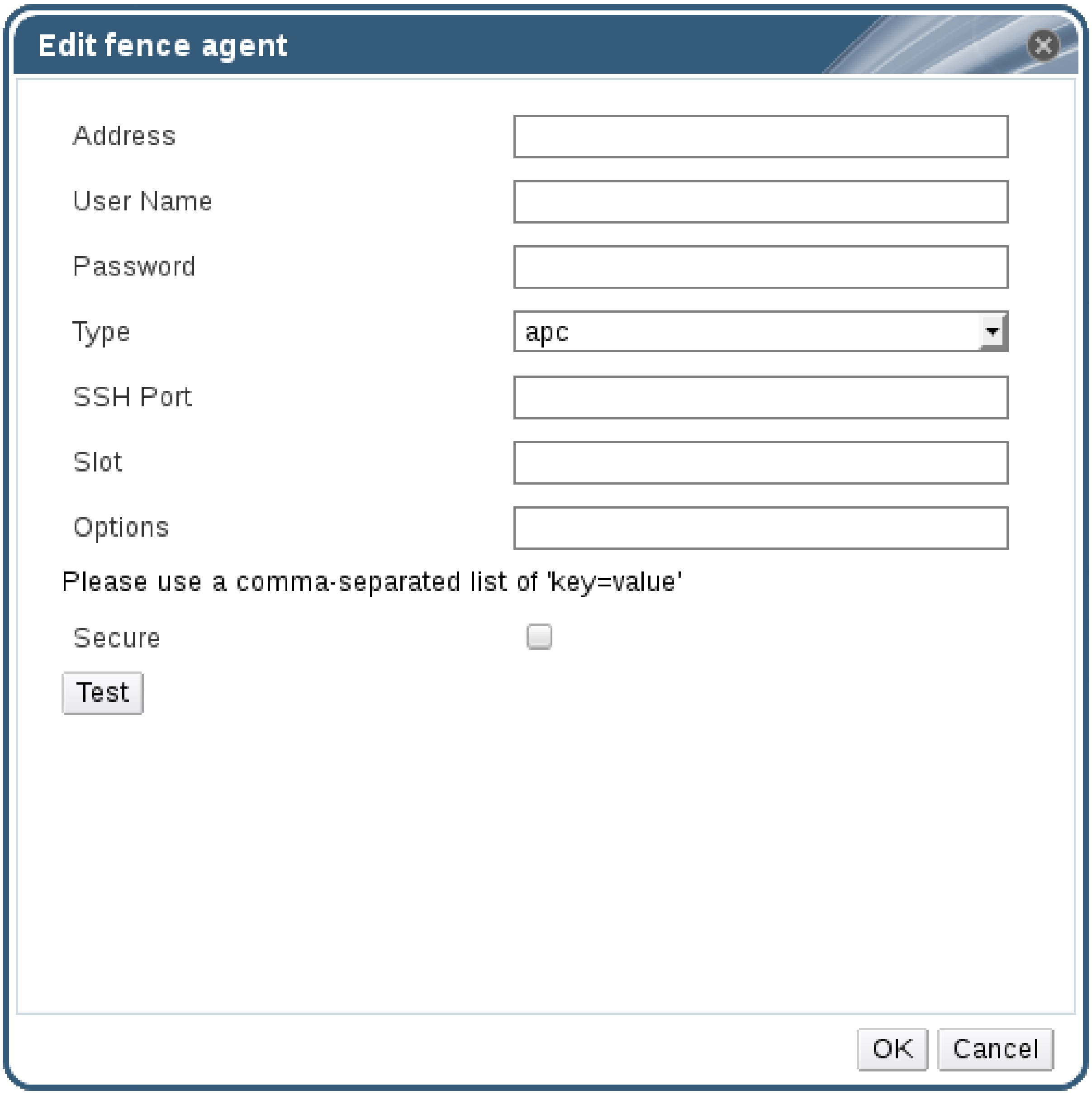
5. Check the **Kdump integration** check box to prevent the host from fencing while performing a kernel crash dump.

IMPORTANT

When you enable Kdump integration on an existing host, the host must be reinstalled for kdump to be configured. See Chapter 7, *Reinstalling a virtualization host* for instructions on reinstalling a host.

1. Click the plus (+) button to add a new power management device. The **Edit fence agent** window opens.

Figure 2.2. Edit fence agent



Edit fence agent

Address	
User Name	
Password	
Type	apc
SSH Port	
Slot	
Options	

Please use a comma-separated list of 'key=value'

Secure □

a. Enter the **Address**, **User Name**, and **Password** of the power management device.

b. Select the power management device **Type** from the drop-down list.

a. Enter the **SSH Port** number used by the power management device to communicate with the host.

b. Enter the **Slot** number used to identify the blade of the power management device.

c. Enter the **Options** for the power management device. Use a comma-separated list of *key=value* entries.

d. Check the **Secure** check box to enable the power management device to connect securely to the host.

e. Click the **Test** button to ensure the settings are correct. *Test Succeeded, Host Status is: on* displays upon successful verification.

WARNING

Power management parameters (userid, password, options, etc.) are tested by Red Hat Virtualization Manager in two situations: during setup, and when parameter values are manually changed in Red Hat Virtualization Manager. If you choose to ignore alerts about incorrect parameters, or if the parameters are changed on the power management hardware without the corresponding change in Red Hat Virtualization Manager, fencing is likely to fail.

f. Click **OK** to close the **Edit fence agent** window.

1. Click **OK** to save your configuration.

You are returned to the list of hosts. Note that the exclamation mark next to the host’s name has now disappeared, signifying that power management has been successfully configured.

CHAPTER 3. CONFIGURE DISASTER RECOVERY USING GEO-REPLICATION

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

IMPORTANT

When used in conjunction with Red Hat Hyperconverged Infrastructure, geo-replication for disaster recovery is supported only as a method of backing up data. Failover (promoting a slave volume) and failback (syncing to a restored master volume) are not supported for the Red Hat Hyperconverged Infrastructure use case.

3.1. CONFIGURING GEO-REPLICATION FOR DISASTER RECOVERY

3.1.1. Before you begin

- Prepare a remote backup volume to hold the geo-replicated copy of your local volume.
  - Ensure that the local volume you want to back up has shared storage enabled. Run the following command on the master node to enable shared storage:

    ```
    # gluster volume set all cluster.enable-shared-storage enable
    ```

  - Ensure that your remote backup volume has sharding enabled. Enable sharding by running the following command on any of the nodes that host the remote backup volume:

    ```
    # gluster volume set VOLNAME features.shard enable
    ```

3.1.2. Configuring a geo-replication session

1. Create (but do not start) the geo-replication session
   From any master node, create (but do not start) a geo-replication session from a local volume to the remote backup volume.

   See the Red Hat Gluster Storage 3.3 *Administration Guide* for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/sect-Preparing_to_Deploy_Geo-replication#Setting_Up_the_Environment_for_Geo-replication_Session

3.1.3. Configuring synchronization schedule

1. Verify that geo-replication is configured
   In Red Hat Virtualization Manager for the master node, click the **Volumes** tab.

   Check the **Info** column for the geo-replication icon. If present, a geo-replication session is configured for that volume.

2. In the **Storage Domain** tab, select the storage domain to back up.

3. Click the **Remote Data Sync Setup** sub-tab
   The *Setup Remote Data Synchronization* window opens.

   a. In the **Geo-replicated to** field, select the backup destination.

   b. In the **Recurrence** field, select a recurrence interval type.
      Valid values are **WEEKLY** with at least one weekday checkbox selected, or **DAILY**.

   c. In the **Hours** and **Minutes** field, specify the time to start synchronizing.

      > **NOTE**
      >
      > This time is based on the Hosted Engine’s timezone.

   d. Click **OK**.

4. Check the *Events* pane at the time you specified to verify that synchronization works correctly.

3.1.4. Deleting synchronization schedule

1. In the **Storage Domain** tab, select the storage domain to back up.

2. Click the **Remote Data Sync Setup** sub-tab
   The *Setup Remote Data Synchronization* window opens.

   a. In the **Recurrence** field, select a recurrence interval type of **NONE**.

   b. Click **OK**.

3. (Optional) Remove the geo-replication session
   Run the following command from the geo-replication master node:

   ```
   # gluster volume geo-replication MASTER_VOL SLAVE_HOST::SLAVE_VOL
   delete
   ```

   You can also run this command with the `reset-sync-time` parameter. For further information about this parameter and geo-replication in general, see the Red Hat Gluster Storage *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-managing_geo-replication.

CHAPTER 4. CONFIGURE ENCRYPTION WITH TRANSPORT LAYER SECURITY (TLS/SSL)

Transport Layer Security (TLS/SSL) can be used to encrypt management and storage layer communications between nodes. This helps ensure that your data remains private.

Encryption can be configured using either self-signed certificates or certificates signed by a Certificate Authority.

This document assumes that you want to enable encryption on an existing deployment. However, encryption can also be configured as part of the deployment process. See *Deploying Red Hat Hyperconverged Infrastructure* for details: https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/deploying_red_hat_hyperconverged_infrastructure/.

4.1. CONFIGURING TLS/SSL USING SELF-SIGNED CERTIFICATES

IMPORTANT

Enabling or disabling encryption is a disruptive process that requires virtual machines and the Hosted Engine to be shut down.

1. Shut down all virtual machines
   See *Shutting Down a Virtual Machine* in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/virtual_machine_management_guide/chap-administrative_tasks.

2. Move all storage domains **except the hosted engine storage domain** into Maintenance mode
   See *Moving Storage Domains to Maintenance Mode*in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/administration_guide/sect-storage_tasks.

3. Move the hosted engine into global maintenance mode
   Run the following command on the virtualization host that hosts the hosted engine:

   ```
   # hosted-engine --set-maintenance --mode=global
   ```

4. Shut down the hosted engine virtual machine
   Run the following command on the virtualization host that hosts the hosted engine:

   ```
   # hosted-engine --vm-shutdown
   ```

   Verify that the hosted engine has shut down by running the following command:

   ```
    # hosted-engine --vm-status
   ```

5. Stop all high availability services
   Run the following command on all virtualization hosts:

   ```
   # systemctl stop ovirt-ha-agent
   # systemctl stop ovirt-ha-broker
   ```

6. Unmount the hosted engine storage domain from all virtualization hosts

   ```
   # hosted-engine --disconnect-storage
   ```

7. Verify that all volumes are unmounted
   On each virtualization host, verify that all gluster volumes are no longer mounted.

   ```
   # mount
   ```

8. Create a gdeploy configuration file
   Use the template file in Section B.1, “Example gdeploy configuration file for setting up TLS/SSL” to create a new configuration file that will set up TLS/SSL on your deployment.

9. Run gdeploy using your new configuration file
   On the first physical machine, run gdeploy using the configuration file you created in the previous step:

   ```
   # gdeploy -c set_up_encryption.conf
   ```

   This may take some time to complete.

10. Verify that no TLS/SSL errors occurred
    Check the **/var/log/glusterfs/glusterd.log** file on each physical machine to ensure that no TLS/SSL related errors occurred, and setup completed successfully.

11. Start all high availability services
    Run the following commands on all virtualization hosts:

    ```
    # systemctl start ovirt-ha-agent
    # systemctl start ovirt-ha-broker
    ```

12. Move the hosted engine out of Global Maintenance mode

    ```
    # hosted-engine --set-maintenance --mode=none
    ```

    The hosted engine starts automatically after a short wait.

13. Wait for nodes to synchronize
    Run the following command on the first virtualization host to check synchronization status. If engine status is listed as **unknown stale-data**, synchronization requires several more minutes to complete.

    The following output indicates completed synchronization.

    ```
    # hosted-engine --vm-status | grep 'Engine status'
    Engine status   : {"health": "good", "vm": "up", "detail": "up"}
    Engine status   : {"reason": "vm not running on this host",
      "health": "bad", "vm": "down", "detail": "unknown"}
    Engine status   : {"reason": "vm not running on this host",
      "health": "bad", "vm": "down", "detail": "unknown"}
    ```

14. Activate all storage domains
    Activate the master storage domain first, followed by all other storage domains.

For details on activating storage domains, see *Activating Storage Domains from Maintenance Mode* in the Red Hat Virtualization documentation: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/administration_guide/sect-storage_tasks.

15. Start all virtual machines
    See *Starting a Virtual Machine* in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/virtual_machine_management_guide/sect-starting_the_virtual_machine.

4.2. CONFIGURING TLS/SSL USING CERTIFICATE AUTHORITY SIGNED CERTIFICATES

IMPORTANT

Enabling or disabling encryption is a disruptive process that requires virtual machines and the Hosted Engine to be shut down.

IMPORTANT

Ensure that you have appropriate certificates signed by a Certificate Authority before proceeding. Obtaining certificates is outside the scope of this document, but further details are available in the Red Hat Gluster Storage *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-network_encryption#chap-Network_Encryption-Prereqs.

1. Shut down all virtual machines
   See *Shutting Down a Virtual Machine* in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/virtual_machine_management_guide/chap-administrative_tasks.

2. Move all storage domains **except the hosted engine storage domain** into Maintenance mode
   See *Moving Storage Domains to Maintenance Mode*in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/administration_guide/sect-storage_tasks.

3. Move the hosted engine into global maintenance mode
   Run the following command on the virtualization host that hosts the hosted engine:

   ```
   # hosted-engine --set-maintenance --mode=global
   ```

4. Shut down the hosted engine virtual machine
   Run the following command on the virtualization host that hosts the hosted engine:

   ```
   # hosted-engine --vm-shutdown
   ```

   Verify that the hosted engine has shut down by running the following command:

   ```
    # hosted-engine --vm-status
   ```

5. Stop all high availability services
   Run the following command on all virtualization hosts:

   ```
   # systemctl stop ovirt-ha-agent
   # systemctl stop ovirt-ha-broker
   ```

6. Unmount the hosted engine storage domain from all virtualization hosts

   ```
   # hosted-engine --disconnect-storage
   ```

7. Verify that all volumes are unmounted
   On each virtualization host, verify that all gluster volumes are no longer mounted.

   ```
   # mount
   ```

8. Configure Certificate Authority signed encryption

   IMPORTANT

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

   a. Place certificates 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.

   b. Stop all volumes

      ```
      # gluster volume stop all
      ```

   c. Restart glusterd on all nodes

      ```
      # systemctl restart glusterd
      ```

   d. Enable TLS/SSL encryption on all volumes

      ```
      # gluster volume set <volname> client.ssl on
      # gluster volume set <volname> server.ssl on
      ```

   e. Specify access permissions on all hosts

      ```
      # gluster volume set <volname> auth.ssl-allow "host1,host2,host3"
      ```

   f. Start all volumes

```
# gluster volume start all
```

9. Verify that no TLS/SSL errors occurred
   Check the **/var/log/glusterfs/glusterd.log** file on each physical machine to ensure that no TLS/SSL related errors occurred, and setup completed successfully.

10. Start all high availability services
    Run the following commands on all virtualization hosts:

    ```
    # systemctl start ovirt-ha-agent
    # systemctl start ovirt-ha-broker
    ```

11. Move the hosted engine out of Global Maintenance mode

    ```
    # hosted-engine --set-maintenance --mode=none
    ```

    The hosted engine starts automatically after a short wait.

12. Wait for nodes to synchronize
    Run the following command on the first virtualization host to check synchronization status. If engine status is listed as **unknown stale-data**, synchronization requires several more minutes to complete.

    The following output indicates completed synchronization.

    ```
    # hosted-engine --vm-status | grep 'Engine status'
    Engine status   : {"health": "good", "vm": "up", "detail": "up"}
    Engine status   : {"reason": "vm not running on this host",
      "health": "bad", "vm": "down", "detail": "unknown"}
    Engine status   : {"reason": "vm not running on this host",
      "health": "bad", "vm": "down", "detail": "unknown"}
    ```

13. Activate all storage domains
    Activate the master storage domain first, followed by all other storage domains.

    For details on activating storage domains, see *Activating Storage Domains from Maintenance Mode* in the Red Hat Virtualization documentation: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/administration_guide/sect-storage_tasks.

14. Start all virtual machines
    See *Starting a Virtual Machine* in the Red Hat Virtualization documentation for details: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/virtual_machine_management_guide/sect-starting_the_virtual_machine.

CHAPTER 5. CONFIGURE PERFORMANCE IMPROVEMENTS

Some deployments benefit from additional configuration to achieve optimal performance. This section covers recommended additional configuration for certain deployments.

5.1. IMPROVING VOLUME PERFORMANCE BY CHANGING SHARD SIZE

The default value of the **`shard-block-size`** parameter changed from **`4MB`** to **`64MB`** between Red Hat Hyperconverged Infrastructure version 1.0 and 1.1. This means that all new volumes are created with a **`shard-block-size`** value of 64MB. However, existing volumes retain the original **`shard-block-size`** value of 4MB.

There is no safe way to modify the **`shard-block-size`** value on volumes that contain data. Because shard block size applies only to writes that occur after the value is set, attempting to change the value on a volume that contains data results in a mixed shard block size, which results in poor performance.

This section shows you how to safely modify the shard block size on an existing volume after upgrading to Red Hat Hyperconverged Infrastructure 1.1, in order to take advantage of the performance benefits of a larger shard size.

5.1.1. Prerequisites

- A logical thin pool with sufficient free space to create additional logical volumes that are large enough to contain all existing virtual machines.
- A complete backup of your data. For details on how to achieve this, see Chapter 3, *Configure Disaster Recovery using Geo-replication*.

5.1.2. Safely changing the shard block size parameter value

A. Create a new storage domain

1. **Create new thin provisioned logical volumes**

   a. For an arbitrated replicated volume:

      i. Create an **`lv_create_arbitrated.conf`** file with the following contents:

```
[lv10:{<Gluster_Server_IP1>,<Gluster_Server_IP2>}]
action=create
lvname=<lv_name>
ignore_lv_errors=no
vgname=<volgroup_name>
mount=<brick_mountpoint>
lvtype=thinlv
poolname=<thinpool_name>
virtualsize=<size>

[lv11:<Gluster_Server_IP3>]
action=create
lvname=<lv_name>
ignore_lv_errors=no
vgname=<volgroup_name>
```

```
mount=<brick_mountpoint>
lvtype=thinlv
poolname=<thinpool_name>
virtualsize=<size>
```

ii. Run the following command:

```
# gdeploy -c lv_create_arbitrated.conf
```

b. For a normal replicated volume:

i. Create an **lv_create_replicated.conf** file with the following contents:

```
[lv3]
action=create
lvname=<lv_name>
ignore_lv_errors=no
vgname=<volgroup_name>
mount=<brick_mountpoint>
lvtype=thinlv
poolname=<thinpool_name>
virtualsize=<size>
```

ii. Run the following command:

```
# gdeploy -c lv_create_replicated.conf
```

2. **Create new gluster volumes on the new logical volumes**

a. For an arbitrated replicated volume

i. Create a **gluster_arb_volume.conf** file with the following contents:

```
[volume4]
action=create
volname=data_one
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,server.ssl,client.ssl,auth.ssl-allow
value=virt,36,36,30,on,off,enable,64MB,on,on,"
<Gluster_Server_IP1>;<Gluster_Server_IP2>;
<Gluster_Server_IP3>"
brick_dirs=<Gluster_Server_IP1>:<brick1_mountpoint>,
<Gluster_Server_IP2>:<brick2_mountpoint>,<Gluster_Server_IP3>:
<brick3_mountpoint>
ignore_volume_errors=no
arbiter_count=1
```

ii. Run the following command:

```
# gdeploy -c gluster_arb_volume.conf
```

b. For a normal replicated volume:

i. Create a gluster_rep_volume.conf file with the following contents:

```
[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=<Gluster_Server_IP1>:<brick1_mountpoint>,
<Gluster_Server_IP2>:<brick2_mountpoint>,<Gluster_Server_IP3>:
<brick3_mountpoint>
ignore_volume_errors=no
```

ii. Run the following command:

```
# gdeploy -c gluster_rep_volume.conf
```

3. **Create a new storage domain using the new gluster volumes**
   Browse to the engine and follow the steps in Create the master storage domain to add a new storage domain consisting of the new gluster volume.

B. Migrate any virtual machine templates

If your virtual machines are created from templates, copy each template to the new Storage Domain.

Click the **Template** tab. For each template to migrate:

1. Select the template to migrate.
2. Click the **Disks** tab.
3. Click **Copy**, and select the new storage domain as the target domain.

C. Migrate virtual machine disks to the new storage domain

For each virtual machine:

1. Right-click **VM Data Disks** → **Move**.
2. Select the new storage domain as the target domain.

You can monitor progress in the **Tasks** tab.

D. Verify that disk images migrated correctly

Click the **Disks** tab. For each migrated disk:

1. Select the disk to check.
2. Click the **Storage** sub-tab.

3. Verify that the domain listed is the new storage domain.

IMPORTANT

Do not skip this step. There is no way to retrieve a disk image after a domain is detached and removed, so be sure that all disk images have correctly migrated before you move on.

E. Remove and reclaim the old storage domain

1. Move the old storage domain into maintenance mode.
2. Detach the old storage domain from the data center.
3. Remove the old storage domain from the data center.

5.2. CONFIGURING A LOGICAL VOLUME CACHE (LVMCACHE) FOR IMPROVED PERFORMANCE

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

1. **Create the gdeploy configuration file**
   Create a gdeploy configuration file named `lvmcache.conf` that contains at least the following information. Note that the **ssd** value should be the device name, not the device path (for example, use **sdb** not `/dev/sdb`).

   ### Example lvmcache.conf file

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

   [lv1]
   action=setup-cache
   ssd=sdb
   vgname=gluster_vg_sdb
   poolname=gluster_thinpool_sdb
   cache_lv=lvcache
   cache_lvsize=220GB
   #cachemode=writethrough
   ```

   IMPORTANT

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

IMPORTANT

The default cache mode is **`writethrough`**, but **`writeback`** mode is also supported. To avoid the potential for data loss when implementing lvmcache in **`writeback`** mode, two separate SSD/NVMe devices are highly recommended. By configuring the two devices in a RAID-1 configuration (via software or hardware), the potential of data loss from lost writes is reduced significantly.

2. **Run gdeploy**
   Run the following command to apply the configuration specified in **`lvmcache.conf`**.

   ```
   # gdeploy -c lvmcache.conf
   ```

For further information about lvmcache configuration, see *Red Hat Enterprise Linux 7 LVM Administration*: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Logical_Volume_Manager_Administration/LV.html#lvm_cache_volun

PART II. MAINTENANCE TASKS

CHAPTER 6. UPGRADING TO RED HAT HYPERCONVERGED INFRASTRUCTURE 1.1

Upgrading involves moving from one version of a product to a newer major release of the same product. This section shows you up to upgrade to Red Hat Hyperconverged Infrastructure 1.1 from version 1.0.

From a component standpoint, this involves:

- upgrading the Hosted Engine virtual machine to version 4.1.8
- updating the physical hosts to Red Hat Virtualization 4.1.8
  Updating the physical hosts includes an upgrade of Red Hat Gluster Storage from version 3.2 to version 3.3.1.

6.1. MAJOR DIFFERENCES IN RED HAT HYPERCONVERGED INFRASTRUCTURE 1.1

Be aware of the following differences between Red Hat Hyperconverged Infrastructure 1.1 and previous versions.

Changed shard-block-size parameter default value

The default value of the **`shard-block-size`** parameter is now **64MB**, instead of the previous default value of **4MB**. Existing volumes retain the previous value of 4MB when a deployment is upgraded from Red Hat Hyperconverged Infrastructure 1.0 to 1.1.
There is no safe way to modify the **`shard-block-size`** value on volumes that contain data. Because shard block size applies only to writes that occur after the value is set, attempting to change the value on a volume that contains data results in a mixed shard block size, which results in poor performance.

Customers upgrading from Red Hat Hyperconverged Infrastructure 1.0 who want to use the new shard block size of 64MB can follow the steps in Section 5.1, “Improving volume performance by changing shard size” to create a new volume, migrate data to the new volume, and then change the block size on the original volume when all data has been removed.

6.2. UPGRADE WORKFLOW

Red Hat Hyperconverged Infrastructure is a software solution comprised of several different components. Upgrade the components in the following order to minimize disruption to your deployment.

1. Hosted Engine virtual machine
2. Physical hosts

6.3. PREPARING TO UPGRADE

- Ensure that your Hosted Engine virtual machine is subscribed to the following repositories.

```
# subscription-manager repos --enable=rhel-7-server-rhv-4.1-rpms
# subscription-manager repos --enable=rhel-7-server-rhv-4-tools-rpms
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-supplementary-
```

```
rpms
# subscription-manager repos --enable=jb-eap-7-for-rhel-7-server-
rpms
```

- Ensure that all physical machines are subscribed to the **rhel-7-server-rhvh-4-rpms** repository.

  ```
  # subscription-manager repos --enable=rhel-7-server-rhvh-4-rpms
  ```

- If geo-replication is configured, ensure that data is not being synchronized.

  a. Check the **Tasks** subtab and ensure that there are no ongoing tasks related to Data Synchronization. If data synchronization tasks are present, wait until they are complete before beginning the update.

  b. Stop all geo-replication sessions so that synchronization will not occur during the update. Click the **Geo-replication** subtab and select the session that you want to stop, then click **Stop**.
  Alternatively, run the following command to stop a geo-replication session.

  ```
  # gluster volume geo-replication MASTER_VOL SLAVE_HOST::SLAVE_VOL
  stop
  ```

6.4. UPGRADING RED HAT HYPERCONVERGED INFRASTRUCTURE

6.4.1. Upgrading the Hosted Engine virtual machine

Follow the steps in the following section of the Red Hat Virtualization *Self-Hosted Engine Guide* to upgrade the Hosted Engine virtual machine: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/single/self-hosted_engine_guide/#Upgrading_the_Self-Hosted_Engine.

6.4.2. Upgrading the physical hosts

Follow the steps in the sections linked below to upgrade the physical hosts one at a time.

Between upgrades, ensure that you wait for any heal operations to complete before upgrading the next host. You can view heal status in the **Bricks** subtab. Alternatively, run the following command for every volume, and ensure that **Number of entries: 0** is displayed for each brick before upgrading the next host.

```
# gluster volume heal VOLNAME info
```

Most upgrades can be applied using Red Hat Virtualization Manager. Follow the steps in the following section of the Red Hat Virtualization *Upgrade Guide* to update the physical host machines one at a time: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/upgrade_guide/updating_virtualization_hosts.

If you need to apply a security fix, apply upgrades manually instead. Follow the steps in the following section of the Red Hat Virtualization *Upgrade Guide*: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/upgrade_guide/Manually_Updating_Virtualization_Hosts

IMPORTANT

When you set a gluster server into maintenance mode, ensure that you check the **`Stop Gluster service`** checkbox.

IMPORTANT

Remember to move your hosts out of maintenance mode when their updates have been applied by running the following command:

```
# hosted-engine --set-maintenance --mode=none
```

CHAPTER 7. REINSTALLING A VIRTUALIZATION HOST

Some configuration changes require a virtualization host to be reinstalled before the configuration change can take effect. Follow these steps to reinstall a virtualization host.

1. Select the host and click **`Management`** > **`Maintenance`** > **`OK`** to place this host in Maintenance mode.

2. Click **`Installation`** > **`Reinstall`** to open the Reinstall window.

3. On the General tab, uncheck the **`Automatically Configure Host firewall`** checkbox.

4. On the Hosted Engine tab, set the value of **`Choose hosted engine deployment action`** to **`deploy`**.

5. Click **`OK`** to reinstall the host.

CHAPTER 8. REPLACING THE PRIMARY GLUSTER STORAGE NODE

IMPORTANT

When self-signed encryption is enabled, replacing a node is a disruptive process that requires virtual machines and the Hosted Engine to be shut down.

1. (Optional) If encryption using a Certificate Authority is enabled, follow the steps at the following link before continuing: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/ch22s04.

2. Move the node to be replaced into Maintenance mode

   a. In Red Hat Virtualization Manager, click the **Hosts** tab and select the Red Hat Gluster Storage node in the results list.

   b. Click **Maintenance** to open the *Maintenance Host(s)* confirmation window.

   c. Click **OK** to move the host to Maintenance mode.

3. Install the replacement node
   Follow the instructions in *Deploying Red Hat Hyperconverged Infrastructure* to install the physical machine and configure storage on the new node.

   a. Installing host physical machines

   b. Configuring Public Key based SSH Authentication

   c. Configuring RHGS for Hosted Engine using the Cockpit UI

4. Prepare the replacement node

   a. Create a file called **replace_node_prep.conf** based on the template provided in Section B.2, “Example gdeploy configuration file for preparing to replace a node”.

   b. From a node with **gdeploy** installed (usually the node that hosts the Hosted Engine), run gdeploy using the new configuration file:

   ```
   # gdeploy -c replace_node_prep.conf
   ```

5. (Optional) If encryption with self-signed certificates is enabled

   a. Generate the private key and self-signed certificate on the replacement node. See the Red Hat Gluster Storage *Administration Guide* for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-network_encryption#chap-Network_Encryption-Prereqs.

   b. On a healthy node, make a backup copy of the /etc/ssl/glusterfs.ca file:

   ```
   # cp /etc/ssl/glusterfs.ca /etc/ssl/glusterfs.ca.bk
   ```

   c. Append the new node’s certificate to the content of the /etc/ssl/glusterfs.ca file.

   d. Distribute the /etc/ssl/glusterfs.ca file to all nodes in the cluster, including the new node.

   e. Run the following command on the replacement node to enable management encryption:

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

   f. Include the new server in the value of the **`auth.ssl-allow`** volume option by running the following command for each volume.

   ```
   # gluster volume set <volname> auth.ssl-allow "<old_node1>,
   <old_node2>,<new_node>"
   ```

   g. Restart the glusterd service on all nodes

   ```
   # systemctl restart glusterd
   ```

   h. Follow the steps in Section 4.1, “Configuring TLS/SSL using self-signed certificates” to remount all gluster processes.

6. Add the replacement node to the cluster
   Run the following command from any node already in the cluster.

   ```
   # peer probe <new_node>
   ```

7. Move the Hosted Engine into Maintenance mode:

   ```
   # hosted-engine --set-maintenance --mode=global
   ```

8. Stop the ovirt-engine service

   ```
   # systemctl stop ovirt-engine
   ```

9. Update the database

   ```
   # sudo -u postgres psql
   \c engine;
   UPDATE storage_server_connections SET connection
   ='<replacement_node_IP>:/engine' WHERE connection =
   ‘<old_server_IP>:/engine';
   UPDATE storage_server_connections SET connection
   ='<replacement_node_IP>:/vmstore' WHERE connection =
   ‘<old_server_IP>:/vmstore';
   UPDATE storage_server_connections SET connection
   ='<replacement_node_IP>:/data' WHERE connection =
   '<old_server_IP>:/data';
   ```

10. Start the ovirt-engine service

    ```
    # systemctl start ovirt-engine
    ```

11. Stop all virtual machines except the Hosted Engine.

12. Move all storage domains **except** the Hosted Engine domain into Maintenance mode

13. Stop the Hosted Engine virtual machine
    Run the following command on the existing node that hosts the Hosted Engine.

    ```
    # hosted-engine --vm-shutdown
    ```

14. Stop high availability services on all nodes

    ```
    # systemctl stop ovirt-ha-agent
    # systemctl stop ovirt-ha-broker
    ```

15. Disconnect Hosted Engine storage from the virtualization host
    Run the following command on the existing node that hosts the Hosted Engine.

    ```
    # hosted-engine --disconnect-storage
    ```

16. Update the Hosted Engine configuration file
    Edit the **storage** parameter in the **`/etc/ovirt-hosted-engine/hosted-engine.conf`** file to use the replacement server.

    ```
    storage=<replacement_server_IP>:/engine
    ```

17. Reboot the existing and replacement nodes
    Wait until both nodes are available before continuing.

18. Take the Hosted Engine out of Maintenance mode

    ```
    # hosted-engine --set-maintenance --mode=none
    ```

19. Verify replacement node is used
    On all virtualization hosts, verify that the **engine** volume is mounted from the replacement node by checking the IP address in the output of the **`mount`** command.

20. Activate storage domains
    Verify that storage domains mount using the IP address of the replacement node.

21. Remove the old node

    a. Using the RHV Management UI, remove the old node.

    b. Detach the old host from the cluster.

       ```
       # gluster peer detach <old_node_IP> force
       ```

22. Using the RHV Management UI, add the replacement node
    Specify that the replacement node be used to host the Hosted Engine.

23. Move the replacement node into Maintenance mode.

    ```
    # hosted-engine --set-maintenance --mode=global
    ```

24. Update the Hosted Engine configuration file
    Edit the **storage** parameter in the **`/etc/ovirt-hosted-engine/hosted-engine.conf`** file to use the replacement node.

```
storage=<replacement_node_IP>:/engine
```

25. Reboot the replacement node.
    Wait until the node is back online before continuing.

26. Activate the replacement node from the RHV Management UI.
    Ensure that all volumes are mounted using the IP address of the replacement node.

27. Replace engine volume brick
    Replace the brick on the old node that belongs to the engine volume with a new brick on the replacement node.

    a. Click the **Volumes** tab.

    b. Click the **Bricks** sub-tab.

    c. Select the brick to replace, and then click **Replace brick**.

    d. Select the node that hosts the brick being replaced.

    e. In the *Replace brick* window, provide the new brick’s path.

28. On the replacement node, run the following command to remove metadata from the previous host.

```
# hosted-engine --clean-metadata --host-id=<old_host_id> --force-
clean
```

CHAPTER 9. REPLACING A GLUSTER STORAGE NODE

If a Red Hat Gluster Storage node needs to be replaced, there are two options for the replacement node:

1. Replace the node with a new node that has a different fully-qualified domain name by following the instructions in Section 9.1, “Replacing a Gluster Storage Node (Different FQDN)”.

2. Replace the node with a new node that has the same fully-qualified domain name by following the instructions in Section 9.2, “Replacing a Gluster Storage Node (Same FQDN)”.

Follow the instructions in whichever section is appropriate for your deployment.

9.1. REPLACING A GLUSTER STORAGE NODE (DIFFERENT FQDN)

IMPORTANT

When self-signed encryption is enabled, replacing a node is a disruptive process that requires virtual machines and the Hosted Engine to be shut down.

1. Prepare the replacement node
   Follow the instructions in *Deploying Red Hat Hyperconverged Infrastructure* to install the physical machine.

   a. Installing host physical machines

   b. Configuring Public Key based SSH Authentication

2. Stop any existing geo-replication sessions

   ```
   # gluster volume geo-replication MASTER_VOL SLAVE_HOST::SLAVE_VOL
   stop
   ```

   For further information, see the Red Hat Gluster Storage *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/sect-starting_geo-replication#Stopping_a_Geo-replication_Session.

3. Move the node to be replaced into Maintenance mode
   Perform the following steps in Red Hat Virtualization Manager:

   a. Click the **Hosts** tab and select the Red Hat Gluster Storage node in the results list.

   b. Click **Maintenance** to open the *Maintenance Host(s)* confirmation window.

   c. Click **OK** to move the host to Maintenance mode.

4. Prepare the replacement node

   a. Configure key-based SSH authentication
      Configure key-based SSH authentication from a physical machine still in the cluster to the replacement node. For details, see https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/deploying_red_hat_hyperconverged_infras
      configure-key-based-ssh-auth.

   b. Prepare the replacement node

Create a file called **replace_node_prep.conf** based on the template provided in Section B.2, “Example gdeploy configuration file for preparing to replace a node”.

From a node with **gdeploy** installed (usually the node that hosts the Hosted Engine), run gdeploy using the new configuration file:

```
# gdeploy -c replace_node_prep.conf
```

5. Create replacement brick directories
   Ensure the new directories are owned by the **vdsm** user and the **kvm** group.

```
# mkdir /gluster_bricks/engine/engine
# chmod vdsm:kvm /gluster_bricks/engine/engine
# mkdir /gluster_bricks/data/data
# chmod vdsm:kvm /gluster_bricks/data/data
# mkdir /gluster_bricks/vmstore/vmstore
# chmod vdsm:kvm /gluster_bricks/vmstore/vmstore
```

6. (Optional) If encryption is enabled

   a. Generate the private key and self-signed certificate on the new server using the steps in the Red Hat Gluster Storage *Administration Guide*: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-network_encryption#chap-Network_Encryption-Prereqs.
      If encryption using a Certificate Authority is enabled, follow the steps at the following link before continuing: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/ch22s04.

   b. Add the new node’s certificate to existing certificates.

      i. On one of the healthy nodes, make a backup copy of the **/etc/ssl/glusterfs.ca** file.

      ii. Add the new node’s certificate to the **/etc/ssl/glusterfs.ca** file on the healthy node.

      iii. Distribute the updated **/etc/ssl/glusterfs.ca** file to all other nodes, including the new node.

   c. Enable management encryption
      Run the following command on the new node to enable management encryption:

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

   d. Include the new server in the value of the **auth.ssl-allow** volume option by running the following command for each volume.

```
# gluster volume set <volname> auth.ssl-allow "<old_node1>,
<old_node2>,<new_node>"
```

   e. Restart the glusterd service on all nodes

```
# systemctl restart glusterd
```

   f. If encryption uses self-signed certificates, follow the steps in Section 4.1, “Configuring TLS/SSL using self-signed certificates” to remount all gluster processes.

7. Add the new host to the existing cluster

   a. Run the following command from one of the healthy cluster members:

      ```
      # gluster peer probe <new_node>
      ```

   b. Add the new host to the existing cluster

      i. Click the **Hosts** tab and then click **New** to open the *New Host* dialog.

      ii. Provide a **Name**, **Address**, and **Password** for the new host.

      iii. Uncheck the **Automatically configure host firewall** checkbox, as firewall rules are already configured by gdeploy.

      iv. In the **Hosted Engine** tab of the *New Host* dialog, set the value of **Choose hosted engine deployment action** to **deploy**.

      v. Click **Deploy**.

      vi. When the host is available, click the **Network Interfaces** subtab and then click **Setup Host Networks**.

      vii. Drag and drop the network you created for gluster to the IP associated with this host, and click **OK**.
      See the Red Hat Virtualization 4.1 Self-Hosted Engine Guide for further details: https://access.redhat.com/documentation/en/red-hat-virtualization/4.1/paged/self-hosted-engine-guide/chapter-7-installing-additional-hosts-to-a-self-hosted-environment.

8. Configure and mount shared storage on the new host

   ```
   # cp /etc/fstab /etc/fstab.bk
   # echo "<new_host>:/gluster_shared_storage
   /var/run/gluster/shared_storage/ glusterfs defaults 0 0" >>
   /etc/fstab
   # mount /gluster_shared_storage
   ```

9. Replace the old brick with the brick on the new host

   a. In Red Hat Virtualization Manager, click the **Hosts** tab and select the volume.

   b. Click the **Bricks** sub-tab.

   c. Click **Replace Brick** beside the old brick and specify the replacement brick.

   d. Verify that brick heal completes successfully.

10. In the **Hosts** tab, right-click on the old host and click **Remove**.
    Use **`gluster peer status`** to verify that that the old host no longer appears. If the old host is still present in the status output, run the following command to forcibly remove it:

    ```
    # gluster peer detach <old_node> force
    ```

11. Clean old host metadata

    ```
    # hosted-engine --clean-metadata --host-id=<old_host_id> --force-
    clean
    ```

12. Set up new SSH keys for geo-replication of new brick

    ```
    # gluster system:: execute gsec_create
    ```

13. Recreate geo-replication session and distribute new SSH keys.

    ```
    # gluster volume geo-replication <MASTER_VOL> <SLAVE_HOST>::
    <SLAVE_VOL> create push-pem force
    ```

14. Start the geo-replication session.

    ```
    # gluster volume geo-replication <MASTER_VOL> <SLAVE_HOST>::
    <SLAVE_VOL> start
    ```

9.2. REPLACING A GLUSTER STORAGE NODE (SAME FQDN)

IMPORTANT

When self-signed encryption is enabled, replacing a node is a disruptive process that requires virtual machines and the Hosted Engine to be shut down.

1. (Optional) If encryption using a Certificate Authority is enabled, follow the steps at the following link before continuing: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/ch22s04.
2. Move the node to be replaced into Maintenance mode
    a. In Red Hat Virtualization Manager, click the **Hosts** tab and select the Red Hat Gluster Storage node in the results list.
    b. Click **Maintenance** to open the *Maintenance Host(s)* confirmation window.
    c. Click **OK** to move the host to Maintenance mode.
3. Prepare the replacement node
   Follow the instructions in *Deploying Red Hat Hyperconverged Infrastructure* to install the physical machine and configure storage on the new node.
    a. Installing host physical machines
    b. Configuring Public Key based SSH Authentication
    c. Configuring RHGS for Hosted Engine using the Cockpit UI
4. Prepare the replacement node
    a. Create a file called **replace_node_prep.conf** based on the template provided in Section B.2, “Example gdeploy configuration file for preparing to replace a node”.

   b. From a node with **gdeploy** installed (usually the node that hosts the Hosted Engine), run gdeploy using the new configuration file:

   ```
   # gdeploy -c replace_node_prep.conf
   ```

5. (Optional) If encryption with self-signed certificates is enabled

   a. Generate the private key and self-signed certificate on the replacement node. See the Red Hat Gluster Storage *Administration Guide* for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/chap-network_encryption#chap-Network_Encryption-Prereqs.

   b. On a healthy node, make a backup copy of the /etc/ssl/glusterfs.ca file:

   ```
   # cp /etc/ssl/glusterfs.ca /etc/ssl/glusterfs.ca.bk
   ```

   c. Append the new node’s certificate to the content of the /etc/ssl/glusterfs.ca file.

   d. Distribute the /etc/ssl/glusterfs.ca file to all nodes in the cluster, including the new node.

   e. Run the following command on the replacement node to enable management encryption:

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

6. Replace the host machine
   Follow the instructions in the Red Hat Gluster Storage *Administration Guide* to replace the host: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/administration_guide/sect-replacing_hosts#Replacing_a_Host_Machine_with_the_Same_Hostname.

7. Restart the glusterd service on all nodes

   ```
   # systemctl restart glusterd
   ```

8. Verify that all nodes reconnect

   ```
   # gluster peer status
   ```

9. (Optional) If encryption uses self-signed certificates, follow the steps in Section 4.1, “Configuring TLS/SSL using self-signed certificates” to remount all gluster processes.

10. Verify that all nodes reconnect and that brick heal completes successfully

   ```
   # gluster peer status
   ```

11. Refresh fingerprint

   a. In Red Hat Virtualization Manager, click the **Hosts** tab and select the new host.

   b. Click **Edit Host**.

   c. Click **Advanced** on the details screen.

   d. Click **Fetch fingerprint**.

12. Click **Reinstall** and provide the root password when prompted.

13. Click the **Hosted Engine** tab and click **Deploy**

14. Attach the gluster network to the host

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

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

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

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

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

    f. Click **OK** to save.

15. Verify the health of the network
    Click the Hosts tab and select the host. Click the Networks subtab and check the state of the host’s network.

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

CHAPTER 10. RESTORING A VOLUME FROM A GEO-REPLICATED BACKUP

1. Install and configure a replacement Hyperconverged Infrastructure deployment
   For instructions, refer to *Deploying Red Hat Hyperconverged Infrastructure*: https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure/1.1/html/deploying_red_hat_hyperconverged_infrastruct

2. Import the backup of the storage domain
   From the new Hyperconverged Infrastructure deployment, in Red Hat Virtualization Manager:

   a. Click the **Storage** tab.

   b. Click **Import Domain**. The *Import Pre-Configured Domain* window opens.

   c. In the **Storage Type** field, specify **GlusterFS**.

   d. In the **Name** field, specify a name for the new volume that will be created from the backup volume.

   e. In the **Path** field, specify the path to the backup volume.

   f. Click **OK**. The following warning appears, with any active data centers listed below:

   ```
   This operation might be unrecoverable and destructive!

   Storage Domain(s) are already attached to a Data Center.
   Approving this operation might cause data corruption if
   both Data Centers are active.
   ```

   g. Check the **Approve operation** checkbox and click **OK**.

3. Determine a list of virtual machines to import

   a. Determine the imported domain's identifier
      The following command returns the domain identifier.

   ```
   # curl -v -k -X GET -u "admin@internal:password" -H "Accept:
   application/xml" https://$ENGINE_FQDN/ovirt-
   engine/api/storagedomains/
   ```

   For example:

   ```
   # curl -v -k -X GET -u "admin@example.com:mybadpassword" -H
   "Accept: application/xml" https://10.0.2.1/ovirt-
   engine/api/storagedomains/
   ```

   b. Determine the list of unregistered disks by running the following command:

   ```
   # curl -v -k -X GET -u "admin@internal:password" -H "Accept:
   application/xml" "https://$ENGINE_FQDN/ovirt-
   engine/api/storagedomains/DOMAIN_ID/vms;unregistered"
   ```

   For example:

```
# curl -v -k -X GET -u "admin@example.com:mybadpassword" -H
"Accept: application/xml" "https://10.0.2.1/ovirt-
engine/api/storagedomains/5e1a37cf-933d-424c-8e3d-
eb9e40b690a7/vms;unregistered"
```

4. Perform a partial import of each virtual machine to the storage domain

   a. Determine cluster identifier
   The following command returns the cluster identifier.

   ```
   # curl -v -k -X GET -u "admin@internal:password" -H "Accept:
   application/xml" https://$ENGINE_FQDN/ovirt-engine/api/clusters/
   ```

   For example:

   ```
   # curl -v -k -X GET -u "admin@example:mybadpassword" -H "Accept:
   application/xml" https://10.0.2.1/ovirt-engine/api/clusters/
   ```

   b. Import the virtual machines
   The following command imports a virtual machine without requiring all disks to be available in the storage domain.

   ```
   # curl -v -k -u 'admin@internal:password' -H "Content-type:
   application/xml" -d '<action> <cluster id="CLUSTER_ID"></cluster>
   <allow_partial_import>true</allow_partial_import> </action>'
   "https://ENGINE_FQDN/ovirt-
   engine/api/storagedomains/DOMAIN_ID/vms/VM_ID/register"
   ```

   For example:

   ```
   # curl -v -k -u 'admin@example.com:mybadpassword' -H "Content-
   type: application/xml" -d '<action> <cluster id="bf5a9e9e-5b52-
   4b0d-aeba-4ee4493f1072"></cluster>
   <allow_partial_import>true</allow_partial_import> </action>'
   "https://10.0.2.1/ovirt-engine/api/storagedomains/8d21980a-a50b-
   45e9-9f32-cd8d2424882e/e164f8c6-769a-4cbd-ac2a-
   ef322c2c5f30/register"
   ```

   For further information, see the Red Hat Virtualization *REST API Guide*: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/rest_api_guide/.

5. Migrate the partially imported disks to the new storage domain
   On the **Disks** tab, click on the **Move Disk** option. Move the imported disks from the synced volume to the replacement cluster's storage domain. For further information, see the Red Hat Virtualization Administration Guide.

6. Attach the restored disks to the new virtual machines
   Follow the instructions in the Red Hat Virtualization Virtual Machine Management Guide to attach the replacement disks to each virtual machine.

PART III. REFERENCE MATERIAL

APPENDIX A. FENCING POLICIES FOR RED HAT GLUSTER STORAGE

The following fencing policies are required for Red Hat Hyperconverged Infrastructure (RHHI) deployments. They ensure that hosts are not shut down in situations where brick processes are still running, or when shutting down the host would remove the cluster's ability to reach a quorum.

These policies can be set in the **New Cluster** or **Edit Cluster** window in Red Hat Virtualization Manager when Red Hat Gluster Storage functionality is enabled.

Skip fencing if gluster bricks are up

Fencing is skipped if bricks are running and can be reached from other peers.

Skip fencing if gluster quorum not met

Fencing is skipped if bricks are running and shutting down the host will cause loss of quorum

These policies are checked after all other fencing policies when determining whether a node is fenced.

Additional fencing policies may be useful for your deployment. For further details about fencing, see the Red Hat Virtualization *Technical Reference*: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html/technical_reference/fencing.

APPENDIX B. EXAMPLE GDEPLOY CONFIGURATION FILES

B.1. EXAMPLE GDEPLOY CONFIGURATION FILE FOR SETTING UP TLS/SSL

set_up_encryption.conf

```
# IPs that corresponds to the Gluster Network
[hosts]
<Gluster_Network_NodeA>
<Gluster_Network_NodeB>
<Gluster_Network_NodeC>

# STEP-1: Generate Keys, Certificates & CA files
# The following section generates the keys,certicates, creates
# ca file and distributes it to all the hosts
[volume1]
action=enable-ssl
volname=engine
ssl_clients=<Gluster_Network_NodeA>,<Gluster_Network_NodeB>,
<Gluster_Network_NodeC>
ignore_volume_errors=no

# As the certificates are already generated, its enough to stop
# rest of the volumes,set TLS/SSL related volume options, and
# start the volume

# STEP-2: Stop all the volumes
[volume2]
action=stop
volname=vmstore

[volume3]
action=stop
volname=data

# STEP-3: Set volume options on all the volumes to enable TLS/SSL on the
volumes
[volume4]
action=set
volname=vmstore
key=client.ssl,server.ssl,auth.ssl-allow
value=on,on,"<Gluster_Network_NodeA>;<Gluster_Network_NodeB>;
<Gluster_Network_NodeC>"
ignore_volume_errors=no

[volume5]
action=set
volname=data
key=client.ssl,server.ssl,auth.ssl-allow
value=on,on,"<Gluster_Network_NodeA>;<Gluster_Network_NodeB>;
<Gluster_Network_NodeC>"
ignore_volume_errors=no
```

```
# STEP-4: Start all the volumes
[volume6]
action=start
volname=vmstore

[volume7]
action=start
volname=data
```

B.2. EXAMPLE GDEPLOY CONFIGURATION FILE FOR PREPARING TO REPLACE A NODE

IMPORTANT

If the disks must be replaced as well as the node, ensure that the **[pv]**, **[vg]**, and **[lv]** sections are not commented out of this file.

For details about how to safely replace a node, see Chapter 9, *Replacing a Gluster Storage Node*.

replace_node_prep.conf

```
# EDITME: @1: Change to IP addresses of the network intended for gluster
traffic
# Values provided here are used to probe the gluster hosts.
[hosts]
10.70.X1.Y1

#EDITME : @2: Change to IP addresses of the network intended for gluster
traffic
#of the node which is going to be replaced.
[script1]
action=execute
ignore_script_errors=no
file=/usr/share/ansible/gdeploy/scripts/grafton-sanity-check.sh -d sdc -h
10.70.X1.Y1

# EDITME: @3: Specify the number of data disks in RAID configuration
[disktype]
raid6

[diskcount]
4

[stripesize]
256

# EDITME : @4:  UNCOMMENT SECTION (RHEL ONLY) :Provide the subscription
details
# Register to RHSM only on the node which needs to be replaced
#[RH-subscription1:10.70.X1.Y1]
#action=register
#username=<username>
#password=<passwd>
#pool=<pool-id>
```

```
#[RH-subscription2]
#action=disable-repos
#repos=

#[RH-subscription3]
#action=enable-repos
#repos=rhel-7-server-rpms,rh-gluster-3-for-rhel-7-server-rpms,rhel-7-
server-rhv-4-mgmt-agent-rpms

#[yum1]
#action=install
#packages=vdsm,vdsm-gluster,ovirt-hosted-engine-setup,screen,gluster-
nagios-addons
#update=yes

[service1]
action=enable
service=chronyd

[service2]
action=restart
service=chronyd

[shell1]
action=execute
command=gluster pool list

[shell2]
action=execute
command=vdsm-tool configure --force

# Disable multipath
[script3]
action=execute
file=/usr/share/ansible/gdeploy/scripts/disable-multipath.sh

#EDIT ME: @5: UNCOMMENT SECTIONS ONLY: if original brick disks have to be
replaced.
#[pv1]
#action=create
#devices=sdc
#ignore_pv_errors=no

#[vg1]
#action=create
#vgname=gluster_vg_sdc
#pvname=sdc
#ignore_vg_errors=no


#[lv2:10.70.X1:Y1]
#action=create
#poolname=gluster_thinpool_sdc
#ignore_lv_errors=no
#vgname=gluster_vg_sdc
```

```
#lvtype=thinpool
#poolmetadatasize=16GB
#size=14TB

#[lv3:10.70.X1:Y1]
#action=create
#lvname=gluster_lv_engine
#ignore_lv_errors=no
#vgname=gluster_vg_sdc
#mount=/gluster_bricks/engine
#size=100GB
#lvtype=thick


#[lv5:10.70.X1:Y1]
#action=create
#lvname=gluster_lv_data
#ignore_lv_errors=no
#vgname=gluster_vg_sdc
#mount=/gluster_bricks/data
#lvtype=thinlv
#poolname=gluster_thinpool_sdc
#virtualsize=12TB


#[lv7:10.70.X1:Y1]
#action=create
#lvname=gluster_lv_vmstore
#ignore_lv_errors=no
#vgname=gluster_vg_sdc
#mount=/gluster_bricks/vmstore
#lvtype=thinlv
#poolname=gluster_thinpool_sdc
#virtualsize=1TB

#[selinux]
#yes

#[lv9:10.70.X1:Y1]
#action=setup-cache
#ssd=sdb
#vgname=gluster_vg_sdc
#poolname=lvthinpool
#cache_lv=lvcache
#cache_lvsize=180GB

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

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

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

B.3. EXAMPLE GDEPLOY CONFIGURATION FILE FOR SCALING TO ADDITIONAL NODES

add_nodes.conf

```
# Add the hosts to be added
[hosts]
<Gluster_Network_NodeD>
<Gluster_Network_NodeE>
<Gluster_Network_NodeF>

# If using RHEL 7 as platform, enable required repos
# RHVH has all the packages available
#[RH-subscription]
#ignore_register_errors=no
#ignore_attach_pool_errors=no
#ignore_enable_errors=no
#action=register
#username=<username>
#password=<mypassword>
#pool=<pool-id>
#repos=rhel-7-server-rpms,rh-gluster-3-for-rhel-7-server-rpms,rhel-7-
server-rhv-4-mgmt-agent-rpms
#disable-repos=yes

# If using RHEL 7 as platform, have the following section to install
packages
[yum1]
action=install
packages=vdsm-gluster,ovirt-hosted-engine-setup,screen
update=yes
gpgcheck=yes
ignore_yum_errors=no

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

# start chronyd service
[service2]
action=restart
service=chronyd

# Setup glusterfs slice
[service3]
action=restart
service=glusterd
slice_setup=yes
```

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

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