Oracle Linux Virtualization Manager

Administrator's Guide

F52197-07
July 2023

Oracle Linux Virtualization Manager Administrator's Guide,

F52197-07

Contents

1 Preface

Conventions 1-1
Documentation Accessibility 1-2
Access to Oracle Support for Accessibility 1-2
Diversity and Inclusion 1-2

2 Global Configuration

Administering User Accounts from the Administration Portal 2-1
- Adding VM Portal Permissions to a User 2-1
- Removing Users and Groups 2-1
- Assigning Permissions to Users and Groups 2-2
- Creating a Custom Role 2-2

Administering User and Group Accounts from the Command Line 2-3
- Creating a New User Account 2-3
- Setting the Password for a User Account 2-4
- Editing User Information 2-5
- Viewing User Information 2-5
- Removing a User 2-6
- Disabling User Accounts 2-6
- Creating Group Accounts 2-7
- Removing a Group Account 2-8
- Querying Users and Groups 2-9
- Managing Account Settings 2-12

Creating a Scheduling Policy 2-12

3 Administrative Tasks

Data Centers 3-1
- Creating a New Data Center 3-1

Clusters 3-1
- Creating a New Cluster 3-1

Hosts 3-3

Moving a Host to Maintenance Mode 3-3
Activating a Host from Maintenance Mode 3-4
Removing a Host 3-4
Networks 3-5
Customizing vNIC Profiles for Virtual Machines 3-5
Attaching and Configuring a Logical Network to a Host Network Interface 3-6
Storage 3-9
Using Local Storage on a KVM Host 3-9
Preparing Local Storage for a KVM Host 3-9
Configuring a KVM Host to Use Local Storage 3-9
Using NFS Storage 3-10
Preparing NFS Storage 3-10
Attaching an NFS Data Domain 3-11
Using iSCSI Multipathing 3-12
Configuring iSCSI Multipathing 3-12
Migrating a Logical Network to an iSCSI Bond 3-13
Adding an FC Data Domain 3-13
Deploying GlusterFS Storage 3-14
Deploying GlusterFS Storage Using Cockpit 3-14
Creating a GlusterFS Storage Domain Using the Manager 3-16
Detaching a Storage Domain from a Data Center 3-17
Virtual Machines 3-18
Live Editing a Virtual Machine 3-18
Migrating Virtual Machines between Hosts 3-20
Configuring Your Environment for Live Migration 3-20
Automatic Virtual Machine Migration 3-21
Setting Virtual Machine Migration Mode 3-21
Manually Migrate a Virtual Machine 3-22
Importing an Oracle Linux Template 3-22
Working with Virtual Machine Snapshots 3-23
Creating a Snapshot of a Virtual Machine 3-23
Restoring a Virtual Machine from a Snapshot 3-24
Creating a Virtual Machine from a Snapshot 3-25
Deleting a Snapshot 3-25
Encrypted Communication 3-26
Replacing the Oracle Linux Virtualization Manager Apache SSL Certificate 3-27
Event Notifications 3-28
Configuring Event Notification Services on the Engine 3-28
Creating Event Notifications in the Administration Portal 3-30
Canceling Event Notifications in the Administration Portal 3-31

Configuring the Engine to Send SNMP Traps 3-31

4 Deployment Optimization

Optimizing Clusters, Hosts and Virtual Machines 4-1
Configuring Memory and CPUs 4-1
Configuring Cluster Memory and CPUs 4-2
Changing Memory Overcommit Manually 4-3
Configuring Virtual Machine Memory and CPUs 4-3
Configuring a Highly Available Host 4-4
Configuring Power Management and Fencing on a Host 4-4
Preventing Host Fencing During Boot 4-6
Checking Fencing Parameters 4-7
Configuring a Highly Available Virtual Machine 4-7
Configuring High-Performance Virtual Machines 4-8
Creating a High Performance Virtual Machine 4-9
Configuring Huge Pages 4-9
Hot Plugging Devices on Virtual Machines 4-10
Hot Plugging vCPUs 4-10
Hot Plugging Virtual Memory 4-11

5 Upgrading Your Environment to 4.4

Before You Begin 5-1
Updating Engine or Self-Hosted Engine to 4.3 5-1
Upgrading Engine or Self-Hosted Engine 5-3
Prerequisites 5-3
Upgrading the Engine 5-4
Upgrading the Self-Hosted Engine 5-5
Migrating Hosts and Virtual Machines 5-7
Changing Data Center and Cluster Compatibility Versions After Upgrading 5-8
Compatibility Version Restrictions 5-8
Changing Cluster Compatibility Versions 5-9
Changing Data Center Compatibility Versions 5-10

6 Updating Engines and KVM Hosts within Versions

Updating the Engine 6-1
Updating the Self-Hosted Engine 6-2
Updating KVM Hosts 6-4

7 Disaster Recovery

Active-Active Disaster Recovery 7-1
- Network Considerations 7-2
- Storage Considerations 7-2
- Configuring a Standalone Engine Stretch Cluster Environment 7-2
- Configuring a Self-Hosted Engine Stretch Cluster Environment 7-3

Active-Passive Disaster Recovery 7-4
- Network Considerations 7-5
- Storage Considerations 7-5
- Creating the Ansible Playbooks 7-6
  - Simplifying Ansible Tasks Using the ovirt-dr Script 7-7
  - Generating the Mapping File Using an Ansible Playbook 7-7
  - Creating Failover and Failback Playbooks 7-8
- Executing a Failover 7-9
- Cleaning the Primary Site 7-9
- Executing a Failback 7-10
- Testing the Active-Passive Configuration 7-10
  - Discreet Failover Test 7-11
  - Discreet Failover and Failback Tests 7-11
  - Full Failover and Failback Tests 7-12
- Mapping File Attributes 7-13

1 Preface

Oracle Linux Virtualization Manager Release 4.4 is based on oVirt, which is a free, open-source virtualization solution. The product documentation comprises:

- **Release Notes** - A summary of the new features, changes, fixed bugs, and known issues in the Oracle Linux Virtualization Manager. It contains last-minute information, which might not be included in the main body of documentation.
- **Architecture and Planning Guide** - An architectural overview of Oracle Linux Virtualization Manager, prerequisites, and planning information for your environment.
- **Getting Started Guide** - How to install, configure, and get started with the Oracle Linux Virtualization Manager. The document includes an example scenario covering basic procedures for setting up the environment, such as adding hosts and storage, creating virtual machines, configuring networks, working with templates, and backup and restore tasks. In addition, there is information on upgrading your engine and hosts as well as deploying a self-hosted configuration.
- **Administration Guide** - Provides common administrative tasks for Oracle Linux Virtualization Manager and information on setting up users and groups, configuring high-availability, memory and CPUs, configuring and using event notifications, configuring vCPUs and virtual memory.

You can also refer to:

- REST API Guide, which you can access from the Welcome Dashboard or directly through its URL `https://`***`manager-fqdn`***`ovirt-engine/apidoc`.
- Upstream oVirt Documentation.

To access the Release 4.3.10 documentation, PDFs are available at:

- https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-releasenotes.pdf
- https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-gettingstarted.pdf
- https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-architecture-planning.pdf
- https://www.oracle.com/a/ocom/docs/olvm43/olvm-43-administration.pdf

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at https://www.oracle.com/corporate/accessibility/templates/t2-11535.html.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

2

Global Configuration

For Oracle Linux Virtualization Manager, global configuration options are set from the **Configure** dialog box. This dialog box is accessed by selecting **Administration** and then clicking **Configure**. From the **Configure** dialog box, you can configure a number of global resources for your virtualization environment, such as users, roles, system permissions, scheduling policies, instance types, and MAC address pools. You can also customize the way in which users interact with resources in the environment and configure options that can be applied to multiple clusters from a central location.

Administering User Accounts from the Administration Portal

The following tasks describe common user administration tasks that are performed in the Administration Portal.

Adding VM Portal Permissions to a User

Users must be created already before they can be added and assigned roles and permissions. For more information, refer to Administering User and Group Accounts from the Command Line.

In the following example procedure, a user is assigned the roles and permissions associated with the `UserRole`. This role gives the user the permission to log in to the VM Portal and to start creating virtual machines. The procedure also applies to group accounts.

1. Click **Administration** and then select **Configure**.

   The **Configure** dialog box opens with the **Roles** tab selected on the sidebar menu.
2. Click the **System Permissions** tab on the sidebar.
3. Click **Add**.

   The **Add System Permission to User** dialog box opens.
4. Select a profile from the **Search** drop-down list and click **Go**.
5. Select the check box next to the user or group account.
6. Under the **Role to Assign** drop-down list, select **UserRole**.
7. Click **OK**.
8. **(Optional)** Log in to the VM Portal to verify the permissions of the user account.

Removing Users and Groups

To use the Administration Portal to remove a user or group:

1. Go to **Administration** and then click **Users**.

   The **Users** pane opens.

2. On the **Users** pane, select either the **User** or **Group** tab to display the added users or groups.
3. Select the user or group to be removed.
4. Click **Remove**.

   The **Remove User(s)** dialog box opens.
5. Click **OK** to confirm the removal of the user.

   The user or group is removed and no longer appears on the **Users** pane.

Assigning Permissions to Users and Groups

Users and groups must be created already before they can be assigned roles and permissions. For more information, refer to Administering User and Group Accounts from the Command Line.

1. Go to **Administration** and then click **Users**.

   The **Users** pane opens.
2. Click **Add**.

   The **Add Users and Groups** dialog box opens.
3. Select either the **Users** option.
4. In the **Search** field, enter the name of the user or group to be added and then select **Go**.

   The dialog box updates to display the search results.
5. Select the check box next to the user or group to be added.
6. Click **Add**.

   The user or group is added and appears on the **Users** pane.
7. On the **Users** pane, select either the **User** or **Group** tab to display the added users or groups.
8. Display the detailed view for the user or group by clicking the name of the user under the **User Name** column or the name of the group under the **Group Name** column.
9. Click the **Permissions** tab.
10. Click **Add System Permissions**.

    The **Add System Permission to User** dialog box opens.
11. From the **Add System Permission to User** drop-down list, select the role to assign to the user.

Creating a Custom Role

If you require a role that is not available in the default set of roles provided by the Manager, you can create a custom role.

> **Note:**
>
> For more information about the default set of roles provided by the Manager, the *Administration Guide* in oVirt Documentation.

To create a custom role:

1. Click **Administration** and then select **Configure**.

   The **Configure** dialog box opens with the **Roles** tab selected on the sidebar menu. The **Roles** tab displays a list of administrator and user roles, and any custom roles that have been created.

2. Click **New**.

   The **New Role** dialog box opens.

3. For the **Name** and **Description** fields, enter an appropriate name and description for the role.

4. Under **Account Type**, select either **Admin** or **User**.

5. Under **Check Boxes to Allow Action**, select the appropriate objects whose permissions to assign to the user.

   Click **Expand All** to see the objects under each permissions group. Click **Collapse All** to collapse the list of objects under each of the permission group.

6. For each of the objects, select or clear the objects the actions to be permitted or denied for the custom role that is being created.

7. Click **OK** to create the custom role.

The custom role now appears on the **Roles** tab.

Administering User and Group Accounts from the Command Line

The following sections describe the common tasks that can be performed to administer user accounts using the `ovirt-aaa-jdbc-tool` command utility. This utility is used to manage user and group accounts on the internal domain. To view a list all available options for managing user and group accounts, run the `ovirt-aaa-jdbc-tool --help` command.

> **Note:**
>
> Changes made using `ovirt-aaa-jdbc-tool` command utility take effect immediately and do not require you to restart the Manager.

Creating a New User Account

The `ovirt-aaa-jdbc-tool user add` command is used to create user accounts.

To create a new user account:

1. Log in to the host that is running the Manager.
2. Create a new user account.

   ```
   ovirt-aaa-jdbc-tool user add username option
   ```

   To view a full list of options available for creating a user account, run the `ovirt-aaa-jdbc-tool user add --help` command.

   The following example shows how to create a new user account and add a first and last name to associate with the account.

   ```
   # ovirt-aaa-jdbc-tool user add test1 --attribute=firstName=John --
   attribute=lastName=Doe
   adding user test1...
   user added successfully
   Note: by default created user cannot log in. see:
   /usr/bin/ovirt-aaa-jdbc-tool user password-reset --help.
   ```

   > **Note:**
   >
   > After creating a new user account, you must set a password so that the user can log in. See Setting the Password for a User Account.

3. Add the newly created user in the Administration Portal and assign the group appropriate roles and permissions. See Assigning Permissions to Users and Groups.

Setting the Password for a User Account

The `ovirt-aaa-jdbc-tool password-reset` command is used to set (or reset) passwords for a user account.

To set (or reset) the password for a user account:

1. Log in to the host that is running the Manager.
2. Set (or reset) the password for a user account.

   ```
   ovirt-aaa-jdbc-tool user password-reset username --password-valid-to yyyy-MM-dd HH:mm:ssX
   ```

   > **Note:**
   >
   > You must set a value for the `--password-valid-to` option; otherwise the password expiry time defaults to the time of the last login.

   By default, the password policy for user accounts on the internal domain has the following restrictions:

   - A user password must be a minimum length of 6 characters.
   - When resetting a password, you cannot use the three previous passwords used for the user account.

For more information on the password policy and other default settings, run the `ovirt-aaa-jdbc-tool settings show` command.

The following example shows how to set a user password. In the example, `0800` stands for GMT minus 8 hours.

```
# ovirt-aaa-jdbc-tool user password-reset test1 --password-valid-to="2025-08-01
12:00:00-0800"
Password:
Reenter password:
updating user test1...
user updated successfully
```

Editing User Information

The `ovirt-aaa-jdbc-tool user edit` command is used to edit user information associated with a user account.

To edit user information:

1. Log in to the host that is running the Manager.
2. Edit the user account.

   `ovirt-aaa-jdbc-tool user edit` ***username option***

   To view a full list of options available for editing user information, run the `ovirt-aaa-jdbc-tool user edit --help` command.

   The following example shows to edit a user account by adding an email address to associate with this user.

   ```
   # ovirt-aaa-jdbc-tool user edit test1 --attribute=email=jdoe@example.com
   updating user test1...
   user updated successfully
   ```

Viewing User Information

The `ovirt-aaa-jdbc-tool user show` command is used to display user information.

To view detailed user information:

1. Log in to the host that is running the Manager.
2. Display information about a user.

   `ovirt-aaa-jdbc-tool user show` ***username***

   The following example shows how to view details about a user account.

   ```
   # ovirt-aaa-jdbc-tool user show test1
   -- User test1(e9e4b7d0-8ffd-45a3-b6ea-1f519238e766) --
   Namespace: *
   Name: test1
   ID: e9e4b7d0-8ffd-45a3-b6ea-1f519238e766
   Display Name:
   Email: jdoe@example.com
   First Name: John
   Last Name: Doe
   Department:
   Title:
   Description:
   ```

```
Account Disabled: false
Account Locked: false
Account Unlocked At: 1970-01-01 00:00:00Z
Account Valid From: 2019-08-26 18:59:16Z
Account Valid To: 2219-08-26 18:59:16Z
Account Without Password: false
Last successful Login At: 2019-08-27 15:21:20Z
Last unsuccessful Login At: 2019-08-27 15:20:59Z
Password Valid To: 2025-08-01 20:00:00Z
```

Removing a User

The `ovirt-aaa-jdbc-tool user delete` command is used to remove a user.

To remove a user account:

1. Log in to the host that is running the Manager.
2. Remove a user.

   ```
   ovirt-aaa-jdbc-tool user delete username
   ```

   The following example shows how to remove a user account.

   ```
   # ovirt-aaa-jdbc-tool user delete test1
   deleting user test1...
   user deleted successfully
   ```

Disabling User Accounts

You can disable users on the local domains, including the internal `admin` user created that is created when you run the `engine-setup` command.

> **Important:**
>
> Make sure you have at least one user in the environment with full administrative permissions before disabling the default internal administrative user account (`admin` user). The `SuperUser` role gives a user full administrative permissions.

To disable a user:

1. Log in to the host that is running the Manager.
2. Disable the user.

   ```
   ovirt-aaa-jdbc-tool user edit username --flag=+disabled
   ```

   The following example shows how to disable the `admin` user.

   ```
   # ovirt-aaa-jdbc-tool user edit admin --flag=+disabled
   updating user admin...
   user updated successfully
   ```

> **Note:**
>
> If for some reason you need to re-enable the internal `admin` user after it has been disabled, you can do so by running the `ovirt-aaa-jdbc-tool user edit admin --flag=-disabled` command.

Creating Group Accounts

The `ovirt-aaa-jdbc-tool` command is used to create and manage group accounts on the internal domain. Managing group accounts is similar to managing user accounts. To view all available options for managing group accounts, run the `ovirt-aaa-jdbc-tool group --help` command. Common examples are provided in this section.

Creating a Group
To create a group account:

1. Log in to the host that is running the Manager.
2. Create a new group account.

   `ovirt-aaa-jdbc-tool group add` ***group-name***

   > **Note:**
   >
   > Users must be created before they can be added to groups.

   The following examples shows how to add a new group account.

   ```
   # ovirt-aaa-jdbc-tool group add group1
   adding group group1...
   group added successfully
   ```

3. Add users to the group:

   `ovirt-aaa-jdbc-tool group-manage useradd` ***group-name*** `--user=` ***username***

   To view a full list of the options for adding or removing members to and from groups, run the `ovirt-aaa-jdbc-tool group-manage --help` command.

   The following example shows how to add users to a group.

   ```
   # ovirt-aaa-jdbc-tool group-manage useradd group1 --user test1
   updating user group1...
   user updated successfully
   ```

4. Display group account details.

   `ovirt-aaa-jdbc-tool group show` ***group-name***

   The following example shows how to display details about a group account.

   ```
   # ovirt-aaa-jdbc-tool group show group1
   -- Group group1(f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829) --
   ```

```
Namespace: *
Name: group1
ID: f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829
Display Name:
Description:
```

5. Add the newly created group in the Administration Portal and assign the group appropriate roles and permissions. See Assigning Permissions to Users and Groups.

   The users in the group inherit the roles and permissions of the group.

Creating Nested Groups

To create nested groups:

1. Log in to the host that is running the Manager.
2. Create the first group account.

   ```
   ovirt-aaa-jdbc-tool group add group1
   ```

   The following examples shows how to add a new group account.

   ```
   # ovirt-aaa-jdbc-tool group add group1
   adding group group1...
   group added successfully
   ```

3. Create the second group account.

   ```
   ovirt-aaa-jdbc-tool group add group2
   ```

   The following examples shows how to create the second group account.

   ```
   # ovirt-aaa-jdbc-tool group add group2
   adding group group2...
   group added successfully
   ```

4. Add the second group to the first group.

   ```
   ovirt-aaa-jdbc-tool group manage group add group1 --
   group=group2
   ```

   The following examples shows how to add the second group to the first group.

   ```
   # ovirt-aaa-jdbc-tool group-manage groupadd group1 --group=group2
   updating group group1...
   group updated successfully
   ```

5. Add the first group in the Administration Portal and assign the group appropriate roles and permissions. See Assigning Permissions to Users and Groups.

Removing a Group Account

To remove a group account:

1. Log in to the host that is running the Manager.
2. Remove a group account.

   ```
   ovirt-aaa-jdbc-tool group delete group-name
   ```

   The following example shows how to remove a group account.

```
# ovirt-aaa-jdbc-tool group delete group3
deleting group group3...
group deleted successfully
```

Querying Users and Groups

The `ovirt-aaa-jdbc-tool query` command is used to query user and group information. To view a full list of options available for querying users and groups, run the `ovirt-aaa-jdbc-tool query --help` command.

Listing All User or Group Account Details
To list all account information:

1. Log in to the host that is running the Manager.
2. Display account details.
   - List all user account details.

     ```
     ovirt-aaa-jdbc-tool query --what=user
     ```

     The following example shows sample output from the `ovirt-aaa-jdbc-tool query --what=user` command.

     ```
     # ovirt-aaa-jdbc-tool query --what=user
     -- User test2(35e8b35e-2320-45da-b59e-1076b521d13f) --
     Namespace: *
     Name: test2
     ID: 35e8b35e-2320-45da-b59e-1076b521d13f
     Display Name:
     Email:
     First Name: Jane
     Last Name: Doe
     Department:
     Title:
     Description:
     Account Disabled: false
     Account Locked: false
     Account Unlocked At: 1970-01-01 00:00:00Z
     Account Valid From: 2019-09-06 16:51:32Z
     Account Valid To: 2219-09-06 16:51:32Z
     Account Without Password: false
     Last successful Login At: 2019-09-06 17:12:08Z
     Last unsuccessful Login At: 1970-01-01 00:00:00Z
     Password Valid To: 2025-08-01 20:00:00Z
     -- User admin(89559d7f-3b48-420b-bd4d-2790122c199b) --
     Namespace: *
     Name: admin
     ID: 89559d7f-3b48-420b-bd4d-2790122c199b
     Display Name:
     Email:
     First Name: admin
     Last Name:
     Department:
     Title:
     Description:
     ```

```
Account Disabled: false
Account Locked: false
Account Unlocked At: 2019-03-07 11:09:07Z
Account Valid From: 2019-01-24 21:18:11Z
Account Valid To: 2219-01-24 21:18:11Z
Account Without Password: false
Last successful Login At: 2019-09-06 18:10:11Z
Last unsuccessful Login At: 2019-09-06 18:09:36Z
Password Valid To: 2025-08-01 20:00:00Z
-- User test1(e75956a8-6ebf-49d7-94fa-504afbfb96ad) --
Namespace: *
Name: test1
ID: e75956a8-6ebf-49d7-94fa-504afbfb96ad
Display Name:
Email: jdoe@example.com
First Name: John
Last Name: Doe
Department:
Title:
Description:
Account Disabled: false
Account Locked: false
Account Unlocked At: 1970-01-01 00:00:00Z
Account Valid From: 2019-08-29 18:15:20Z
Account Valid To: 2219-08-29 18:15:20Z
Account Without Password: false
Last successful Login At: 1970-01-01 00:00:00Z
Last unsuccessful Login At: 1970-01-01 00:00:00Z
Password Valid To: 2025-08-01 20:00:00Z
```

- List all group account details. `ovirt-aaa-jdbc-tool query --what=group`

  The following example shows sample output from the `ovirt-aaa-jdbc-tool query --what=group` command.

```
# ovirt-aaa-jdbc-tool query --what=group
-- Group group2(d6e0b913-d038-413a-b732-bc0c33ea1ed4) --
Namespace: *
Name: group2
ID: d6e0b913-d038-413a-b732-bc0c33ea1ed4
Display Name:
Description:
-- Group group1-1(e43ba527-6256-4c29-bd7a-0fb08b990b72) --
Namespace: *
Name: group1-1
ID: e43ba527-6256-4c29-bd7a-0fb08b990b72
Display Name:
Description:
-- Group group1(f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829) --
Namespace: *
Name: group1
ID: f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829
Display Name:
Description:
```

Listing Filtered Account Details

To apply filters when listing account information:

1. Log in to the host that is running the Manager.

2. Filter account details using the `--pattern` keyword.

   - List user account based on a pattern.

     ```
     ovirt-aaa-jdbc-tool query --what=user --
     pattern=attribute =value
     ```

     The following example shows how to filter the output of the `ovirt-aaa-jdbc-tool query` command to display only user account details that start with the character `J`.

     ```
     # ovirt-aaa-jdbc-tool query --what=user --pattern="firstName=J*"
     -- User test1(e75956a8-6ebf-49d7-94fa-504afbfb96ad) --
     Namespace: *
     Name: test1
     ID: e75956a8-6ebf-49d7-94fa-504afbfb96ad
     Display Name:
     Email: jdoe@example.com
     First Name: John
     Last Name: Doe
     Department:
     Title:
     Description:
     Account Disabled: false
     Account Locked: false
     Account Unlocked At: 1970-01-01 00:00:00Z
     Account Valid From: 2019-08-29 18:15:20Z
     Account Valid To: 2219-08-29 18:15:20Z
     Account Without Password: false
     Last successful Login At: 1970-01-01 00:00:00Z
     Last unsuccessful Login At: 1970-01-01 00:00:00Z
     Password Valid To: 2025-08-01 20:00:00Z
     -- User test2(35e8b35e-2320-45da-b59e-1076b521d13f) --
     Namespace: *
     Name: test2
     ID: 35e8b35e-2320-45da-b59e-1076b521d13f
     Display Name:
     Email:
     First Name: Jane
     Last Name: Doe
     Department:
     Title:
     Description:
     Account Disabled: false
     Account Locked: false
     Account Unlocked At: 1970-01-01 00:00:00Z
     Account Valid From: 2019-09-06 16:51:32Z
     Account Valid To: 2219-09-06 16:51:32Z
     Account Without Password: false
     Last successful Login At: 2019-09-06 17:12:08Z
     Last unsuccessful Login At: 1970-01-01 00:00:00Z
     Password Valid To: 2025-08-01 20:00:00Z
     ```

   - List groups based on a pattern.

     ```
     ovirt-aaa-jdbc-tool-query --what=group --pattern=attribute =value
     ```

The following example shows how to filter the output of the `ovirt-aaa-jdbc-tool query` command to display only group account details that match the description `documentation-group`.

```
# ovirt-aaa-jdbc-tool query --what=group --
pattern="description=documentation-group"
-- Group group1(f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829) --
Namespace: *
Name: group1
ID: f23ca27c-1d6a-4f6e-8c3e-1e03e8e56829
Display Name:
Description: documentation-group
```

Managing Account Settings

The `ovirt-aaa-jdbc-tool settings` command is used to change the default account settings.

To change the default account settings:

1. Log in to the host that is running the Manager.
2. **(Optional)** Display all the settings that are available.

   ```
   ovirt-aaa-jdbc-tool setting show
   ```

3. Change the desired settings.

   ```
   ovirt-aaa-jdbc-tool setting set --name=setting-name --value=value
   ```

Creating a Scheduling Policy

If you require a scheduling policy that is not available in the default set provided by the Manager, you can create a custom scheduling policy.

> **Note:**
>
> To learn about the default scheduling policies and for conceptual information, see *High Availability and Optimization* in the Oracle Linux Virtualization Manager: Architecture and Planning Guide. For detailed information on scheduling policies and other policy types, refer to the *Administration Guide* in oVirt Documentation.

To create a scheduling policy:

1. Click **Administration** and then select **Configure**.

   The **Configure** dialog box opens.

2. Click **Scheduling Policies**.
3. Click **New**.

   The **New Scheduling Policy** dialog box opens.

4. For the **Name** and **Description** fields, enter an appropriate name and description for the policy.
5. In **Filter Modules**:

- Drag and drop modules from the **Disabled Filters** section to the **Enabled Filters** section.
- Optionally, set the module priority by right-clicking on a filter module name, hover over **Position** and then select **First** or **Last**.

6. In **Weights Modules**:
   - Drag and drop modules from the **Disabled Weights** section to the **Enabled Weights & Factors** section.
   - Optionally, use the plus (+) and minus (-) to increase or decrease module weight.
7. In **Load Balancer**:
   - Select the load balancing policy.
   - Select a load balancing property and then enter a property value.
   - Optionally, use the plus (+) and minus (-) to add or remove additional properties.
8. Click **OK** to create the scheduling policy.

3
Administrative Tasks

The following are common Oracle Linux Virtualization Manager administration tasks. For conceptual information about these topics, refer to the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

For additional administrative tasks, see the oVirt Documentation.

Databases

Oracle Linux Virtualization Manager creates a PostgreSQL database called `engine` during installation. Optionally, you might have the `ovirt_engine_history` database if you installed the data warehouse.

Occasionally, you should perform maintenance on these databases. Running the Engine Vacuum tool updates tables and removes dead rows, allowing disk space to be reused.

Reclaiming Database Storage

To reclaim database storage using the Engine Vacuum tool, you must log into the engine host as the **root** user and provide the administration credentials for the oVirt environment.

1. Check the current database size:

   ```
   # /usr/share/ovirt-engine/dbscripts/engine-psql.sh -c "SELECT datname as db_name,
   pg_size_pretty(pg_database_size(datname)) as db_usage FROM pg_database"
   ```

2. Vacuum the Engine database.

   a. Stop the ovirt-engine, ovirt-engine-dwhd, and grafana-server services:

      ```
      # systemctl stop ovirt-engine ovirt-engine-dwhd grafana-server
      ```

   b. Backup the engine database:

      ```
      # grep 'ENGINE_DB_PASSWORD=' /etc/ovirt-engine/engine.conf.d/10-setup-
      database.conf

      # PGPASSWORD=your-engine-db-pw  /usr/bin/pg_dump \
       -E UTF8 \
       --disable-dollar-quoting \
       --disable-triggers \
       -U engine \
       -h localhost \
       -p 5432 \
       --format=custom \
       --file=/var/lib/ovirt-engine/backups/engine-$(date +%Y%m%d%H%M%S).$$.dump
      engine
      ```

   c. Vacuum the engine database:

      ```
      /usr/share/ovirt-engine/bin/engine-vacuum.sh -f -v
      ```

   d. Start the ovirt-engine, ovirt-engine-dwhd, and grafana-server services:

```
# systemctl start ovirt-engine ovirt-engine-dwhd grafana-server
```

3. Vacuum the data warehouse (`ovirt_engine_history`) database.

   a. Stop the ovirt-engine, ovirt-engine-dwhd, and grafana-server services:

   ```
   # systemctl stop ovirt-engine ovirt-engine-dwhd grafana-server
   ```

   b. Backup the `ovirt_engine_history` database:

   ```
   # grep 'DWH_DB_PASSWORD=' /etc/ovirt-engine/engine.conf.d/10-setup-dwh-
   database.conf

   # PGPASSWORD=your-datawarehouse-db-pw  /usr/bin/pg_dump \
    -E UTF8 \
    --disable-dollar-quoting \
    --disable-triggers \
    -U ovirt_engine_history \
    -h localhost \
    -p 5432 \
    --format=custom \
    --file=/var/lib/ovirt-engine-dwh/backups/dwh-$(date +
   %Y%m%d%H%M%S).$$.dump ovirt_engine_history
   ```

   c. Vacuum the `ovirt_engine_history` database:

   ```
   # /usr/share/ovirt-engine-dwh/bin/dwh-vacuum.sh -f -v
   ```

   d. Start the ovirt-engine, ovirt-engine-dwhd, and grafana-server services:

   ```
   # systemctl start ovirt-engine ovirt-engine-dwhd grafana-server
   ```

4. Check the post-vacuum database size:

   ```
   # /usr/share/ovirt-engine/dbscripts/engine-psql.sh -c "SELECT datname as
   db_name, pg_size_pretty(pg_database_size(datname)) as db_usage FROM
   pg_database"
   ```

Data Centers

Oracle Linux Virtualization Manager creates a default data center during installation. You can configure the default data center, or set up new appropriately named data centers.

A data center requires a functioning cluster, host, and storage domain to operate in your virtualization environment.

Creating a New Data Center

1. Go to **Compute** and then select **Data Centers**.

   The **Data Centers** pane opens.

2. Click **New**.
3. Enter a **Name** and optional **Description**.
4. Select the storage **Type**, **Compatibility Version**, and **Quota Mode** of the data center from the respective drop-down menus.
5. Click **OK** to create the data center.

3

The new data center is added to the virtualization environment and the **Data Center - Guide Me** menu opens to guide you through the entities that are required be configured for the data center to operate.

The new data center remains in `Uninitialized` state until a cluster, host, and storage domain are configured for it.

You can postpone the configuration of these entities by clicking the **Configure Later** button. You can resume the configuration of these entities by selecting the respective data center and clicking **More Actions** and then choosing **Guide Me** from the drop-down menu.

Clusters

Oracle Linux Virtualization Manager creates a default cluster in the default data center during installation. You can configure the default cluster, or set up new appropriately named clusters.

Creating a New Cluster

1. Go to **Compute** and then select **Clusters**.

   The **Clusters** pane opens.

2. Click **New**.

   The **New Cluster** dialog box opens with the **General** tab selected on the sidebar.

3. From the **Data Center** drop-down list, choose the Data Center to associate with the cluster.
4. For the **Name** field, enter an appropriate name for the data center.
5. For the **Description** field, enter an appropriate description for the cluster.
6. From the **Management Network** drop-down list, choose the network for which to assign the management network role.
7. From the **CPU Architecture** and **CPU Type** drop-down lists, choose the CPU processor family and minimum CPU processor that match the hosts that are to be added to the cluster.

   For both Intel and AMD CPU types, the listed CPU models are in logical order from the oldest to the newest. If your cluster includes hosts with different CPU models, choose the oldest CPU model from the list to ensure that all hosts can operate in the cluster.

8. From the **Compatibility Version** drop-down list, choose the compatibility version of the cluster.

   > **Note:**
   >
   > For more information on compatibility versions, see Changing Data Center and Cluster Compatibility Versions After Upgrading.

9. From the **Switch Type** drop-down list, choose the type of switch to be used for the cluster.

   By default, **Linux Bridge** is selected from the drop-down list.

10. From the **Firewall Type** drop-down list, choose the firewall type for hosts in the cluster.

    The firewall types available are either **iptables** or **firewalld**. By default, the **firewalld** option is selected from the drop-down list.

11. The **Enable Virt Service** check box is selected by default. This check box designates that the cluster is to be populated with virtual machine hosts.

12. **(Optional)** Review the other tabs to further configure your cluster:

    a. Click the **Optimization** tab on the sidebar to select the memory page sharing threshold for the cluster, and optionally enable CPU thread handling and memory ballooning on the hosts in the cluster. See Deployment Optimization.

    b. Click the **Migration Policy** tab on the sidebar menu to define the virtual machine migration policy for the cluster.

    c. Click the **Scheduling Policy** tab on the sidebar to optionally configure a scheduling policy, configure scheduler optimization settings, enable trusted service for hosts in the cluster, enable HA Reservation, and add a custom serial number policy.

    d. Click the **Fencing policy** tab on the sidebar to enable or disable fencing in the cluster, and select fencing options.

    e. Click the **MAC Address Pool** tab on the sidebar to specify a MAC address pool other than the default pool for the cluster.

13. Click **OK** to create the data center.

    The cluster is added to the virtualization environment and the **Cluster - Guide Me** menu opens to guide you through the entities that are required to be configured for the cluster to operate.

    You can postpone the configuration of these entities by clicking the **Configure Later** button. You can resume the configuration of these entities by selecting the respective cluster and clicking **More Actions** and then choosing **Guide Me** from the drop-down menu.

Hosts

Hosts, also known as hypervisors, are the physical servers on which virtual machines run. Full virtualization is provided by using a loadable Linux kernel module called Kernel-based Virtual Machine (KVM). KVM can concurrently host multiple virtual machines. Virtual machines run as individual Linux processes and threads on the host machine and are managed remotely by the engine.

Moving a Host to Maintenance Mode

Place a host into maintenance mode when performing common maintenance tasks, including network configuration and deployment of software updates, or before any event that might cause VDSM to stop working properly, such as a reboot, or issues with networking or storage.

When you place a host into maintenance mode the engine attempts to migrate all running virtual machines to alternative hosts. The standard prerequisites for live migration apply, in particular there must be at least one active host in the cluster with capacity to run the migrated virtual machines.

> **Note:**
>
> Virtual machines that are pinned to the host and cannot be migrated are shut down. You can check which virtual machines are pinned to the host by clicking **Pinned to Host** in the **Virtual Machines** tab of the host's details view.

1. Click **Compute** and then select **Hosts**.
2. Select the desired host.
3. Click **Management** and then select **Maintenance**.
4. Optionally, enter a **Reason** for moving the host into maintenance mode, which will appear in the logs and when the host is activated again. Then, click **OK**.

   The host maintenance Reason field will only appear if it has been enabled in the cluster settings.
5. Optionally, select the required options for hosts that support Gluster.

   Select the **Ignore Gluster Quorum** and **Self-Heal Validations** option to avoid the default checks. By default, the Engine checks that the Gluster quorum is not lost when the host is moved to maintenance mode. The Engine also checks that there is no self-heal activity that will be affected by moving the host to maintenance mode. If the Gluster quorum will be lost or if there is self-heal activity that will be affected, the Engine prevents the host from being placed into maintenance mode. Only use this option if there is no other way to place the host in maintenance mode.

   Select the **Stop Gluster Service** option to stop all Gluster services while moving the host to maintenance mode.

   These fields will only appear in the host maintenance window when the selected host supports Gluster.

6. Click **OK** to initiate maintenance mode.
7. All running virtual machines are migrated to alternative hosts. If the host is the Storage Pool Manager (SPM), the SPM role is migrated to another host. The Status field of the host changes to `Preparing for Maintenance`, and finally `Maintenance` when the operation completes successfully. VDSM does not stop while the host is in maintenance mode.

> **Note:**
>
> If migration fails on any virtual machine, click **Management** and then select **Activate** on the host to stop the operation placing it into maintenance mode, then click **Cancel Migration** on the virtual machine to stop the migration.

Activating a Host from Maintenance Mode

You must activate a host from maintenance mode before using it.

1. Click **Compute** and then select **Hosts**.
2. Select the host.
3. Click **Management** and then select **Activate**.
4. When complete, the host status changes to `Unassigned`, and finally `Up`.

   Virtual machines can now run on the host. Virtual machines that were migrated off the host when it was placed into maintenance mode are not automatically migrated back to the host when it is activated, but can be migrated manually. If the host was the Storage Pool Manager (SPM) before being placed into maintenance mode, the SPM role does not return automatically when the host is activated.

Removing a Host

You may need to remove a host from the Oracle Linux Virtualization Manager environment when upgrading to a newer version.

1. Click **Compute** and then select **Hosts** and select the host.
2. Select the host.
3. Click **Management** and then select **Maintenance**.
4. Once the host is in maintenance mode, click **Remove**.

   Select the **Force Remove** check box if the host is part of a Gluster Storage cluster and has volume bricks on it, or if the host is non-responsive.
5. Click **OK**.

Networks

With Oracle Linux Virtualization Manager, you can create custom vNICs for your virtual machines.

Customizing vNIC Profiles for Virtual Machines

To customize vNICs for virtual machines:

1. Go to **Compute** and then click **Virtual Machines**.

   The **Virtual Machines** pane opens with the list of virtual machines that have been created.

2. Under the **Name** column, select the virtual machine for which to add the virtual machine network.

   The **General** tab opens with details about the virtual machine.

3. Click the **Network Interfaces** tab.

   The **Network Interfaces** tab opens with the available network interface to be used for the network.

4. Highlight the network interface by clicking the row for the respective interface and then click **Edit** on the right side above the interface listing.

   The **Edit Network Interface** dialog box opens.

5. In the **Edit Network Interface** dialog box, update the following fields:

   a. From the **Profile** drop-down list, select the network to be added to the virtual machine.

   b. Click the **Custom MAC address** check box, and then enter or update the MAC address that is allocated for this virtual machine in the text entry field.

6. Click **OK** when you are finished editing the network interface settings for the virtual machine.

7. Go to **Compute** and then click **Virtual Machines**.

   The **Virtual Machines** pane opens.

   > **Important:**
   >
   > Since virtual machines can start on any host in a data center/cluster, all hosts must have the customized VM network assigned to one of its NICs. Ensure that you assign this customized VM network to each host before booting the virtual machine. For more information, see Assigning a Virtual Machine Network to a KVM Host in the Oracle Linux Virtualization Manager: Getting Started Guide.

8. Highlight the virtual machine where you added the network and then click **Run** to boot the virtual machine.

   The red down arrow icon to the left of the virtual machine turns green and the **Status** column displays `UP` when the virtual machine is up and running on the network.

Attaching and Configuring a Logical Network to a Host Network Interface

You can change the settings of physical host network interfaces, move the management network from one physical host network interface to another, and assign logical networks to physical host network interfaces.

Before you begin the steps below, keep in mind the following:

- To change the IP address of a host, you must remove the host and then re-add it.
- To change the VLAN settings of a host, see Editing a Host's VLAN Settings in oVirt Documentation.
- You cannot assign logical networks offered by external providers to physical host network interfaces; such networks are dynamically assigned to hosts as they are required by virtual machines.
- If a switch has been configured to provide Link Layer Discovery Protocol (LLDP) information, you can hover your cursor over a physical network interface to view the switch port's current configuration.

> **Note:**
>
> Before assigning logical networks, check the configuration. To help detect to which ports and on which switch the host's interfaces are patched, review **Port Description (TLV type 4)** and **System Name (TLV type 5)**. The **Port VLAN ID** shows the native VLAN ID configured on the switch port for untagged ethernet frames. All VLANs configured on the switch port are shown as **VLAN Name** and **VLAN ID** combinations.

To edit host network interfaces and assign logical networks:

1. Click **Compute Hosts**.
2. Click the host's name. This opens the details view.
3. Click the **Network Interfaces** tab.
4. Click **Setup Host Networks**.
5. Optionally, hover your cursor over host network interface to view configuration information provided by the switch.
6. Attach a logical network to a physical host network interface by selecting and dragging the logical network into the **Assigned Logical Networks** area next to the physical host network interface.

   If a NIC is connected to more than one logical network, only one of the networks can be non-VLAN. All the other logical networks must be unique VLANs.

7. Configure the logical network.

   a. Hover your cursor over an assigned logical network and click the pencil icon. This opens the **Edit Management Network** window.

   b. Configure IPv4 or IPv6:

      - From the **IPv4** tab, set the **Boot Protocol**. If you select **Static**, enter the **IP**, **Netmask / Routing Prefix**, and the **Gateway**.
      - From the **IPv6** tab:
        - Set the**Boot Protocol** to **Static**.
        - For Routing Prefix, enter the length of the prefix using a forward slash and decimals. For example: /48 IP:
        - In the IP field, enter the complete IPv6 address of the host network interface. For example: 2001:db8::1:0:0:6
        - In the Gateway field, enter the source router's IPv6 address. For example: 2001:db8::1:0:0:1

      Note:

      If you change the host's management network IP address, you must reinstall the host for the new IP address to be configured.

      Each logical network can have a separate gateway defined from the management network gateway. This ensures traffic that arrives on the logical network is forwarded using the logical network's gateway instead of the default gateway used by the management network.

      Set **all** hosts in a cluster to use the same IP stack for their management network; either IPv4 or IPv6 only.

   c. To configure a network bridge, click the **Custom Properties** tab, select **bridge_opts** from the list, and enter a valid key and value with the syntax of `key=value`.

      The following are valid keys with example values:

      ```
      forward_delay=1500
      group_addr=1:80:c2:0:0:0
      group_fwd_mask=0x0
      hash_max=512
      hello_time=200
      max_age=2000
      multicast_last_member_count=2
      multicast_last_member_interval=100
      multicast_membership_interval=26000
      multicast_querier=0
      multicast_querier_interval=25500
      multicast_query_interval=13000
      multicast_query_response_interval=1000
      multicast_query_use_ifaddr=0
      multicast_router=1
      multicast_snooping=1
      multicast_startup_query_count=2
      multicast_startup_query_interval=3125
      ```

      Separate multiple entries with a whitespace character.

d. To configure ethernet properties, click the **Custom Properties** tab, select **ethtool_opts** from the list, and enter a valid value using the format of the command-line arguments of `ethtool`. For example:

```
--coalesce em1 rx-usecs 14 sample-interval 3 --offload em2 rx on lro on
tso off \
--change em1 speed 1000 duplex half
```

You can use wildcard to apply the same option to all of a network's interfaces, for example:

```
--coalesce * rx-usecs 14 sample-interval 3
```

The `ethtool_opts` option is not available by default; you need to add it using the engine configuration tool. To view `ethtool` properties, from a command line type `man ethtool` to open the man page. For more information, see How to Set Up oVirt Engine to Use Ethtool in oVirt Documentation.

e. To configure Fibre Channel over Ethernet (FCoE), click the **Custom Properties** tab, select **fcoe** from the list, and enter `enable=yes`. Separate multiple entries with a whitespace character.

The `fcoe` option is not available by default; you need to add it using the engine configuration tool. For more information, see How to Set Up oVirt Engine to Use FCoE in oVirt Documentation.

f. To change the default network used by the host from the management network (ovirtmgmt) to a non-management network, configure the non-management network's default route. For more information, see Configuring a Non-Management Logical Network as the Default Route in oVirt Documentation.

g. If your logical network definition is not synchronized with the network configuration on the host, select the **Sync network** check box. For more information about unsynchronized hosts and how to synchronize them, see Synchronizing Host Networks in oVirt Documentation.

8. To check network connectivity, select the **Verify connectivity between Host and Engine** check box.

> **Note:**
>
> The host must be in maintenance mode.

9. Click **OK**.

> **Note:**
>
> If not all network interface cards for the host are displayed, click **Management** and then **Refresh Capabilities** to update the list of network interface cards available for that host.

Storage

Oracle Linux Virtualization Manager uses a centralized storage system for virtual machine disk images, ISO files, and snapshots. You can use Network File System (NFS), Internet Small Computer System Interface (iSCSI), or Fibre Channel Protocol (FCP) storage. You can also configure local storage attached directly to hosts.

This following administration tasks cover preparing and adding local, NFS, and FCP storage. For information about attaching iSCSI storage, see Attaching an iSCSI Data Domain in the Oracle Linux Virtualization Manager: Getting Started Guide.

Using Local Storage on a KVM Host

Before you begin, ensure the following prerequisites have been met:

- You have allocated disk space for local storage. You can allocate an entire physical disk on the host or you can use a portion of the disk.
- You have created a filesystem on the block device path to be used for local storage. Local storage should always be defined on a file system that is separate from the root directory (`/root`).

Preparing Local Storage for a KVM Host

To prepare local storage for a KVM host:

1. Create the directory to be used for the local storage on the host.

   ```
   Copy# mkdir -p /data/images
   ```

2. Ensure that the directory has permissions that allows read-write access to the `vdsm` user (UID 36) and `kvm` group (GID 36).

   ```
   Copy# chown 36:36 /data /data/images
   # chmod 0755 /data /data/images
   ```

   The local storage can now be added to your virtualization environment.

Configuring a KVM Host to Use Local Storage

When you configure a KVM host to use local storage, it is automatically added to a new data center and cluster that can contain no other hosts. With local storage, features, such as live migration, fencing, and scheduling, are not available.

To configure a KVM host to use local storage:

1. Go to **Compute**, and then click **Hosts**.

   The **Hosts** pane opens.

2. Highlight the host on which to add the local storage domain.
3. Click **Management** and then select **Maintenance** from the drop-down list.

   The **Status** column for the host displays `Maintenance` when the host has successfully entered into Maintenance mode.

4. After the host is in `Maintenance` mode, click **Management** and then select **Configure Local Storage** from the drop-down list.

   The **Configure Local Storage** pane opens with the **General** tab selected.

5. Click **Edit** next to the **Data Center**, **Cluster**, and **Storage** fields to configure and name the local storage domain.

6. In the **Set the path to your local storage** text input field, specify the path to your local storage domain.

   For more information, refer to Preparing Local Storage for a KVM Host.

7. Click **OK** to add the local storage domain.

   When the virtualization environment is finished adding the local storage, the new data center, cluster, and storage created for the local storage appears on the **Data Center**, **Clusters**, and **Storage** panes, respectively.

   You can click **Tasks** to monitor the various processing steps that are completed to add the local storage to the host.

   You can also verify the successful addition of the local storage domain by viewing the `/var/log/ovirt-engine/engine.log` file.

Using NFS Storage

Before preparing the NFS share, ensure your environment meets the following conditions:

- Ensure that the Manager and KVM host installation are running the Oracle Linux 8.5 or later in an environment with two or more servers where one acts as the Manager host and the other servers act as KVM hosts.

  The installation creates a `vdsm:kvm` (`36:36`) user and group in the `/etc/passwd` and `/etc/group` directories, respectively.

  ```
  # cat /etc/passwd | grep vdsm
  vdsm:x:36:36:Node Virtualization Manager:/:/sbin/nologin

  # cat /etc/group | grep kvm
  kvm:x:36:qemu,sanlock
  ```

- An Oracle Linux NFS File server that is reachable by your virtualization environment.

Preparing NFS Storage

To prepare NFS storage:

1. On a Linux fileserver that has access to the virtualization environment, create a directory that is to be used for the data domain.

   ```
   # mkdir -p /nfs/olv_ovirt/data
   ```

2. Set the required permissions on the new directory to allow read-write access to the `vdsm` user (UID 36) and `kvm`group (GID 36).

   ```
   # chown -R 36:36 /nfs/olv_ovirt
   # chmod -R 0755 /nfs/olv_ovirt
   ```

3. Add an entry for the newly created NFS share in the `/etc/exports` directory on the NFS file server that uses the following format: ***full-path-of-share-created*** `*(rw,sync,no_subtree_check,all_squash,anonuid=36,anongid=36)`.

   For example:

   ```
   # vi /etc/exports
   # added the following entry
   /nfs/olv_ovirt/data *(rw,sync,no_subtree_check,all_squash,anonuid=36,anongid=36)
   ```

   Verify that the entry has been added.

   ```
   # grep "/nfs/olv_ovirt/data" /etc/exports
   /nfs/ol_ovirt/data *(rw,sync,no_subtree_check,all_squash,anonuid=36,anongid=36)
   ```

   If you do not want to export the domain share to all servers on the network (denoted by the * before the left parenthesis), you can specify each individual host in your virtualization environment by using the following format: `/nfs/ol_ovirt/data` ***hostname-or-ip-address*** `(rw,sync,no_subtree_check,all_squash,anonuid=36,anongid=36)`.

   For example:

   ```
   /nfs/olv_ovirt/data
   hostname
   (rw,sync,no_subtree_check,all_squash,anonuid=36,anongid=36)
   ```

4. Export the NFS share.

   ```
   # exportfs -rv
   ```

5. Confirm that the added export is available to Oracle Linux Virtualization Manager hosts by using the following `showmount` commands on the NFS File Server.

   ```
   # showmount -e | grep pathname-to-domain-share-added
   # showmount | grep ip-address-of-host
   ```

Attaching an NFS Data Domain

To attach an NFS data domain:

1. Go to **Storage** and then click **Domains**.

   The **Storage Domains** pane opens.

2. Click **New Domain**.

   The **New Domain** dialog box opens.

3. From the **Data Center** drop-down list, select the Data Center for which to attach the data domain.

4. From the **Domain Function** drop-down list, select **Data**. By default, the **Data** option is selected in the drop-down list.

5. From the **Storage Type** drop-down list, select **NFS**. By default, the **NFS** option is selected in the drop-down list.

   When **NFS** is selected for the Storage Type, the options that are applicable to this storage types (such as the required **Export Path** option) are displayed in the **New Domain** dialog box.

6. For the **Host to Use** drop-down list, select the host for which to attach the data domain.

7. For the **Export Path** option, enter the remote path to the NFS export to be used as the storage data domain in the text input field.

The **Export Path** option must be entered in one of the following formats: **IP:/pathname** or **FQDN:/pathname** (for example, `server.example.com:/nfs/olv_ovirt/data`).

The **/pathname** that you enter must be the same as the path that you created on the NFS file server for the data domain in Preparing NFS Storage.

8. Click **OK** to attach the NFS storage data domain.

   For information about uploading images to the data domain, see Uploading Images to a Data Domain in the Oracle Linux Virtualization Manager: Getting Started Guide.

Using iSCSI Multipathing

Multiple network paths between hosts and iSCSI storage prevent host downtime caused by network path failure. iSCSI multipathing enables you to create and manage groups of logical networks and iSCSI storage connections.

The Engine connects each host in a data center to each storage target using the NICs or VLANs that are assigned to the logical networks in the iSCSI bond.

You can create an iSCSI bond with multiple targets and logical networks for redundancy.

Before you can configure iSCSI multipathing, ensure you have the following:

- One or more iSCSI targets. For more information, see Attaching an iSCSI Data Domain in the Oracle Linux Virtualization Manager: Getting Started Guide.
- One or more logical networks that are:
  - Not defined as Required or VM Network. For more information, see Migrating a Logical Network to an iSCSI Bond.
  - Assigned to a host interface.
  - Assigned a static IP address in the same VLAN and subnet as the other logical networks in the iSCSI bond.

  For more information, see Creating a Logical Network in the Oracle Linux Virtualization Manager: Getting Started Guide.

Configuring iSCSI Multipathing

To configure iSCSI multipathing:

1. Click **Compute Data Centers**.
2. Click the data center name.
3. In the **iSCSI Multipathing** tab, click **Add**.
4. In the **Add iSCSI Bond** window, enter a **Name** and optionally add a **Description**.
5. Select a logical network from **Logical Networks** and a storage domain from **Storage Targets**. You must select all paths to the same target.
6. Click **OK**.

The hosts in the data center are connected to the iSCSI targets through the logical networks in the iSCSI bond.

Migrating a Logical Network to an iSCSI Bond

If you have a logical network that you created for iSCSI traffic and configured on top of an existing network bond, you can migrate the logical network to an iSCSI bond on the same subnet without disruption or downtime.

To migrate a logical network to an iSCSI bond:

1. Modify the current logical network so that it is not required.
    a. Click **Compute** and then click **Clusters**.
    b. Click the cluster name.
    c. In the **Logical Networks** tab of the cluster detail page, select the current logical network (`net-1`) and click **Manage Networks**.
    d. Clear the **Require** check box and click **OK**.
2. Create a new logical network that is not Required and not VM network.
    a. Click **Add Network**. This opens the **New Logical Network** window.
    b. In the **General** tab, enter the **Name** (`net-2`) and clear the **VM network** check box.
    c. In the **Cluster** tab, clear the **Require** check box and click **OK**.
3. Remove the current network bond and reassign the logical networks.
    a. Click **Compute** and then click **Hosts**.
    b. Click the host name.
    c. In the **Network Interfaces** tab of the host detail page, click **Setup Host Networks**.
    d. Drag `net-1` to the right to unassign it.
    e. Drag the current bond to the right to remove it.
    f. Drag `net-1` and `net-2` to the left to assign them to physical interfaces.
    g. To edit the `net-2` network, click its pencil icon.
    h. In the **IPV4** tab of the **Edit Network** window, select **Static**.
    i. Enter the **IP** and **Netmask/Routing Prefix** of the subnet and click **OK**.
4. Create the iSCSI bond.
    a. Click **Compute** and then click **Data Centers**.
    b. Click the data center name.
    c. In the **iSCSI Multipathing** tab of the data center details page, click **Add**.
    d. In the **Add iSCSI Bond** window, enter a **Name**, select the networks `net-1` and `net-2` and click **OK**.

Your data center has an iSCSI bond containing the old and new logical networks.

Adding an FC Data Domain

To add an FC data domain:

1. Go to **Storage** and then click **Domains**.

The **Storage Domains** pane opens.

2. On the **Storage Domains** pane, click the **New Domain** button.

   The **New Domain** dialog box opens.

3. For the **Name** field, enter a name for the data domain.
4. From the **Data Center** drop-down list, select the Data Center for which to attach the data domain. By default, the **Default** option is selected in the drop-down list.
5. From the **Domain Function** drop-down list, select the domain function. By default, the **Data** option is selected in the drop-down list.

   For this step, leave **Data** as the domain function because you are creating a data domain in this example.
6. From the **Storage Type** drop-down list, select **Fibre Channel**.
7. For the **Host to Use** drop-down list, select the host for which to attach the data domain.
8. When **Fibre Channel** is selected for the **Storage Type**, the **New Domain** dialog box automatically displays the known targets with unused LUNs.
9. Click **Add** next to the LUN ID that is connect to the target.
10. **(Optional)** Configure the advanced parameters.
11. Click **OK**.

    You can click **Tasks** to monitor the various processing steps that are completed to attach the FC data domain to the data center.

Deploying GlusterFS Storage

Oracle Linux Virtualization Manager has been integrated with GlusterFS, an open source scale-out distributed filesystem, to provide a hyperconverged solution where both compute and storage are provided from the same hosts. Gluster volumes residing on the hosts are used as storage domains in the Manager to store the virtual machine images. In this scenario, the Manager is run as a self-hosted engine within a virtual machine on these hosts; although, you can deploy GlusterFS within a standalone environment.

For instructions on creating a GlusterFS storage domain, refer to the My Oracle Support (MOS) article *How to Create Glusterfs Storage Domain (Doc ID 2679824.1)*.

> **Note:**
>
> If you are deploying a self-hosted engine as hyperconverged infrastructure with GlusterFS storage, you must deploy GlusterFS *BEFORE* you deploy the self-hosted engine. For more information about using GlusterFS, including prerequisites, see the Oracle Linux GlusterFS documentation.

Deploying GlusterFS Storage Using Cockpit

To deploy GlusterFS storage using the Cockpit web interface, complete the following steps.

> **Important:**
>
> To deploy GlusterFS, you must have at least three (3) KVM hosts. If you want more than three KVM hosts, they must be added in factors of three.

1. Ensure that on all KVM hosts you have installed the following packages:
   - `cockpit-ovirt-dashboard` to provide a UI for installation
   - `vdsm-gluster` to manage gluster services
   - `ovirt-host` on the KVM host used for cockpit deployment
2. Go to **Compute**, and then click **Hosts**.
3. Under the **Name** column, click the host to be used as the designated server.
4. Click **Host Console**.
5. Enter your login credentials (the user name and password of the root account.).
6. Go to **Virtualization** and then click **Hosted Engine**.
7. Click **Redeploy** under **Hosted Engine Setup**.
8. Click **Start** under **Hyperconverged**.
9. On the **Hosts** screen, enter 3 (or more) KVM hosts that are in the data center to be used for GlusterFS, with the main designated KVM host entered first and click **Next** when finished.
10. On the **FQDNs** screen, enter the FQDN (or IP address) for the hosts to be managed by the Hosted Engine and click **Next** when finished.

    > **Note:**
    >
    > The FQDN of the designated server is input during the Hosted Engine deployment process and is not asked for here.

11. Click **Next** on the **Packages** screen.
12. On the **Volumes** screen, create the minimum storage domains that are required: `engine`, `data`, `export`, and `iso`. Click **Next** when finished.

    For example:

    engine

    - **Name**: `engine`
    - **Volume Type**: `Replicate` (default)
    - **Arbiter**: Ensure the check box is selected.
    - **Brick Dirs**: **/gluster_bricks/engine/engine** (default)

    data

    - **Name**: `data`
    - **Volume Type**: `Replicate` (default)

- **Arbiter**: Ensure the check box is selected.
- **Brick Dirs**: `/gluster_bricks/data/data` (default)

export

- **Name**: `export`
- **Volume Type**: `Replicate` (default)
- **Arbiter**: Ensure the check box is selected.
- **Brick Dirs**: `/gluster_bricks/export/export` (default)

iso

- **Name**: `iso`
- **Volume Type**: `Replicate` (default)
- **Arbiter**: Ensure the check box is selected.
- **Brick Dirs**: `/gluster_bricks/iso/iso` (default)

13. On the **Brick Locations** screen, specify the brick locations for your volumes and click **Next** when finished.

    For this step, you specify the brick locations for your volumes (`engine`, `data`, `export`, and `iso`).

14. Review the screen and click **Deploy**.

    - If you are using an internal disk as the Gluster disk, no edits are required and you can simply click **Deploy** to continue with the deployment.
    - If you are using an external iSCSI ZFS external drive as the Gluster disk, click **Edit** to edit the `gdeployConfig.conf` file and specify the block device on each server that is being used for storage. Click **Save** and then click **Deploy** to continue with the deployment.

    This process takes some time to complete, as the gdeploy tool installs required packages and configures Gluster volumes and their underlying storage.

    A message display on the screen when the deployment completes successfully.

Creating a GlusterFS Storage Domain Using the Manager

To add a GlusterFS storage volume as a storage domain:

1. Go to **Storage** and then click **Domains**.
2. On the **Storage Domains** pane, click the **New Domain** button.
3. For the **Name** field, enter a name for the data domain.
4. From the **Data Center** drop-down list, select the data center where the GlusterFS volume is deployed. By default, the **Default** option is selected in the drop-down list.
5. From the **Domain Function** drop-down list, select the domain function. By default, the **Data** option is selected in the drop-down list.

   For this step, leave **Data** as the domain function because a data domain is being created in this example.
6. From the **Storage Type** drop-down list, select **GlusterFS**.

7. For the **Host to Use** drop-down list, select the host for which to attach the data domain.
8. When **GlusterFS** is selected for the **Storage Type**, the **New Domain** dialog box updates to display additional configuration fields associated with GlusterFS storage domains.
9. Ensure the **Use managed gluster volume** check box is not selected.
10. From the **Gluster** drop-down list, select the path to which domain function you are creating.
11. For the **Mount Options** option, specify additional mount options in a comma-separated list, as you would using the mount -o command.
12. **(Optional)** Configure the advanced parameters.
13. Click **OK** to mount the volume as a storage domain.

    You can click **Tasks** to monitor the various processing steps that are completed to add the GlusterFS storage domain to the data center.

Detaching a Storage Domain from a Data Center

A storage domain must be in maintenance mode before it can be detached and removed. This is required to redesignate another data domain as the master data domain.

You cannot move a storage domain into maintenance mode if a virtual machine has a lease on the storage domain. The virtual machine needs to be shut down, or the lease needs to be to removed or moved to a different storage domain first.

To detach a storage domain from one data center to migrate it to another data center:

1. Shut down all the virtual machines running on the storage domain.
2. Go to **Storage** and then click **Domains**.

   The **Storage Domains** pane opens.
3. Click the storage domain's name.

   The details view of the storage domain opens.
4. Click the **Data Center** tab.
5. Click **Maintenance**.

   The `Ignore OVF update failure` check box allows the storage domain to go into maintenance mode even if the OVF update fails.

   > **Note:**
   >
   > The OVF_STORE disks are images that contain the metadata of virtual machines and disks that reside on the storage data domain.

6. Click **OK**.

   The storage domain is deactivated and has an `Inactive` status in the results list. You can now detach the inactive storage domain from the data center.
7. Click **Detach**.
8. Click **OK** to detach the storage domain.

Now that the storage domain is detached from the data center, it can be attached to another data center.

Virtual Machines

Oracle Linux Virtualization Manager lets you perform basic administration of your virtual machines, including live editing, creating and using snapshots and live migration.

> **Note:**
>
> For information on creating virtual machines, see the Oracle Linux Virtualization Manager: Getting Started Guide.

Live Editing a Virtual Machine

You can optionally change many settings for a virtual machine while it is running.

1. From the **Administration Portal**, click **Compute** and then select **Virtual Machines**.
2. Under the **Name** column, select the virtual machine you want to make changes to and then click **Edit**.
3. On the bottom left of the **Edit Virtual Machine** window, click **Show Advanced Options**.
4. Change any of the following properties while the virtual machine is running without restarting the virtual machine.

   Select the **General** tab, to modify:

   - **Optimized for**

     You can select from three options:

     – **Desktop** - the virtual machine has a sound card, uses an image (thin allocation), and is stateless.

     – **Server** - the virtual machine does not have a sound card, uses a cloned disk image, and is not stateless. In contrast, virtual machines optimized to act as desktop machines.

     – **High Performance** - the virtual machine is pre-configured with a set of suggested and recommended configuration settings for reaching the best efficiency.

     – **Name**

       A virtual machine's name must be unique within the data center. It must not contain any spaces and must contain at least one character from A-Z or 0-9. The maximum length is 255 characters.

       The name can be re-used in different data centers within Oracle Linux Virtualization Manager.

     – **Description** and **Comment**

- **Delete Protection**

  If you want to make it impossible to delete a virtual machine, check this box. If you later decide you want to delete the virtual machine, remove the check.

- **Network Interfaces**

  Add or remove network interfaces or change the network of an existing NIC.

Select the **System** tab, to modify:

- **Memory Size**

  Use to hot plug virtual memory. For more information, see Hot Plugging Virtual Memory.

- **Virtual Sockets** (Under **Advance Parameters**)

  Use to hot plug CPUs to the virtual machine. Do not assign more sockets to a virtual machine than are present on its KVM host. For more information, see Hot Plugging vCPUs.

Select the **Console** tab, to modify:

- **Disable strict user checking**

  By default, strict checking is enabled allowing only one user to connect to the console of a virtual machine until it has been rebooted. The exception is that a SuperUser can connect at any time and replace a existing connection. When a SuperUser has connected, no normal user can connect again until the virtual machine is rebooted.

  > **Important:**
  >
  > Check this box with caution because you can expose the previous user's session to the new user.

Select the **High Availability** tab, to modify:

- **Highly Available**

  Check this box if you want the virtual machine to automatically live migrate to another host if its host crashes or becomes non-operational. Only virtual machines with high availability are restarted on another host. If the virtual machine's host is manually shut down, the virtual machine does not automatically live migrate to another host. For more information, see Configuring a Highly Available Virtual Machine.

  > **Note:**
  >
  > You are not able to check this box if on the **Host** tab you have selected either **Allow manual migration only** or **Do not allow migration** for the Migration mode. For a virtual machine to be highly-available it must be possible for the engine to migrate the virtual machine to another host when needed.

- **Priority for Run/Migration Queue**

  Select the priority level (**Low**, **Medium** or **High**) for the virtual machine to live migrate or restart on another host.

Select the **Icon** tab, to upload a new icon.

5. Click **OK** when you are finished with all tabs to save your changes.

Changes to any other settings are applied when you shut down and restart your virtual machine. Until then, an orange icon displays to indicate pending changes.

Migrating Virtual Machines between Hosts

Virtual machines that share the same storage domain can live migrate between hosts that belong to the same cluster. Live migration allows you to move a running virtual machine between physical hosts with no interruption to service. The virtual machine stays powered on and user applications continue running while the virtual machine is relocated to a new physical host. In the background, the virtual machine's RAM is copied from the source host to the destination host. Storage and network connectivity are not changed.

You use live migration to seamlessly move virtual machines to support a number of common maintenance tasks. Ensure that your environment is correctly configured to support live migration well in advance of using it.

Configuring Your Environment for Live Migration

To enable successful live migrations, you should ensure you correctly configure it. At a minimum, to successfully migrate running virtual machines:

- Source and destination hosts should be in the same cluster
- Source and destination hosts must have a status of `Up`.
- Source and destination hosts must have access to the same virtual networks and VLANs
- Source and destination hosts must have access to the data storage domain where the virtual machines reside
- There must be enough CPU capacity on the destination host to support the virtual machine's requirements.
- There must be enough RAM on the destination host that is not in use to support the virtual machine's requirements

> **Note:**
>
> Live migrations are performed using the management network. The number of concurrent migrations supported is limited by default. Even with these limits, concurrent migrations can potentially saturate the management network. To minimize the risk of network saturation, we recommended that you create separate logical networks for storage, display, and virtual machine data.

To configure virtual machines so they reduce network outage during migration:

- Ensure that the destination host has an available virtual function (VF)
- Set the **Passthrough** and **Migrateable** options in the passthrough vNIC's profile

- Enable hotplugging for the virtual machine's network interface
- Ensure that the virtual machine has a backup VirtIO vNIC to maintain the virtual machine's network connection during migration
- Set the VirtIO vNIC's **No Network Filter** option before configuring the bond
- Add both vNICs as subordinate under an active-backup bond on the virtual machine, with the passthrough vNIC as the primary interface

Automatic Virtual Machine Migration

The Engine automatically initiates live migration of virtual machines in two situations:

- When a host is moved into maintenance mode live migration is initiated for all virtual machines running on the host. The destination host for each virtual machine is assessed as the virtual machine is migrated, in order to spread the load across the cluster.
- To maintain load balancing or power saving levels in line with scheduling policy live migrations are initiated.

You can disable automatic, or even manual, live migration of specific virtual machines if required.

Setting Virtual Machine Migration Mode

Using the **Migration mode** setting for a virtual machine, you can allow automatic and manual migration, disable automatic migration, or disable automatic and manual migration. If a virtual machine is configured to run only on a specific host, you cannot migrate in manually.

To set a virtual machine's migration mode:

From the **Migration mode** drop-down list, select **Allow manual and automatic migration**, **Allow manual migration only** or **Do not allow migration**.

To set the migration mode of a virtual machine:

1. Click **Compute** and select **Virtual Machines**.
2. Select a virtual machine and click **Edit**.
3. Click the **Host** tab.
4. Use the **Start Running On** radio buttons to specify whether the virtual machine should run on any host in the cluster, a specific host, or a group of hosts.

   If the virtual machine has host devices attached to it, and you choose a different host, the host devices from the previous host are removed from the virtual machine.

   > **NOT_SUPPORTED:**
   >
   > Assigning a virtual machine to one specific host and disabling migration is mutually exclusive in Oracle Linux Virtualization Manager high availability (HA). Virtual machines that are assigned to one specific host can only be made highly available using third-party HA products. This restriction does not apply to virtual machines that are assigned to a group of hosts.

5. From the **Migration mode** drop-down list, select **Allow manual and automatic migration**, **Allow manual migration only** or **Do not allow migration**.

6. **(Optional)** Check **Use custom migration downtime** and specify a value in milliseconds.
7. Click **OK**.

Manually Migrate a Virtual Machine

To manually migrate a virtual machine:

1. Click **Compute** and the select **Virtual Machines**.
2. Select a running virtual machine and click **Migrate**.
3. Choose either **Select Host Automatically** or **Select Destination Host** and select the destination host from the drop-down list.

   When you choose **Select Host Automatically**, the system determines the destination host according to the load balancing and power management rules set up in the scheduling policy.
4. Click **OK**.

During migration, progress is shown in the `Status` field. When the virtual machine has been migrated, the `Host` field updates to show the virtual machine's new host.

Importing an Oracle Linux Template

The Oracle Linux Virtualization Manager: Getting Started Guide has instructions on creating an Oracle Linux template. However, Oracle provides pre-installed and pre-configured templates that allow you to deploy a fully configured software stack. Use of Oracle Linux templates eliminates the installation and configuration costs and reduces the ongoing maintenance costs.

To import an Oracle Linux template:

1. Download a the template OVA file from http://yum.oracle.com/oracle-linux-templates.html and copy to your KVM host.
2. Assign permissions to the file.

   ```
   # chown 36:36 /tmp/<myfile>.ova
   ```
3. Ensure that the `kvm` user has access to the OVA file's path, for example:

   ```
   # -rw-r--r-- 1 vdsm kvm 872344576 Jan 15 17:43 OLKVM_OL7U7_X86_64.ova
   ```
4. In the Admistration Portal, click **Compute** and then select **Templates**.
5. Click **Import**.
6. From the **Import Template(s)** window, select the following options:
   - Data Center: **<datacenter>**
   - Source: **Virtual Appliance (OVA)**
   - Host: **<kvm_host_containing_ova>**
   - File Path: **<full_path_to_ova_file>**
7. Click **Load**.
8. From the **Virtual Machines on Source** list, select the virtual appliance's check box.

> **Note:**
>
> You can select more than one virtual appliance to import.

9. Click the right arrow to move the appliance(s) to the **Virtual Machines to Import** list and then click **Next**.
10. Click the **Clone** field for the template you want to import and review its **General**, **Network Interfaces**, and **Disks** configuration.
11. Click **OK**.

The import process can take several minutes. Once it completes, you can view the template(s) by clicking **Compute** and then **Templates**.

To create a virtual machine from your imported template, see **Creating an Oracle Linux Virtual Machine from a Template** in the Oracle Linux Virtualization Manager: Getting Started Guide.

Working with Virtual Machine Snapshots

A snapshot is a view of a virtual machine's operating system and applications on any or all available disks at a given point in time. You can take a snapshot of a virtual machine before you make a change to it that may have unintended consequences. If needed, you can use the snapshot to return the virtual machine to its previous state.

> **Note:**
>
> For best practices when using snapshots, see **Considerations When Using Snapshots** in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

Creating a Snapshot of a Virtual Machine

To create a snapshot of a virtual machine:

1. Click **Compute** and then select **Virtual Machines**.

   The **Virtual Machines** pane opens with the list of virtual machines that have been created.
2. Under the **Name** column, select the virtual machine for which to take a snapshot.

   The **General** tab opens with details about the virtual machine.
3. Click the **Snapshots** tab.
4. Click **Create**.
5. **(Optional)** For the **Description** field, enter a description for the snapshot.
6. **(Optional)** Select the **Disks to include** checkboxes. By default, all disks are selected.

> **Important:**
>
> Not selecting a disk results in the creation of a partial snapshot of the virtual machine without a disk. Although a saved partial snapshot does not have a disk, you can still preview a partial snapshot to view the configuration of the virtual machine.

7. **(Optional)** Select the **Save Memory** check box to include the virtual machine's memory in the snapshot. By default, this checkbox is selected.
8. Click **OK** to save the snapshot.

   The virtual machine's operating system and applications on the selected disks are stored in a snapshot that can be previewed or restored.

   On the **Snapshots** pane, the `Lock` icon appears next to the snapshot as it is being created. Once complete, the icon changes to the `Snapshot` (camera) icon. You can then display details about the snapshot by selecting the **General**, **Disks**, **Network Interfaces**, and **Installed Applications** drop-down views.

Restoring a Virtual Machine from a Snapshot

A snapshot can be used to restore a virtual machine to a previous state.

> **Note:**
>
> The virtual machine must be in a `Down` state before performing this task.

To restore a virtual machine from a snapshot:

1. Click **Compute** and then select **Virtual Machines**.

   The **Virtual Machines** pane opens with the list of virtual machines that have been created.
2. Under the **Name** column, select the virtual machine that you want to restore from a snapshot.

   The **General** tab opens with details about the virtual machine.
3. Click the **Snapshots** tab.
4. On the **Snapshots** pane, select the snapshot to be used to restore the virtual machine.
5. From the **Preview** drop-down list, select **Custom**.

   On the **Virtual Machines** pane, the status of the virtual machine briefly changes to `Image Locked` before returning to `Down`.

   On the **Snapshots** pane, the `Preview` (eye) icon appears next to the snapshot when the preview of the snapshot is completed.
6. Click **Run** to start the virtual machine.

   The virtual machine runs using the disk image of the snapshot. You can preview the snapshot and verify the state of the virtual machine.

7. Click **Shutdown** to stop the virtual machine.
8. From the **Snapshot** pane, perform one of the following steps:
   a. Click **Commit** to permanently restore the virtual machine to the condition of the snapshot. Any subsequent snapshots are erased.
   b. Alternatively, click **Undo** to deactivate the snapshot and return the virtual machine to its previous state.

Creating a Virtual Machine from a Snapshot

Before performing this task, you must create a snapshot of a virtual machine. For more information, refer to Creating a Snapshot of a Virtual Machine.

To create a virtual machine from a snapshot:

1. Click **Compute** and then select **Virtual Machines**.

   The **Virtual Machines** pane opens with the list of virtual machines that have been created.
2. Under the **Name** column, select the virtual machine with the snapshot that you want to use as the basis from which to create another virtual machine.

   The **General** tab opens with details about the virtual machine.
3. Click the **Snapshots** tab.
4. On the **Snapshots** pane, select the snapshot from which to create the virtual machine.
5. Click **Clone**.

   The **Clone VM from Snapshot** dialog box opens.
6. For the **Name** field, enter a name for the virtual machine.

> **Note:**
>
> The **Name** field is the only required field on this dialog box.

After a short time, the cloned virtual machine appears on the **Virtual Machines** pane with a status of `Image Locked`. The virtual machine remains in this state until the Manager completes the creation of the virtual machine. When the virtual machine is ready to use, its status changes from `Image Locked` to `Down` on the **Virtual Machines** pane.

Deleting a Snapshot

You can delete a virtual machine snapshot and permanently remove it from your virtualization environment. This operation is supported on a running virtual machine and does not require the virtual machine to be in a `Down` state.

Important:

- When you delete a snapshot from an image chain, there must be enough free space in the storage domain to temporarily accommodate both the original volume and the newly merged volume; otherwise, the snapshot deletion fails. This is due to the data from the two volumes being merged in the resized volume and the resized volume growing to accommodate the total size of the two merged images. In this scenario, you must export and reimport the volume to remove the snapshot.
- If the snapshot being deleted is contained in a base image, the volume subsequent to the volume containing the snapshot being deleted is extended to include the base volume.
- If the snapshot being deleted is contained in a QCOW2 (thin-provisioned), non-base image hosted on internal storage, the successor volume is extended to include the volume containing the snapshot being deleted.

To delete a snapshot:

1. Click **Compute** and then select **Virtual Machines**.

   The **Virtual Machines** pane opens with the list of virtual machines that have been created.
2. Under the **Name** column, select the virtual machine with the snapshot that you want to delete.

   The **General** tab opens with details about the virtual machine.
3. Click the **Snapshots** tab.
4. On the **Snapshots** pane, select the snapshot to delete.
5. Select the snapshot to delete.
6. Click **Delete**.
7. Click **OK**.

   On the **Snapshots** pane, a `Lock` icon appears next to the snapshot until the snapshot is deleted.

Encrypted Communication

You can configure your organization's third-party CA certificate to identify the Oracle Linux Virtualization Manager to users connecting over HTTPS.

Using a third-party CA certificate for HTTPS connections does not affect the certificate that is used for authentication between the engine host and KVM hosts. They continue to use the self-signed certificate generated by the Manager.

Replacing the Oracle Linux Virtualization Manager Apache SSL Certificate

Before you begin you must obtain a third-party CA certificate, which is a digital certificate issued by a certificate authority (CA). The certificate is provided as a PEM file. The certificate chain must be complete up to the root certificate. The chain's order is critical and must be from the last intermediate certificate to the root certificate.

> **Caution:**
>
> Do not change the permissions and ownerships for the `/etc/pki` directory or any subdirectories. The permission for the `/etc/pki` and `/etc/pki/ovirt-engine` directories must remain as the default value of `755`.

To replace the Oracle Linux Virtualization Manager Apache SSL Certificate:

1. Copy the new third-party CA certificate to the host-wide trust store and update the trust store.

   ```
   # cp third-party-ca-cert .pem /etc/pki/ca-trust/source/anchors/
   # update-ca-trust export
   ```

2. Remove the symbolic link to `/etc/pki/ovirt-engine/apache-ca.pem`.

   The Engine has been configured to use `/etc/pki/ovirt-engine/apache-ca.pem`, which is symbolically linked to `/etc/pki/ovirt-engine/ca.pem`.

   ```
   # rm /etc/pki/ovirt-engine/apache-ca.pem
   ```

3. Copy the CA certificate into the PKI directory for the Manager.

   ```
   # cp third-party-ca-cert .pem /etc/pki/ovirt-engine/apache-ca.pem
   ```

4. Back up the existing private key and certificate.

   ```
   # cp /etc/pki/ovirt-engine/keys/apache.key.nopass/etc/pki/ovirt-engine/keys/ \
     apache.key.nopass.bck
   # cp /etc/pki/ovirt-engine/certs/apache.cer/etc/pki/ovirt-engine/certs/
   apache.cer.bck
   ```

5. Copy the new Apache private key into the PKI directory for the Manager by entering the following command and respond to prompt.

   ```
   # cp apache.key /etc/pki/ovirt-engine/keys/apache.key.nopass
   cp: overwrite \u2018/etc/pki/ovirt-engine/keys/apache.key.nopass\u2019? y
   ```

6. Copy the new Apache certificate into the PKI directory for the Manager by entering the following command and respond to the prompt.

   ```
   # cp apache.cer /etc/pki/ovirt-engine/certs/apache.cer
   cp: overwrite \u2018/etc/pki/ovirt-engine/certs/apache.cer\u2019? y
   ```

7. Restart the Apache HTTP server (`httpd`) and the Manager.

   ```
   # systemctl restart httpd
   # systemctl restart ovirt-engine
   ```

8. Create a new trust store configuration file (or edit the existing one) at `/etc/ovirt-engine/engine.conf.d/99-custom-truststore.conf` by adding the following parameters.

```
ENGINE_HTTPS_PKI_TRUST_STORE="/etc/pki/java/cacerts"
ENGINE_HTTPS_PKI_TRUST_STORE_PASSWORD=""
```

9. Back up the existing Websocket configuration file.

```
# cp /etc/ovirt-engine/ovirt-websocket-proxy.conf.d/10-setup.conf/etc/ovirt-engine/ \
ovirt-websocket-proxy.conf.d/10-setup.conf.bck
```

10. Edit the Websocket configuration file at `/etc/ovirt-engine/ovirt-websocket-proxy.conf.d/10-setup.conf` by adding the following parameters.

```
SSL_CERTIFICATE=/etc/pki/ovirt-engine/certs/apache.cer
SSL_KEY=/etc/pki/ovirt-engine/keys/apache.key.nopass
```

11. Restart the `ovirt-provider-ovn` service.

```
# systemctl restart ovirt-provider-ovn
```

12. Restart the `ovirt-engine` service.

```
# systemctl restart ovirt-engine
```

Event Notifications

The following section explains how to set up event notifications to monitor events in your virtualization environment. You can configure the Manager to send event notifications in email to alert designated users when certain events occur or enable Simple Network Management Protocol (SNMP) traps to monitor your virtualization environment.

Configuring Event Notification Services on the Engine

For event notifications to be sent properly to email recipients, you must configure the mail server on the Engine and enable `ovirt-engine-notifier` service. For more information about creating event notifications in the Administration portal, see Creating Event Notifications in the Administration Portal.

To configure notification services on the Engine:

1. Log in to the host that is running the Manager.
2. Copy the `ovirt-engine-notifier.conf` to a new file named `90-email-notify.conf`.

```
# cp /usr/share/ovirt-engine/services/ovirt-engine-notifier/ovirt-engine-notifier.conf/ \
etc/ovirt-engine/notifier/notifier.conf.d/90-email-notify.conf
```

3. Edit the `90-email-notify.conf` file by deleting everything except the `EMAIL Notifications` section.

> **Note:**
>
> If you plan to also configure SNMP traps in your virtualization environment, you can also copy the values from the `SNMP_TRAP Notifications` section of the `ovirt-notifier.conf` file to a file named `20-snmp.conf`. For more information, see Configuring the Engine to Send SNMP Traps.

4. Enter the correct email variables. This file overrides the values in the original `ovirt-engine-notifier.conf` file.

```
--------------------
# EMAIL Notifications #
--------------------

# The SMTP mail server address. Required.
MAIL_SERVER=myemailserver.mycompany.com

# The SMTP port (usually 25 for plain SMTP, 465 for SMTP with SSL, 587 for SMTP
with TLS)
MAIL_PORT=25

# Required if SSL or TLS enabled to authenticate the user. Used also to specify
'from'
  user address if mail server
# supports, when MAIL_FROM is not set. Address is in RFC822 format
MAIL_USER=email.example.com

# Required to authenticate the user if mail server requires authentication or if
SSL or
  TLS is enabled
SENSITIVE_KEYS="${SENSITIVE_KEYS},MAIL_PASSWORD"
MAIL_PASSWORD=

# Indicates type of encryption (none, ssl or tls) should be used to communicate
with
  mail server.
MAIL_SMTP_ENCRYPTION=none

# If set to true, sends a message in HTML format.
HTML_MESSAGE_FORMAT=false

# Specifies 'from' address on sent mail in RFC822 format, if supported by mail
server.
MAIL_FROM=myovirtengine@mycompany.com

# Specifies 'reply-to' address on sent mail in RFC822 format.
MAIL_REPLY_TO=myusername@mycompany.com

# Interval to send smtp messages per # of IDLE_INTERVAL
MAIL_SEND_INTERVAL=1

# Amount of times to attempt sending an email before failing.
MAIL_RETRIES=4
```

> **Note:**
>
> For information about the other parameters available for event notification in the `ovirt-engine-notifier.conf` file, refer to oVirt Documentation.

5. Enable and restart the `ovirt-engine-notifier` service to activate your changes.

```
# systemctl daemon-reload
# systemctl enable ovirt-engine-notifier.service
# systemctl restart ovirt-engine-notifier.service
```

Creating Event Notifications in the Administration Portal

Before creating event notifications, you must have access to an email server that can handle incoming automated messages and deliver these messages to a distribution list. You should also configure event notification services on the Engine. For more information, see Configuring Event Notification Services on the Engine.

To create event notifications in the Administration Portal:

1. Go to **Administration** and then click **Users**.

   The **Users** pane opens.

2. Under the **User Name** column, click the name of the user to display the detailed view for the user.

> **Note:**
>
> A user does not appear in the Administration Portal until the user is created and assigned appropriate permissions. For more information, refer to Creating a New User Account.

3. Click the **Event Notifier** tab.
4. Click **Manage Events**.

   The **Add Event Notification** dialog box opens.

5. Select the events for which you want to create notifications by selecting the check box next to individual events or event topic areas for notification.

   The events available for notification are grouped under topic areas. By default, selecting the check box for a top-level topic area, such as **General Host Events**, selects all events under that topic area. You can optionally expand or collapse all the event topic areas by clicking **Expand All** or **Collapse All**. Additionally, you can click the arrow icon next to a specific top-level topic area to expand or collapse the events associated with a specific topic area.

6. For the **Mail Recipient** field, enter an email address.
7. Click **OK** to save the changes.

Canceling Event Notifications in the Administration Portal

To cancel event notifications in the Administration Portal:

1. Go to **Administration** and then click **Users**.

   The **Users** pane opens.
2. Under the **User Name** column, click the name of the user to display the detailed view for the user.
3. Click the **Event Notifier** tab.
4. Click **Manage Events**.

   The **Add Event Notification** dialog box opens.
5. Click **Expand All**, or the topic-specific expansion options, to display the events.
6. Clear the appropriate check boxes to cancel the notification for that event.
7. Click **OK** to save your changes.

Configuring the Engine to Send SNMP Traps

You can configure the Manager to send SNMP traps to one or more external SNMP managers. SNMP traps contain system event information that are used to monitor your virtualization environment. The number and type of traps sent to the SNMP manager can be defined within the Engine.

Before performing this task, you must have configured one or more external SNMP managers to receive traps, and know the following details:

- The IP addresses or fully qualified domain names of machines that act as SNMP managers. Optionally, determine the port through which the SNMP manager receives trap notifications; the default UDP port is `162`.
- The SNMP community. Multiple SNMP managers can belong to a single community. Management systems and agents can communicate only if they are within the same community. The default community is `public`.
- The trap object identifier for alerts. The Engine provides a default OID of `1.3.6.1.4.1.2312.13.1.1`. All trap types are sent, appended with event information, to the SNMP manager when this OID is defined.

> **Note:**
>
> – Changing the default trap prevents generated traps from complying with the Engine's management information base.
>
> – The Engine provides management information bases at `/usr/share/doc/ovirt-engine/mibs/OVIRT-MIB.txt` and `/usr/share/doc/ovirt-engine/mibs/REDHAT-MIB.txt`. Load the MIBs in your SNMP manager before proceeding.

To configure SNMP traps on the Engine:

1. Log in to the host that is running the Manager.
2. On the Engine, create the SNMP configuration file:

   ```
   # vi /etc/ovirt-engine/notifier/notifier.conf.d/20-snmp.conf
   ```

   Default SNMP configuration values exist on the Engine in the events notifications configuration file (`ovirt-engine-notifier.conf`), which is available at the following directory path: `/usr/share/ovirt-engine/services/ovirt-engine-notifier/ovirt-engine-notifier.conf`. The values provided in this step are based on the default or example values provided in that file. To persist that your configuration settings persist across reboots, define an override file for your SNMP configuration (`20-snmp.conf`), rather than edit the `ovirt-engine-notifier.conf` file, For more information, see Configuring Event Notification Services on the Engine.
3. Specify the SNMP manager, the SNMP community, and the OID in the following format:

   ```
   SNMP_MANAGERS="manager1.example.com manager2.example.com:162"
   SNMP_COMMUNITY=public
   SNMP_OID=1.3.6.1.4.1.2312.13.1.1
   ```

   The following values can be configured in the `20-snmp.conf` file.

   ```
   #-------------------------#
   # SNMP_TRAP Notifications #
   #-------------------------#
   # Send v2c snmp notifications

   # Minimum SNMP configuration
   #
   # Create /etc/ovirt-engine/notifier/notifier.conf.d/20-snmp.conf with:
   # SNMP_MANAGERS="host"
   # FILTER="include:*(snmp:) ${FILTER}"

   # Default whitespace separated IPv4/[IPv6]/DNS list with optional port,
   default is 162.
   # SNMP_MANAGERS="manager1.example.com manager2.example.com:164"
   SNMP_MANAGERS=

   # Default SNMP Community String.
   SNMP_COMMUNITY=public

   # SNMP Trap Object Identifier for outgoing notifications.
   # { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) redhat(2312)
   ovirt(13)
     engine(1) notifier(1) }
   #
   # Note: changing the default will prevent generated traps from complying
   with
     OVIRT-MIB.txt.
   SNMP_OID=1.3.6.1.4.1.2312.13.1.1

   # Default SNMP Version. SNMP version 2 and version 3 traps are supported
   # 2 = SNMPv2
   # 3 = SNMPv3
   SNMP_VERSION=2

   # The engine id used for SNMPv3 traps
   SNMP_ENGINE_ID=
   ```

```
# The user name used for SNMPv3 traps
SNMP_USERNAME=

# The SNMPv3 auth protocol. Supported values are MD5 and SHA.
SNMP_AUTH_PROTOCOL=

# The SNMPv3 auth passphrase, used when SNMP_SECURITY_LEVEL is set to AUTH_NOPRIV
  and AUTH_PRIV
SNMP_AUTH_PASSPHRASE=

# The SNMPv3 privacy protocol. Supported values are AES128, AES192 and AES256.
# Be aware that AES192 and AES256 are not defined in RFC3826, so please verify
# that your SNMP server supports those protocols before enabling them.
SNMP_PRIVACY_PROTOCOL=

# The SNMPv3 privacy passphrase, used when SNMP_SECURITY_LEVEL is set to AUTH_PRIV
SNMP_PRIVACY_PASSPHRASE=

# The SNMPv3 security level.
# 1 = NOAUTH_NOPRIV
# 2 = AUTH_NOPRIV
# 3 = AUTH_PRIV
SNMP_SECURITY_LEVEL=1

# SNMP profile support
#
# Multiple SNMP profiles are supported.
# Specify profile settings by using _profile suffix,
# for example, to define a profile to sent specific
# message to host3, specify:
# SNMP_MANAGERS_profile1=host3
# FILTER="include:VDC_START(snmp:profile1) ${FILTER}"
```

4. Define which events to send to the SNMP Manager.

   By default, the following default filter is defined in the `ovirt-engine-notifier.conf` file; if you do not override this filter or apply overriding filters, no notifications are sent.

   ```
   FILTER="exclude:\*"
   ```

   The following are other common examples of event filters.

   - Send all events to the default SNMP profile.

     ```
     FILTER="include:\*(snmp:) ${FILTER}"
     ```

   - Send all events with the severity `ERROR` or `ALERT` to the default SNMP profile:

     ```
     FILTER="include:\*:ERROR(snmp:) ${FILTER}"
      FILTER="include:\*:ALERT(snmp:) ${FILTER}"
     ```

5. Save the file.
6. Start the `ovirt-engine-notifier` service, and ensure that this service starts on boot.

   ```
   # systemctl start ovirt-engine-notifier.service
   # systemctl enable ovirt-engine-notifier.service
   ```

7. **(Optional)** Validate that traps are being sent to the SNMP Manager.

4
Deployment Optimization

You can configure Oracle Linux Virtualization Manager so that your cluster is optimized and your hosts and virtual machine are highly available. You can also enable or disable devices (hot plug) while a virtual machine is running.

Optimizing Clusters, Hosts and Virtual Machines

Whether you have a new cluster, host, or virtual machine or existing ones, you can optimize resources such as CPU and memory and configure hosts and virtual machines for high availability.

Configuring Memory and CPUs

Using the **Optimization** tab when creating or editing a cluster, you can select the memory page sharing threshold for the cluster, and optionally enable CPU thread handling and memory ballooning on the hosts in the cluster. Some of the benefits are:

- Virtual machines run on hosts up to the specified overcommit threshold. Higher values conserve memory at the expense of great CPU usage.
- Hosts to run virtual machines with a total number of CPU cores greater than the number of cores in the host.
- Memory overcommitment on virtual machines running on the hosts in the cluster.
- Memory Overcommitment Manager (MoM) runs Kernel Same-page Merging (KSM) when it can yield a memory saving benefit.

> **Note:**
>
> If a virtual machine is running Oracle products, such as Oracle Database or other Oracle applications, that require dedicated memory, configuring memory overcommitment is not an available option.

Using the Resource Allocation tab when creating or editing a virtual machine, you can:

- set the maximum amount of processing capability a virtual machine can access on its host.
- pin a virtual CPU to a specific physical CPU.
- guarantee an amount of memory for the virtual machine.
- enable the memory balloon device for the virtual machine. Enable **Memory Balloon Optimization** must also be selected for the cluster.
- improve the speed of disks that have a VirtIO interface by pinning them to a thread separate from the virtual machine's other functions.

For more information, refer to High Availability and Optimization in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

Configuring Cluster Memory and CPUs

Use the Administration Portal to optimize the usage of memory and CPUs at the cluster level:

1. Select the **Optimization** tab of the **New Cluster** or **Edit Cluster** window.
2. Choose a setting for **Memory Optimization**:
   - **None** - Disable memory overcommit

     Disables memory page sharing, which allows you to commit 100% of the physical memory to virtual machines.
   - **For Server Load** - Allow scheduling of 150% of physical memory

     Sets memory page sharing threshold to 150% of the system memory on each host.
   - **For Desktop Load** - Allow scheduling of 200% of physical memory

     Sets memory page sharing threshold to 200% of the system memory on each host.
3. Under **CPU Threads**, check **Count Threads As Cores** to allow guests to use host threads as virtual CPU cores.

   Allowing hosts to run virtual machines with the total number of processing cores greater than the number of cores in the host may be useful for less CPU-intensive workloads.
4. Under **Memory Balloon**, check **Enable Memory Balloon Optimization** to enable memory overcommitment on virtual machines running on hosts in this cluster.

   The MoM starts ballooning where and when possible. It is only limited by the guaranteed memory size of every virtual machine. Each virtual machine in the cluster needs to have a balloon device with relevant drivers, which is included unless you specifically remove it. Every host in the cluster receives a balloon policy update when its Status changes to Up.

   > **Note:**
   >
   > Enable ballooning on virtual machines that have applications and loads that slowly consume memory, occasionally release memory, or stay dormant for long periods of time, such as virtual desktops.
5. Under **KSM Control**, check **Enable KSM** to enable MoM to run KSM when necessary and when it can yield a memory saving benefit that outweighs its CPU cost.
6. Click **OK** to save your changes.

Changing Memory Overcommit Manually

The memory overcommit settings in the Administration Portal allow you to disable overcommit or set it to 150% or 200%. If you require a different value for your cluster, you can change the setting manually.

1. From a a command line, log into the Engine.
2. Check the current memory overcommit settings:

```
# engine-config -a | grep -i maxvdsmem
MaxVdsMemOverCommit: 200 version: general
MaxVdsMemOverCommitForServers: 150 version: general
```

3. Change the memory overcommit settings:

```
# engine-config -s MaxVdsMemOverCommitForServers=percentage

# engine-config -s MaxVdsMemOverCommit=percentage
```

Configuring Virtual Machine Memory and CPUs

To optimize the usage of memory and CPUs for a virtual machine:

1. Select the **Resource Allocation** tab of the **New VM** or **Edit VM** window.
2. Under **CPU Allocation**, for the **CPU Shares** drop-down list select the level of CPU resources a virtual machine can demand relative to other virtual machines in the cluster.
   - **Low**=512
   - **Medium**=1024
   - **High**=2048
   - **Custom**=Enter a number in the field next to the drop-down list
3. Under **Memory Allocation**, for **Physical Memory Guaranteed** enter an amount of memory.

   The amount of physical memory guaranteed for a virtual machine should be any number between 0 and its defined memory.
4. Check **Memory Balloon Device Enabled** to enable the device for the virtual machine and allow memory overcommitment.

> **Important:**
>
> Since this check box is selected by default, make sure you have enabled memory ballooning for the cluster where the virtual machine's host resides.

5. Under **I/O Threads**, check **I/O Threads Enabled** to improve the speed of disks that have a VirtIO interface by pinning them to a thread separate from the virtual machine's other functions.

   This check box is selected by default.
6. Under **Queues**, check **Multi Queues Enabled** to create up to four queues per vNIC, depending on how many vCPUs are available.

This check box is selected by default.

To define a different number of queues per vNIC, you can create a custom property:

```
# engine-config -s "CustomDeviceProperties={type=interface;prop={other-nic-properties;queues=[1-9][0-9]\*}}"
```

where ***other-nic-properties*** is a list of pre-existing NIC custom properties separated by semicolons.

7. Under **Queues**, check **VirtIO-SCSI Enabled** to enable or disable the use of VirtIO-SCSI on the virtual machine.

   This check box is selected by default.

8. Click **OK** to save your changes.

Configuring a Highly Available Host

If you want the hosts in a cluster to be responsive and available when unexpected failures happen, you should use fencing. Fencing allows a cluster to react to unexpected host failures and enforce power saving, load balancing, and virtual machine availability policies.

A *non-operational* host is different from a *non-responsive* host. A non-operational host can communicate with the Manager, but isn't configured correctly, for example a missing logical network. A non-responsive host cannot communicate with the Manager. In a fencing operation, if a host becomes non-responsive, it is rebooted. If after a prescribed period of time the host remains non-responsive, manual intervention needs to be taken.

The Manager can perform management operations after it reboots, by a proxy host, or manually in the **Administration Portal**. All the virtual machines running on the non-responsive host are stopped, and highly available virtual machines are restarted on a different host. At least two hosts are required for power management operations.

> **Important:**
>
> If a host runs virtual machines that are highly available, power management must be enabled and configured.

For more information, refer to High Availability and Optimization in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

Configuring Power Management and Fencing on a Host

The Manager uses a proxy to send power management commands to a host power management device because the engine does not communicate directly with fence agents. The host agent (VDSM) executes power management device actions and another host in the environment is used as a fencing proxy. This means that you must have at least two hosts for power management operations.

When you configure a fencing proxy host, make sure the host is in:

- the same cluster as the host requiring fencing.

- the same data center as the host requiring fencing.
- `UP` or `Maintenance` status to remain viable.

Power management operations can be performed in three ways:

- by the Manager after it reboots
- by a proxy host
- manually in the **Administration Portal**

To configure power management and fencing on a host:

1. Click **Compute** and select **Hosts**.
2. Select a host and click **Edit**.
3. Click the **Power Management** tab.
4. Check **Enable Power Management** to enable the rest of the fields.
5. Check **Kdump integration** to prevent the host from fencing while performing a kernel crash dump. Kdump integration is enabled by default.

   > **Important:**
   >
   > If you enable or disable Kdump integration on an existing host, you must reinstall the host.

6. **(Optional)** Check **Disable policy control of power management** if you do not want your host's power management to be controlled by the scheduling policy of the host's cluster.
7. To configure a fence agent, click the plus sign (+) next to **Add Fence Agent**.

   The **Edit fence agent** pane opens.
8. Enter the **Address** (IP Address or FQDN) to access the host's power management device.
9. Enter the **User Name** and **Password** of the of the account used to access the power management device.
10. Select the power management device **Type** from the drop-down list.
11. Enter the **Port** (SSH) number used by the power management device to communicate with the host.
12. Enter the **Slot** number used to identify the blade of the power management device.
13. Enter the **Options** for the power management device. Use a comma-separated list of key-value pairs.
    - If you leave the **Options** field blank, you are able to use both IPv4 and IPv6 addresses
    - To use only IPv4 addresses, enter `inet4_only=1`
    - To use only IPv6 addresses, enter `inet6_only=1`
14. Check **Secure** to enable the power management device to connect securely to the host.

    You can use ssh, ssl, or any other authentication protocol your power management device supports.

15. Click **Test** to ensure the settings are correct and then click **OK**.

    Test Succeeded, Host Status is: on displays if successful.

    NOT_SUPPORTED:

    Power management parameters (userid, password, options, etc.) are tested by the Manager only during setup and manually after that. If you choose to ignore alerts about incorrect parameters, or if the parameters are changed on the power management hardware without changing in the Manager as well, fencing is likely to fail when most needed.

16. Fence agents are sequential by default. To change the sequence in which the fence agents are used:
    a. Review your fence agent order in the **Agents by Sequential Order** field.
    b. To make two fence agents concurrent, next to one fence agent click the **Concurrent with** drop-down list and select the other fence agent.

       You can add additional fence agents to this concurrent fence agent group.
17. Expand the **Advanced Parameters** and use the up and down buttons to specify the order in which the Manager searches the host's **cluster** and **dc** (data center) for a power management proxy.
18. To add an additional power management proxy:
    a. Click the plus sign (+) next to **Add Power Management Proxy**.

       The **Select fence proxy preference type to add** pane opens.
    b. Select a power management proxy from the drop-down list and then click **OK**.

       Your new proxy displays in the **Power Management Proxy Preference** list.

    Note:

    By default, the Manager searches for a fencing proxy within the same cluster as the host. If The Manager cannot find a fencing proxy within the cluster, it searches the data center.

19. Click **OK**.

From the list of hosts, the exclamation mark next to the host's name disappeared, signifying that you have successfully configured power management and fencing.

Preventing Host Fencing During Boot

After you configure power management and fencing, when you start the Manager it automatically attempts to fence non-responsive hosts that have power management enabled *after* the quiet time (5 minutes by default) has elapsed. You can opt to extend the quiet time to prevent, for example, a scenario where the Manager attempts to fence hosts while they boot up. This can happen after a data center outage because a host's boot process is normally longer than the Manager boot process.

You can configure quiet time using the `engine-config` command option `DisableFenceAtStartupInSec`:

```
# engine-config -s DisableFenceAtStartupInSec=number
```

Checking Fencing Parameters

To automatically check the fencing parameters, you can configure the `PMHealthCheckEnabled` (false by default) and `PMHealthCheckIntervalInSec` (3600 sec by default) engine-config options.

```
# engine-config -s PMHealthCheckEnabled=True

# engine-config -s PMHealthCheckIntervalInSec=number
```

When set to true, `PMHealthCheckEnabled` checks all host agents at the interval specified by `PMHealthCheckIntervalInSec` and raises warnings if it detects issues.

Configuring a Highly Available Virtual Machine

If you have virtual machines that run critical workloads, you might consider configuring these virtual machines for high availability. Only a highly available virtual machine automatically restarts on its original host or migrates to another host in the cluster if its original host:

- has a hardware failure and becomes non-operational.
- has scheduled downtime and is put in maintenance mode.
- loses communication with external storage and becomes unavailable.

If a virtual machine's host is manually shut down, the virtual machine does not automatically migrate to another host. Further, virtual machines that share the same storage domain can live migrate between hosts that belong to the same cluster. For more information, see Migrating Virtual Machines between Hosts in the Oracle Linux Virtualization Manager: Administration Guide.

> **Note:**
>
> A highly available virtual machine does not restart if you shut it down cleanly from within the virtual machine or the Manager or if you shut down a host without first putting it into maintenance mode.

To enable a virtual machine to migrate to another available host in the cluster:

- Configure power management and fencing for the host running the highly available virtual machine
- Ensure the highly available virtual machine's host is part of a cluster of two or more available hosts
- Check that the destination host is operational
- Ensure the source and destination hosts can access the data domain where the virtual machine resides
- Ensure the source and destination hosts can access the same virtual networks and VLANs

- Check that the destination host has enough RAM and CPUs that are not in use to support the virtual machine's requirements

Virtual machines can also be restarted on another host even if the original host loses power if you have configured it to acquire a lease on a special volume on the storage domain. Acquiring a lease prevents the virtual machine from being started on two different hosts, which could result in virtual machine disk corruption.

If you configure high availability:

- there is minimal service interruption because virtual machines are restarted within seconds and with no user intervention.
- your resources are balanced by restarting virtual machines on a host with low current resource utilization.
- you are ensured that there is sufficient capacity to restart virtual machines at all times.

You must configure high availability for each virtual machine using the following steps:

1. Click **Compute** and then **Virtual Machines**.
2. In the list of virtual machines, click to highlight a virtual machine and then click **Edit**.
3. In the **Edit Virtual Machine** window, click the **High Availability** tab.
4. Check **Highly Available** to enable high availability for the virtual machine.
5. From the **Target Storage Domain for VM Lease** drop-down list, select **No VM Lease** (default) to disable the functionality or select a storage domain to hold the virtual machine lease.

   Virtual machines are able to acquire a lease on a special volume on the storage domain. This enables a virtual machine to start on another host even if the original host loses power. For more information, see Storage Leases in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.
6. From the **Resume Behavior** drop-down list, select **AUTO_RESUME**, **LEAVE_PAUSED**, OR **KILL**. If you defined a VM lease, KILL is the only option available.
7. From the **Priority** list, select **Low**, **Medium**, or **High**.

   When virtual machine migration is triggered, a queue is created in which the high priority virtual machines are migrated first. If a cluster is running low on resources, only the high-priority virtual machines are migrated.
8. Click **OK**.

Configuring High-Performance Virtual Machines

You can configure a virtual machine for high performance, so that it runs with performance metrics as close to bare metal as possible. When you choose high performance optimization, the virtual machine is configured with a set of automatic, and recommended manual, settings for maximum efficiency.

The high performance option is only accessible in the Administration Portal, by selecting **High Performance** from the **Optimized for** dropdown list in the **Edit** or **New** virtual machine, template, or pool window. This option is not available in the VM Portal.

If you change the optimization mode of a running virtual machine to high performance, some configuration changes require restarting the virtual machine. To change the optimization mode of a new or existing virtual machine to high performance, you may need to make manual changes to the cluster and to the pinned host configuration first.

A high performance virtual machine has certain limitations, because enhanced performance has a trade-off in decreased flexibility:

- If pinning is set for CPU threads, IO threads, emulator threads, or NUMA nodes, according to the recommended settings, only a subset of cluster hosts can be assigned to the high performance virtual machine.
- Many devices are automatically disabled, which limits the virtual machine's usability.

Creating a High Performance Virtual Machine

To create a high performance virtual machine:

1. In the **New** or **Edit** window, select **High Performance** from the **Optimized for** drop-down menu.

   Selecting this option automatically performs certain configuration changes to this virtual machine.

2. Click **OK**.

   If you have not set any manual configurations, the **High Performance Virtual Machine/ Pool Settings** screen describing the recommended manual configurations appears.

   If you have set some of the manual configurations, the **High Performance Virtual Machine/Pool Settings** screen displays the settings you have not made.

   If you have set all the recommended manual configurations, the **High Performance Virtual Machine/Pool Settings** screen does not appear.

3. If the **High Performance Virtual Machine/Pool Settings** screen appears, click **Cancel** to return to the **New** or **Edit** window to perform the manual configurations. For details, see Configuring the Recommended Manual Settings in oVirt Documentation.

   Alternatively, click **OK** to ignore the recommendations. The result may be a drop in the level of performance.

4. Click **OK**.

   You can view the optimization type in the **General** tab of the details view of the virtual machine, pool, or template.

Certain configurations can override the high performance settings. For example, if you select an instance type for a virtual machine before selecting **High Performance** from the **Optimized for** drop-down menu and performing the manual configuration, the instance type configuration will not affect the high performance configuration. If, however, you select the instance type after the high performance configurations, you should verify the final configuration in the different tabs to ensure that the high performance configurations have not been overridden by the instance type.

The last-saved configuration usually takes priority.

Configuring Huge Pages

You can configure a virtual machine for high performance, so that it runs with performance metrics as close to bare metal as possible. When you choose high performance optimization,

the virtual machine is configured with a set of automatic and recommended manual settings for maximum efficiency. By using huge pages, you increase the page size which reduces the page table, reduces the pressure on the Translation Lookaside Buffer cache, and improves performance.

Huge pages are pre-allocated when a virtual machine starts to run (dynamic allocation is disabled by default).

> **Note:**
>
> If you configure huge pages for a virtual machine, you cannot hotplug or hot uplug memory.

To configure huge pages:

1. In the **Custom Properties** tab, select **hugepages** from the custom properties list, which displays **Please select a key…** by default.
2. Enter the huge page size in KB.

   You should set the huge page size to the largest size supported by the pinned host. The recommended size for x86_64 is 1 GB.

   The huge page size has the following requirements:

   - The virtual machine's huge page size must be the same size as the pinned host's huge page size.
   - The virtual machine's memory size must fit into the selected size of the pinned host's free huge pages.
   - The NUMA node size must be a multiple of the huge page's selected size.

To enable dynamic allocation of huge pages:

1. Disable the HugePages filter in the scheduler.
2. In the `[performance]` section in `/etc/vdsm/vdsm.conf` set the following:

   use_dynamic_hugepages = true

Hot Plugging Devices on Virtual Machines

You can enable or disable devices while a virtual machine is running.

Hot Plugging vCPUs

Hot plugging vCPUs means enabling or disabling devices while a virtual machine is running.

> **Note:**
>
> Hot unplugging a vCPU is only supported if the vCPU was previously hot plugged. A virtual machine's vCPUs cannot be hot unplugged to less vCPUs than it was originally created with.

Before you can hot plug vCPUs, you must meet the following prerequisites:

- The virtual machine's operating system must be explicitly set and must support CPU hot plug. For details, see oVirt Documentation.
- The virtual machine must have at least four vCPUs
- Windows virtual machines must have the guest agents installed. See *Windows Virtual Machines Lose Functionality Due To Deprecated Guest Agent* in the *Known Issues* section of the Oracle Linux Virtualization Manager: Release Notes for more information.

Create vm with 4 cpus Hotplug 2 more (cpu count 6) Hot unplug cpus that you added (cpu count 4) Note that only previously hot plugged CPUs can be hot unplugged

To hot plug a vCPU:

1. Click **Compute** and then select **Virtual Machines**.
2. Select a virtual machine that is running and click **Edit**.
3. Click the **System** tab.
4. Change the value of **Virtual Sockets** as required.
5. Click **OK**.

Hot Plugging Virtual Memory

Hot plugging memory means enabling or disabling devices while a virtual machine is running. Each time you hot plug memory, it appears as a new memory device under **Vm Devices** on the virtual machine's details page, up to a maximum of 16.

When you shut down and restart a virtual machine, these devices are cleared from **Vm Devices** without reducing the virtual machine's memory, allowing you to hot plug more memory devices.

> **Note:**
>
> This feature is only available for the self-hosted engine Engine virtual machine, which is currently a technology preview feature.

To hot plug virtual memory:

1. Click **Compute** and then select **Virtual Machines**.
2. Select a virtual machine that is running and click **Edit**.
3. Click the **System** tab.

4. Enter a new number for **Memory Size**. You can add memory in multiples of 256 MB. By default, the maximum memory allowed for the virtual machine is set to 4x the memory size specified.
5. Click **OK**.

   The **Pending Virtual Machine changes** window opens.
6. Click **OK** for the changes to take place immediately or check **Apply later** and then **OK** to wait for the next virtual machine restart.
7. Click **OK**.

   You can see the virtual machine's updated memory in the **Defined Memory** field of the virtual machine's details page and you can see the added memory under **Vm Devices**.

You can also hot unplug virtual memory, but consider:

- Only memory added with hot plugging can be hot unplugged.
- The virtual machine's operating system must support memory hot unplugging.
- The virtual machine must not have a memory balloon device enabled.

To hot unplug virtual memory:

1. Click **Compute** and then select **Virtual Machines**.
2. Click on the name of a virtual machine that is running.

   The virtual machine's details page opens.
3. Click **Vm Devices**.
4. In the **Hot Unplug** column, click **Hot Unplug** beside any memory device you want to remove.

   The **Memory Hot Unplug** windows opens with a warning.
5. Click **OK**.

   Under **General** on the virtual machine details page, the **Physical Memory Guaranteed** value for the virtual machine is decremented automatically.

5

Upgrading Your Environment to 4.4

You can upgrade Oracle Linux Virtualization Manager from 4.3.10 to 4.4 by upgrading your engine or self-hosted engine and KVM hosts. Upgrading from 4.3 to 4.4 with Gluster 8 storage in your environment is supported. However, if your 4.3 installation uses Gluster 6 storage, you must upgrade to Gluster 8 before upgrading to 4.4.

Optionally, you can use the Leapp utility to upgrade the engine from 4.3 to 4.4. Refer to the My Oracle Support (MOS) article *Leapp Upgrade from OLVM 4.3.10 to 4.4.x (Doc ID 2900355.1)* for instructions.

If you want to update your engine, KVM hosts, or self-hosted engine within versions, such as from 4.4.8 to 4.4.10, see Updating Engines and KVM Hosts within Versions.

> **Important:**
>
> Upgrading from 4.2 to 4.4 is not supported. You must first upgrade to 4.3.10.

Before You Begin

Before upgrading your environment, consider the following:

- Plan for any necessary virtual machine downtime. After you update the clusters' compatibility versions, a new hardware configuration is automatically applied to each virtual machine once it reboots. You must reboot any running or suspended virtual machines as soon as possible to apply the configuration changes.
- Ensure your environment meets the requirements for 4.4. See Requirements and Scalability Limits in the Oracle Linux Virtualization Manager: Release Notes.
- When upgrading the engine, Oracle recommends using one of the existing hosts. If you decide to use a new host, you must assign a unique name to the new host and then add it to the existing cluster before upgrading.

Updating Engine or Self-Hosted Engine to 4.3

Before upgrading to 4.4, you must update the engine or self-hosted engine to the latest version of 4.3.

1. ***(Self-hosted engine only)*** Migrate virtual machines and enable global maintenance mode.
    - Migrate all other virtual machines off the host that contains the self-hosted engine virtual machine. Move the virtual machines to another host within the same cluster. During the upgrade, the host can only contain the self-hosted engine virtual machine (no other virtual machines can be on the host). Use Live Migration to minimize virtual machine down-time. See Migrating Virtual Machines between Hosts.
    - Enable global maintenance mode:

a. Log in to your self-hosted engine host and enable global maintenance mode:

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

b. Confirm that the environment is in global maintenance mode before proceeding:

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

c. You should see the following message:

```
!! Cluster is in GLOBAL MAINTENANCE mode !!
```

2. Install and update `oracle-ovirt-release-el7.rpm` to the latest version. This package includes the necessary repositories.
3. On the engine machine, check for updated packages:

```
# engine-upgrade-check
```

4. Update the setup packages:

```
# dnf update ovirt\*setup\*
```

5. Update the engine:

```
# engine-setup
```

> **Important:**
>
> The update process might take some time. Do not stop the process before it completes.

The `engine-setup` script:

- Prompts you with some configuration questions
- Stops the `ovirt-engine` service
- Downloads and installs the updated packages
- Backs up and updates the database
- Performs post-installation configuration
- Starts the `ovirt-engine` service

When the script completes successfully, you'll see:

```
Execution of setup completed successfully
```

> **Note:**
>
> The `engine-setup` script displays configuration values supplied during the initial engine installation process. These values may not be up to date, if you used `engine-config` after the initial installation. However, `engine-setup` will not overwrite your updated values.
>
> For example, if you used `engine-config` to update `SANWipeAfterDelete` to `true` after installation, `engine-setup` will display "Default SAN wipe after delete: False" in the configuration preview. However, it will not apply this value. It will keep the `SANWipeAfterDelete` to `true` setting.

6. Update the base operating system and any optional packages installed on the engine:

   ```
   # dnf update
   ```

7. If any kernel packages were updated, reboot the machine to complete the update.
8. You can now upgrade the engine or self-hosted engine to 4.4.

Upgrading Engine or Self-Hosted Engine

The 4.4 engine is only supported on Oracle Linux 8.5 or later. You need a clean installation of Oracle Linux 8.5 or later and the 4.4 engine, even if you are using the same physical machine that you use to run the 4.3 engine.

The upgrade process requires restoring the 4.3 engine backup files onto the 4.4 engine machine.

> **Note:**
>
> Connected hosts and virtual machines can continue to work while you upgrade the engine.

Prerequisites

> **NOT_SUPPORTED:**
>
> Before you begin the upgrade process, ensure you have updated your engine or self-hosted engine to the latest version of 4.3.

There are upgrade prerequisites that are common to both a standard environment and a self-hosted engine environment. There are additional prerequisites for a self-hosted engine environment.

For all environments:

- The engine must be updated to the latest version of 4.3.

- All data centers and clusters in the environment must have the cluster compatibility level set to version 4.2 or 4.3.
- All virtual machines in the environment must have the cluster compatibility level set to version 4.3.
- If you use an external CA to sign HTTPS certificates, follow the steps in Replacing the Oracle Linux Virtualization Manager Apache SSL Certificate. The backup and restore include the 3rd-party certificate, so you should be able to log in to the Administration portal after the upgrade. Ensure the CA certificate is added to system-wide trust stores of all clients to ensure the foreign menu of virt-viewer works.

Additionally, for self-hosted engine environments:

- Make note of the MAC address of the self-hosted engine if you are using DHCP and want to use the same IP address. The deploy script prompts you for this information.
- During the deployment you need to provide a new storage domain for the engine machine. The deployment script renames the 4.3 storage domain and retains its data to enable disaster recovery.
- Set the cluster scheduling policy to `cluster_maintenance` in order to prevent automatic virtual machine migration during the upgrade.

> **Caution:**
>
> In an environment with multiple highly available self-hosted engine nodes, you need to detach the storage domain hosting the version 4.3 engine after upgrading the engine to 4.4. Use a dedicated storage domain for the 4.4 self-hosted engine deployment.

Upgrading the Engine

> **Important:**
>
> If you have a self-hosted engine environment, see Upgrading the Self-Hosted Engine.

1. Verify the prerequisites.
2. Log in to the engine machine.
3. Back up the 4.3 engine environment.

   ```
   # engine-backup --scope=all --mode=backup --file=backup.bck --
   log=backuplog.log
   ```

4. Copy the backup file to a storage device outside of the Oracle Linux Virtualization Manager environment.
5. See Installing the Engine in the Oracle Linux Virtualization Manager: Getting Started Guide.

Install Oracle Linux 8.5 or later and complete the steps to install the 4.4 engine, including running the command `dnf install ovirt-engine`, but do not run `engine-setup`.

6. Copy the backup file to the 4.4 engine machine and restore it.

```
# engine-backup --mode=restore --file=backup.bck --provision-all-databases
```

If the backup contained grants for extra database users, this command creates the extra users with random passwords. You must change these passwords manually if the extra users require access to the restored system.

7. Install optional extension packages if they were installed on the 4.3 engine machine.

```
# dnf install ovirt-engine-extension-aaa-ldap ovirt-engine-extension-aaa-misc
```

The backup and restore process does not migrate configuration settings. You must manually reapply the configuration for these package extensions.

8. Configure the engine by running the `engine-setup` command:

```
# engine-setup
```

9. Decommission the 4.3 engine machine if a different machine is used for the 4.4 engine. Two different engines must not manage the same hosts or storage.
10. You can now update the KVM hosts. Proceed to Migrating Hosts and Virtual Machines.

Upgrading the Self-Hosted Engine

> **Important:**
>
> If you do not have a self-hosted engine environment, see Upgrading the Engine.

1. Verify the prerequisites.
2. Log in to the engine machine and shut down the engine service.

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

3. Back up the 4.3 engine environment.

```
# engine-backup --scope=all --mode=backup --file=backup.bck --log=backuplog.log
```

4. Copy the backup file to a storage device outside of the Oracle Linux Virtualization Manager environment.
5. Shut down the self-hosted engine.

```
# shutdown
```

If you want to reuse the self-hosted engine virtual machine to deploy the 4.4 engine, note the MAC address of the self-hosted engine network interface before you shut it down.

6. Make sure that the self-hosted engine is shut down.

```
# hosted-engine --vm-status | grep 'Engine status|Hostname'
```

If any of the hosts report the `detail` field as `Up`, log in to that specific host and shut it down with the `hosted-engine --vm-shutdown` command.

7. Install Oracle Linux 8.5 or later on the existing host currently running the engine virtual machine to use it as the self-hosted engine deployment host. See Deploying the Self-Hosted Engine in the Oracle Linux Virtualization Manager: Getting Started Guide for more information.

   > **Note:**
   >
   > Oracle recommends that you use one of the existing hosts. If you decide to use a new host, you must assign a unique name to the new host and then add it to the existing cluster before you begin the upgrade procedure.

8. Install the self-hosted engine deployment tool.

   ```
   # dnf install ovirt-hosted-engine-setup
   ```

9. Copy the backup file to the host.

10. Log in to the Engine host and deploy the self-hosted engine with the backup file:

    ```
    # hosted-engine --deploy --restore-from-file=/path/backup.bck
    ```

    tmux enables the deployment script to continue if the connection to the server is interrupted, so you can reconnect and attach to the deployment and continue. Otherwise, if the connection is interrupted during deployment, the deployment fails.

    To run the deployment script using tmux, enter the tmux command before you run the deployment script:

    ```
    # tmux
    # hosted-engine --deploy --restore-from-file=backup.bck
    ```

    The deployment script automatically disables global maintenance mode and calls the HA agent to start the self-hosted engine virtual machine. The upgraded host with the 4.4 self-hosted engine reports that HA mode is active, but the other hosts report that global maintenance mode is still enabled as they are still connected to the old self-hosted engine storage.

11. Detach the storage domain that hosts the 4.3 engine machine. For details, see Detaching a Storage Domain from a Data Center.

12. Log in to the Engine virtual machine and shut down the engine service.

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

13. Install optional extension packages if they were installed on the 4.3engine machine.

    ```
    # dnf install ovirt-engine-extension-aaa-ldap ovirt-engine-extension-aaa-misc
    ```

    > **Note:**
    >
    > The configuration for these package extensions must be manually reapplied because they are not migrated as part of the backup and restore process.

14. Configure the engine by running the engine-setup command:

    ```
    # engine-setup
    ```

    The 4.4 engine is now installed, with the cluster compatibility version set to 4.2 or 4.3, whichever was the preexisting cluster compatibility version.

15. You can now update the self-hosted engine host and then any standard hosts. Proceed to Migrating Hosts and Virtual Machines.

Migrating Hosts and Virtual Machines

Migrate hosts and virtual machines from 4.3 to 4.4 such that you minimize the downtime of virtual machines in your environment. This process requires migrating all virtual machines from one host so as to make that host available to upgrade to 4.4. After the upgrade, you can reattach the host to the engine.

> **NOT_SUPPORTED:**
>
> When installing or reinstalling the host's operating system, Oracle strongly recommends that you first detach any existing non-OS storage from the host to avoid potential data loss from accidental initialization of these disks.
>
> Oracle Linux Virtualization Manager 4.3 and 4.4 are based on Oracle Linux 7 and Oracle Linux 8, respectively, which have different kernel versions with different CPU flags and microcodes. This can cause problems in migrating CPU-passthrough virtual machines and they might not migrate properly.

Procedure

1. Verify the 4.4 engine is installed and running.
2. Verify the compatibility level of all data centers and clusters in the environment is 4.2 or 4.3.
3. Pick a host to upgrade and migrate that host's virtual machines to another host in the same cluster.

   You can use Live Migration to minimize virtual machine downtime. See Migrating Virtual Machines between Hosts.
4. Put the host into maintenance mode and remove the host from the engine.

   See Moving a Host to Maintenance Mode and Removing a Host.
5. Install Oracle Linux 8.5 or later and install the appropriate packages to enable the host for 4.4. Even if you are using the same physical machine as for 4.3, your 4.4 hosts require a clean installation of Oracle Linux 8.5 or later.

> **Caution:**
>
> **Before you install**, review the prerequisites and follow the instructions in *Configuring a KVM Host* in the Oracle Linux Virtualization Manager: Getting Started Guide. Ensure you select **Minimal Install** as the base environment for the installation. If you do not, your hosts will have incorrect qemu and libvirt versions, incorrect repositories configured, and no access to virtual machine consoles.

6. Add this host to the engine, assigning it to the same cluster. You can now migrate virtual machines onto this host.

   See Adding a KVM Host in the Oracle Linux Virtualization Manager: Getting Started Guide.

7. Repeat these steps to migrate virtual machines and upgrade hosts for the rest of the hosts in the same cluster, one by one, until all are running 4.4.

8. Update the compatibility version to 4.6

   See Changing Data Center and Cluster Compatibility Versions After Upgrading.

Changing Data Center and Cluster Compatibility Versions After Upgrading

Oracle Linux Virtualization Manager data centers and clusters have a compatibility version. The data center compatibility version indicates the version of Oracle Linux Virtualization Manager that the data center is intended to be compatible with. The cluster compatibility version indicates the features supported by all of the hosts in the cluster. The cluster compatibility is set according to the version of the least capable host operating system in the cluster.

> **Important:**
>
> Although the Oracle Linux Virtualization Manager version is 4.4, the corresponding compatibility version is 4.6.

Compatibility Version Restrictions

Consider these restrictions to ensure you do not have issues with compatibility versions after you upgrade.

Data Center Compatibility Versions

The data center compatibility level is the minimum version you can use for all clusters in your data center. For example:

- If your data center compatibility level is 4.6, you can only have clusters with compatibility level 4.6.
- If your data center compatibility level is 4.3, you can have 4.3 and 4.6 compatibility level clusters.

Cluster Compatibility Versions

The cluster compatibility level is the minimum version of any host you add to the cluster. For example:

- If you have a 4.3 compatibility version cluster, you can add 4.3 or 4.4 hosts.
- If you have a 4.6 compatibility version cluster, you can only add 4.4 hosts.

Possible Errors When Changing Compatibility Versions

- If you try to change the data center compatibility version from 4.3 to 4.6 when you have a 4.3 compatibility version cluster, you get the following error:

  ```
  Cannot update Data Center compatibility version to a value that is greater than
  its
  cluster's version. The following clusters should be upgraded: [clustername]
  ```

- If you try to change the cluster compatibility version from 4.3 to 4.6 when you have 4.3 hosts running, you get the following error:

  ```
  Error while executing action: Cannot change Cluster Compatibility Version to
  higher version
  when there are active Hosts with lower version. -Please move Host [hostname] with
  lower
  version to maintenance first.
  ```

- When you put a 4.3 host in maintenance mode, you can change the cluster and then data center compatibility version to 4.6. However, the host shows non-operational with the following event:

  ```
  Host [hostname] is compatible with versions ([version levels]) and cannot join
  Cluster
  [clustername] which is set to version [version level].
  ```

Possible Issues When Adding Hosts

- If you attempt to add a new 4.3 host to a 4.4 engine you might get an error message in the ansible log similar to the following:

  ```
  ValueError: need more than 1 value to unpack.
  ```

  To resolve this error, log onto the host as root and execute the following two commands and then attempt to add the host to the engine again.

  ```
  # sed 's|enabled=1|enabled=0|g' /etc/yum/pluginconf.d/enabled_repos_upload.conf -i
  # sed 's|enabled=1|enabled=0|g' /etc/yum/pluginconf.d/package_upload.conf -i
  ```

> **Note:**
>
> The preferred approach after upgrading your engine to 4.4 is to upgrade all hosts to 4.4 and then change the cluster compatibility to 4.6. You can then add new hosts as 4.4 hosts.

Changing Cluster Compatibility Versions

To change the cluster compatibility version, you must have first upgraded all the hosts in your cluster to a level that supports your desired compatibility level.

1. Verify all hosts are running a version level that supports your desired compatibility level. See Compatibility Version Restrictions.
2. In the **Administration Portal**, go to **Compute** and click **Clusters**.
3. Select the cluster to change and click **Edit**.
4. From the **Edit Cluster** dialog box, select **General**.
5. For **Compatibility Version**, select desired value and click **OK**.
6. On the **Change Cluster Compatibility Version** confirmation window, click **OK**.

> **Important:**
>
> You might get an error message warning that some virtual machines and templates are incorrectly configured. To fix this error, edit each virtual machine manually. The Edit Virtual Machine window provides additional validations and warnings that show what to correct. Sometimes the issue is automatically corrected and the virtual machine's configuration just needs to be saved again. After editing each virtual machine, you will be able to change the cluster compatibility version.

7. Update the cluster compatibility version of all running or suspended virtual machines by restarting them from within the Administration Portal.

> **Note:**
>
> Virtual machines continue to run in the previous cluster compatibility level until you restart them. The **Next-Run** icon (triangle with an exclamation mark) indentifies virtual machines that require a restart. However, the self-hosted engine virtual machine does not need to be restarted.
>
> You cannot change the cluster compatibility version of a virtual machine snapshot that is in preview. You must first commit or undo the preview.

Changing Data Center Compatibility Versions

After updating the compatibility version of all clusters in a data center, you can change the compatibility version of the data center itself.

1. Verify that all clusters are at the proper compatibility version. If not, change the version of the clusters, see Changing Cluster Compatibility Versions.
2. In the **Administration Portal**, go to **Compute** and click **Data Centers**.
3. Select the data center to change and click **Edit**.
4. From the **Edit Data Center** dialog box, change the **Compatibility Version** to the desired value and then click **OK**.
5. On the **Change Data Center Compatibility Version** confirmation window, click **OK**.

6

Updating Engines and KVM Hosts within Versions

You can update your engine, KVM hosts, and self-hosted engine within versions, such as from 4.4.8 to 4.4.10.

If you want to move from one version to another, such as 4.3.10 to 4.4, this is considered an upgrade. See Upgrading Your Environment to 4.4.

Updating the Engine

> **Important:**
>
> If the update fails, the engine-setup command attempts to rollback your installation to its previous state. If you encounter a failed update, detailed instructions display explaining how to restore your installation.

To update your engine:

1. Update the release rpm:

   ```
   # dnf update oracle-ovirt-release-el8
   ```

2. Check to see if your engine is eligible to update and if there are updates for any packages.

   ```
   # engine-upgrade-check
   ...
   Upgrade available.
   ```

3. Update the setup packages and resolve dependencies.

   ```
   # dnf update ovirt\*setup\*
   ...
   Complete!
   ```

4. Run the `engine-setup` command. The update process may take some time, so allow it to complete and do not stop the process once initiated.

   ```
   # engine-setup
   ...
   [ INFO  ] Execution of setup completed successfully
   ```

   The `engine-setup` script prompts you with some configuration questions, then stops the **ovirt-engine** service, downloads and installs the updated packages, backs up and updates the database, performs post-installation configuration, and starts the **ovirt-engine** service. For more information, see Engine Configuration Options in the Oracle Linux Virtualization Manager: Getting Started.

> **Note:**
>
> When you run the engine-setup script during the installation process your configuration values are stored. During an upgrade, these stored values display when previewing the configuration and they might not be up-to-date if you ran engine-config after installation. For example, if you ran engine-config to update `SANWipeAfterDelete` to `true` after installation, engine-setup outputs `Default SAN wipe after delete: False` in the configuration preview. However, your updated values are not overwritten by engine-setup.

5. Update the base operating system and any optional packages installed.

```
# dnf update
```

> **Note:**
>
> If the update upgraded any kernel packages, reboot the system to complete the changes.

Updating the Self-Hosted Engine

Before you can update your self-hosted engine, you must place the self-hosted engine environment in global maintenance mode.

1. Log into your self-hosted engine host and enable global maintenance mode.

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

2. Confirm that the environment is in maintenance mode .

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

   You should see the following message indicating that the cluster is in maintenance mode.

```
!! Cluster is in GLOBAL MAINTENANCE mode !!
```

When you run the `engine-setup` script during the installation process your configuration values are stored. During an upgrade, these stored values display when previewing the configuration and they might not be up-to-date if you ran engine-config after installation. For example, if you ran `engine-config` to update `SANWipeAfterDelete` to `true` after installation, engine-setup outputs `Default SAN wipe after delete: False` in the configuration preview. However, your updated value of `true` is not overwritten by `engine-setup`.

1. Update the release rpm:

```
# dnf update oracle-ovirt-release-el8
```

2. Log in to the engine virtual machine and check to see if your engine is eligible to update and if there are updates for any packages.

```
# engine-upgrade-check
...
Upgrade available.
```

3. Update the setup packages and resolve dependencies.

```
# dnf update ovirt\*setup\*
...
Complete!
```

4. Run the `engine-setup` command.

```
# engine-setup
...
[ INFO  ] Execution of setup completed successfully
```

The `engine-setup` script prompts you with some configuration questions, then stops the **ovirt-engine** service, downloads and installs the updated packages, backs up and updates the database, performs post-installation configuration, and starts the **ovirt-engine** service. For more information, see Engine Configuration Options in the Oracle Linux Virtualization Manager: Getting Started.

5. Update the base operating system and any optional packages installed on the engine.

```
# dnf update
```

> **Important:**
>
> If any kernel packages were updated, disable global maintenance mode and reboot the machine to complete the update.

After you update your self-hosted engine, you must disable global maintenance mode for the self-hosted engine environment.

1. Log in to the engine virtual machine and shut it down.
2. Log in to the self-hosted engine host and disable global maintenance mode.

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

> **Note:**
>
> When you exit global maintenance mode, `ovirt-ha-agent` starts the engine virtual machine, and then the engine automatically starts. This process can take up to ten minutes.

3. Confirm that the environment is running.

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

The status information shows **Engine Status** and its value should be:

```
{"health": "good", "vm": "up", "detail": "Up"}
```

When the virtual machine is still booting and the engine hasn't started yet, the **Engine status** is:

```
{"reason": "bad vm status", "health": "bad", "vm": "up", "detail": "Powering up"}
```

If this happens, wait a few minutes and try again.

Updating KVM Hosts

Before you update a KVM host, here are a few considerations.

- If migration is enabled at the cluster level, virtual machines are automatically migrated to another host in the cluster.
- The cluster must contain more than one host before performing an update.
- Do not attempt to update all hosts at the same time because one host must remain available to perform Storage Pool Manager (SPM) tasks.
- The cluster must have sufficient memory reserve in order for its hosts to perform maintenance. If a cluster lacks sufficient memory, the virtual machine migration hangs and then fails. You can reduce the memory usage of virtual machine migration by shutting down some or all virtual machines before updating the host.
- You cannot migrate a virtual machine using a vGPU to a different host. Virtual machines with vGPUs installed must be shut down before updating the host.

To update a KVM host, complete the following steps in the Administration Portal:

1. In the Administration portal, go to **Compute** and then click **Hosts**.
2. In the **Hosts** pane, select a host, click **Installation** and then **Check for Upgrade**.
3. From the Upgrade Host window, click **OK**.

   The engine checks the KVM host to see if it requires an upgrade.
4. To proceed with the upgrade, click **Installation** and then **Upgrade**.
5. From the Upgrade Host window, click **OK** to begin the upgrade process.

   On the **Hosts** pane you can watch the host transition through the upgrade stages: **Maintenance**, **Installing**, **Up**. The host is rebooted after the upgrade and displays a status of Up if successful. If any virtual machines were migrated off the host, they are migrated back.

   > **Note:**
   >
   > If the update fails, the host's status changes to **Install Failed** and you must click **Installation** and then **Upgrade** again.
6. **(Optional)** Repeat the previous steps for any KVM host in your environment that you want to upgrade or update.

7

Disaster Recovery

Oracle Linux Virtualization Manager supports active-active and active-passive disaster recovery solutions to ensure that environments can recover when a site outage occurs. Both solutions support two sites and require replicated storage.

Active-Active Disaster Recovery

Active-active disaster recovery uses a stretch cluster configuration. This means that there is a single Oracle Linux Virtualization Manager environment with a cluster that contains hosts capable of running the required virtual machines in the primary and secondary site. The virtual machines in the primary site automatically migrate to hosts in the secondary site if an outage occurs. However, the environment must meet latency and networking requirements.

Active-Passive Disaster Recovery

Active-passive disaster recovery is a site-to-site failover solution. Two separate Oracle Linux Virtualization Manager environments are configured: the active primary environment and the passive secondary (backup) environment. With active-passive disaster recovery, you must manually execute failover and failback (when needed) both of which are performed using Ansible.

> **Important:**
>
> When using clustering applications, such as RAC, Pacemaker/Corosync, set virtual machines to Kill for Resume Behaviour (which you can find in the edit VM dialog under High Availability). Otherwise, the clustering applications might try to fence a suspended or paused virtual machine.

Active-Active Disaster Recovery

Oracle Linux Virtualization Manager supports an active-active disaster recovery failover configuration that can span two sites, both of which are active. If the primary site becomes unavailable, the Oracle Linux Virtualization Manager environment smoothly transitions to the secondary site to ensure business continuity.

To support active-active failover, you must configure a stretch cluster where hosts capable of running all the virtual machines in the cluster are located in the primary and secondary site. All the hosts belong to the same Oracle Linux Virtualization Manager cluster. You can implement a stretched cluster configuration using a self-hosted engine environment or a standalone Engine environment.

With active-active disaster recovery you must also have replicated storage that is writable on both sites. This enables virtual machines to migrate between sites and continue running on the site's storage.

Virtual machines migrate to the secondary site if the primary site becomes unavailable. When the primary site becomes available and the storage is replicated in both sites, virtual machines automatically failback.

To ensure virtual machine failover and failback works, you must configure:

- virtual machines for highly availability, and each virtual machine must have a lease on a target storage domain to ensure the virtual machine can start even without power management.
- soft enforced virtual machine to host affinity to ensure the virtual machines only start on the selected hosts.

Network Considerations

All hosts in the stretch cluster must be on the same broadcast domain over a Layer 2 (L2) network, which means that connectivity between the two sites needs to be L2.

The maximum latency requirements between the sites across the L2 network is different for the standalone Engine environment and the self-hosted engine environment:

- A maximum latency of 100ms is required for the standalone Engine environment
- A maximum latency of 7ms is required for self-hosted engine environment

Storage Considerations

The storage domain for Oracle Linux Virtualization Manager can be either block devices (iSCSI or FCP) or a file system (NAS/NFS or GlusterFS).

Both sites require synchronously replicated storage that is writable with shared L2 network connectivity to allow virtual machines to migrate between sites and continue running on the site's storage. All storage replication options supported by Oracle Linux 8 and later can be used in the stretch cluster.

For more information, see the storage topics in the Administration Guide and the Architecture and Planning Guide.

Configuring a Standalone Engine Stretch Cluster Environment

Before you begin configuring your standalone engine environment for a stretch cluster, review the following prerequisites and limitations:

- A writable storage server in both sites with L2 network connectivity.
- Real-time storage replication service to duplicate the storage.
- Maximum 100ms latency between sites.

  The Engine must be highly available for virtual machines to failover and failback between sites. If the Engine goes down with the site, the virtual machines do not failover.
- The standalone Engine is only highly available when managed externally, for example:
  - As a highly available virtual machine in a separate virtualization environment
  - In a public cloud

To configure a standalone engine stretch cluster:

1. Install and configure the Oracle Linux Virtualization Manager engine.

For more information, see Installation and Configuration in the Oracle Linux Virtualization Manager: Getting Started Guide.

2. Install hosts in each site and add them to the cluster.

   For more information, see Configuring a KVM Host in the Oracle Linux Virtualization Manager: Getting Started Guide.

3. Configure the storage pool manager (SPM) priority to be higher on all hosts in the primary site to ensure SPM failover to the secondary site occurs only when all hosts in the primary site are unavailable.

   For more information, see Storage Pool Manager in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

4. Configure all virtual machines that need to failover as highly available and ensure that a virtual machine has a lease on the target storage domain.

   For more information, see Optimizing Clusters, Hosts and Virtual Machines.

5. Configure virtual machine to host soft affinity and define the behavior you expect from the affinity group.

   For more information, see Affinity Groups in the oVirt Virtual Machine Management Guide.

   Important:

   With VM Affinity Rule **Enforcing** enabled (shown as **Hard** in the list of Affinity Groups), the system does not migrate a virtual machine to a host different from where the other virtual machines in its affinity group are running. For more information, see Virtual Machine Issues in the Oracle Linux Virtualization Manager: Release Notes.

The active-active failover can be manually performed by placing the main site's hosts into maintenance mode.

Configuring a Self-Hosted Engine Stretch Cluster Environment

Before you begin configuring your self-hosted engine environment for a stretch cluster, review the following prerequisites and limitations:

- A writable storage server in both sites with L2 network connectivity
- Real-time storage replication service to duplicate the storage
- Maximum 7ms latency between sites

To configure a self-hosted engine stretch cluster:

1. Deploy the Oracle Linux Virtualization Manager self-hosted engine.

   For more information, see Self-hosted Engine Deployment in the Oracle Linux Virtualization Manager: Getting Started Guide.

2. Optionally, install additional hosts in each site and add them to the cluster.

   For more information, see Adding a KVM Host in the Oracle Linux Virtualization Manager: Getting Started Guide.

3. Configure the storage pool manager (SPM) priority to be higher on all hosts in the primary site to ensure SPM failover to the secondary site occurs only when all hosts in the primary site are unavailable.

   For more information, see Storage Pool Manager in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

4. Configure all virtual machines that need to failover as highly available and ensure that a virtual machine has a lease on the target storage domain.

   For more information, see Optimizing Clusters, Hosts and Virtual Machines.

5. Configure a virtual machine to host soft affinity and define the affinity group's behaviour.

   For more information, see Affinity Groups in the oVirt Virtual Machine Management Guide.

> **Important:**
>
> With VM Affinity Rule **Enforcing** enabled (shown as **Hard** in the list of Affinity Groups), the system does not migrate a virtual machine to a host different from where the other virtual machines in its affinity group are running. For more information, see Virtual Machine Issues in the Oracle Linux Virtualization Manager: Release Notes.

The active-active failover can be manually performed by placing the main site's hosts into maintenance mode.

Active-Passive Disaster Recovery

Oracle Linux Virtualization Manager active-passive disaster recovery solution can span two sites. If the primary site becomes unavailable, the Oracle Linux Virtualization Manager environment can be forced to failover to the secondary (backup) site.

Failover is achieved by configuring a secondary site with:

- An active Oracle Linux Virtualization Manager Engine.
- A data center and clusters.
- Networks with the same general connectivity as the primary site.
- Active hosts capable of running critical virtual machines after failover.

> **Important:**
>
> You must ensure that the secondary environment has enough resources to run the failed over virtual machines, and that both the primary and secondary environments have identical Engine versions, data center and cluster compatibility levels, and PostgreSQL versions.
>
> Storage domains that contain virtual machine disks and templates in the primary site must be replicated. These replicated storage domains must not be attached to the secondary site.

The failover and failback processes are executed manually using Ansible playbooks that map entities between the sites and manage the failover and failback processes. The mapping file instructs the Oracle Linux Virtualization Manager components where to failover or failback to.

Network Considerations

You must ensure that the same general connectivity exists in the primary and secondary sites. If you have multiple networks or multiple data centers then you must use an empty network mapping in the mapping file to ensure that all entities register on the target during failover.

Storage Considerations

The storage domain for Oracle Linux Virtualization Manager can be made of either block devices (iSCSI or FCP) or a file system (NAS/NFS or GlusterFS). Local storage domains are unsupported for disaster recovery.

Your environment must have a primary and secondary storage replica. The primary storage domain's block devices or shares that contain virtual machine disks or templates must be replicated. The secondary storage must not be attached to any data center and is added to the backup site's data center during failover.

If you are implementing disaster recovery using a self-hosted engine, ensure that the storage domain used by the self-hosted engine's Engine virtual machine does not contain virtual machine disks because the storage domain will not failover.

You can use any storage solutions that have replication options supported by Oracle Linux 8 and later.

Important:

Metadata for all virtual machines and disks resides on the storage data domain as OVF_STORE disk images. This metadata is used when the storage data domain is moved by failover or failback to another data center in the same or different environment.

By default, the metadata is automatically updated by the Engine in 60 minute intervals. This means that you can potentially lose all data and processing completed during an interval. To avoid such loss, you can manually update the metadata from the Administration Portal by navigating to the storage domain section and clicking **Update OVFs**. Or, you can modify the Engine parameters to change the update frequency, for example:

```
# engine-config -s OvfUpdateIntervalInMinutes=30 && systemctl restart ovirt-
engine
```

For more information, see the Storage topics in the Oracle Linux Virtualization Manager: Administration Guide and the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

Creating the Ansible Playbooks

You use Ansible to initiate and manage the active-passive disaster recovery failover and failback through Ansible playbooks that you create. For more information about Ansible playbooks, see the Ansible documentation.

Before you begin creating your Ansible playbooks, review the following prerequisites and limitations:

- Primary site has a fully functioning Oracle Linux Virtualization Manager environment.
- A backup environment in the secondary site with the same data center and cluster compatibility level as the primary environment. The backup environment must have:
  - An Oracle Linux Virtualization Manager Engine
  - Active hosts capable of running the virtual machines and connecting to the replicated storage domains
  - A data center with clusters
  - Networks with the same general connectivity as the primary site
- Replicated storage that contains virtual machines and templates not attached to the secondary site.
- The `ovirt-ansible-collection` package must be installed on the highly available Ansible Engine machine to automate the failover and failback.
- The machine running the Ansible Engine must be able to use SSH to connect to the Engine in the primary and secondary site.

> **Note:**
>
> We recommended that you create environment properties that exist in the primary site, such as affinity groups, affinity labels, users, on the secondary site. The default behaviour of the Ansible playbooks can be configured in the `/usr/share/ansible/collections/ansible_collections/ovirt/ovirt/roles/disaster_recovery/defaults/main.yml` file.

You must create the following required Ansible playbooks:

- Playbook that creates the file to map entities on the primary and secondary sites
- Failover playbook
- Failback playbook

The playbooks and associated files that you create must reside in `/usr/share/ansible/collections/ansible_collections/ovirt/ovirt/roles/disaster_recovery` on the Ansible machine that is managing the failover and failback. If you have multiple Ansible machines that can manage it, ensure that you copy the files to all of them.

After configuring active-passive disaster recovery, you should test and verify the configuration. See Testing the Active-Passive Configuration.

Simplifying Ansible Tasks Using the `ovirt-dr` Script

You can use the `ovirt-dr` script, located in **/usr/share/ansible/collections/ansible_collections/ovirt/ovirt/roles/disaster_recovery**, to simplify these Ansible tasks:

- Generating a `var` mapping file of the primary and secondary sites for failover and fallback
- Validating the `var` mapping file
- Executing failover on a target site
- Executing failback from a target site to a source site

The following is an example of the `ovirt-dr` script:

```
# ./ovirt-dr generate/validate/failover/failback

[--conf-file=dr.conf]
[--log-file=ovirt-dr-log_number.log]
[--log-level=DEBUG/INFO/WARNING/ERROR]
```

You optionally can make the following customizations:

- Set parameters for the script's actions in the configuration file: **/usr/share/ansible/collections/ansible_collections/ovirt/ovirt/roles/disaster_recovery/files/dr.conf**.
- Change location of the configuration file using the `--conf-file` option
- Set location of log file using the `--log-file` option
- Set level of logging detail using the `--log-level` option

Generating the Mapping File Using an Ansible Playbook

The Ansible playbook used to generate the mapping file prepopulates the file with the primary site's entities. Then, you need to manually add to the file the backup site's entities, such as IP addresses, cluster, affinity groups, affinity label, external LUN disks, authorization domains, roles, and vNIC profiles.

> **Important:**
>
> Generating the mapping file will fail if you have any virtual machine disks on the self-hosted engine's storage domain. Also, the generated mapping file will not contain an attribute for this storage domain because it must not be failed over.

To create the mapping file, complete the following steps.

1. Create an Ansible playbook using a yaml file (such as `dr-olvm-setup.yml`) to generate the mapping file. For example:

```
---
- name: Setup oVirt environment
  hosts: localhost
  connection: local
  vars:
```

```
    site: https://manager1.mycompany.com/ovirt-engine/api
    username: admin@internal
    password: Mysecret1
    ca: /etc/pki/ovirt-engine/ca.pem
    var_file: disaster_recovery_vars.yml
  roles:
    - disaster_recovery
  collections:
    - ovirt.ovirt
```

For extra security you can encrypt your Engine password in a `.yml` file.

2. Run the Ansible command to generate the mapping file. The primary site's configuration will be prepopulated.

```
# ansible-playbook dr-olvm-setup.yml --tags "generate_mapping"
```

3. Configure the generated mapping .yml file with the backup site's configuration. For more information, see Mapping File Attributes.

If you have multiple Ansible machines that can perform failover and failback, then copy the mapping file to all relevant machines.

Creating Failover and Failback Playbooks

Before creating the failover and failback playbooks, ensure you have created and configured the mapping file, which must be added to the playbooks.

To create the failover and failback playbooks, complete the following steps.

1. Optionally, define a password file (for example `passwords.yml`) to store the Engine passwords of the primary and secondary site, for example:

```
---
# This file is in plain text, if you want to
# encrypt this file, please execute following command:
#
# $ ansible-vault encrypt passwords.yml
#
# It will ask you for a password, which you must then pass to
# ansible interactively when executing the playbook.
#
# $ ansible-playbook myplaybook.yml --ask-vault-pass
#
dr_sites_primary_password: primary_password
dr_sites_secondary_password: secondary_password
```

For extra security you can encrypt the password file. However, you will need to use the `--ask-vault-pass` parameter when running the playbook.

2. Create an Ansible playbook using a failover yaml file (such as `dr-olvm-failover.yml`) to failover the environment, for example:

```
---
- name: oVirt Failover
  hosts: localhost
  connection: local
  vars:
    dr_target_host: secondary
    dr_source_map: primary
  vars_files:
    - disaster_recovery_vars.yml
```

```
  roles:
     - disaster_recovery
  collections:
     - ovirt.ovirt
```

3. Create an Ansible playbook using a failback yaml file (such as `dr-olvm-failback.yml`) to failback the environment, for example:

```
---
- name: oVirt Failback
  hosts: localhost
  connection: local
  vars:
     dr_target_host: primary
     dr_source_map: secondary
  vars_files:
     - disaster_recovery_vars.yml
  roles:
     - disaster_recovery
  collections:
     - ovirt.ovirt
```

Executing a Failover

Before executing a failover, ensure you have read and understood the Network Considerations and Storage Considerations. You must also ensure that:

- the Engine and hosts in the secondary site are running.
- replicated storage domains are in read/write mode.
- no replicated storage domains are attached to the secondary site.
- a machine running the Ansible Engine that can connect via SSH to the Engine in the primary and secondary site, with the required packages and files:
  - The `ovirt-ansible-collection` package.
  - The mapping file and failover playbook.

Sanlock must release all storage locks from the replicated storage domains before the failover process starts. These locks should be released automatically approximately 80 seconds after the disaster occurs.

To execute a failover, run the failover playbook on the Engine host using the following command:

```
# ansible-playbook dr-olvm-failover.yml --tags "fail_over"
```

When the primary site becomes active, ensure that you clean the environment before failing back. For more information, see Cleaning the Primary Site.

Cleaning the Primary Site

After you failover, you must clean the environment in the primary site before failing back to it. Cleaning the primary site's environment:

- Reboots all hosts in the primary site.
- Ensures the secondary site’s storage domains are in read/write mode and the primary site’s storage domains are in read only mode.

- Synchronizes the replication from the secondary site's storage domains to the primary site's storage domains.
- Cleans the primary site of all storage domains to be imported. This can be done manually in the Engine. For more information, see Detaching a Storage Domain from a Data Center.

For example, create a cleanup yml file (such as `dr_cleanup_primary_site.yml`):

```
---
- name: oVirt Cleanup Primary Site
  hosts: localhost
  connection: local
  vars:
     dr_source_map: primary
  vars_files:
     - disaster_recovery_vars.yml
  roles:
     - disaster_recovery
  collections:
     - ovirt.ovirt
```

Once you have cleaned the primary site, you can now failback the environment to the primary site. For more information, see Executing a Failback.

Executing a Failback

After failover, you can failback to the primary site when it is active and you have performed the necessary steps to clean the environment by ensuring:

- The primary site's environment is running and has been cleaned. For more information, see Cleaning the Primary Site.
- The environment in the secondary site is running and has active storage domains.
- The machine running the Ansible Engine that can connect via SSH to the Engine in the primary and secondary site, with the required packages and files:
  - The `ovirt-ansible-collection` package.
  - The mapping file and required failback playbook.

To execute a failback, complete the following steps.

1. Run the failback playbook on the Engine host using the following command:

   ```
   #  ansible-playbook dr-olvm-failback.yml --tags "fail_back"
   ```

2. Enable replication from the primary storage domains to the secondary storage domains.

Testing the Active-Passive Configuration

You must test your disaster recovery solution after configuring it using one of the provided options:

1. Test failover while the primary site remains active and without interfering with virtual machines on the primary site's storage domains. See Discreet Failover Test.

2. Test failover and failback using specific storage domains attached to the primary site which allows the primary site to remain active. See Discreet Failover and Failback Tests.
3. Test failover and failback for an unplanned shutdown of the primary site or an impending disaster where you have a grace period to failover to the secondary site. See Full Failover and Failback Tests.

> **Important:**
>
> Ensure that you have completed all the steps to configure your active-passive disaster recovery before running any of these tests.

Discreet Failover Test

The discreet failover test simulates a failover while the primary site and all its storage domains remain active which allows users to continue working in the primary site. To perform this test, you must disable replication between the primary storage domains and the replicated (secondary) storage domains. During this test the primary site is unaware of the failover activities on the secondary site.

This test does not allow you to test the failback functionality.

> **Important:**
>
> Ensure that no production tasks are performed after the failover. For example, ensure that email systems are blocked from sending emails to real users or redirect emails elsewhere. If systems are used to directly manage other systems, prohibit access to the systems or ensure that they access parallel systems in the secondary site.

To perform a discreet failover test, complete the following steps.

1. Disable storage replication between the primary and replicated storage domains and ensure that all replicated storage domains are in read/write mode.
2. Run the following command to failover to the secondary site:

   ```
   # ansible-playbook playbook --tags "fail_over"
   ```

3. Verify that all relevant storage domains, virtual machines, and templates are registered and running successfully on the secondary site.

To restore the environment to its active-passive state, complete the following steps.

1. Detach the storage domains from the secondary site.
2. Enable storage replication between the primary and secondary storage domains.

Discreet Failover and Failback Tests

The discreet failover and failback tests use testable storage domains that you specifically define for testing failover and failback. These storage domains must be replicated so that the

replicated storage can be attached to the secondary site which allows you to test the failover while users continue to work in the primary site.

> **Note:**
>
> You should define the testable storage domains on a separate storage server that can be offline without affecting the primary storage domains used for production in the primary site.

To perform a discreet failover test, complete the following steps.

1. Stop the test storage domains in the primary site. For example, shut down the server host or block it with a firewall rule.
2. Disable the storage replication between the testable storage domains and ensure that all replicated storage domains used for the test are in read/write mode.
3. Place the test primary storage domains into read-only mode.
4. Run the command to failover to the secondary site:

   ```
   # ansible-playbook playbook --tags "fail_over"
   ```

5. Verify that all relevant storage domains, virtual machines, and templates are registered and running successfully on the secondary site.

To perform a discreet failback test, complete the following steps.

1. Clean the primary site and remove all inactive storage domains and related virtual machines and templates. For more information, see Cleaning the Primary Site.
2. Run the command to failback to the primary site:

   ```
   # ansible-playbook playbook --tags "fail_back"
   ```

3. Enable replication from the primary storage domains to the secondary storage domains.
4. Verify that all relevant storage domains, virtual machines, and templates are registered and running successfully in the primary site.

Full Failover and Failback Tests

The full failover and failback tests allow you to simulate a primary site disaster, failover to the secondary site, and failback to the primary site. To simulate a primary site disaster, you can shut down the primary site's hosts or by add firewall rules to block writing to the storage domains.

To perform a full failover test, complete the following steps.

1. Disable storage replication between the primary and replicated storage domains and ensure that all replicated storage domains are in read/write mode.
2. Run the command to failover to the secondary site:

   ```
   # ansible-playbook playbook --tags "fail_over"
   ```

3. Verify that all relevant storage domains, virtual machines, and templates are registered and running successfully in the secondary site.

To perform a full failback test, complete the following steps.

1. Synchronize replication between the secondary site's storage domains and the primary site's storage domains. The secondary site's storage domains must be in read/write mode and the primary site's storage domains must be in read-only mode.
2. Clean the primary site and remove all inactive storage domains and related virtual machines and templates. For more information, see Cleaning the Primary Site.
3. Run the command to failback to the primary site:

   ```
   # ansible-playbook playbook --tags "fail_back"
   ```

4. Enable replication from the primary storage domains to the secondary storage domains.
5. Verify that all relevant storage domains, virtual machines, and templates are registered and running successfully on the primary site.

Mapping File Attributes

The attributes in the mapping file are used to failover and failback between the two sites in an active-passive disaster recovery solution.

- Site details

  Attributes that map the Engine details in the primary and secondary site, for example:

  ```
  dr_sites_primary_url: https://manager1.mycompany.com/ovirt-engine/api
  dr_sites_primary_username: admin@internal
  dr_sites_primary_ca_file: /etc/pki/ovirt-engine/ca.pem

  # Please fill in the following properties for the secondary site:
  dr_sites_secondary_url: https://manager2.mycompany.com/ovirt-engine/api
  dr_sites_secondary_username: admin@internal
  dr_sites_secondary_ca_file: /etc/pki/ovirt-engine/ca.pem
  ```

- Storage domain details

  Attributes that map the storage domain details between the primary and secondary site, for example:

  ```
  dr_import_storages:
  - dr_domain_type: nfs
    dr_primary_name: DATA
    dr_master_domain: True
    dr_wipe_after_delete: False
    dr_backup: False
    dr_critical_space_action_blocker: 5
    dr_warning_low_space: 10
    dr_primary_dc_name: Default
    dr_discard_after_delete: False
    dr_primary_path: /storage/data
    dr_primary_address: 10.64.100.xxx
    # Fill in the empty properties related to the secondary site
    dr_secondary_dc_name: Default
    dr_secondary_path: /storage/data2
    dr_secondary_address:10.64.90.xxx
    dr_secondary_name: DATA
  ```

- Cluster details

  Attributes that map the cluster names between the primary and secondary site, for example:

```
dr_cluster_mappings:
  - primary_name: cluster_prod
    secondary_name: cluster_recovery
  - primary_name: fc_cluster
    secondary_name: recovery_fc_cluster
```

- Affinity group details

  Attributes that map the affinity groups that virtual machines belong to, for example:

  ```
  dr_affinity_group_mappings:
  - primary_name: affinity_prod
    secondary_name: affinity_recovery
  ```

- Affinity label details

  Attributes that map the affinity labels that virtual machines belong to, for example:

  ```
  dr_affinity_label_mappings:
  - primary_name: affinity_label_prod
    secondary_name: affinity_label_recovery
  ```

- Domain authentication, authorization and accounting details

  Attributes that map authorization details between the primary and secondary site, for example:

  ```
  dr_domain_mappings:
  - primary_name: internal-authz
    secondary_name: recovery-authz
  - primary_name: external-authz
    secondary_name: recovery2-authz
  ```

- Role details

  Attributes that provide mapping for specific roles, for example:

  ```
  dr_role_mappings:
  - primary_name: admin
    Secondary_name: newadmin
  ```

- Network details

  Attributes that map the vNIC details between the primary and secondary site, for example:

  ```
  dr_network_mappings:
  - primary_network_name: ovirtmgmt
    primary_profile_name: ovirtmgmt
    primary_profile_id: 0000000a-000a-000a-000a-000000000398
    # Fill in the correlated vnic profile properties in the secondary site for
  profile 'ovirtmgmt'
    secondary_network_name: ovirtmgmt
    secondary_profile_name: ovirtmgmt
    secondary_profile_id:  0000000a-000a-000a-000a-000000000410
  ```

  If you have multiple networks or multiple data centers then you must use an empty network mapping in the mapping file to ensure that all entities register on the target during failover, for example:

  ```
  dr_network_mappings:
  # No mapping should be here
  ```

- External LUN disk details

  LUN attributes allow virtual machines to be registered with the appropriate external LUN disk after failover and failback, for example:

```
dr_lun_mappings:
- primary_logical_unit_id: 460014069b2be431c0fd46c4bdce29b66
  primary_logical_unit_alias: My_Disk
  primary_wipe_after_delete: False
  primary_shareable: False
  primary_logical_unit_description: 2b66
  primary_storage_type: iscsi
  primary_logical_unit_address: 10.35.xx.xxx
  primary_logical_unit_port: 3260
  primary_logical_unit_portal: 1
  primary_logical_unit_target: iqn.2017-12.com.prod.example:444
  secondary_storage_type: iscsi
  secondary_wipe_after_delete: False
  secondary_shareable: False
  secondary_logical_unit_id: 460014069b2be431c0fd46c4bdce29b66
  secondary_logical_unit_address: 10.35.x.xxx
  secondary_logical_unit_port: 3260
  secondary_logical_unit_portal: 1
  secondary_logical_unit_target: iqn.2017-12.com.recovery.example:444
```