Oracle Linux Virtualization Manager

Administrator's Guide

F52197-28
April 2026

Oracle Linux Virtualization Manager Administrator's Guide,

F52197-28

Contents

4 Administrative Tasks

5 Deployment Optimization

1

About the Docs

Oracle Linux Virtualization Manager Release 4.5 is based on oVirt, which is a free, open source virtualization solution. The product documentation consists of:

- **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 the environment.
- **Getting Started Guide** - How to install, configure, and get started with the Oracle Linux Virtualization Manager using standard or self-hosted configuration. It also provides information for configuring KVM hosts and deploying GlusterFS storage.
- **Administration Guide** - Provides common administrative tasks for Oracle Linux Virtualization Manager such as:
  - Setting up users and groups
  - Configuring Keycloak authentication
  - Creating data centers, clusters, and virtual machines
  - Using virtual machine templates and snapshots
  - Migrating virtual machines
  - Configuring logical and virtual networks
  - Using local, NFS, iSCSI, and FC storage
  - Backing up and restoring
  - Configuring high-availability, vCPUs, and virtual memory
  - Monitoring with event notifications and Grafana dashboards
  - Upgrading and updating the environment
  - Active-active and active-passive disaster recovery solutions

See also:

- 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 provide feedback about this documentation, please complete the Oracle Help Center feedback form.

To access Oracle Linux Virtualization Manager Release 4.4 documentation, PDFs are available at:

- Release Notes
- Getting Started Guide
- Architecture and Planning Guide

- Administration Guide

Documentation License

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.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through Oracle Accessibility Learning and Support.

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 selecting **Configure**.

From the **Configure** dialog box, you can configure global resources for the virtualization environment, such as users, roles, system permissions, scheduling policies, 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 many 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, see 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 sign in to the VM Portal and to start creating virtual machines. The procedure also applies to group accounts.

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

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

   The **Add System Permission to User** dialog box opens.
4. Select a profile from the **Search** dropdown list and select **Go**.
5. Select the checkbox next to the user or group account.
6. Under the **Role to Assign** dropdown list, select **UserRole**.
7. Select **OK**.
8. **(Optional)** Sign 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 select **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. Select **Remove**.

   The **Remove User(s)** dialog box opens.
5. Select **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, see Administering User and Group Accounts from the Command Line.

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

   The **Users** pane opens.
2. Select **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 checkbox next to the user or group to be added.
6. Select **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 selecting the name of the user under the **User Name** column or the name of the group under the **Group Name** column.
9. Select the **Permissions** tab.
10. Select **Add System Permissions**.

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

Creating a Custom Role

If you require a role that isn't 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, see the *Administration Guide* in oVirt Documentation.

To create a custom role:

1. Select **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. Select **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 to assign user permissions to.

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

6. a. For each object type, review the list of actions.

   b. Select the actions to allow for the new role.

   c. Clear any actions you want to deny.

7. Select **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 don't 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. Sign in to the host that's running the Manager.
2. Create a user account.

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

For example, create a 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.
```

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

> **Note**
>
> After creating a new user account, you must set a password so that the user can sign 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.

> **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 can't 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.

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

1. From a command line, sign in to the host 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:ssZ"
```

For example, 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. Sign in to the host that's running the Manager.
2. Edit the user account.

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

For example, 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
```

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

Viewing User Information

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

To view detailed user information:

1. From a command line, sign in to the host running the Manager.
2. Show information about a user.

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

For example:

```
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. From a command line, sign in to the host running the Manager.
2. Remove a user.

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

For example:

```
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's created when you run the `engine-setup` command.

> ⚠ **Caution**
>
> Ensure that 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. Sign in to the host that's 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 reenable 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. Sign in to the host that's running the Manager.
2. Create a group account.

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

For example:

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

adding group group1...
group added successfully
```

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

3. Add users to the group:

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

For example:

```
ovirt-aaa-jdbc-tool group-manage useradd group1 --user test1

updating user group1...
user updated successfully
```

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.

4. Show group account details.

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

For example:

```
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. Sign in to the host that's running the Manager.
2. Create the first group account.

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

For example:

```
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
```

For example:

```
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
```

For example:

```
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. Sign in to the host that's running the Manager.
2. Remove a group account.

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

For example:

```
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. From a command line, sign in to the host running the Manager.
2. Show account details.

- List all user account details.

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

  For example, the 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`
  For example, the 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. From a command line, sign in to the host 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
     ```

For example, 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
```

For example, 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. Sign in to the host that's running the Manager.
2. **(Optional)** Display all the settings that are available.

   ```
   ovirt-aaa-jdbc-tool settings show
   ```

3. Change the required settings.

   ```
   ovirt-aaa-jdbc-tool settings 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. Select **Administration** and then select **Configure**.

   The **Configure** dialog box opens.
2. Select **Scheduling Policies**.
3. Select **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 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 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 properties.
8. Select **OK** to create the scheduling policy.

3
Keycloak Integration and Management

Oracle Linux Virtualization Manager uses Keycloak as the default Identity and Access Management (IAM) service for Single Sign-On (SSO). Keycloak replaces the legacy AAA authentication method and provides modern authentication capabilities such as user federation (Active Directory or LDAP), multifactor authentication (MFA), and centralized user and group management.

This chapter describes how to configure Keycloak for Oracle Linux Virtualization Manager, including federation with Microsoft Active Directory, importing groups, mapping groups to roles in the Manager, enabling SSO access to the Monitoring Portal (Grafana), and configuring OTP-based MFA.

> **Note**
>
> In new deployments, we recommend using Keycloak SSO for authentication and user/group federation. The legacy internal domain (AAA) can be retained for recovery and backward compatibility, but isn't the recommended option for production access.

Integrate Microsoft Active Directory with Keycloak

This section describes how to federate Microsoft Active Directory (AD) with the Keycloak instance installed with Oracle Linux Virtualization Manager (the internal SSO provider). After you configure federation, you can synchronize selected AD users and groups into Keycloak and then assign permissions to those groups in the Manager.

User and group recommendations

We recommend assigning permissions to groups and placing users into those groups. You can tailor the group schema for the environment. The following groups are commonly used:

- `olvm-admins`: Global administrators (oVirt SuperUser) and `grafana-admin` access.
- `ovirt-admins`: oVirt administrators (SuperUser or delegated admin roles).
- `ovirt-users`: VM Portal users.
- `grafana-admins`, `grafana-editors`, `grafana-viewers`: Monitoring Portal (Grafana) roles.

> **Note**
>
> Grafana uses roles (Admin, Editor, Viewer). Group-based access is implemented by mapping Keycloak groups to those roles.

Prerequisites

- Microsoft Active Directory 2016 or later.

- Oracle Linux Virtualization Manager 4.5 or later installed with Keycloak enabled during `engine-setup`.
- If you use LDAPS, ensure the AD certificate chain is trusted by the Manager host.

Export the Active Directory certificate and trust it on the Manager host (LDAPS)

If the AD server uses LDAPS, you must trust the AD certificate chain on the Manager host. The exact procedure depends on the organization. The following example shows a common approach for importing the AD root certificate into the system trust store.

1. Export the AD root CA certificate (Base-64 encoded X.509) from the Windows certificate store and copy it to the Manager host.
2. On the Manager host, rename the certificate to use a `.pem` extension (if required), then copy it to the trust anchors directory:

   ```
   sudo cp activedirectory.pem /etc/pki/ca-trust/source/anchors/
   sudo chmod 600 /etc/pki/ca-trust/source/anchors/activedirectory.pem
   sudo chown root:root /etc/pki/ca-trust/source/anchors/activedirectory.pem
   sudo update-ca-trust extract
   ```

3. (Optional) Verify connectivity to the directory service. For example, install `openldap-clients` and run `ldapsearch`:

   ```
   sudo dnf install -y openldap-clients

   ldapsearch -H ldaps://ad-server-fqdn -x -LLL \
     -D "CN=bind-user,CN=Users,DC=example,DC=com" -W \
     -b "DC=example,DC=com"
   ```

Collect required information

Before you start, collect the following information:

- AD server host name (FQDN) and connection URL (LDAP or LDAPS).
- Directory base DN (for example, `DC=example,DC=com`).
- Bind account DN and password for Keycloak to query the directory.
- AD groups to synchronize (for example, `olvm-admins`, `ovirt-admins`, `ovirt-users`, `grafana-*` groups).

Configure user federation (LDAP) in Keycloak

1. Open the Keycloak administration console: `https://manager-fqdn/ovirt-engine-auth/`.
2. In the navigation menu, select **User Federation**, then select **Add provider** and choose **LDAP**.
3. Configure the LDAP provider. The following settings are commonly required for Active Directory:
   - **Enabled**: ON
   - **Console display name**: a friendly name for the directory
   - **Vendor**: Active Directory
   - **Username LDAP attribute**: `sAMAccountName`
   - **UUID LDAP attribute**: `objectGUID`

- **Connection URL**: `ldap://ad-server` or `ldaps://ad-server`
- **Users DN**: for example, `CN=Users,DC=example,DC=com`
- **Bind type**: Simple
- **Bind DN** and **Bind credential**: the bind account DN and password
- **Edit mode**: READ_ONLY

4. Select **Test connection** and **Test authentication**, and confirm both tests succeed.
5. Select **Synchronize all users** to import directory users into Keycloak.
6. Select **Save**.

Import selected Active Directory groups

To avoid importing all directory groups, configure a group mapper with an LDAP filter that matches only the groups you want to synchronize.

1. In Keycloak, select **User Federation**, select the LDAP provider, and then select the **Mappers** tab.
2. Select **Create**, then configure the mapper:
   - **Mapper type**: `group-ldap-mapper`
   - **LDAP groups DN**: the DN that contains the AD groups
   - **Group name LDAP attribute**: `cn`
   - **Membership LDAP attribute**: `member`
   - **Membership user LDAP attribute**: `sAMAccountName`
   - **LDAP filter** (example):

     ```
     (|(cn=olvm-admins)(cn=ovirt-admins)(cn=ovirt-users)(cn=grafana-admins)
     (cn=grafana-editors)(cn=grafana-viewers))
     ```

3. Select **Save**, then select **Sync LDAP groups to Keycloak**.
4. Verify that the groups are present under **Groups** in Keycloak.

Allow group lookup by base DN in internal authorization

On the Manager host, configure the authorization extension to use the directory base DN. This enables group lookup when adding federated groups in the Administration Portal.

1. On the Manager host, edit `/etc/ovirt-engine/extensions.d/internalkeycloak-authz.properties` and add the following line at the end of the file:

   ```
   config.globals.baseDN.simple_baseDN = DC=example,DC=com
   ```

2. Restart the engine:

   ```
   sudo systemctl restart ovirt-engine
   ```

Add federated groups in the Administration Portal and assign permissions

1. Sign in to the Administration Portal as `admin@ovirt`.
2. Go to **Administration** > **Users**.

3. Select **Groups**, then select **Add**.
4. From the domain list, select **internalsso** (internal Keycloak authorization provider).
5. Search for the group, select it, and add it.
6. Select the group name, open the **Permissions** tab, and assign roles (for example, assign **SuperUser** to `olvm-admins` and `ovirt-admins`).
7. Sign out and then sign in as a user from the federated directory to verify access.

> **Note**
>
> Oracle Linux Virtualization Manager might not display a federated user's first name, last name, or email address in the Administration Portal user list. This is a known issue.

Manage Users and Groups Locally in Keycloak

This section describes how to create and manage users and groups directly in Keycloak, without using Active Directory or LDAP federation. You can assign realm roles to Keycloak groups and then map those groups to roles and permissions in the Administration Portal. You can also use the same Keycloak accounts for Monitoring Portal (Grafana) Single Sign-On (SSO).

Prerequisites

- Keycloak is enabled and working for the Manager.
- You can sign in to the Keycloak administration console at `https://`*`manager-fqdn`*`/ovirt-engine-auth/`.

> **Note**
>
> If you plan to use Grafana SSO, ensure that each Keycloak user has an email address. Grafana requires an email address for authentication.

Create a Local Group in Keycloak

This section describes how to create groups in Keycloak and map those groups to realm roles. Users that you add to a group inherit the group's role mappings.

Group naming recommendations

Use group names that are meaningful for the environment. The following example names are commonly used:

- `ovirt-admins`
- `ovirt-users`
- `grafana-admins`, `grafana-editors`, `grafana-viewers`

Create a group and map it to realm roles

1. Sign in to the Keycloak administration console.
2. In the navigation menu, select **Groups**.
3. Select **New**, enter a group name, and then select **Create**.
4. Select the group name, and then select the **Role Mappings** tab.
5. From the role list, select the required realm roles, and then select **Add selected**. For example:
   - For OLVM administrative access, map `SuperUser` to `ovirt-admins`.
   - For OLVM portal access, map `UserRole` to `ovirt-users`.
   - For Grafana access, map `grafana-admin`, `grafana-editor`, or `grafana-viewer` to the corresponding `grafana-*` group.

Manage group membership

To view and manage users assigned to the group, select the group name and then select the **Members** tab.

Create a Local User in Keycloak

This section describes how to create users directly in Keycloak. You can assign a user to one or more Keycloak groups so the user inherits the realm roles mapped to those groups.

Create a user

1. Sign in to the Keycloak administration console.
2. In the navigation menu, select **Users**, and then select **Add user**.
3. Enter the user details:
   - **Username** (required)
   - **Email**
   - **First name** and **Last name**
   - **User enabled**: ON
   - (Optional) In **Groups**, select one or more groups to assign the user.
4. Select **Save**.

Set credentials

1. In the user details page, select the **Credentials** tab.
2. Set the password as required by the environment.

Assign the user to groups

If you didn't assign the user to a group when you created the user, you can add group membership later.

1. In the user details page, select the **Groups** tab.
2. Select the required group and add the user.

> **Note**
>
> If you plan to use Grafana SSO, ensure that the user has an email address configured.

Assign Keycloak Groups in OLVM and Configure Permissions

After you create groups in Keycloak, add those groups in the Administration Portal and assign the required OLVM roles and permissions.

Add a Keycloak group in the Administration Portal

1. Sign in to the Administration Portal as `admin@ovirt`.
2. Go to **Administration** > **Users**.
3. Select **Groups**, and then select **Add**.
4. From the domain list, select **internalsso**.
5. Search for the Keycloak group, select it, and then add it.

Assign roles and permissions

1. Select the group name, and then select the **Permissions** tab.
2. Assign the required role. For example:
   - For full administrative access, assign **SuperUser**.
   - For basic portal access, assign **UserRole**.

Assign Grafana Roles for Keycloak Local Accounts

If you use Grafana Single Sign-On (SSO) with Keycloak local users, ensure that Keycloak groups are mapped to the realm roles that Grafana expects.

Configure group-to-role mappings in Keycloak

1. In Keycloak, select **Groups** and select the group that you want to use for Grafana. For example, select `grafana-admins`.
2. Select the **Role Mappings** tab.
3. Add the appropriate realm role, and then select **Add selected**. For example, map `grafana-admin` to the `grafana-admins` group.

Result

After mapping is complete, users that are members of the Grafana groups inherit the corresponding Grafana role when they sign in to Grafana using SSO.

Sign in and Test Access

After you create local users and groups in Keycloak and assign permissions in the Administration Portal, test sign-in to the portals to confirm that role mappings work as expected.

Test Administration Portal access

1. Sign out of the Administration Portal.
2. Sign in using a Keycloak local user account.
3. Verify that the user has the access associated with the user's group permissions.

Test Grafana SSO access

1. Open the Monitoring Portal (Grafana) and select the SSO sign-in option.
2. Sign in using a Keycloak local user account.
3. Verify that the user's role in Grafana matches the mapped Keycloak group.

Enable Grafana Single Sign-On with Keycloak

The Monitoring Portal (Grafana) can use Keycloak as an OAuth 2.0/OpenID Connect identity provider. This section describes the configuration that's required to let Grafana users sign in with Keycloak and to ensure that logout ends the SSO session.

Prerequisites

- Keycloak is enabled and working for the Manager.
- Users are available in Keycloak either as local Keycloak users or federated directory users.
- Keycloak groups are mapped to Grafana roles (Admin, Editor, Viewer) as required by the environment.

Configure Grafana for OAuth sign-up

For Keycloak-based SSO, Grafana must be allowed to create a local Grafana user on first sign-in. Set `allow_sign_up` to `true`.

1. On the Manager host, edit `/etc/grafana/grafana.ini`.
2. In the `[auth.generic_oauth]` section, set:

```
################################ Generic OAuth ######################
[auth.generic_oauth]
name = oVirt Engine Auth
enabled = true
allow_sign_up = true
```

3. Restart Grafana:

```
sudo systemctl restart grafana-server
```

Configure single logout (recommended)

If users aren't prompted to authenticate after logging out, configure a sign-out redirect URL so that logout ends the Keycloak session and returns the user to Grafana.

1. On the Manager host, edit `/etc/grafana/grafana.ini`.

2. In the `[auth.generic_oauth]` section, add the following setting (replace *fqdn* with the Manager host name):

   ```
   signout_redirect_url = https://fqdn/ovirt-engine-auth/realms/ovirt-
   internal/protocol/openid-connect/logout?post_logout_redirect_uri=https://
   fqdn/ovirt-engine-grafana/
   ```

3. Restart Grafana:

   ```
   sudo systemctl restart grafana-server
   ```

Troubleshooting

If users can't sign in and Grafana logs show an error similar to `Error getting email address`, ensure that the user has an email address populated and (if required) verified in the identity source (Active Directory/LDAP or Keycloak).

> **Note**
>
> As a best practice, ensure all IAM user accounts have an associated email address.

Configure Multifactor Authentication (MFA) in Keycloak

Keycloak works withs multifactor authentication (MFA) with One-Time Password (OTP) tokens. This section describes a basic configuration that requires users to configure an OTP device (for example: Oracle Authenticator, FreeOTP, or Google Authenticator) and to provide OTP codes when signing in.

Configure OTP in the authentication flow

1. Sign in to the Keycloak administration console: `https://manager-fqdn/ovirt-engine-auth/`.
2. Select **Authentication**, then select the **Flows** tab.
3. Find **Browser – Conditional OTP** and change the requirement from **Conditional** to **Required**.
4. Select the **Required Actions** tab. Enable **Configure OTP** and set it as a **Default Action** so new users are required to configure OTP during first sign-in.

Require OTP for existing users (optional)

To require OTP for a user that already exists in Keycloak:

1. Select **Users**, find the user, and open the user details.
2. On the **Details** tab, under **Required User Actions**, add **Configure OTP**.

User sign-in experience

When a user signs in to the Administration Portal, the user is prompted to configure an OTP device. The user scans a QR code using an authenticator application and completes the setup. Later sign-ins require an OTP code.

Reset an OTP credential

If a user must reset OTP (for example, when replacing a phone), you can remove the OTP credential in Keycloak.

1. In Keycloak, open **Users**, select the user, and select the **Credentials** tab.
2. In **Manage Credentials**, delete the OTP credential.

4
Administrative Tasks

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

For other 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.

Perform periodic maintenance on these databases. Running the Engine Vacuum tool updates tables and removes dead rows, letting disk space be reused.

Reclaiming Database Storage

To reclaim database storage using the Engine Vacuum tool, you must sign in to 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 the virtualization environment.

Creating a New Data Center

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

   The **Data Centers** pane opens.
2. Select **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 dropdown menus.
5. Select **OK** to create the data center.

   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 selecting the **Configure Later** button. You can resume the configuration of these entities by selecting the respective data center and selecting **More Actions** and then choosing **Guide Me** from the dropdown 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. Select **New**.
   The **New Cluster** dialog box opens with the **General** tab selected on the sidebar.
3. From the **Data Center** dropdown list, select 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** dropdown list, select the network for which to assign the management network role.
7. From the **CPU Architecture** and **CPU Type** dropdown lists, select 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 the cluster includes hosts with different CPU models, select the oldest CPU model from the list to ensure that all hosts can operate in the cluster.
8. From the **Compatibility Version** dropdown list, select 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** dropdown list, select the type of switch to be used for the cluster. By default, **Linux Bridge** is selected from the dropdown list.
10. From the **Firewall Type** dropdown list, select 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** checkbox is selected by default. This checkbox specifies that the cluster is to be populated with virtual machine hosts.
12. **(Optional)** Review the other tabs to further configure the cluster:
    a. Select 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. Select the **Migration Policy** tab on the sidebar menu to define the virtual machine migration policy for the cluster.
    c. Select 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. Select the **Fencing policy** tab on the sidebar to enable or disable fencing in the cluster, and select fencing options.
    e. Select the **MAC Address Pool** tab on the sidebar to specify a MAC address pool other than the default pool for the cluster.
13. Select **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 selecting the **Configure Later** button. You can resume the configuration of these entities by selecting the respective cluster and selecting **More Actions** and then choosing **Guide Me** from the dropdown 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 many 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 tries to migrate all running virtual machines to other 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 can't be migrated and must be shut down before moving the host to maintenance mode. You can check which virtual machines are pinned to the host by selecting **Pinned to Host** in the **Virtual Machines** tab of the host's details view.

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

   The host maintenance Reason field only appears 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 isn't lost when the host is moved to maintenance mode. The Engine also checks that there's no self-heal activity that will be affected by moving the host to maintenance mode. If the Gluster quorum will be lost or if self-heal activity 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 only appear in the host maintenance window when the selected host supports Gluster.
6. Select **OK** to start maintenance mode.
7. All running virtual machines are migrated to other 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 doesn't stop while the host is in maintenance mode.

   > **Note**
   >
   > If migration fails on any virtual machine, select **Management** and then select **Activate** on the host to stop the operation placing it into maintenance mode, then select **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. Select **Compute** and then select **Hosts**.
2. Select the host.
3. Select **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 aren't automatically migrated back to the host when it's activated, but can be migrated manually. If the host was the Storage Pool Manager (SPM) before being placed into maintenance mode, the SPM role doesn't return automatically when the host is activated.

Removing a Host

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

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

   Select the **Force Remove** checkbox if the host is part of a Gluster Storage cluster and has volume bricks on it, or if the host is nonresponsive.
5. Select **OK**.

Networks

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

> **Note**
>
> If you plan to use VLANs on top of bonded interfaces, see the My Oracle Support (MOS) article *How to Configure 802.1q VLAN on NIC (Doc ID 1642456.1)* for instructions.

Creating a Logical Network

To create a logical network:

1. Go to **Network** and then select **Networks**.
2. On the **Networks** pane, select **New**.

   The **New Logical Network** dialog box opens with the **General** tab selected on the sidebar.
3. From the **Data Center** dropdown list, select the Data Center for the network.

   The **Default** data center is preselected in the dropdown list.

   For the procedures to create new data centers or a new clusters, see Data Centers or Clusters tasks.
4. For the **Name** field, enter a name for the new network.

5. Under the **Network Parameters** section, the **VM Network** checkbox is selected by default. Leave the **VM Network** checkbox selected to create a new virtual machine network.
6. **(Optional)** Configure other settings for the new logical network from the other tabs on the **New Logical Network** sidebar.
7. Select **OK** to create the network.

Assigning a Logical Network to a KVM Host

To assign a logical network to a KVM host:

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

   The **Hosts** pane opens.
2. Under the **Name** column, select the name of the host for which to add the network.

   The following screenshot shows the **Hosts** pane with the name of the host highlighted in a red rectangular box to emphasize where you need to select to set up a network on a host.

Figure 4-1 Hosts Pane

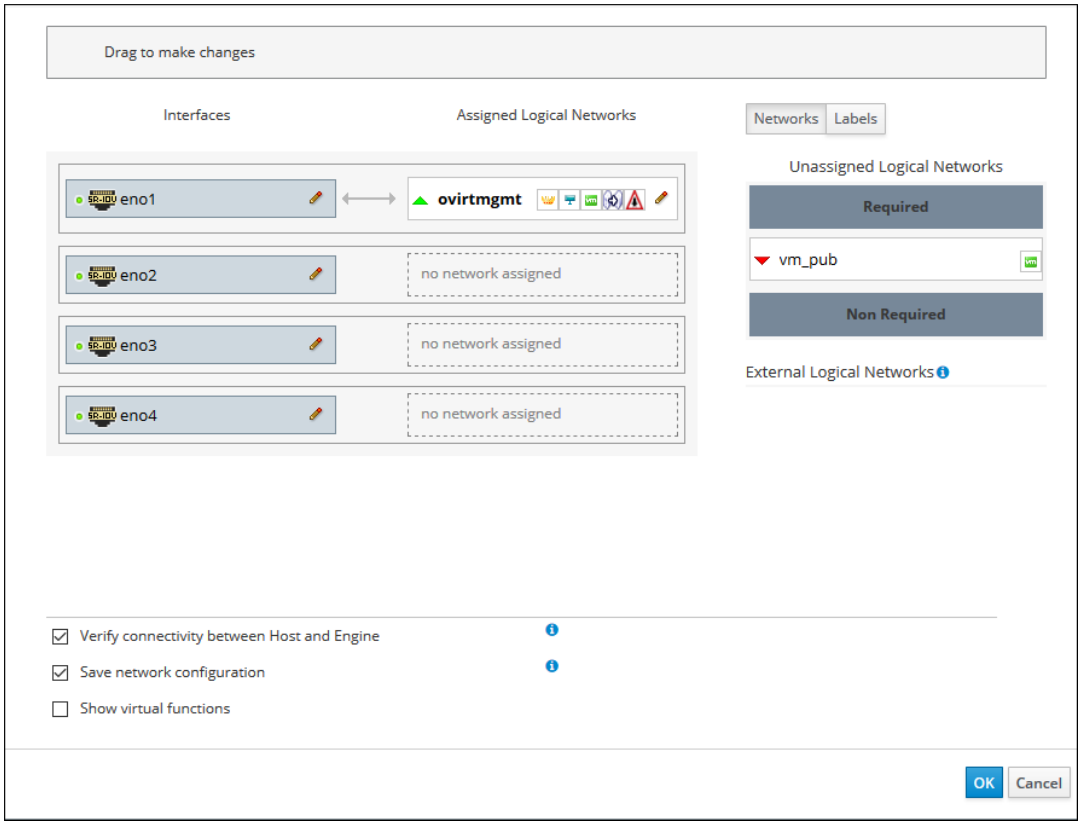
   After selecting the name of the host, the **General** tab opens with details about the host.
3. Select the **Network Interfaces** tab on the horizontal menu.

   The **Network Interfaces** tab opens with details about the network interfaces on the available host.
4. Highlight the network interface that you want to use for the network being added by selecting the row for the respective interface.
5. Select **Setup Host Networks**.

   The **Setup Host Networks** dialog box opens for the host. The physical interfaces on the host are listed under the **Interfaces** column and any logical networks assigned to the interface are displayed under the **Assigned Logical Networks** column. Unassigned logical networks are displayed under the **Unassigned Logical Networks** column.

   In the following example screenshot, a logical network named `vm_pub` is displayed under the **Unassigned Logical Networks** column.

Figure 4-2 Setup Host Dialog Box: Unassigned Logical Networks

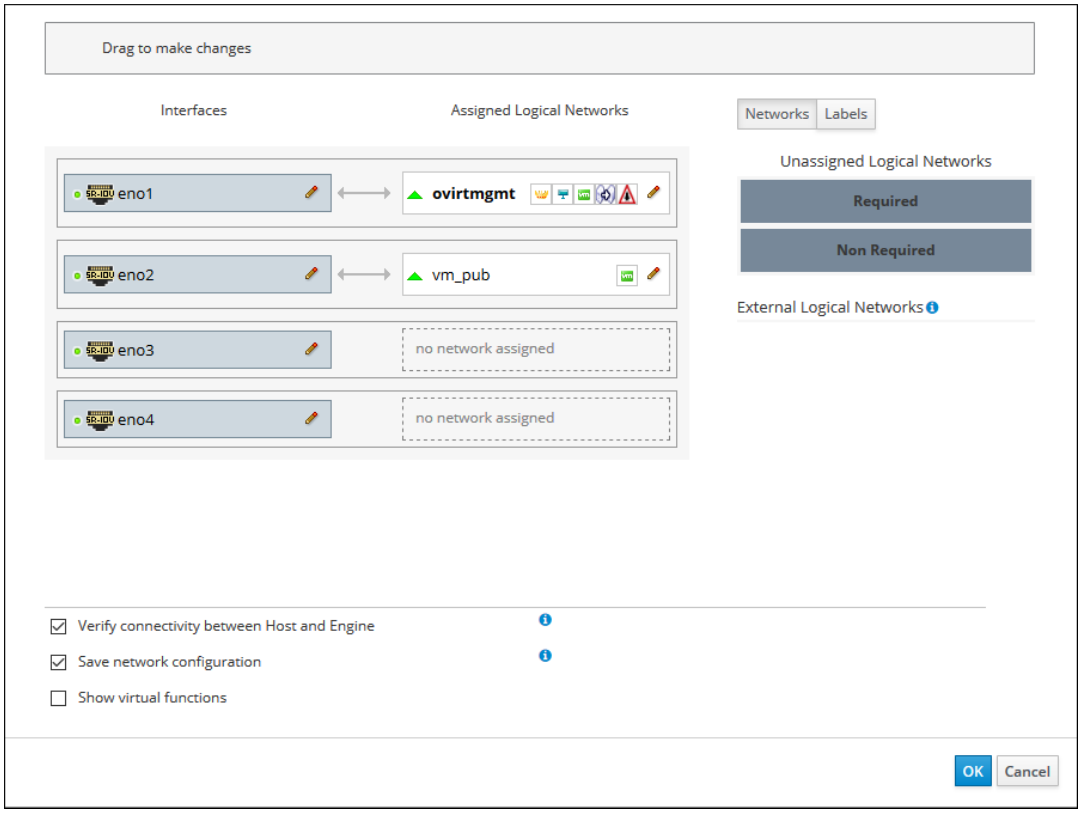


6. Select the network you want to add from the **Unassigned Logical Networks** column by selecting the network and, while holding down the mouse, drag the network over to the box to the right of the available network interface where you want to add the network.

   Or, you can right-click the network and select the available interface from a dropdown list.

   For example, the logical network named `vm_pub` is assigned to the available network interface named `eno2`. In the following screenshot, after dragging the network from **Unassigned Logical Networks** over to this interface, the network named `vm_pub` appears under the **Assigned Logical Networks** column as assigned to the network interface named `eno2`.

Figure 4-3 Setup Host Dialog Box: Assigned Logical Networks



7. After editing the network settings, select **OK** to save the settings.
8. Select **OK** to add the network.

Customizing vNIC Profiles for Virtual Machines

To customize vNICs for virtual machines:

1. Go to **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 add the virtual machine network.

   The **General** tab opens with details about the virtual machine.
3. Select 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 selecting the row for the respective interface and then select **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** dropdown list, select the network to be added to the virtual machine.
   b. Select the **Custom MAC address** checkbox. Then enter or update the MAC address that's allocated for this virtual machine in the text entry field.

6. Select **OK** when you're finished editing the network interface settings for the virtual machine.
7. Go to **Compute** and then select **Virtual Machines**.

   The **Virtual Machines** pane opens.

> **Important**
>
> Because 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 Logical Network to a KVM Host.

8. Highlight the virtual machine where you added the network and then select **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, consider the following:

- To change the IP address of a host, you must remove the host and then readd it.
- To change the VLAN settings of a host, see Editing a Host's VLAN Settings in oVirt Documentation.
- You can't assign logical networks offered by external providers to physical host network interfaces; such networks are dynamically assigned to hosts as they're required by virtual machines.
- If a switch has been configured to provide Link Layer Discovery Protocol (LLDP) information, you can hover the 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. Select **Compute Hosts**.
2. Select the host's name. This opens the details view.
3. Select the **Network Interfaces** tab.

4. Select **Setup Host Networks**.
5. Optionally, hover the 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 the cursor over an assigned logical network and select 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, select 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, select the **Custom Properties** tab, select **ethtool_opts** from the list, and enter a valid value using the format of the commandline 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 a network's interfaces, for example:

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

The `ethtool_opts` option isn't 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), select the **Custom Properties** tab, select **fcoe** from the list, and enter `enable=yes`. Separate multiple entries with a whitespace character.

The `fcoe` option isn't 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 the logical network definition isn't synchronized with the network configuration on the host, select the **Sync network** checkbox. 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** checkbox.

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

9. Select **OK**.

> **Note**
>
> If not all network interface cards for the host are displayed, select **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, iSCSI, and FCP storage.

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's 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.

   ```
   mkdir -p /data/images
   ```

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

   ```
   chown 36:36 /data /data/images
   ```

   ```
   chmod 0755 /data /data/images
   ```

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

Configuring a KVM Host to Use Local Storage

When you configure a KVM host to use local storage, it's 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 select **Hosts**.

The **Hosts** pane opens.

2. Highlight the host on which to add the local storage domain.
3. Select **Management** and then select **Maintenance** from the dropdown 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, select **Management** and then select **Configure Local Storage** from the dropdown list.

   The **Configure Local Storage** pane opens with the **General** tab selected.
5. Select **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 the local storage** text input field, specify the path to the local storage domain.

   For more information, see Preparing Local Storage for a KVM Host.
7. Select **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 select **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 the environment meets the following conditions:

- Ensure that the Manager and KVM host installation are running the Oracle Linux 8.8 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.

  ```
  grep vdsm /etc/passwd
  ```

  ```
  vdsm:x:36:36:Node Virtualization Manager:/:/sbin/nologin
  ```

  ```
  grep kvm /etc/group
  ```

  ```
  kvm:x:36:qemu,sanlock
  ```

- An Oracle Linux NFS File server that's reachable by the virtualization environment.

Preparing NFS Storage

To prepare NFS storage:

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

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

2. Set the required permissions on the new directory to provide 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 don't 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 the 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 -e | 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 the Data Domain.

Using iSCSI Storage

For iSCSI storage, a storage domain is created from a volume group that's composed of preexisting LUNs. Neither volume groups nor LUNs can be attached to more than one storage domain at a time.

Multiple network paths between hosts and iSCSI storage prevent host downtime caused by network path failure. iSCSI multipathing lets you create and manage groups of logical networks and iSCSI storage connections. After configuration, the Manager 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.

Attaching an iSCSI Data Domain

For iSCSI storage, a storage domain is created from a volume group that's composed of preexisting LUNs.

To attach an iSCSI data domain to the virtualization environment:

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

   The **Storage Domains** pane opens.

2. Select **New Domain**.

   The **New Domain** dialog box opens.

3. From the **Data Center** dropdown list, select the data center for which to attach the data domain.

   The **Default** data center is preselected in the dropdown list.

   For the procedures to create new data centers or a new clusters, see Data Centers or Clusters tasks.

4. For the **Name** field, enter a name for the data domain.

5. From the **Domain Function** dropdown list, select the domain function. By default, the **Data** option is selected in the dropdown list.

   For this step, leave **Data** as the domain function because you're creating a data domain in this procedure.

6. From the **Storage Type** dropdown list, select **iSCSI**.

7. From the **Host** dropdown list, select the host for which to attach the data domain.

8. When **iSCSI** is selected for the **Storage Type**, the **Discover Targets** dialog box opens and the **New Domain** dialog box automatically displays the known targets with unused LUNs under the **Target Name** column.

   If the target from which you're adding storage isn't listed, complete the following fields in the **Discover Targets** dialog box:

   a. For the **Address** field, enter the fully-qualified domain name or IP address of the iSCSI host on the storage array.

   b. For the **Port** field, enter the port to connect to on the host when browsing for targets. By default, this field is automatically populated with the default iSCSI Port, **3260**.

   After completing these fields, select **Discover**.

   The **Target Name** column updates to list all the available targets discovered on the storage array.

9. Under the **Target Name** column, select the required target and select the black right-directional arrow to sign in to the target.

   The **Storage Domains** pane refreshes to list only the targets for which you signed in.

10. Select **+** to expand the required target.

    The target expands to display all the unused LUNS.

11. Select **Add** for each LUN ID that's to connect to the target.

12. **(Optional)** Configure the advanced parameters.

    If you're using ZFS storage, you must clear the **Discard after Delete** option.

13. Select **OK**.

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

    After the iSCSI data domain has been added to the virtualization environment, you can then upload the ISO images that are used for creating virtual machines.

Configuring iSCSI Multipathing

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.
- 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.

To configure iSCSI multipathing:

1. Select **Compute Data Centers**.
2. Select the data center name.
3. In the **iSCSI Multipathing** tab, select **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. Select **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. Change the current logical network so that it's not required.
   a. Select **Compute** and then select **Clusters**.
   b. Select the cluster name.
   c. In the **Logical Networks** tab of the cluster detail page, select a current logical network and select **Manage Networks**.

      As an example, `net-1` is the name of the current logical network.

   d. Clear the **Require** checkbox and select **OK**.
2. Create a logical network that's not Required and not a VM network.

a. Select **Add Network**. This opens the **New Logical Network** window.
b. In the **General** tab, enter a **Name** (for example, `net-2`) and clear the **VM network** checkbox.

   As an example, `net-2` is the name of the new logical network.
c. In the **Cluster** tab, clear the **Require** checkbox and select **OK**.

3. Remove the current network bond and reassign the logical networks.
   a. Select **Compute** and then select **Hosts**.
   b. Select the host name.
   c. In the **Network Interfaces** tab of the host detail page, select **Setup Host Networks**.
   d. Drag the old logical network (for example, `net-1`) to the right to unassign it.
   e. Drag the current bond to the right to remove it.
   f. Drag the old logical network (for example, `net-1`) and the new logical network (for example, `net-2`) to the left to assign them to physical interfaces.
   g. To edit the new logical network (for example, `net-2`), select 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 select **OK**.
4. Create the iSCSI bond.
   a. Select **Compute** and then select **Data Centers**.
   b. Select the data center name.
   c. In the **iSCSI Multipathing** tab of the data center details page, select **Add**.
   d. In the **Add iSCSI Bond** window, enter a **Name**, select the the old and new networks (for example, `net-1` and `net-2`) and select **OK**.

The data center now 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 select **Domains**.

   The **Storage Domains** pane opens.
2. On the **Storage Domains** pane, select 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** dropdown list, select the Data Center for which to attach the data domain. By default, the **Default** option is selected in the dropdown list.
5. From the **Domain Function** dropdown list, select the domain function. By default, the **Data** option is selected in the dropdown list.

   For this step, leave **Data** as the domain function because you're creating a data domain in this example.
6. From the **Storage Type** dropdown list, select **Fibre Channel**.
7. For the **Host to Use** dropdown 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. Select **Add** next to the LUN ID that's connect to the target.
10. **(Optional)** Configure the advanced parameters.
11. Select **OK**.

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

Uploading Images to the Data Domain

Before using the Manager to upload images to the data domain, you must perform the following steps to ensure that the prerequisites for uploading images have been met on the Manager and KVM hosts.

Before You Begin

To ensure that the prerequisites for uploading images to the data domain have been met:

1. On the engine host, verify that the `ovirt-imageio` service has been configured and is running.

   ```
   systemctl status ovirt-imageio.service
   ```

   When the service is running, the output displays as follows.

   ```
   ovirt-imageio.service - oVirt ImageIO
      Loaded: loaded (/usr/lib/systemd/system/ovirt-imageio.service; enabled;
      vendor preset: disabled)
      Active: active (running) since Mon 2019-03-25 13:12:29 PDT; 2 weeks 0
   days ago
    Main PID: 28708 (ovirt-imageio-p)
      CGroup: /system.slice/ovirt-imageio.service
              └─28708 /usr/bin/python2 /usr/bin/ovirt-imageio
   ...
   ```

   This service is automatically configured and is started when you run the engine-setup command during the installation of the Manager.
2. On the KVM host, verify that the `ovirt-imageio` service has been configured and is running, for example:

   ```
   systemctl status ovirt-imageio-daemon
   ```

   ```
   ovirt-imageio-daemon.service - oVirt ImageIO Daemon
      Loaded: loaded (/usr/lib/systemd/system/ovirt-imageio-daemon.service;
   disabled;
      vendor preset: disabled)
      Active: active (running) since Wed 2019-03-27 18:38:36 EDT; 3 weeks 4
   days ago
    Main PID: 366 (ovirt-imageio-d)
       Tasks: 4
      CGroup: /system.slice/ovirt-imageio-daemon.service
   ```

```
             └─366 /usr/bin/python /usr/bin/ovirt-imageio-daemon

Mar 27 18:38:36 myserver systemd[1]: Starting oVirt ImageIO Daemon...
Mar 27 18:38:36 myserver systemd[1]: Started oVirt ImageIO Daemon.
```

3. Verify that the certificate authority has been imported into the web browser used to access the Manager by browsing to the following URL and enabling the trust settings:

   ```
   https://engine_address/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA
   ```

4. Verify that you're using a browser that meets the browser requirement to access the Administration Portal.

   For more information, see *Logging into the Administration Portal* in the Oracle Linux Virtualization Manager: Getting Started Guide.

Uploading an ISO Image to the Data Domain

To upload an ISO image to data domain using the Manager:

1. Download or copy an ISO image file that you want to upload into the environment to a location on the desktop, laptop, or a system where the Manager is accessible from a Web browser.
2. Go to **Storage** and then select **Disks**.

   The **Disks** pane opens.
3. Select **Upload** and then select **Start** from the dropdown list.

   The **Upload Image** dialog box opens.
4. Select **Choose File** and navigate to the location where you saved the ISO image.
5. Complete the **Disk Options** section of the dialog box.
6. Ensure that the prerequisites have been met by selecting **Test Connection**.

   If the test returns a warning or error message, see Before You Begin to review the prerequisites.
7. Select **OK** to start uploading the ISO image.

   The status field on the **Disks** pane tracks the progress of the upload.

   After the ISO image upload is completed successfully, you can attach the image to virtual machines as CDROMs or use the image to boot virtual machines.

> **Note**
>
> For information on uploading ISO images to data domains from the command line, see the My Oracle Support article *Sample Script to Upload Disk/ISO To Storage Domain From Remote Linux Server (Doc ID 2830534.1)*.

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 can't move a storage domain into maintenance mode if a virtual machine has a lease on the storage domain. The virtual machine must be shut down, or the lease must be 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 select **Domains**.

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

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

   The `Ignore OVF update failure` checkbox lets the storage domain 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. Select **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. Select **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 create virtual machines and perform basic administration tasks such as live editing, live migration, and creating and using templates and snapshots.

Creating a New Virtual Machine

This section shows you how to install a remove viewer, create Oracle Linux or Microsoft Windows virtual machines, and install the respective guest OS, agents and drivers.

For detailed information on the supported guest operating systems, see the Oracle® Linux: KVM User's Guide.

Before creating Microsoft Windows virtual machines, ensure the following prerequisites are met.

Obtain the Oracle VirtIO Drivers for Microsoft Windows.

1. Download Oracle VirtIO Drivers for Microsoft Windows to the Manager host from Oracle Software Delivery Cloud or My Oracle Support (MOS). See Oracle VirtIO Drivers for Microsoft Windows for Use With KVM for more information.
2. Upload the Oracle VirtIO Drivers for Microsoft Windows ISO image to an Oracle Linux Virtualization Manager storage domain. See Uploading an ISO Image to the Data Domain for more information.

Download the QEMU guest agent to the Manager host.

- **For ULN registered hosts or using Oracle Linux Manager**, download `qemu-ga-win` from the *oVirt Release 4.5 on Oracle Linux 8 (x86_64) - Extra* channel.
- **For Oracle Linux yum server configured KVM hosts**, download `qemu-ga-win` from the Oracle Linux 8 (x86_64) oVirt 4.5 Extra repository.

> **Note**
>
> In addition to creating virtual machines, you can import an Open Virtual Appliance (OVA) file into the environment from any host in the data center. For more information, see oVirt Virtual Machine Management in oVirt Documentation.

Installing Remote Viewer on Client Machine

A console is a UI that lets you view and interact with a virtual machine similar to a physical machine. The default console is a Remote Viewer application that provides users with a UI for connecting to virtual machines.

Before you begin a Linux or Windows OS installation, download the appropriate install package from the Virtual Machine Manager website.

> **Note**
>
> See 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, see Consoles in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

To install Remote Viewer on Linux:

1. Ensure you have downloaded the `virt-viewer` installation package.
2. Install the `virt-viewer` package using one of the following commands depending on the system.

```
yum install virt-viewer
```

```
dnf install virt-viewer
```

3. Restart the browser for the changes to take effect in the Oracle Linux Virtualization Manager.

You can now connect to the virtual machines using the VNC protocol.

To install Remote Viewer on Windows:

1. Ensure you have downloaded either the 32-bit or 64-bit `virt-viewer` installer depending on the architecture of the system.
2. Go to the folder where you saved the file and double-click the file.
3. If prompted with a security warning, select **Run**.
4. If prompted by User Account Control, select **Yes**.

After installation, you can access Remote Viewer in the **VirtViewer** folder of **All Programs** from the **Start** menu.

Creating a New Linux or Microsoft Windows Virtual Machine

Follow this general procedure to create a new virtual machine:

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

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

   The **New Virtual Machine** dialog box opens with the **General** tab selected on the sidebar.
3. From the **Cluster** dropdown list, select the data center and host cluster for the new host.

   The **Default** data center is preselected in the dropdown list.

   For the procedures to create new data centers or a new clusters, see Data Centers or Clusters tasks.
4. From the **Operating System** dropdown list, select the OS for the virtual machine.
5. Using the **Optimized for** dropdown list, select the type of system for which the virtual machine is to be optimized.
   - Server (default) - have no sound card, use a cloned disk image, and aren't stateless
   - Desktop - have a sound card, use an image (thin allocation), and are stateless
   - High Performance - have several configuration changes; see Optimizing Clusters, Hosts and Virtual Machines.
6. For the **Name** field, enter a name for the new virtual machine.
7. Under **Instance Images**, add storage to the virtual machine by either using an existing virtual disk or creating a new virtual desk.
   - To use an existing virtual disk, select **Attach** and select the virtual disk to use for the virtual machine storage. Then select **OK**.
   - To create a virtual disk, select **Create** and update the fields for the virtual machine storage or accept the default settings. Then select **OK**.

   If you're creating a new virtual disk, the following fields are key:
   - Check **Bootable**.
   - Enter a disk **Size (GiB)**.
   - From the **Interface** dropdown list, select **VirtIO-SCSI**.
   - From the **Allocation Policy** dropdown list, selecting **Preallocated** reserves the disk space, while selecting **Thin Provision** creates a sparse allocated virtual disk.

- From the **Disk Profile** list, select the storage domain where you can save the virtual disk.

> **Note**
>
> Repeat this step if you need to create more virtual disks.

8. Connect the virtual machine to a network by adding a network interface.

   See Creating a Logical Network, Assigning a Logical Network to a KVM Host, and Customizing vNIC Profiles for Virtual Machines.

9. Select **Show Advanced Options** to display other configuration options available for the new virtual machine.
10. **(Optional)** Select the **System** tab on the sidebar to adjust the CPU and memory size for the virtual machine from the defaults.
    - For **Memory Size** field, the default value is `1024 MB`.
    - For the **Maximum memory** field, the default value is `4096 MB`, which is four times the memory size but can be manually configured.
    - For the **Total Virtual CPUs** field, the default value is `1`.

> **Note**
>
> Depending on the OS you're installing, there might be memory and vCPU requirements.

11. Select **OK** to create the virtual machine.
12. Proceed to Installing the Oracle Linux Guest OS or Installing the Microsoft Windows Guest OS.

Installing the Oracle Linux Guest OS

To install the Oracle Linux guest OS:

1. From the **Virtual Machines** pane, select the Oracle Linux virtual machine created in Creating a New Linux or Microsoft Windows Virtual Machine.
2. Using the down arrow next to **Run**, select **Run Once**.
3. Attach the ISO file, for example **OracleLinux-R8-U8-Server-x86_64-dvd.iso**, and select **OK**.
4. Select **Console** to open a console to the virtual machine.

   If you haven't installed the Remote Viewer application, see Installing Remote Viewer on Client Machine.

5. Install the Oracle Linux guest OS.

   See the Oracle® Linux documentation for more information on how to install Oracle Linux.

6. When the installation completes, reboot the virtual machine.
7. **(Optional)** If you use a proxy server for Internet access, configure Yum with the proxy server settings. For more information about configuring `firewalld`, see Configuring Packet-filtering Firewalls in the Oracle Linux 8: Configuring the Firewall or Oracle Linux 9: Configuring the Firewall guide.

8. **(Optional)** If you're using yum to update the host, ensure the host is using the modular yum repository configuration. For more information, see Getting Started with Oracle Linux Yum Server.
9. Proceed to Installing the Oracle Linux Guest Agent.

Installing the Oracle Linux Guest Agent

To install the Oracle Linux guest agent, follow the appropriate steps for the release version.

1. Open a console session for the Oracle Linux guest and sign in to the terminal.
2. Install the latest guest agent package.

   For Oracle Linux 8, Oracle Linux 9, or Oracle Linux 10 guests, run the following command, where *n* is the Oracle Linux major release version:

   ```
   dnf module reset virt
   dnf config-manager --enable oln_kvm_appstream
   dnf -y module enable virt:kvm_utils3
   dnf -y install qemu-guest-agent
   ```

   For Oracle Linux 7 guests:

   ```
   yum install yum-utils -y
   yum-config-manager --enable ol7_latest
   yum install qemu-guest-agent
   ```

   For Oracle Linux 6 guests:

   ```
   yum install yum-utils -y
   yum-config-manager --enable ol6_latest
   yum install qemu-guest-agent
   ```

   For Oracle Linux 5 guests:

   ```
   yum install yum-utils -y
   yum install http://yum.oracle.com/repo/OracleLinux/OL7/ovirt42/x86_64/
   getPackage/ \
     ovirt-guest-agent-1.0.13-2.el5.noarch.rpm
   ```

3. Start the guest agent service for the Oracle Linux guest.

   For Oracle Linux 7, Oracle Linux 8, Oracle Linux 9, or Oracle Linux 10 guests:

   ```
   systemctl start qemu-guest-agent.service
   ```

   For Oracle Linux 6 guests:

   ```
   service qemu-ga enable
   service qemu-ga start
   ```

For Oracle Linux 5 guests:

```
service ovirt-guest-agent enable
service ovirt-guest-agent start
```

4. **(Optional)** Enable an automatic restart of the guest agent service when the virtual machine is rebooted.

   For Oracle Linux 7, Oracle Linux 8, Oracle Linux 9, or Oracle Linux 10 guests:

   ```
   systemctl enable qemu-guest-agent.service
   ```

   For Oracle Linux 6 guests:

   ```
   chkconfig qemu-ga on
   ```

   For Oracle Linux 5 guests:

   ```
   chkconfig ovirt-guest-agent on
   ```

Installing the Microsoft Windows Guest OS

To install the Microsoft Windows guest OS:

1. From the **Virtual Machines** pane, select a virtual machine.
2. Using the down arrow next to **Run**, select **Run Once**.
3. Expand the **Boot Options** menu, check **Attach CD**, and select the ISO image.
4. Select **OK** to boot the virtual machine.
5. Select **Console** to open a console to the virtual machine.

   If you haven't installed the Remote Viewer application, refer to Installing Remote Viewer on Client Machine.
6. Install the Microsoft Windows guest OS.

   See the applicable Microsoft Windows documentation for instructions on how to install the OS.
7. When the installation completes, reboot the virtual machine.
8. Proceed to Installing the VirtIO Drivers and then to Installing the QEMU Guest Agent.

Installing the VirtIO Drivers

Before attempting to install the Oracle VirtIO Drivers for Microsoft Windows on a new Microsoft Windows virtual machine, ensure that you have downloaded the drivers onto the Manager host and uploaded the ISO image to an Oracle Linux Virtualization Manager storage domain. For more information, see the prerequisites.

To install the Oracle VirtIO Drivers for Microsoft Windows:

1. After you finish installing the Microsoft Windows guest OS, return to the **Virtual Machines** pane, highlight the row for this virtual machine, and select **Edit**.

   The **Edit Virtual Machines** dialog box opens.

2. Select the **Boot Options** tab on the sidebar of the dialog box to specify the boot sequence for the virtual device.
    a. From the **First Device** dropdown list, change **CD-ROM** to **Hard Disk**.
    b. From the **Second Device** dropdown list, select **CD-ROM**.
    c. Select the **Attach CD** checkbox and select **virtio** from the dropdown list.
3. Select **OK** to save the changes to the virtual machine configuration.
4. Select **OK** when the **Pending Virtual Machine changes** dialog box appears.
5. From the **Virtual Machines** pane, reboot the virtual machine.
6. Select **Console** to open a console to the virtual machine and navigate to the CDROM.
7. Select the **virtio** folder and then select **Setup** to start the Oracle VirtIO Drivers for Microsoft Windows installer.

    The installer window is displayed.
8. Select **Install** to start the Oracle VirtIO Drivers for Microsoft Windows installer.

    The installer copies the Oracle VirtIO Drivers for Microsoft Windows installer files and then installs the drivers on the Microsoft Microsoft Windows guest OS.
9. Select **Yes, I want to restart my computer now** and select **Finish**.

    The virtual machine is restarted.
10. Stop the virtual machine.
11. Go to **Compute** and then select **Virtual Machines**.

    The **Virtual Machines** pane opens with the list of virtual machines that have been created.
12. Select the virtual machine where you installed the Microsoft Windows guest OS and select **Edit**.
13. Edit the virtual disk. From the **Interface** dropdown list, change **SATA** to **VirtIO-SCSI** .
14. Select the **Boot Options** tab on the sidebar.
    a. Don't make any changes to the **First Device** dropdown list. The **Hard Disk** option is selected from a previous step.
    b. From the **Second Device** dropdown list, select **None**.
    c. Clear the **Attach CD** checkbox.
15. Select **OK** to save the changes to the virtual machine configuration.
16. Restart the virtual machine.
17. Proceed to Installing the QEMU Guest Agent.

Installing the QEMU Guest Agent

Before attempting to install the QEMU guest agent on a new Microsoft Windows virtual machine, ensure that you have downloaded the drivers onto the Manager host. For more information, see the prerequisites.

1. On the Manager host, install the QEMU guest agent.

```
dnf install qemu-ga-win
```

2. Verify the installation.

```
ls -lat /usr/i686-w64-mingw32/sys-root/mingw/bin/


total 9280
drwxr-xr-x. 2 root root      30 Nov  3 13:56 .
-rw-r--r--. 1 root root 9499648 Nov  2 09:45 qemu-ga-i386.msi
drwxr-xr-x. 3 root root      17 Sep 23 19:02 ..

# ls -lat /usr/x86_64-w64-mingw32/sys-root/mingw/bin/
total 9472
drwxr-xr-x. 2 root root      32 Nov  3 13:56 .
-rw-r--r--. 1 root root 9697280 Nov  2 09:45 qemu-ga-x86_64.msi
```

Important

- If you have access to the virtual machine, you can copy the appropriate MSI (32-bit or 64-bit) to the virtual machine and then run the installer to install the QEMU guest agent.
- If you don't have access to the virtual machine, use the following steps to build and upload an ISO and then install the QEMU guest agent.

Build the ISO and install the QEMU guest agent on the virtual machine.

1. Build the QEMU guest agent ISO.

```
dnf install genisoimage -y


pwd
/root


mkdir build-iso


cp /usr/i686-w64-mingw32/sys-root/mingw/bin/qemu-ga-i386.msi build-iso/


cp /usr/x86_64-w64-mingw32/sys-root/mingw/bin/qemu-ga-x86_64.msi build-iso/


mkisofs -R -J -o qemu-ga-windows.iso build-iso/*


I: -input-charset not specified, using utf-8 (detected in locale settings)
Using QEMU_000.MSI;1 for  /qemu-ga-x86_64.msi (qemu-ga-i386.msi)
 52.36% done, estimate finish Thu Nov  3 14:20:49 2022
Total translation table size: 0
Total rockridge attributes bytes: 347
Total directory bytes: 0
Path table size(bytes): 10
```

```
Max brk space used 0
9549 extents written (18 MB)


ll qemu-ga-windows.iso


-rw-r--r--. 1 root root 19556352 Nov  3 14:20 qemu-ga-windows.iso
```

2. Upload the QEMU guest agent ISO image to an Oracle Linux Virtualization Manager storage domain. See Uploading an ISO Image to the Data Domain for more information.
3. From the **Virtual Machines** pane, select a virtual machine.
4. Highlight the row for this virtual machine, and select **Edit**.

   The **Edit Virtual Machines** dialog box opens.
5. Select the **Boot Options** tab on the sidebar of the dialog box to specify the boot sequence for the virtual device.
   a. From the **First Device** dropdown list, change **CD-ROM** to **Hard Disk**.
   b. From the **Second Device** dropdown list, select **CD-ROM**.
   c. Select the **Attach CD** checkbox and select the **qemu** executable from the dropdown list.
6. Select **OK** to save the changes to the virtual machine configuration.
7. Select **OK** when the **Pending Virtual Machine changes** dialog box appears.
8. From the **Virtual Machines** pane, reboot the virtual machine.
9. Select **Console** to open a console to the virtual machine and navigate to the CDROM.
10. Select the the **qemu** executable to open the installation program.
11. When installation completes, select **Yes, I want to restart my computer now** and select **Finish**.

    The virtual machine is restarted.
12. Stop the virtual machine.
13. Go to **Compute** and then select **Virtual Machines**.

    The **Virtual Machines** pane opens with the list of virtual machines that have been created.
14. Select the virtual machine where you installed the Microsoft Windows guest OS and select **Edit**.
15. Select the **Boot Options** tab on the sidebar.
    a. Do not make any changes to the **First Device** dropdown list. The **Hard Disk** option is selected from a previous step.
    b. From the **Second Device** dropdown list, select **None**.
    c. Clear the **Attach CD** checkbox.
16. Select **OK** to save the changes to the virtual machine configuration.
17. From the **Virtual Machines** pane, reboot the virtual machine.
18. Run the Microsoft Windows virtual machine.

For more information, see the Oracle® Linux: KVM User's 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**, select **Compute** and then select **Virtual Machines**.
2. Under the **Name** column, select the virtual machine you want to change and then select **Edit**.
3. On the bottom left of the **Edit Virtual Machine** window, select **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 change:

   - **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 doesn't have a sound card, uses a cloned disk image, and isn't stateless. In contrast, virtual machines optimized to act as desktop machines.
     - **High Performance** - the virtual machine is preconfigured 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**

       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 change:

   - **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. Don't 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 change:

   - **Disable strict user checking**

By default, strict checking is enabled which lets 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 change:

- **Highly Available**

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

  > **Note**
  >
  > You can't 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. Select **OK** when you're finished with all tabs to save the changes.

Changes to any other settings are applied when you shut down and restart the 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 lets you 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 aren't changed.

You use live migration to seamlessly move virtual machines for several common maintenance tasks. Ensure that the 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, ensure you correctly configure it. At a minimum, to successfully migrate running virtual machines:

- Source and destination hosts are 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's 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 starts live migration of virtual machines in two situations:

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

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 enable 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 can't migrate in manually.

To set a virtual machine's migration mode:

From the **Migration mode** dropdown 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. Select **Compute** and select **Virtual Machines**.
2. Select a virtual machine and select **Edit**.
3. Select the **Host** tab.
4. Use the **Start Running On** radio buttons to specify whether the virtual machine can 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 select 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 doesn't apply to virtual machines that are assigned to a group of hosts.

5. From the **Migration mode** dropdown 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. Select **OK**.

Manually Migrate a Virtual Machine

To manually migrate a virtual machine:

1. Select **Compute** and the select **Virtual Machines**.
2. Select a running virtual machine and select **Migrate**.
3. Select either **Select Host Automatically** or **Select Destination Host** and select the destination host from the dropdown list.

   When you select **Select Host Automatically**, the system decides which is the destination host according to the load balancing and power management rules set up in the scheduling policy.

4. Select **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.

Working with Templates

A template is a copy of a virtual machine that you can use to simplify the future creation of similar virtual machines. Templates capture the configuration of software, the configuration of hardware, and the software installed on the virtual machine on which the template is based, which is known as the source virtual machine.

Virtual machines that are created based on a template use the same NIC type and driver as the original virtual machine but are assigned separate, unique MAC addresses.

Working with Oracle Linux Templates

For this example scenario, you seal the Oracle Linux virtual machine created in Creating a New Virtual Machine and then you create an Oracle Linux template based on that virtual machine. You can then use that template as the basis for a Cloud-Init enabled template to automate the initial setup of a virtual machine.

> **Important**
>
> Oracle provides preinstalled and preconfigured templates that let 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. For more information, see Importing an Oracle Linux Template.

Sealing an Oracle Linux Virtual Machine for Use as a Template

Sealing is the process of removing all system-specific details from a virtual machine before creating a template based on that virtual machine. Sealing is necessary to prevent the same details from appearing on several virtual machines that are created based on the same template. It also ensures the functionality of other features, such as predictable vNIC order.

To seal an Oracle Linux virtual machine for use as a template:

1. From a command line, sign in to the Oracle Linux virtual machine as the `root` user.
2. Flag the system for reconfiguration.

   ```
   touch /.unconfigured
   ```

3. Remove the SSH host keys.

   ```
   rm -rf /etc/ssh/ssh_host_*
   ```

4. Do one of the following:
   a. For Oracle Linux 6 (or earlier), set the host name value of the `HOSTNAME=localhost.localdomain` in the `/etc/sysconfig/network` file.
   b. For Oracle Linux 7 remove the `/etc/hostname` file.
5. Remove `/etc/udev/rules.d/70-*`.

   ```
   rm -rf /etc/udev/rules.d/70-*
   ```

6. Remove the `HWADDR` and `UUID` lines in the `/etc/sysconfig/network-scripts/ifcfg-eth*` file.
7. **(Optional)** Delete all the logs from `/var/log` and build logs from `/root`.
8. Cleanup the command history.

   ```
   history -c
   ```

9. Shutdown the virtual machine.

   ```
   poweroff
   ```

   The Oracle Linux virtual machine is now sealed and ready to be made into a template.

Creating an Oracle Linux Template

When you create a template based on a virtual machine, a read-only copy of the virtual machine's disk is created. This read-only disk becomes the base disk image of the new template, and of any virtual machines that are created based on the template.

To create an Oracle Linux template:

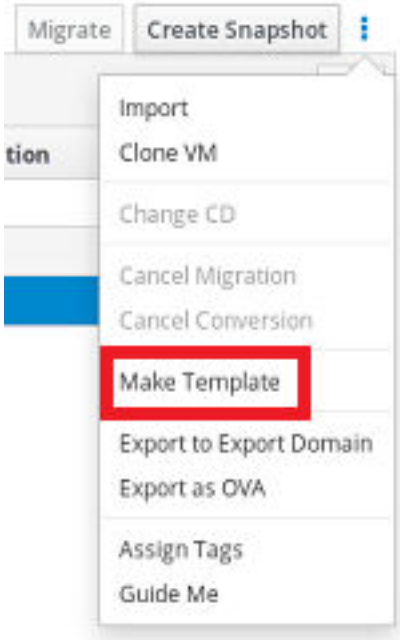
1. Go to **Compute**, and then select **Virtual Machines**.

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

2. Select **More Actions** to expand the dropdown list and select **Make Template** from the dropdown list.

   The following screenshot shows the **More Actions** dropdown list expanded to display the **Make Template** option. The **Make Template** option is highlighted with a red rectangular box for emphasis.

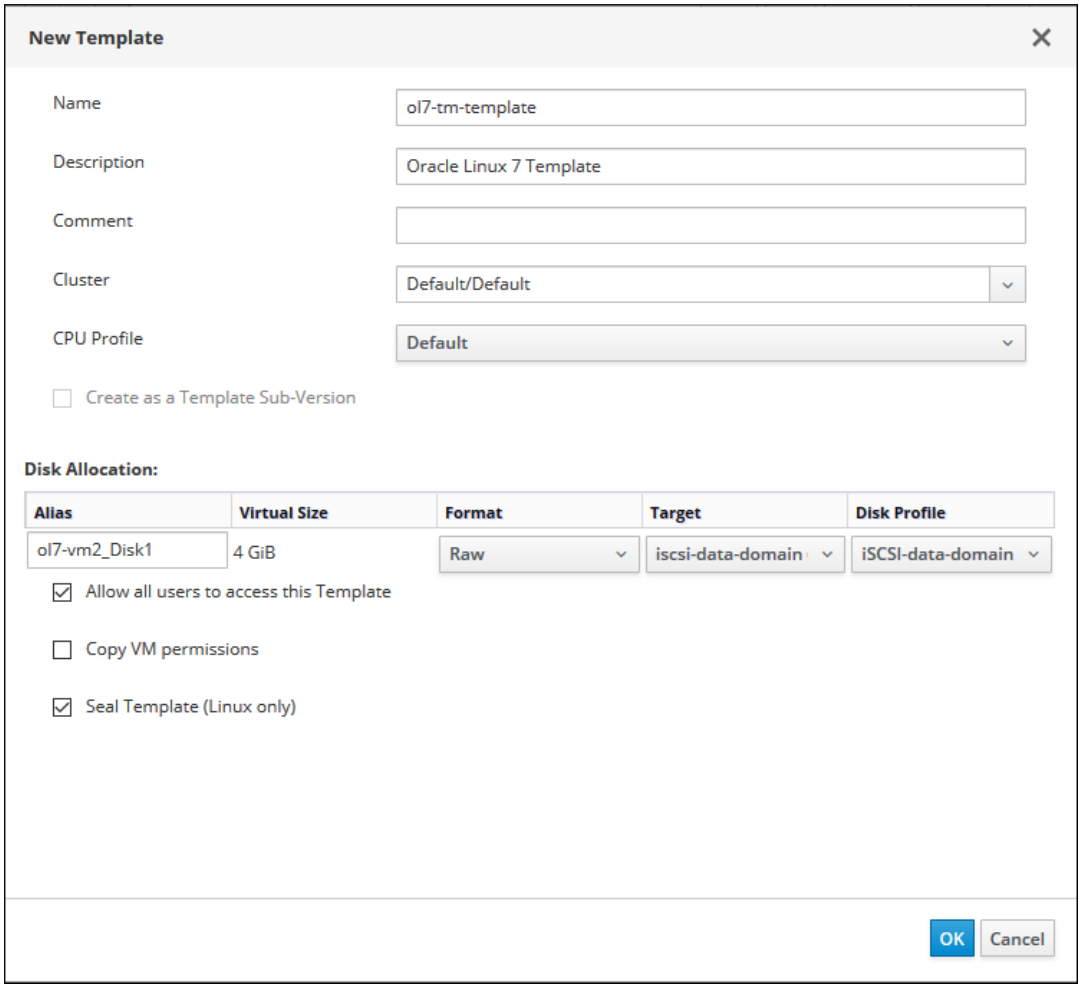
Figure 4-4 Make Template Option



3. For the **Name** field, enter a name for the new virtual machine template.
4. In the **Disc Allocation:** section under the **Alias** column, rename the disk alias to be the same as the template name entered for the **Name** field.
5. Select the **Seal Template (Linux only)** checkbox.

   The following screenshot shows the **New Template** dialog box completed for a new template named `ol7-vm-template`. In the dialog box, the disk alias has been renamed to `ol7-vm-template` and the **Seal Template (Linux only)** checkbox is selected.

Figure 4-5 New Template Dialog Box



6. Select the **OK** button to create the template.

   The virtual machine displays a status of image `Locked` while the template is being created. The time it takes for the template to be created depends on the size of the virtual disk and the capabilities of the storage hardware. When the template creation process completes, the template is added to the list of templates displayed on the **Templates** pane.

   You can now create new Oracle Linux virtual machines that are based on this template.

Creating a Cloud-Init Enabled Template

For Oracle Linux 7 (and later) virtual machines, you can use the Cloud-Init tool to automate the initial setup of virtual machines. Common tasks, such as configuring host names, network interfaces, and authorized keys, can be automated by using this tool. When provisioning virtual

machines that have been deployed based on a template, the Cloud-Init tool can be used to prevent conflicts on the network.

Before you create Cloud-Init enabled templates, ensure the following prerequisites are met:

- You must have sealed an Oracle Linux for use as a template. For more information, see Sealing an Oracle Linux Virtual Machine for Use as a Template.
- You must create a template. For more information, see Creating an Oracle Linux Template.
- The `cloud-init` package must first be installed on the virtual machine. After installation, the Cloud-Init service starts during the boot process and searches for instructions on what to configure. Use the **Run Once** window to provide these instructions on a one-time only basis.

Installing the Cloud-Init Package

> **Note**
>
> The following procedure assumes your OS is Oracle Linux 8 or later.
> To install Cloud-Init on a virtual machine:

1. Sign in to a Oracle Linux virtual machine.
2. List the `cloud-init` package.

   ```
   dnf list cloud-init
   ```

3. Install the `cloud-init` package.

   ```
   dnf install cloud-init
   ```

4. Run the following command to enable the `cloud-init` service.

   ```
   systemctl enable cloud-init
   ```

5. Run the following command to start the cloud-init service.

   ```
   systemctl start cloud-init
   ```

Using Cloud-Init to Automate the Initial Setup of a Virtual Machine

To use Cloud-Init to automate the initial setup of a virtual machine:

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

   The **Templates** pane opens with the list of templates that have been created.
2. Select a template and select the **Edit** button.
3. Select **Show Advanced Options**.
4. Select the **Initial Run** tab and select the **Use Cloud-Init/Sysprep** checkbox.
5. Expand the **Authentication** section.

- Select the **Use already configured password** checkbox to use the existing credentials, or clear that checkbox and enter a `root` password in the **Password** and **Verify Password** text fields to specify a new `root` password.
- Enter any SSH keys to be added to the authorized hosts file on the virtual machine in the **SSH Authorized Keys** text area.
- Select the **Regenerate SSH Keys** checkbox to regenerate SSH keys for the virtual machine.

6. Expand the **Networks** section.
   - Enter any DNS servers in the **DNS Servers** text field.
   - Enter any DNS search domains in the **DNS Search Domains** text field.
   - Select the **In-guest Network Interface** checkbox and use the **+ Add new** and **- Remove selected** buttons to add or remove network interfaces to or from the virtual machine.

   > ⚠ **Caution**
   >
   > You must specify the correct network interface name and number (for example, `eth0`, `eno3`, `enp0s`); otherwise, the virtual machine's interface connection will be up but will not have the Cloud-Init network configuration.

7. Expand the **Custom Script** section and enter any custom scripts in the **Custom Script** text area.

Importing an Oracle Linux Template

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, select **Compute** and then select **Templates**.
5. Select **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. Select **Load**.
8. From the **Virtual Machines on Source** list, select the virtual appliance's checkbox.

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

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

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

You can now create a virtual machine from your imported template.

Creating an Oracle Linux Virtual Machine from a Template

To create an Oracle Linux virtual machine from a template:

1. Go to **Compute** and then select **Virtual Machines**.
2. Select **New VM**.
3. From the **Template** dropdown list, select the required template from the dropdown list. For example, select the template created in Creating an Oracle Linux Template.
4. On the **Cluster** dropdown list, select the data center and host cluster for the new host.

   The **Default** data center is preselected in the dropdown list.

   For the procedures to create new data centers or a new clusters, see Data Centers or Clusters tasks.
5. At a minimum, complete the following key fields.

   For example, if the new Oracle Linux virtual machine that's being created is based on the template that was created in Creating an Oracle Linux Template:

   - **Name** - enter a name for the virtual machine, for example `ol7-vm2`.
   - **Cluster** - select a cluster or leave **Default** option selected.
   - **Template** - select a template, for example, `ol7-vm-template`.
   - **Operating System** - select an OS, for example, `Oracle Linux 7.x x64`.
   - **nic1** - select a logical network, for example, `vm_pub`.
6. Select **OK** to create the virtual machine from the template.

   After creation, the new virtual machine appears in the **Virtual Machines** pane and shows a status of `Down`.
7. Highlight the virtual machine that you created from the template. From the dropdown arrow next to **Run**, select **Run Once** to customize the template on-the-fly to create users, set passwords, configure the network.

   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.

Depending on the template, you might need to configure the Cloud-Init option when you run the virtual machine for the first time:

a. From the dropdown arrow next to **Run**, select **Run Once**

b. Expand **Initial Run** and check **Use Cloud-init**,

c. The hostname is populated for you. Fill in other options such as a new user and password, network configuration, timezone.

d. Add a cloud-init script.

Working with Windows Templates

For this example scenario, you prepare the Microsoft Windows virtual machine created in Creating a New Virtual Machine and then you create a Microsoft Windows template based on that virtual machine. You can then use that template as the basis to automate the initial setup of a virtual machine.

Preparing a Microsoft Windows VM for Use as a Template

The Microsoft Windows Systems Preparation Tool (Sysprep) is used to remove all system-specific details from a virtual machine before creating a template based on that virtual machine. This is necessary to prevent the same details from appearing on several virtual machines that are created based on the same template. It also ensures the functionality of other features, such as predictable vNIC order.

To use Sysprep to prepare a Microsoft Windows virtual machine for use as a template:

You have two ways to run Sysprep successfully.

1. The first way is to do the following:
   a. Open a command prompt as an Administrator.
   b. Change to the Sysprep directory:

      ```
      C:\Windows\System32\Sysprep
      ```

   c. Open Sysprep by running **sysprep.exe**
   d. In the Sysprep GUI configure the following:
      i. **System Cleanup Action**: Enter System Out-of-Box Experience (OOBE).
      ii. **Generalize**: Check to remove unique system information.
      iii. **Shutdown Options**: Select **Shutdown** to power off after Sysprep completes.
2. Or, run the Sysprep configuration using a single command in the Command Prompt:

   ```
   C:\Windows\System32\Sysprep\sysprep.exe /generalize /shutdown /oobe
   ```

3. Allow Sysprep to complete and power off the VM before continuing the template creation process.

> **Note**
>
> Generalizing the image using Sysprep is required for reliable deployment of Microsoft VMs from a template. Also, ensure that all Windows Updates and any other required drivers are present before running Sysprep.

Creating a Microsoft Windows Template

When you create a template based on a virtual machine, a read-only copy of the virtual machine's disk is created. This read-only disk becomes the base disk image of the new template, and of any virtual machines that are created based on the template.

To create a Microsoft Windows template:

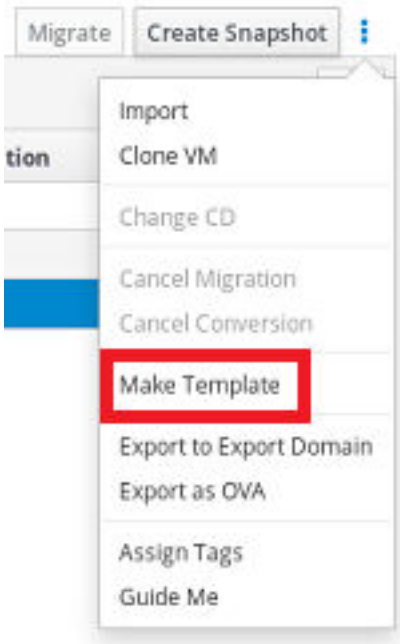
1. Go to **Compute**, and then select **Virtual Machines**.

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

2. Select **More Actions** to expand the dropdown list and select **Make Template**.

   The following screenshot shows the **More Actions** dropdown list expanded to display the **Make Template** option. The **Make Template** option is highlighted with a red rectangular box for emphasis.

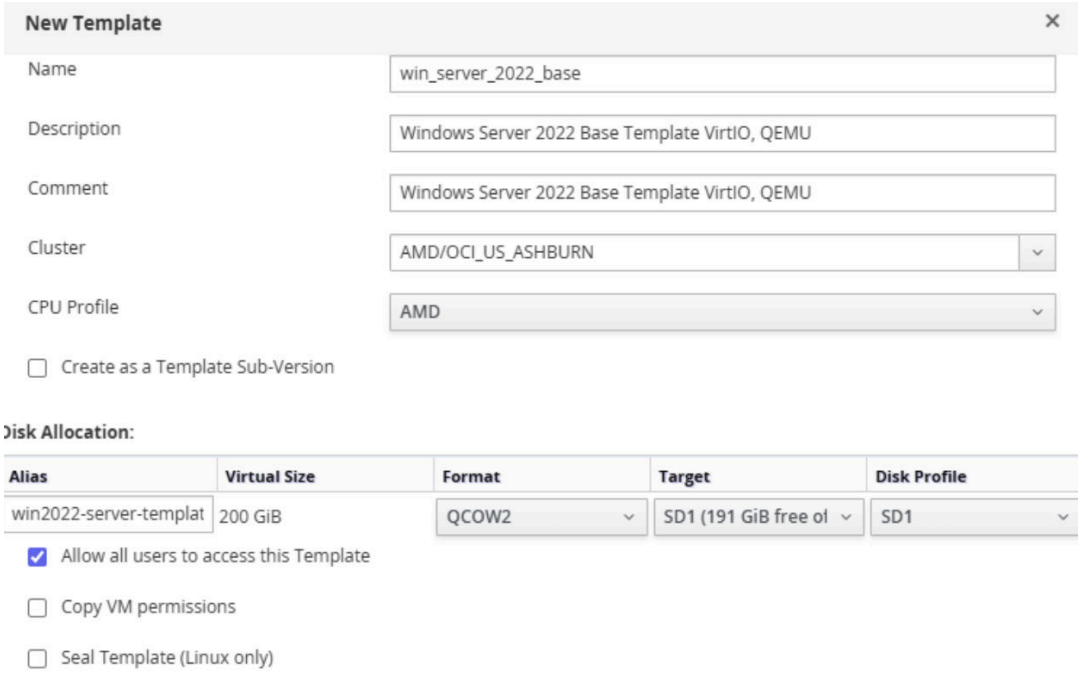
Figure 4-6 Make Template Option



3. For the **Name** field, enter a name for the new virtual machine template.
4. In the **Disc Allocation:** section under the **Alias** column, rename the disk alias to be the same as the template name entered for the **Name**

   The following screenshot shows the **New Template** dialog box completed for a new template named `win_server_2022_base`. The disk alias has been renamed to `win2022-server-template`.

Figure 4-7 New Template Dialog Box



5. Select the **OK** button to create the template.

   The virtual machine displays a status of image `Locked` while the template is being created. The time it takes for the template to be created depends on the size of the virtual disk and the capabilities of the storage hardware. When the template creation process completes, the template is added to the list of templates displayed in the **Templates** pane.

   You can now create new Microsoft Windows virtual machines that are based on this template.

Creating a Microsoft Windows Virtual Machine from a Template

To create an Microsoft Windows virtual machine from a template:

1. Go to **Compute** and then select **Virtual Machines**.
2. Select **New VM**.
3. From the **Template** dropdown list, select the required template. For example, select the template created in Creating a Microsoft Windows Template.
4. From the **Cluster** dropdown list, select the data center and host cluster for the new host.

   The **Default** data center is preselected in the dropdown list.

   For the procedures to create new data centers or new clusters, see Data Centers or Clusters.
5. Fill in the basic details for the new VM:
   - **Name** - enter a name for the virtual machine, for example `WinSvr2022-vm2`.
   - **Cluster** - select a cluster or leave **Default** option selected.
   - **Template** - select a template, for example, `WinSvr2022-vm-template`.
   - **Operating System** - select an OS, for example, `Windows Server 2022`.

- **nic1** - select a logical network, for example, `vm_pub`.

6. (Optional) Add a Sysprep answer or unattended XML file.
A Sysprep file automates Windows first boot setup on the VM, applying settings such as product key, locale, admin password, hostname, and network for rapid deployment of standardized, compliant images at scale.

> **Note**
>
> Creating a Sysprep answer or unattended XML file is beyond the scope of this document. For more information, see Microsoft's documentation at the Windows Hardware Developer website.

To specify a Sysprep XML file:

a. Select the **Initial Run** tab.

b. Select the **Sysprep** checkbox.

c. Paste the XML content of the Sysprep file directly into the **Sysprep** field.

7. Select **OK** to create the virtual machine from the template.

After creation, the new virtual machine appears in the **Virtual Machines** pane and shows a status of `Down`.

8. Highlight the virtual machine that you created from the template. From the dropdown arrow next to **Run**, select **Run Once** to customize the template on-the-fly to create users, set passwords, and configure the network.

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.

Working with Virtual Machine Snapshots

A snapshot is a view of a virtual machine's OS and applications on any or all available disks at a specific point in time. You can take a snapshot of a virtual machine before you make a change to it that might 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

> **Note**
>
> This procedure is for taking a live snapshot. The QEMU guest agent must be installed and the `qemu-guest-agent` service must be up and running.

To create a snapshot of a virtual machine:

1. Select **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. Select the **Snapshots** tab.
4. Select **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 doesn't have a disk, you can still preview a partial snapshot to view the configuration of the virtual machine.
7. **(Optional)** Select the **Save Memory** checkbox to include the virtual machine's memory in the snapshot. By default, this checkbox is selected.
8. Select **OK** to save the snapshot.

   The virtual machine's OS 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's being created. After creation, 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** dropdown 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. Select **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. Select the **Snapshots** tab.
4. On the **Snapshots** pane, select the snapshot to be used to restore the virtual machine.

5. From the **Preview** dropdown list, select **Custom**.

   On the **Virtual Machines** pane, the status of the virtual machine 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. Select **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. Select **Shutdown** to stop the virtual machine.

8. From the **Snapshot** pane, perform one of the following steps:

   a. Select **Commit** to permanently restore the virtual machine to the condition of the snapshot. Any subsequent snapshots are erased.

   b. Or, select **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, see Creating a Snapshot of a Virtual Machine.

To create a virtual machine from a snapshot:

1. Select **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. Select the **Snapshots** tab.

4. On the **Snapshots** pane, select the snapshot from which to create the virtual machine.

5. Select **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 the virtualization environment. This operation is supported on a running virtual machine and doesn't require the virtual machine to be in a `Down` state.

> **⚠ Caution**
>
> - 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 because of 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. Select **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. Select the **Snapshots** tab.
4. On the **Snapshots** pane, select the snapshot to delete.
5. Select the snapshot to delete.
6. Select **Delete**.
7. Select **OK**.

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

Security

You can encrypt communications by configuring 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's used for authentication between the engine host and KVM hosts. They continue to use the self-signed certificate generated by the Manager.

You can also enable HTTP Strict Transport Security (HSTS) to help protect websites against man-in-the-middle attacks such as protocol downgrade attacks and cookie hijacking.

> **Note**
>
> If you're required to be compliant with the Federal Information Processing Standard (FIPS), you can enable FIPS mode for the Oracle Linux Virtualization Manager deployment. See *FIPS Mode Deployment* in the Oracle Linux Virtualization Manager: Getting Started.

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**
>
> Don't 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/certs/apache.cer /etc/pki/ovirt-engine/certs/
apache.cer.bck
```

```
cp /etc/pki/ovirt-engine/keys/apache.key.nopass /etc/pki/ovirt-engine/keys/
apache.key.nopass.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 /etc/pki/ovirt-engine/keys/apache.key.nopass? 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 /etc/pki/ovirt-engine/certs/apache.cer? y
```

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

```
systemctl restart httpd
```

```
systemctl restart ovirt-engine
```

8. Create a 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
```

Enabling HTTP Strict Transport Security

To enable HTTP Strict Transport Security, complete the following steps.

1. For the ovirt-engine service port 443, create a configuration file for `httpd`, for example:

```
cat ovirt-enable-strict-transport-security.conf


LoadModule headers_module modules/mod_headers.so
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains"
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</IfModule>


systemctl restart httpd
```

For the ovirt-imageio service port, edit the `_internal/http.py` file:

```
vi /usr/lib64/python3.6/site-packages/ovirt_imageio/_internal/http.py


class Response:

    def __init__(self, con):
        self._con = con
        self.status_code = OK
        self.headers = Headers({"content-length": 0, "Strict-Transport-
Security": "max-age=31536000"})
        self._started = False


systemctl restart ovirt-imageio


curl -s -I -k https://localhost:54323


HTTP/1.1 404 Not Found
server: imageio/2.4.7
date: Wed, 13 Sep 2023 16:56:45 GMT
content-length: 19
Strict-Transport-Security: max-age=31536000
content-type: text/plain; charset=UTF-8
```

For the ovirt-provider-ovn service port, edit the `server.py` file:

```
vi /usr/lib64/python3.6/http/server.py


def send_response(self, code, message=None):
        """Add the response header to the headers buffer and log the
        response code.

        Also send two standard headers with the server software
        version and the current date.

        """
        self.log_request(code)
        self.send_response_only(code, message)
        self.send_header('Server', self.version_string())
        self.send_header('Date', self.date_time_string())
       # Oracle Bug-33308887: added below header for security scans
        self.send_header("Strict-Transport-Security", "max-age=31536000")


systemctl restart ovirt-provider-ovn


curl -s -I -k https://localhost:35357


HTTP/1.0 501 Unsupported method ('HEAD')
Server: BaseHTTP/0.6 Python/3.6.8
Date: Wed, 13 Sep 2023 17:34:32 GMT
Strict-Transport-Security: max-age=31536000
Connection: close
Content-Type: application/json
Content-Length: 137
```

For the ovirt-websocket-proxy service port, edit the `response.py` file.

```
vi /usr/lib/python3.6/site-packages/webob/response.py


Initialize headers
        self._headers = None
        if headerlist is None:
            self._headerlist = []
        else:
            self._headerlist = headerlist
```

```
        self._headerlist.append(('Strict-Transport-Security', 'max-
age=31536000'))


systemctl restart ovirt-websocket-proxy


curl -s -I -k https://localhost:6100


HTTP/1.1 405 Method Not Allowed
Server: WebSockify Python/3.6.8
Date: Wed, 13 Sep 2023 18:31:12 GMT
Strict-Transport-Security: max-age=31536000
Connection: close
Content-Type: text/html;charset=utf-8
Content-Length: 472
```

Monitoring and Diagnostics

The following section explains how to setup and use Grafana dashboards and event notifications for the virtualization environment, and about Performance Co-Pilot metrics.

Using Event Notifications

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

For more information, see Event Logging and Notifications in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

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. Sign in to the host that's 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 the 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, see oVirt Documentation.

5. Enable and restart the `ovirt-engine-notifier` service to activate the 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. 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 select **Users**.

   The **Users** pane opens.

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

> **Note**
>
> A user doesn't appear in the Administration Portal until the user is created and assigned appropriate permissions. For more information, see Creating a New User Account.

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

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

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

   The events available for notification are grouped under topic areas. By default, selecting the checkbox 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 selecting **Expand All** or **Collapse All**. You can also select 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. Select **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 select **Users**.

   The **Users** pane opens.

2. Under the **User Name** column, select the name of the user to display the detailed view for the user.
3. Select the **Event Notifier** tab.
4. Select **Manage Events**.

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

5. Select **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. Select **OK** to save the 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 the 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. Several SNMP managers can belong to a single community. Management systems and agents can communicate only if they're 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 the SNMP manager before proceeding.

To configure SNMP traps on the Engine:

1. Sign in to the host that's 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 configuration settings across reboots, define an override file for the 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 don't 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.

Using Grafana

The following section explains how to setup and use Grafana dashboards in the virtualization environment.

> ⚠ **Caution**
>
> You must install the Data Warehouse database, the Data Warehouse service and Grafana all on the same machine, even though you can install each of these components on separate machines from each other.

For more information, see Monitoring with Grafana in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

For more information on using Grafana, see the Grafana website.

Installing Grafana

Grafana integration is enabled and installed by default when you run `engine-setup` in a standard engine or a self-hosted engine installation. In some scenarios, you might need to install Grafana manually, such as performing an upgrade, restoring a backup, or after migrating the data warehouse to a separate machine.

To install Grafana manually:

1. ***(Self-hosted engine only)*** Put the environment in global maintenance mode:

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

2. Sign in to the machine where you want to install Grafana. This must be the same machine where the data warehouse is configured. This is usually the engine machine.
3. Run the `engine-setup` command as follows to start the reconfiguration for Grafana:

```
engine-setup --reconfigure-optional-components
```

4. Press Enter to answer `Yes` to install Grafana on this machine.

   ```
   Configure Grafana on this host (Yes, No) [Yes]:
   ```

5. ***(Self-hosted engine only)*** Disable global maintenance mode.

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

After installation, you can access the Grafana dashboards in one of the following ways:

- Go to `https://<engine FQDN or IP address>/ovirt-engine-grafana`.
- Click **Monitoring Portal** in the web administration welcome page.

For more information, see Default Grafana Dashboards in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

Configuring Users for Single Sign-On with Grafana

Even though `engine-setup` automatically configures Grafana to let existing users sign in from the Administration Portal, it doesn't automatically create these users within Grafana.

To configure a user for single sign-on with Grafana:

1. Sign in to the host that's running the Manager.
2. Edit the user account to add an email address if not already defined, for example:

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


   updating user test1...
   user updated successfully
   ```

3. Sign in to Grafana with an existing admin user (the initially configured admin).
4. Navigate to Configuration and then Users and select Invite.
5. Enter the email address and name of the user account and select a Role.
6. Send the invitation using one of these options:
   - Select **Send invite mail** and select **Submit**. For this option, you need an operational local mail server configured on the Grafana machine.
   - Select **Pending Invites**
     - Find the entry you want
     - Select **Copy invite**
     - Use this link to create the account by pasting it directly into a browser address bar, or by sending it to another user.

> **Note**
>
> If you use the Pending Invites option, no email is sent, and the email address isn't important. Any valid-looking address works, if it's configured as the email address of a Manager user.

To sign in with this account:

1. Sign in to the Administration Portal using the account that has the email address configured in Grafana.
2. Select `Monitoring Portal` to open the Grafana dashboard.
3. Select **Sign in with oVirt Engine Auth**.

Using Performance Co-Pilot

Performance Co-Pilot (PCP) is enabled by default for all Oracle Linux KVM installs and collects OS and network metrics that you can use to diagnose performance issues and that can help the support team to resolve issues. The logs generated and archived by the `pmlogger` service are stored in the `/var/log/pcp/pmlogger/<hostname>` directory and are collected using `sosreport`.

By default, PCP keeps logs for 7 days and compresses them after 1 day as defined in the `pmlogger_timers` file. For example:

```
grep PMLOGGER_DAILY_PARAMS /etc/sysconfig/pmlogger_timers |grep -v '#'


PMLOGGER_DAILY_PARAMS="-x 1 -k 7 -R"
```

The archive logs are rotated after 100MB and can be controlled using the following parameter:

```
grep LOCALHOSTNAME /etc/pcp/pmlogger/control.d/local


LOCALHOSTNAME y n PCP_ARCHIVE_DIR/LOCALHOSTNAME -r -T24h10m -c config.ora -v
100Mb
```

When the `pcp-oracle-conf` package is installed, only metrics collected by the `pmlogger` service are those listed in the `/var/lib/pcp/config/pmlogger/config.ora` configuration file in which the default metrics collection time is 10 seconds.

For more information, see:

- Oracle Linux 8: Collecting and Analyzing Metrics With Performance Co-Pilot.
- Oracle Linux 9: Collecting and Analyzing Metrics With Performance Co-Pilot.

Backup and Restore

You can use the `engine-backup` command utility to take regular backups of Oracle Linux Virtualization Manager. The tool backs up the engine database and configuration files into a single file and can be run without interrupting the `ovirt-engine` service.

The `engine-backup` command has two modes:

```
engine-backup --mode=backup


engine-backup --mode=restore
```

Run `engine-backup --help` for a full list of options and their function.

The basic options are:

`--mode`

Specifies whether the command performs a backup operation or a restore operation. The available options are: `backup` (default), `restore`, and `verify`.

`--file`

Specifies the path and name of the backup file, for example, *`file_name`*`.backup`. For backup mode, the file is where backups are saved. For restore mode, the fileis read as backup data. The default path is `/var/lib/ovirt-engine-backup/`.

`--log`

Specifies the path and name of the log file, for example, *`log_file_name`*. This file logs the backup or restore operations. The default path is `/var/log/ovirt-engine-backup/`.

`--scope`

Specifies the scope of the backup or restore operation and can be specified many times in the same `engine-backup` command. The options are:

- `all` (default) - back up or restore all databases and configuration data
- `files` - back up or restore only files on the system
- `db` - back up or restore only the Engine database
- `dwhdb` - back up or restore only the Data Warehouse database

For more information on backup and restore, see the oVirt documentation Administration Guide.

Backing Up the Manager

To backup the Manager:

1. Sign in to the host that's running the Manager.

   > **Note**
   >
   > When running the Manager within a virtual machine (standalone or self-hosted engine) sign in to the virtual machine that's running the engine.

2. Create a full backup of the Manager. You do *not* need to stop the ovirt-engine service before creating the backup.

   ```
   engine-backup --mode=backup --scope=all --file=path --log=path
   ```

For example, use the engine-backup command to create a full backup of the Manager. A backup file and log file for the Manager backup is created in the path specified.

```
engine-backup --mode=backup --scope=all --file=backup/file/ovirt-engine-
backup --log=backup/log/ovirt-engine-backup.log


Backing up:
Notifying engine
- Files
- Engine database 'engine'
- DWH database 'ovirt_engine_history'
Packing into file 'backup/file/ovirt-engine-backup'
Notifying engine
Done.
```

3. **(Optional)** Set up a `cron` job to take regular backups.

   By default, the Manager doesn't take automatic backups. We recommend that you take you regular backups of the Manager.

   The following example shows a sample `cron` job defined in a `crontab`-format file.

```
today=`date +'%Y%m%d-%H%M'`
engine-backup --mode=backup --scope=all --file=/backup/file/ovirt-engine-
backup-${today}
--log=/backup/log/ovirt-engine-backup-${today}.log
```

Restoring a Full Backup of the Manager

To restore a full backup of the Manager:

1. From a command line, sign in to the host running the Manager.

   > **Note**
   >
   > When running the Manager within a virtual machine (standalone or self-hosted engine) sign in to the virtual machine running the engine.

2. Clean up the objects associated with the Manager using the `engine-cleanup` command. This removes the configuration files and cleans the database associated with the Manager.

```
engine-cleanup
```

   For example, the output from the `engine-cleanup` command:

```
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
          Configuration files: ...
          Log file: ...
          Version: otopi-1.7.8 (otopi-1.7.8-1.el7)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
[ INFO  ] Stage: Environment customization
```

```
          Do you want to remove all components? (Yes, No) [Yes]: Yes
          The following files were changed since setup:
          /etc/ovirt-engine/engine.conf.d/11-setup-sso.conf
          Remove them anyway? (Yes, No) [Yes]: Yes

          --== PRODUCT OPTIONS ==--

[ INFO  ] Stage: Setup validation
          During execution engine service will be stopped (OK, Cancel)
[OK]: OK
          All the installed ovirt components are about to be removed ...
(OK, Cancel)
          [Cancel]: OK
[ INFO  ] Stage: Transaction setup
[ INFO  ] Stopping engine service
[ INFO  ] Stopping ovirt-fence-kdump-listener service
[ INFO  ] Stopping dwh service
[ INFO  ] Stopping Image I/O Proxy service
[ INFO  ] Stopping vmconsole-proxy service
[ INFO  ] Stopping websocket-proxy service
[ INFO  ] Stage: Misc configuration
[ INFO  ] Stage: Package installation
[ INFO  ] Stage: Misc configuration
[ INFO  ] Backing up PKI configuration and keys
...
[ INFO  ] Clearing Engine database engine
...
[ INFO  ] Clearing DWH database ovirt_engine_history
[ INFO  ] Removing files
[ INFO  ] Reverting changes to files
...
[ INFO  ] Stage: Transaction commit
[ INFO  ] Stage: Closing up

          --== SUMMARY ==--

          Engine setup successfully cleaned up
          A backup of PKI configuration and keys is available at ...
          ovirt-engine has been removed
          A backup of the Engine database is available at ...
          A backup of the DWH database is available at ...

          --== END OF SUMMARY ==--

[ INFO  ] Stage: Clean up
          Log file is located at ...
[ INFO  ] Generating answer file ...
[ INFO  ] Stage: Pre-termination
[ INFO  ] Stage: Termination
[ INFO  ] Execution of cleanup completed successfully
```

3. Restore a full backup of the Manager using the `engine-backup` command.

```
engine-backup --mode=restore --scope=all --file=path --log=path --restore-
permissions
```

For example:

```
engine-backup --mode=restore --scope=all --file=backup/file/ovirt-engine-
backup --log=backup/log/ovirt-engine-backup.log --restore-permissions


Preparing to restore:
- Unpacking file 'backup/file/ovirt-engine-backup'
Restoring:
- Files
- Engine database 'engine'
  - Cleaning up temporary tables in engine database 'engine'
  - Updating DbJustRestored VdcOption in engine database
  - Resetting DwhCurrentlyRunning in dwh_history_timekeeping in engine
database
  - Resetting HA VM status
------------------------------------------------------------------------
---
Please note:

The engine database was backed up at 2019-03-25 12:48:02.000000000 -0700 .

Objects that were added, removed or changed after this date, such as
virtual
machines, disks, etc., are missing in the engine, and will probably require
recovery or recreation.
------------------------------------------------------------------------
---
- DWH database 'ovirt_engine_history'
You should now run engine-setup.
Done.
```

4. Run the engine-setup command to complete the setup of the restored Manager. This reconfigures the firewall and ensures that the Manager service is correctly configured.

   ```
   engine-setup
   ```

5. Sign in to the Manager and verify that the Manager has been restored to the backup.

5

Deployment Optimization

You can configure Oracle Linux Virtualization Manager so that the cluster is optimized and the 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 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 or memory ballooning or Kernel Same-Page Merging (KSM) aren't available options.

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, see 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. Select a setting for **Memory Optimization**:
   - **None** - Disable memory overcommit

     Disables memory page sharing, which lets you 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 enable guests to use host threads as virtual CPU cores.

   Enabling hosts to run virtual machines with the total number of processing cores greater than the number of cores in the host might 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. Select **OK** to save the 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 the cluster, you can change the setting manually.

1. From a a command line, sign in to 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 must 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 checkbox is selected by default, ensure 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 checkbox 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 checkbox 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 preexisting 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 checkbox is selected by default.

8. Select **OK** to save the changes.

Configuring a Highly Available Host

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

A *nonoperational* host is different from a *nonresponsive* host. A nonoperational host can communicate with the Manager, but isn't configured correctly, for example a missing logical network. A nonresponsive host can't communicate with the Manager. In a fencing operation, if a host becomes nonresponsive, it's rebooted. If after a prescribed period the host remains nonresponsive, manual intervention must 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 nonresponsive 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, see 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, ensure 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. Select **Compute** and select **Hosts**.
2. Select a host and select **Edit**.
3. Select 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.

    Note

    See *Supported Power Management (Fencing) Devices* (Doc ID 3078282.1)
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 tries to fence nonresponsive 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 tries 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 nonoperational.
- 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 doesn't 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's always enough capacity to restart virtual machines.

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

1. Select **Compute** and then **Virtual Machines**.
2. In the list of virtual machines, select to highlight a virtual machine and then select **Edit**.
3. In the **Edit Virtual Machine** window, select 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** dropdown list, select **No VM Lease** (default) to disable the functionality or select a storage domain to hold the virtual machine lease.

   Virtual machines can acquire a lease on a special volume on the storage domain. This lets a virtual machine 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** dropdown 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. Select **OK**.

Optimizing Virtual Machine Performance

You can optimize a virtual machine for high performance, so that it runs with performance metrics as close to bare metal as possible. When you select 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 isn't 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 might 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.

Configuring a High Performance Virtual Machine

To configure a high performance virtual machine:

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

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

2. Select **OK**.

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

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

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

3. If the **High Performance Virtual Machine/Pool Settings** screen appears, select **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.

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

4. Select **OK**.

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

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 select 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 preallocated 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 unplug 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.

   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'Download the QEMU guest agent to the Manager hosts 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 can't be hot unplugged to less vCPUs than it was created with.

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

- The virtual machine's OS 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. Select **Compute** and then select **Virtual Machines**.
2. Select a virtual machine that's running and select **Edit**.

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

   > **Note**
   >
   > Hot plugging virtual memory is only possible if the Max Memory Size is greater than the configured memory size.

5. Select **OK**.

   The **Pending Virtual Machine changes** window opens.
6. Select **OK** for the changes to take place immediately or check **Apply later** and then **OK** to wait for the next virtual machine restart.
7. Select **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 OS must support memory hot unplugging.
- The virtual machine must not have a memory balloon device enabled.

To hot unplug virtual memory:

1. Select **Compute** and then select **Virtual Machines**.
2. Select on the name of a virtual machine that's running.

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

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

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

Configuring Windows Failover Clustering

You can configure Windows Failover Clustering for Windows server nodes running as virtual machines within Oracle Linux Virtualization Manager.

The Windows Failover Clustering software requires a Windows domain controller running the Active Directory Domain Services (ADDS) and Domain Name System (DNS). The ADDS domain controller must run separate to the Windows Failover Clustering nodes.

> **Note**
>
> Windows Server 2016 introduced Workgroup Cluster, an alternative configuration that doesn't require a domain controller running ADDS; in a Workgroup Cluster, the nodes join a workgroup instead of a domain. A DNS server is, however, still required. A Workgroup Cluster can be less expensive and require fewer hardware resources than a traditional Failover Cluster without sacrificing high availability.

The procedure requires that the Windows Failover Clustering Node VMs are already created, are running the same Windows server version, are in the same time zone, and are joined to the same Windows domain. The Windows Failover Clustering tool must be installed on all the node VMs.

At least one shared LUN is required to attach to each of the cluster nodes.

KVM Host Configuration

Configure each KVM host using the following steps:

1. On each KVM host, create `/etc/multipath/conf.d/scsi3-reservation.conf` with the following content:

   ```
   defaults {
      reservation_key file
   }
   ```

2. Reload the `multipathd.service`:

   ```
   systemctl reload multipathd.service
   ```

3. Confirm the `reservation_key file` option is enabled, by running the following command:

   ```
   multipath -t | grep reservation_key
   ```

Configuring the Engine

Configure the engine using the following steps:

1. Enable disk error propagation for storage reliability:

   ```
   engine-config -s PropagateDiskErrors=true
   ```

2. Restart the engine to apply changes:

   ```
   systemctl restart ovirt-engine.service
   ```

3. Validate the engine configuration changes:

   ```
   engine-config -g PropagateDiskErrors
   ```

Windows Failover Cluster Node Configuration

1. Configure the first virtual machine node in the cluster to add the direct LUN into the cluster.
   a. Edit the first virtual machine in the cluster and create disk by selecting the **+** button, then selecting the **Create** button.
   b. In the **New Virtual Disk** dialog, select the **Direct LUN** button:
      - Provide a name for the virtual disk.
      - In the Discover Targets section, enter the portal Address and select discover.
      - Login to the LUN and expand the Target Name section. Check the radio box for the LUN to use.
      - Check the **Shareable** and **Enable SCSI-3 Reservation** checkboxes.

      > ⚠ **Caution**
      >
      > – When enabling SCSI-3 reservations, passthrough must be enabled for that LUN. Note that for all other LUNs, passthrough is unsupported and disabled by default.
      > – Live migration of VMs isn't possible when using SCSI-3 reservation.

   c. Select **OK** to confirm and close.

> ⓘ **Note**
>
> You must create the disk by editing the virtual machine and using the **New Virtual Disk** dialog. The required options that you need to enable are only available in this dialog. You can't enable these options if you add the disk by using the **Storage** → **Disks** → **New** dialog.

2. Attach the disk to each of the remaining virtual machines in the cluster. For each virtual machine:
   a. Edit the virtual machine and create disk by selecting the **+** button, then selecting the **Attach** button.
   b. In the **Attach Virtual Disk** dialog, select the **Direct LUN** button:
      - Check the radio button for the LUN from the list, and select **Ok**.
      - Select **Ok** again to close the **Edit Virtual Machine** window and save the configuration.
      - Edit the virtual machine again, and edit the recently added Direct LUN.
      - Check the **Enable SCSI-3 Reservation** checkbox, and select **Ok** to close the Virtual Disk definition.
   c. Select **OK** to confirm and close.
3. Use the Disk Management tool in one of the virtual machines, to bring the virtual disk online, and initialize it.

Validating Windows Failover Clustering Resources

Validate the Windows Failover Clustering resources before creating the Windows Failover Cluster.

You can perform validation by using either the **Failover Cluster Manager** tool, or by running a command in PowerShell on one of the Windows virtual machine guests:

- Run the **Failover Cluster Manager** tool.
  1. Under the **Management** section, go to the **Actions** pane on the right side of the window and select **Validate Configuration**.
  2. On the **Before You Begin** page, select **Next**.
  3. On the **Select Servers or a Cluster** page in the **Enter name** field, enter the hostnames for each of the nodes in the cluster as a comma-separated list and select **Next**.
  4. On the **Testing Options** page, select **Run all tests (recommended)** and then select **Next**.
  5. On the **Confirmation** page, select **Next** to run the tests.
  6. Review the output on the **Summary** page.
- Run PowerShell:
  1. Run the `Test-Cluster` command:

     ```
     Test-Cluster –Node node1, node2, node3
     ```

     Substitute *node1*, *node2*, and *node3* with the resolvable hostnames of the nodes in the cluster.
  2. Review the output.

Create the Windows Failover Cluster

You can create the Windows Failover Cluster by using either the **Failover Cluster Manager** tool or by running a command in PowerShell on one of the Windows virtual machine guests:

- Run the **Failover Cluster Manager** tool.
  1. Select the **Create Cluster** link under the **Management** section.
  2. On the **Before You Begin** page, select **Next**.
  3. On **Select Servers** page, in the **Enter name** field, enter the hostnames of each of the nodes in the cluster, separate the names by a comma or a semicolon. Select **Next**.
  4. On the **Access Point for Administering the Cluster** page, provide a name for the cluster in the **Cluster Name** field. Select **Next**.
  5. Review the final confirmation page, which provides details of the cluster. Create the cluster by selecting **Next**.
  6. Review the summary and select **Finish**.
- Run PowerShell:
  1. Run the `Test-Cluster` command:

     ```
     New-Cluster -Name cluster1 -Node node1,node2,node3
     ```

     Substitute *cluster1* with a suitable name for the cluster. Substitute *node1*, *node2*, and *node3* with the resolvable hostnames of the nodes in the cluster.

6

Upgrading Your Environment to 4.5

You can upgrade Oracle Linux Virtualization Manager environment from the latest version of 4.4 to 4.5 by upgrading the engine or self-hosted engine and KVM hosts.

Warning

Upgrading from 4.4 to 4.5 with Gluster 8 storage in the environment is supported for Oracle 8 KVM hosts, but isn't supported for Oracle 9 KVM hosts.

Important

Before you upgrade, be aware of the following prerequisites.

- The engine, self-hosted engine, and KVM hosts must be updated to the latest version of 4.4. See Updating engine or self-hosted engine to latest version of 4.4 and Updating KVM hosts to the latest version of 4.4.
- All Oracle Linux 7 KVM hosts must be upgraded to Oracle Linux 8 before you upgrade the environment to 4.5.
- Upgrading from 4.3 to 4.5 isn't supported. You must first upgrade to the latest version of 4.4. For more information, see the Oracle Linux Virtualization Manager 4.4: Administration Guide.
- If you use a local yum mirror configuration, you must redo the yum mirror configuration in the engine and hosts *after* updating the ovirt-release package and *before* running the update.

Important

(**ULN registered hosts only**) Before you begin the update process, run the following commands on the engine and KVM hosts:

```
echo "Disabling yum module virt:ol"
/usr/bin/dnf -y module disable virt:ol  > /dev/null

echo "Enabling yum module virt:kvm_utils3"
/usr/bin/dnf -y module enable virt:kvm_utils3  > /dev/null

echo "Enabling module pki-deps"
/usr/bin/dnf -y module enable pki-deps  > /dev/null

echo "Enabling module postgresql:13"
/usr/bin/dnf -y module enable postgresql:13  > /dev/null

echo "Enabling module nodejs:18"
/usr/bin/dnf -y module reset nodejs  > /dev/null
/usr/bin/dnf -y module enable nodejs:18  > /dev/null

echo "Exclude ansible-core updates from OL ol8_appstream"
/usr/bin/dnf -y config-manager --setopt="exclude=ansible-core"  > /dev/
null

echo "Version lock gluster-ansible* packages"
/usr/bin/dnf install -y 'dnf-command(versionlock)' > /dev/null
/usr/bin/dnf versionlock gluster-ansible-cluster-1.0-2.1*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-features-1.0.5-9*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-infra-1.0.4-18*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-maintenance-1.0.1-10*  > /dev/
null
/usr/bin/dnf versionlock gluster-ansible-repositories-1.0.1-3*  > /dev/
null
/usr/bin/dnf versionlock gluster-ansible-roles-1.0.5-23*  > /dev/null
```

Considerations

- 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 after it reboots. You must reboot any running or suspended virtual machines as soon as possible to apply the configuration changes.
- Ensure the environment meets the requirements for 4.5. See Requirements and Scalability Limits in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.
- Clusters must have a minimum of two (2) KVM hosts (in the same cluster) to perform a live migration of virtual machines when you are upgrading the specific KVM host that's in Maintenance mode.

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

Before You Begin

Before you begin the upgrade process:

- Read the upgrade prerequisites that are common to both a standard environment and a self-hosted engine environment.
- If applicable, read the additional prerequisites for a self-hosted engine environment.
- Ensure the engine or self-hosted engine and KVM hosts are updated to the latest version of 4.4 ***before*** you begin the upgrade process. See Updating engine or self-hosted engine to latest version of 4.4 and Updating KVM hosts to the latest version of 4.4.
- Don't upgrade the KVM hosts until after the engine or self-hosted engine upgrade is completed.
- Upgrade all Oracle Linux 7 KVM hosts to Oracle Linux 8.
- (**ULN registered hosts only**) Run the following commands on the engine and KVM hosts:

```
echo "Disabling yum module virt:ol"
/usr/bin/dnf -y module disable virt:ol  > /dev/null

echo "Enabling yum module virt:kvm_utils3"
/usr/bin/dnf -y module enable virt:kvm_utils3  > /dev/null

echo "Enabling module pki-deps"
/usr/bin/dnf -y module enable pki-deps  > /dev/null

echo "Enabling module postgresql:13"
/usr/bin/dnf -y module enable postgresql:13  > /dev/null

echo "Enabling module nodejs:18"
/usr/bin/dnf -y module reset nodejs  > /dev/null
/usr/bin/dnf -y module enable nodejs:18  > /dev/null

echo "Exclude ansible-core updates from OL ol8_appstream"
/usr/bin/dnf -y config-manager --save --setopt="exclude=ansible-core"
> /dev/null

echo "Version lock gluster-ansible* packages"
/usr/bin/dnf install -y 'dnf-command(versionlock)' > /dev/null
/usr/bin/dnf versionlock gluster-ansible-cluster-1.0-2.1*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-features-1.0.5-9*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-infra-1.0.4-18*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-maintenance-1.0.1-10*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-repositories-1.0.1-3*  > /dev/null
/usr/bin/dnf versionlock gluster-ansible-roles-1.0.5-23*  > /dev/null
```

Prerequisites

Ensure the following prerequisites are met before beginning the upgrade:

- The OS for the engine host must be Oracle Linux 8.8 or later (8.x). If it's not, then update the engine host before you upgrade the engine to 4.5.
- All data centers and clusters in the environment must have the cluster compatibility level set to version 4.6 or later.
- All virtual machines in the environment must have the same compatibility level as their cluster.
- 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 can sign 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.
- For self-hosted engine environments:
  - Make note of the MAC address of the self-hosted engine if you're using DHCP and want to use the same IP address. The deploy script prompts you for this information.
  - Set the cluster scheduling policy to `cluster_maintenance` to prevent automatic virtual machine migration during the upgrade.

Updating engine or self-hosted engine to latest version of 4.4

Before upgrading to 4.5, you must update the engine or self-hosted engine to the latest version of 4.4.

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. Sign in to the KVM host where the self-hosted engine is running or any KVM host configured to run the self-hosted engine.
     b. Enable global maintenance mode:

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

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

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

     d. You should see the following message:

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

2. On the engine or self-hosted engine machine, update to the latest Oracle Linux Virtualization Manager Release 4.4 package.

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

3. Check for updated packages:

   ```
   engine-upgrade-check
   ```

4. Update the setup packages:

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

5. Update the engine or self-hosted engine:

   ```
   engine-setup
   ```

   Important

   The update process might take some time. Don't stop the process before it completes.

   The `engine-setup` script:

   - Prompts you with some configuration questions

     For more information, see Engine Configuration Options in the Oracle Linux Virtualization Manager: Getting Started.
   - Stops the `ovirt-engine` service.
   - Downloads and installs the updated packages.
   - Backs up and updates the database.
   - Performs postinstallation configuration.
   - Starts the `ovirt-engine` service.

   Note

   The `engine-setup` script displays stored configuration values supplied during the initial engine installation process. These stored values display when previewing the configuration and might not be up-to-date if you ran `engine-config` after installation. However, `engine-setup` won't overwrite the updated values.
   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, `engine-setup` won't apply this value. Instead it keeps the `SANWipeAfterDelete=true` setting.

   After an update is successful, you see:

   ```
   Execution of setup completed successfully
   ```

   If the update fails, the engine-setup command trie to rollback the installation to its previous state. If you encounter a failed update, detailed instructions display explaining how to restore the installation.

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

   ```
   dnf update
   ```

7. If any core libraries or services were upgraded, reboot the system to complete the changes. Run the following command to confirm a reboot is required: `dnf needs-restarting -r`,
8. Update the KVM hosts.

> **Important**

Updating KVM hosts to the latest version of 4.4

1. In the Administration portal, go to **Compute** and then select **Hosts**.
2. In the **Hosts** pane, select a blank or non-linked cell for a host.
3. Select **Installation** and then **Check for Upgrade**.
4. From the Upgrade Host window, select **OK**.

   The engine checks the KVM host to see if it requires an update.
5. Using the mouse, hover over the icon next to the host name to see if an update is available.
6. To proceed with the update, select **Installation** and then **Upgrade**.
7. From the Upgrade Host window, select **OK** to begin the update process.

Repeat these steps to update the rest of the KVM hosts in the same cluster, one-by-one, until they're all updated to the latest version of 4.4.

When you finish updating the engine or self-hosted engine and KVM hosts, continue to Upgrading the Engine or Self-Hosted Engine.

Upgrading the Engine or Self-Hosted Engine

Before you start the upgrade process,

- Review the environment upgrade overview.
- Verify that you have satisfied the prerequisites.
- Ensure you have updated engine or self-hosted engine to latest version of 4.4.
- Ensure you have updated KVM hosts to the latest version of 4.4.

1. On the engine or self-hosted engine machine, install the Oracle Linux Virtualization Manager Release 4.5 package.

   ```
   dnf install oracle-ovirt-release-45-el8
   ```

2. Verify that the `ovirt-4.5` and `ovirt-4.5-extras` repositories are enabled.

   ```
   dnf repolist
   ```

If either repository isn't enabled, use the `dnf config-manager --enable` ***repository*** command to enable them.

> **Note**
>
> For ULN registered hosts or if you're using Oracle Linux Manager, ensure you have subscribed to `ol8_x86_64_ovirt45` and `ol8_x86_64_ovirt45_extras`.

3. Check for updated packages:

   ```
   engine-upgrade-check
   ```

4. Update the setup packages:

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

5. Check the engine host configuration or the self-hosted engine host configuration.
6. Update the engine or self-hosted engine:

   ```
   engine-setup
   ```

7. Proceed to Migrating KVM Hosts to 4.5.

> **Important**
>
> After the engine and KVM hosts are upgraded and verified, remove the release package for version 4.4 on engine and all KVM hosts:
>
> ```
> dnf remove oracle-ovirt-release-el8
> ```

Migrating KVM Hosts to 4.5

After you upgrade the engine or self-hosted engine to Oracle Linux Virtualization Manager Release 4.5, migrate each Oracle Linux 8 KVM host from the 4.4 repositories to the 4.5 repositories and then upgrade the host from the Administration Portal.

> **Important**
>
> Upgrade the KVM hosts ***after*** you complete the engine or self-hosted engine upgrade.

> **Caution**
>
> Upgrade hosts one-by-one. Ensure that the cluster has at least one other available host so that virtual machines can be migrated when you place a host into maintenance mode.

1. Pick a host to migrate and migrate (or shut down) the virtual machines that are running on the host.

   You can use Live Migration to minimize downtime. See Migrating Virtual Machines between Hosts.

   Any CPU-pinned virtual machines must be shut down and started on another available KVM host.

2. Put the host into maintenance mode.

   See Moving a Host to Maintenance Mode.

3. On the KVM host, install the Oracle Linux Virtualization Manager Release 4.5 release package:

   ```
   dnf install oracle-ovirt-release-45-el8
   ```

   > **Note**
   >
   > For ULN registered hosts or if you're using Oracle Linux Manager, ensure you have subscribed to `ol8_x86_64_ovirt45` and `ol8_x86_64_ovirt45_extras`.

4. If the host is running UEK R7, install the additional kernel modules package, and then reboot the host:

   ```
   dnf install kernel-uek-modules-extra
   reboot
   ```

5. From the Administration Portal, upgrade the host:
   a. Go to **Compute** and then select **Hosts**.
   b. Select a blank or non-linked cell for the host.
   c. Select **Installation** and then **Check for Upgrade**.
   d. From the **Upgrade Host** window, select **OK**.
   e. When an upgrade is available, select **Installation** and then **Upgrade**.
   f. From the **Upgrade Host** window, select **OK** to begin the upgrade process.
6. When the upgrade completes, activate the host.

   See Activating a Host from Maintenance Mode.

7. Repeat these steps for each remaining Oracle Linux 8 KVM host in the cluster.

> **Important**
>
> After the engine and all KVM hosts are upgraded and verified, remove the release package for version 4.4 on the engine and on all KVM hosts:
>
> ```
> dnf remove oracle-ovirt-release-el8
> ```

Upgrading KVM Hosts from Oracle Linux 8 to Oracle Linux 9

During the upgrade process, you must migrate all virtual machines to another host in the environment. Doing so minimizes the downtime of virtual machines in the environment. After the upgrade, you can reattach the KVM host to the engine and migrate the virtual machines back to the host.

Migrating KVM hosts from Oracle Linux 8 to Oracle Linux 9 requires a clean installation of Oracle Linux 9. For information about migrating Oracle Linux 8 KVM hosts to the 4.5 repositories while staying on Oracle Linux 8, see Migrating KVM Hosts to 4.5.

⚠ Caution

- When installing or reinstalling the host's OS, we strongly recommend that you first detach any existing non-OS storage from the host to avoid potential data loss from accidental initialization of these disks.
- Upgrading from 4.4 to 4.5 with Gluster 8 storage in the environment is supported for Oracle 8 KVM hosts, but isn't supported for Oracle 9 KVM hosts. You must remove Gluster before upgrading.

1. Verify the 4.5 engine is installed and running.
2. *(Optional)* Verify that all data centers and clusters in the environment are at the same compatibility level.
3. Pick a host to upgrade and migrate the 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. Any CPU-pinned virtual machines must be shut down and booted into another available KVM host before live migration.
4. Put the host into maintenance mode.

   See Moving a Host to Maintenance Mode.
5. Remove the host from the engine.

   See Removing a Host.
6. Install Oracle Linux 9.6 or later release and install the appropriate packages to enable the host for 4.5. Even if you're using the same physical machine as for 4.3 (Oracle Linux 8 KVM), the 4.4 (Oracle Linux 9 KVM) hosts require a clean installation of Oracle Linux 9.6 or later Oracle Linux 9 release.

   ⚠ Caution

   Before you install, review the prerequisites. Then, follow the instructions in *Configure a KVM Host* in the Oracle Linux Virtualization Manager Getting Started guide.

   Ensure that you select **Minimal Install** as the base environment for the installation. If you don't, the hosts will have incorrect `qemu` and `libvirt` versions, incorrect repositories configured, and no access to virtual machine consoles.

7. Add the host to the engine.

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 hosts in the cluster. The cluster compatibility is set according to the version of the least capable host OS in the cluster.

The preferred approach after upgrading the engine to 4.5 is to upgrade all hosts to 4.5 and then change the cluster compatibility to 4.7. You can then add new hosts as 4.5 hosts.

> **Important**
>
> Although the Oracle Linux Virtualization Manager version is 4.5, the corresponding compatibility version is 4.7.

Compatibility Version Restrictions

Consider these restrictions to ensure you don't 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 the data center. For example:

- If the data center compatibility level is 4.7, you can only have clusters with compatibility level 4.7.
- If the data center compatibility level is 4.4, you can have 4.4 or higher 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 4.6 compatibility version cluster, you can add 4.4 or 4.5 hosts.
- If you have a 4.7 compatibility version cluster, you can only add 4.5 hosts.

> **Note**
>
> Adding Oracle Linux 9 KVM hosts to a cluster requires a compatibility version of 4.7.

Possible Errors When Changing Compatibility Versions

- If you try to change the data center compatibility version from 4.6 to 4.7 when you have a 4.4 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 4.6 to 4.7 when you have 4.4 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 4.4 host in maintenance mode, you can change the cluster and then data center compatibility version to 4.7. However, the host shows nonoperational with the following event:

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

Changing Cluster Compatibility Versions

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

1. Verify all hosts are running a version level that supports the required compatibility level. See Compatibility Version Restrictions.
2. In the **Administration Portal**, go to **Compute** and select **Clusters**.
3. Select the cluster to change and select **Edit**.
4. From the **Edit Cluster** dialog box, select **General**.
5. For **Compatibility Version**, select the required value and select **OK**.
6. On the **Change Cluster Compatibility Version** confirmation window, select **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 other validations and warnings that show what to correct. Sometimes the issue is automatically corrected and you just need to save the virtual machine's configuration again. After editing each virtual machine, you can 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 doesn't need to be restarted.
>
> You can't change the cluster compatibility version of a virtual machine snapshot that's 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 select **Data Centers**.
3. Select the data center to change and select **Edit**.
4. From the **Edit Data Center** dialog box, change the **Compatibility Version** to the required value and then select **OK**.
5. On the **Change Data Center Compatibility Version** confirmation window, select **OK**.

7

Updating Your Environment

You can update the engine, self-hosted engine, and KVM hosts within versions, such as from 4.5 to the latest version of 4.5.

However, moving from one version to another, such as 4.4 to 4.5, is considered an upgrade. See Upgrading Your Environment to 4.5.

Updating the Engine

> **Important**
>
> If you're *upgrading* the environment, see Updating engine or self-hosted engine to 4.4 in the Upgrading Your Environment to 4.5 section.

To update the engine:

1. Update the release rpm.
   - If you're updating from 4.4 to the latest version of 4.4, run:

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

   - If you're updating from 4.5 to the latest version of 4.5, run:

     ```
     dnf update oracle-ovirt-release-45-el8
     ```

2. Verify that the 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. 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

  For more information, see Engine Configuration Options in the Oracle Linux Virtualization Manager: Getting Started.
- Stops the `ovirt-engine` service
- Downloads and installs the updated packages
- Backs up and updates the database
- Performs postinstallation configuration
- Starts the `ovirt-engine` service

> **Note**
>
> The `engine-setup` script displays stored configuration values supplied during the initial engine installation process. These stored values display when previewing the configuration and may not be up-to-date if you ran `engine-config` after installation. However, `engine-setup` won't overwrite the updated values.
> 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, `engine-setup` won't apply this value. Instead, it keeps the `SANWipeAfterDelete=true` setting.

If the update is successful, you will see:

```
Execution of setup completed successfully
```

If the update fails, the engine-setup command tries to rollback the installation to its previous state. If you encounter a failed update, detailed instructions display explaining how to restore the installation.

5. Update the base OS and any optional packages installed.

```
dnf update
```

> **Important**
>
> If any core libraries or services were upgraded, reboot the system to complete the changes. Run the following command to confirm a reboot is required: `dnf needs-restarting -r`,

Updating the Self-Hosted Engine

> **Important**
>
> If you're *upgrading* the environment, see Updating engine or self-hosted engine to 4.4 in the Upgrading Your Environment to 4.5 section.

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

1. Sign in to the 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
   ```

   The following message indicates that the cluster is in maintenance mode.

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

3. Update the release rpm.
   - If you're updating from 4.4 to the latest version of 4.4, run:

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

   - If you're updating from 4.5 to the latest version of 4.5, run:

     ```
     dnf update oracle-ovirt-release-45-el8
     ```

4. Sign in to the engine virtual machine and verify that the engine is eligible to update and if there are updates for any packages.

   ```
   engine-upgrade-check


   ...
   Upgrade available.
   ```

5. Update the setup packages and resolve dependencies.

   ```
   dnf update ovirt\*setup\*


   ...
   Complete!
   ```

6. Update the self-hosted engine:

```
engine-setup
```

> **Important**
>
> The update process might take some time. Don't stop the process before it completes.

The `engine-setup` script:

- Prompts you with some configuration questions

  For more information, see Engine Configuration Options in the Oracle Linux Virtualization Manager: Getting Started.
- Stops the `ovirt-engine` service
- Downloads and installs the updated packages
- Backs up and updates the database
- Performs postinstallation configuration
- Starts the `ovirt-engine` service

> **Note**
>
> The `engine-setup` script displays stored configuration values supplied during the initial engine installation process. These stored values display when previewing the configuration and might not be up-to-date if you ran `engine-config` after installation. However, `engine-setup` won't overwrite the updated values.
> 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, `engine-setup` won't not apply this value; rather, it keeps the `SANWipeAfterDelete=true` setting.

If the update is successful, you will see:

```
Execution of setup completed successfully
```

If the update fails, the engine-setup command tries to rollback the installation to its previous state. If you encounter a failed update, detailed instructions display explaining how to restore the installation.

7. Update the base OS and any optional packages installed on the engine.

```
dnf update
```

> **Important**
>
> If any core libraries or services were upgraded, reboot the system to complete the changes. Run the following command to confirm a reboot is required: `dnf needs-restarting -r`,

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

1. Sign in to the engine virtual machine and shut it down.
2. Sign 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

> **Caution**
>
> If you're *upgrading* the environment, see Before You Beginin the Upgrading Your Environment to 4.5 section.

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.
- Don't try 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 enough memory reserve in order for its hosts to perform maintenance. If a cluster lacks enough 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 can't 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, follow the instructions below.

Connect to the KVM host:

1. Open a terminal.
2. Connect to the KVM host.
3. Update the release rpm.

   Update Oracle Linux Virtualization Manager to the latest 4.5 release by running the following command, where *n* is the major version of Oracle Linux on the KVM host:

   ```
   dnf update oracle-ovirt-release-45-eln
   ```

After you're connected to the KVM host, complete the following steps in the Administration Portal:

1. In the Administration portal, go to **Compute** and then select **Hosts**.
2. In the **Hosts** pane, select a blank or non-linked cell for a host.
3. Select **Installation** and then **Check for Upgrade**.
4. From the Upgrade Host window, select **OK**.

   The engine checks the KVM host to see if it requires an update.
5. Using the mouse, hover over the icon next to the host name to see if an update is available.
6. To proceed with the update, select **Installation** and then **Upgrade**.
7. From the Upgrade Host window, select **OK** to begin the update process.

   On the **Hosts** pane you can watch the host transition through the update stages: **Maintenance**, **Installing**, **Up**. The host is rebooted after the update and displays a status of Up if successful. If any virtual machines were migrated off the host, they're migrated back.

   > **Note**
   >
   > If the update fails, the host's status changes to **Install Failed** and you must select **Installation** and then **Upgrade** again.

8. **(Optional)** Repeat the previous steps for any KVM host in the environment that you want to update.

8
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 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's writable on both sites. This lets virtual machines 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 must 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 100 ms is required for the standalone Engine environment
- A maximum latency of 7 ms 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's writable with shared L2 network connectivity to let virtual machines 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 the 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 100 ms 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 don't 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 doesn't 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 the 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 7 ms 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 extra 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 doesn't 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.
>
> In both environments, Manager hosts must use the same deployment type, either hosted-engine or standalone. Running disaster recovery playbooks across mixed deployment types, such as hosted-engine to standalone, or conversely, isn't supported.
>
> 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 several networks or 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.

The 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're implementing disaster recovery using a self-hosted engine, ensure that the storage domain used by the self-hosted engine's Engine virtual machine doesn't contain virtual machine disks because the storage domain won't 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 might 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 selecting **Update OVFs**. Or, you can change 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 start 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 the 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 can 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's managing the failover and failback. If you have several 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 in `/usr/share/ansible/collections/ansible_collections/ovirt/ovirt/roles/disaster_recovery`, to simplify the following 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 can optionally 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 fails if you have any virtual machine disks on the self-hosted engine's storage domain. Also, the generated mapping file won't 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 the Engine password in a `.yml` file.

2. Run the Ansible command to generate the mapping file. The primary site's configuration is 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 several 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 must 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 are released automatically about 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.

Continue with a failback. See Executing a Failback.

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
```

After 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's 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 the 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.

> **Caution**
>
> Ensure that you have completed all the steps to configure the 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 lets users 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 can't be used 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 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 lets you test the failover while users continue to work in the primary site.

> **Note**
>
> 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 let you 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.

  Define the primary site attributes, 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
  ```

  Copy the primary site attributes to the secondary site:

  ```
  scp manager2:/etc/pki/ovirt-engine/ca.pem /var/tmp/secondary-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: /var/tmp/secondary-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 several networks or 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 let virtual machines 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
```