Oracle Linux Virtualization Manager

Getting Started

F52194-21
April 2026

Oracle Linux Virtualization Manager Getting Started,

F52194-21

Contents

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

Requirements and Scalability Limits

Before you begin the tasks in this guide, review Oracle Linux Virtualization Manager Release 4.5 concepts, environment requirements, and scalability limitations in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

3 Installation and Configuration

To deploy Oracle Linux Virtualization Manager, you install and configure the engine on a host with Oracle Linux 8.8 or later (8.*x*), configure KVM hosts, storage, and networks, and create virtual machines. Ensure that you review the Requirements and Scalability Limits as the requirements for the engine host are different than the KVM hosts.

To review conceptual information and help plan the installation, see the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

> **Note**
>
> If you're required to comply 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.

Install the Engine

Install and configure the engine host.

To install Oracle Linux Virtualization Manager, you perform a fresh installation of Oracle Linux 8.8 or later (8.*x*) release on the host, install the `ovirt-engine` package, and then run the engine-setup command to configure the Manager.

> **Note**
>
> You can install the Manager in a virtual machine if it's not managing that virtual machine, or in a self-hosted engine configuration. For more information, see Self-Hosted Engine Deployment. **Don't configure the same host as a standalone engine and a KVM host.**

You can download the installation ISO for Oracle Linux from the Oracle Software Delivery Cloud at https://edelivery.oracle.com.

Configure the engine host

Complete the following steps to configure the host for installation.

1. Install Oracle Linux 8.8 or later (8.*x*) on the host using the **Minimal Install** base environment.

   Follow the instructions in Oracle® Linux 8: Installing Oracle Linux.

> **Important**
>
> Don't install any extra packages until after you have installed the Manager packages, because they might cause dependency issues.

2. **(Optional)** If you use a proxy server for Internet access, configure Yum with the proxy server settings. For more information, see the Oracle® Linux: Managing Software on Oracle Linux.
3. Complete one of the following sets of steps:
   - **For ULN registered hosts or using Oracle Linux Manager**

     Subscribe the system to the required channels and enable appstream modules.

     a. For ULN registered hosts, sign in to https://linux.oracle.com with a ULN username and password. For Oracle Linux Manager registered hosts, access the internal server URL.

     b. On the Systems tab, select the link named for the host in the list of registered machines.

     c. On the System Details page, select **Manage Subscriptions**.

     d. On the System Summary page, select each required channel from the list of available channels and select the right arrow to move the channel to the list of subscribed channels. Subscribe the system to the following channels:

        - `ol8_x86_64_baseos_latest`
        - `ol8_x86_64_appstream`
        - `ol8_x86_64_kvm_appstream`
        - `ol8_x86_64_addons`
        - `ol8_x86_64_ovirt45`
        - `ol8_x86_64_ovirt45_extras`
        - `ol8_x86_64_gluster_appstream`
        - **(For VDSM)**`ol8_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

     e. Select **Save Subscriptions**.

     f. Install the Oracle Linux Virtualization Manager Release 4.5 package. This automatically enables/disables the required repositories.

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

   - **For Oracle Linux yum server hosts**

     Install the Oracle Linux Virtualization Manager Release 4.5 package and enable the required repositories.

     a. Enable the `ol8_baseos_latest` repository.

        ```
        sudo dnf config-manager --enable ol8_baseos_latest
        ```

b. Install the Oracle Linux Virtualization Manager Release 4.5 package, which automatically enables/disables the required repositories.

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

c. Use the `dnf` command to verify that the required repositories are enabled.

i. Clear the dnf cache.

```
dnf clean all
```

ii. List the configured repositories and verify that the required repositories are enabled.

```
dnf repolist
```

The following repositories must be enabled:

- `ol8_baseos_latest`
- `ol8_appstream`
- `ol8_kvm_appstream`
- `ol8_addons`
- `ovirt-4.5`
- `ovirt-4.5-extra`
- 91 `ol8_gluster_appstream`
- **(For VDSM)** `ol8_UEKR7`

iii. If a required repository isn't enabled, use the `dnf config-manager` command to enable it.

```
sudo dnf config-manager --enable repository
```

4. If the host runs the Unbreakable Linux Kernel (UEK):

a. Install the *Extra kernel modules* package.

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

b. Reboot the host.

Check host configuration

To ensure that the engine host is configured correctly, run the precheck script **BEFORE** you install the engine. You must also run the precheck script on all KVM hosts in the environment.

> **Note**
>
> To run the script on several hosts simultaneously, we recommend using an Ansible playbook.

1. Connect to the engine host from a command line and run the precheck script:

```
sudo olvm-pre-check.py
```

A series of checks begins and you see something similar to

```
-----------------------------------
    OLVM 4.5.5 PRE-CHECK SCRIPT
-----------------------------------

+++ Checking oracle-ovirt-release-45                    [PASS]
+++ Checking if Host is installed                       [WARN]

    The 'ovirt-engine' package is already installed.
    DO NOT configure this Server as a KVM Host.

+++ Checking if a Minimal Installation                  [PASS]
+++ Validating the 'Minimal Install' Group              [PASS]
+++ Checking enabled repositories                       [WARN]

    Extra repositories are enabled:
    update-pcp

    Please run the command:
    dnf config-manager --set-disabled update-pcp

+++ Running 'dnf makecache'                             [PASS]
+++ Dry run 'dnf update --assumeno'                     [PASS]
+++ Checking Linux Kernel                               [PASS]
+++ Checking kernel-uek-modules-extra                   [PASS]
+++ Checking Firewalld status                           [PASS]
+++ Checking SELinux status                             [PASS]
+++ Checking FIPS status                                [PASS]
    FIPS is disabled.
+++ If installed, check ansible version                 [PASS]
+++ If installed, check qemu-kvm version                [PASS]
+++ If installed, check libvirt version                 [PASS]
+++ Checking Hostname/FQDN                              [PASS]
```

2. If any checks are marked **WARN** or **FAIL**, the script output provides information that can help you resolve the issues:

```
+++ Checking if Host is installed              [WARN]

    The 'ovirt-engine' package is already installed.
    DO NOT configure this Server as a KVM Host.

+++ Checking enabled repositories                       [WARN]

    Extra repositories are enabled:
    update-pcp

    Please run the command:
    dnf config-manager --set-disabled update-pcp
```

3. If you had warnings or failures to address, rerun the script to ensure that the system passes all configuration checks. For example:

```
sudo olvm-pre-check.py

---------------------------------
    OLVM 4.5.5 PRE-CHECK SCRIPT
---------------------------------

+++ Checking oracle-ovirt-release-45                    [PASS]
+++ Checking if Host is installed                       [PASS]
+++ Checking if a Minimal Installation                  [PASS]
+++ Validating the 'Minimal Install' Group              [PASS]
+++ Checking enabled repositories                       [PASS]
+++ Running 'dnf makecache'                             [PASS]
+++ Dry run 'dnf update --assumeno'                     [PASS]
+++ Checking Linux Kernel                               [PASS]
+++ Checking kernel-uek-modules-extra                   [PASS]
+++ Checking Firewalld status                           [PASS]
+++ Checking SELinux status                             [PASS]
+++ Checking FIPS status                                [PASS]
    FIPS is disabled.
+++ If installed, check ansible version                 [PASS]
+++ If installed, check qemu-kvm version                [PASS]
+++ If installed, check libvirt version                 [PASS]
+++ Checking Hostname/FQDN                              [PASS]
```

Install the engine

After you have successfully configured and verified the engine host, install the engine using the `ovirt-engine` command.

```
dnf install ovirt-engine
```

Proceed to Configure the Engine.

Configure the Engine

After you install the Oracle Linux Virtualization Manager, you run the `engine-setup` command (the Setup program) to configure the Manager. You're prompted to answer a series of questions whose values are used to configure the Manager. Some of these questions relate to features that are in technology preview. For more information, see Technology Preview in the Oracle Linux Virtualization Manager: Release Notes.

The Manager uses two PostgreSQL databases: one for the engine and one for the data warehouse. By default, Setup creates and configures the engine database locally on the engine host. Or, you can configure the engine host to use a manually-configured local or remote database. To use a manually-configured local or remote database, you must set it up **before** running `engine-setup`. Running the engine or data warehouse database on a remote host is currently a technology preview feature.

To configure the Manager:

1. Run the `engine-setup` command on the host where you installed the Manager.

   ```
   sudo engine-setup
   ```

   The program runs through some initialization steps:

   ```
   [ INFO ] Stage: Initializing
   [ INFO ] Stage: Environment setup
   Configuration files: /etc/ovirt-engine-setup.conf.d/10-packaging-
   jboss.conf, /etc/ovirt-engine-setup.conf.d/10-packaging.conf
   Log file: /var/log/ovirt-engine/setup/ovirt-engine-setup-YYYYMMDDHHMMSS-
   snz1rn.log
   [ INFO ] Stage: Environment packages setup
   [ INFO ] Stage: Programs detection
   [ INFO ] Stage: Environment setup (late)
   [ INFO ] Stage: Environment customization
   ```

   You must then answer a series of questions in the following steps to configure the Manager.

2. Enter Yes to configure Cinderlib integration, which is a Tech Preview feature. The default is *No*.

   ```
   Configure Cinderlib integration (Currently in tech preview) (Yes, No) [No]:
   ```

3. Enter Yes to configure the Manager.

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

   If you enter No, the configuration stops. To restart, rerun the engine-setup command.

4. For the remaining configuration questions, either provide a response or accept the default values, which are shown in square brackets after each question. To accept the default value for a question, press **Enter**.

> **Note**
>
> Setup asks you for the fully-qualified DNS name (FQDN) of the Manager host. Although Setup tries to automatically detect the name, you must ensure the FQDN is correct.
>
> Run `hostname -f` on the host where you installed the Manager to retrieve and confirm its FQDN.
>
> For detailed information on the configuration options, see Engine Configuration Options.

> **Tip**
>
> Keycloak integration is a fully supported feature for internal Single Sign-On (SSO) and it deprecates legacy AAA authentication. When you get to this configuration option, accept the default response of Yes.

5. After you have answered all the questions, Setup displays a list of the values you entered. Review the list and then press **Enter** to configure the Manager.

   The answers are saved to a file that can be used to reconfigure the Manager using the same values. Setup also displays the location of the log file for the configuration process.

6. When the configuration is complete, details about how to sign in to the Administration Portal are displayed. To verify that the configuration process was successful, sign in to the Administration Portal, as described in Log in to the Administration Portal.

Engine Configuration Options

The information in the section describes the options for configuring Oracle Linux Virtualization Manager when you run the engine-setup command.

> ⚠ **Caution**
>
> Some configuration options are in technology preview. For more information, see Technology Preview in the Oracle Linux Virtualization Manager: Release Notes.

OVN Provider

```
Configuring ovirt-provider-ovn also sets the Default cluster's default
network provider to ovirt-provider-ovn.
Non-Default clusters may be configured with an OVN after installation.
Configure ovirt-provider-ovn (Yes, No) [Yes]:
```

Install the Open Virtual Network (OVN) provider on the Manager host and add it as an external network provider. The default cluster is automatically configured to use OVN as its network provider.

OVN is an OVS (Open vSwitch) extension which lets you configure virtual networks.

Using external providers, including the OVN provider, is a technology preview feature.

WebSocket Proxy

```
Configure WebSocket Proxy on this machine? (Yes, No) [Yes]:
```

The WebSocket Proxy lets you connect to virtual machines using the noVNC or HTML 5 consoles.

For security and performance reasons, you can configure the WebSocket Proxy on a remote host.

Data Warehouse

```
Please note: Data Warehouse is required for the engine.
If you choose to not configure it on this host, you have to configure
it on a remote host, and then configure the engine on this host so that it
can
```

```
access the database of the remote Data Warehouse host.
Configure Data Warehouse on this host (Yes, No) [Yes]:
```

The Data Warehouse feature can run on the Manager host or on a remote host. Running Data Warehouse on a remote host reduces the load on the Manager host.

Running the Data Warehouse on a remote host is a technology preview feature.

Keycloak

Keycloak provides internal Single Sign-On (SSO) for the Engine and deprecates legacy AAA authentication. When prompted, accept the default response of `Yes` to enable Keycloak integration.

When Keycloak integration is enabled, the OVN provider and the Monitoring Portal (Grafana) are also reconfigured to use Keycloak SSO.

> **Note**
>
> If you disable Keycloak integration, the Manager uses the legacy internal domain (AAA). This configuration isn't the recommended option for new deployments.

VM Console Proxy

```
Configure VM Console Proxy on this host (Yes, No) [Yes]:
```

The VM Console Proxy lets you access virtual machine serial consoles from a command line. To use this feature, serial consoles must be enabled in the virtual machines.

Grafana

```
Use Engine admin password as initial Grafana admin password (Yes, No) [Yes]:
```

Grafana can be configured to use the Engine password to make signing in easier.

Manager DNS Name

```
Host fully-qualified DNS name of this server [<autodetected-host-name>]:
```

The fully-qualified DNS name of the Manager host. Check that the automatically detected DNS name is correct.

Automatic Firewall Configuration

```
Setup can automatically configure the firewall on this system.
Note: automatic configuration of the firewall may overwrite current settings.
Do you want Setup to configure the firewall? (Yes, No) [Yes]:
```

```
The following firewall managers were detected on this system: firewalld
Firewall manager to configure (firewalld): firewalld
```

Configure the firewall on the host to open the ports used for external communication between Oracle Linux Virtualization Manager and the components it manages.

If Setup configures the firewall, and no firewall managers are active, you're prompted to select a firewall manager from a list.

If you enter No, you must manually configure the firewall. When the Manager configuration is complete, Setup displays a list of ports that need to be opened, see for details.

Data Warehouse Database

```
Where is the DWH database located? (Local, Remote) [Local]:
```

The Data Warehouse database (the history database) can run on the Manager host or on a remote host. Running the database on a remote host reduces the load on the Manager host.

Running the database on a remote host is a technology preview feature.

> ⚠ **Caution**
>
> In this step you configure the name of the database, and the username and password for connecting to it. Make a note of these details.

Enter Local to connect to a local PostgreSQL server, or Remote to connect to an existing PostgreSQL server running on a remote host.

If you enter Local, you can either set up a local PostgreSQL server automatically, or to connect to an existing local PostgreSQL server.

```
Setup can configure the local postgresql server automatically for the DWH to
run.
This may conflict with existing applications.
Would you like Setup to automatically configure postgresql and create DWH
database,
or prefer to perform that manually? (Automatic, Manual) [Automatic]:
```

Enter Automatic to have Setup configure a local database server, or Manual to connect to an existing local database server. If you enter Manual, you're prompted for the details for connecting to the database:

```
DWH database secured connection (Yes, No) [No]:
DWH database name [ovirt_engine_history]:
DWH database user [ovirt_engine_history]:
DWH database password:
```

If you enter Remote to connect to an existing PostgreSQL server running on a remote host, you're prompted for the details for connecting to the database:

```
DWH database host [localhost]:
DWH database port [5432]:
DWH database secured connection (Yes, No) [No]:
DWH database name [ovirt_engine_history]:
DWH database user [ovirt_engine_history]:
DWH database password:
```

Engine Database

```
Where is the Engine database located? (Local, Remote) [Local]:
```

The Oracle Linux Virtualization Manager database (the engine database) can run on the Manager host or on a remote host. Running the database on a remote host reduces the load on the Manager host.

Running the database on a remote host is a technology preview feature.

> ⚠ **Caution**
>
> In this step you configure the name of the database, and the username and password for connecting to it. Make a note of these details.

Enter Local to connect to a local PostgreSQL server, or Remote to connect to an existing PostgreSQL server running on a remote host.

If you enter Local, you can choose whether to set up a local PostgreSQL server automatically, or to connect to an existing local PostgreSQL server.

```
Setup can configure the local postgresql server automatically for the engine
to run.
This may conflict with existing applications.
Would you like Setup to automatically configure postgresql and create Engine
database,
or prefer to perform that manually? (Automatic, Manual) [Automatic]:
```

Enter Automatic to have Setup configure a local database server, or Manual to connect to an existing local database server. If you enter Manual, you're prompted for the details for connecting to the database:

```
Engine database secured connection (Yes, No) [No]:
Engine database name [engine]:
Engine database user [engine]:
Engine database password:
```

If you enter Remote to connect to an existing PostgreSQL server running on a remote host, you're prompted for the details for connecting to the database:

```
Engine database host [localhost]:
Engine database port [5432]:
Engine database secured connection (Yes, No) [No]:
Engine database name [engine]:
Engine database user [engine]:
Engine database password:
```

Admin User Password

```
Engine admin password:
Confirm engine admin password:
```

Enter a password for the default administrative user (`admin@ovirt`). Make a note of the password. If you provide a weak password, you might get the following warning:

```
[WARNING] Password is weak: The password fails the dictionary check - it is
based on a dictionary word
Use weak password? (Yes, No) [No]: Yes
```

Application Mode

```
Application mode (Both, Virt, Gluster) [Both]:
```

The Manager can be configured to manage virtual machines (**Virt**) or manage Gluster clusters (**Gluster**), or **Both**.

OVN Provider Credentials

```
Use default credentials (admin@ovirt) for ovirt-provider-ovn (Yes, No) [Yes]:
oVirt OVN provider user[admin@ovirt]:
oVirt OVN provider password:
```

If you installed the OVN provider, configure the credentials for connecting to the OVN (Open vSwitch) databases.

Using external providers, including the OVN provider, is a technology preview feature.

SAN Wipe After Delete

```
Default SAN wipe after delete (Yes, No) [No]:
```

Enter Yes to set the default value for the `wipe_after_delete` flag to `true`, which wipes the blocks of a virtual disk when it's deleted.

Using the wipe after delete functionality is a technology preview feature.

Web Server Configuration

```
Organization name for certificate [<autodetected-domain-based-name>]:
```

Provide the organization name to use for the automatically generated self-signed SSL certificate used by the Manager web server.

```
Setup can configure the default page of the web server to
present the application home page. This may conflict with existing
applications.
Do you wish to set the application as the default web page of the server?
(Yes, No) [Yes]:
```

Enter Yes to make the Oracle Linux Virtualization Manager landing page the default page presented by the web server.

```
Setup can configure apache to use SSL using a certificate issued
from the internal CA. Do you wish Setup to configure that, or prefer to
perform that manually? (Automatic, Manual) [Automatic]:
```

Enter Automatic to generate a self-signed SSL certificate for the web server. Only use self-signed certificates for testing purposes.

Enter Manual to provide the location of the SSL certificate and private key to use the web server.

> **Note**
>
> For more information, see the following My Oracle Support articles:
>
> - *How to renew OLVM Hosts Certificate in OLVM Environment/Infrastructure (Doc ID 2885203.1)*
> - *VM Migration fails with Error " The server certificate /etc/pki/vdsm/libvirt-vnc/server-cert.pem has expired" (Doc ID 2959537.1)*
> - *Moving From Custom 3rd Party CA Certification to Default certification (Doc ID 2963343.1)*

Data Warehouse Sampling Scale

```
Please choose Data Warehouse sampling scale:
(1) Basic
(2) Full
(1, 2)[1]:
```

Set the Data Warehouse sampling scale to either Basic or Full. If this step is skipped the Data Warehouse isn't configured to run on the Manager host.

Enter 1 for Basic, which reduces the values of `DWH_TABLES_KEEP_HOURLY` to `720` and `DWH_TABLES_KEEP_DAILY` to `0`. Enter 2 for Full.

If the Manager and the Data Warehouse run on the same host, Basic is the recommended sample scale because this reduces the load on the Manager host. Full is recommended only if the Data Warehouse runs on a remote host.

The Full sampling scale is a technology preview feature.

Log in to the Administration Portal

After you run the `engine-setup` command to configure Oracle Linux Virtualization Manager, sign in to the Administration Portal to verify that the configuration was successful.

Prepare to Log In

We recommended that you use the latest version one of the following browsers to access the Administration Portal

- Mozilla Firefox
- Google Chrome
- Microsoft Edge

If Oracle Linux Virtualization Manager was configured to use a self-signed SSL certificate, or an SSL certificate that's signed by a Certificate Authority (CA) that isn't trusted by the browser (for example an Intermediate CA), install the CA certificate in the browser. Consult the browser's instructions for how to import a CA certificate.

You can download the CA certificate by selecting *Engine CA Certificate* on the Welcome dashboard or by navigating directly to `http://`***`manager-fqdn`***`/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA`.

Usually you access the Administration Portal using the fully qualified domain name of the Manager host that you provided during installation. However, you can access the Administration Portal using an alternative host name(s). To do this, add a configuration file to the Manager as follows:

1. Sign in to the Manager host as root.
2. Create the file `/etc/ovirt-engine/engine.conf.d/99-custom-sso-setup.conf` with the following content:

   ```
   SSO_ALTERNATE_ENGINE_FQDNS="alias1.example.com alias2.example.com"
   ```

   The list of alternative host names must be separated by spaces.
3. Restart Oracle Linux Virtualization Manager.

   ```
   systemctl restart ovirt-engine
   ```

Log In and Log Out

Sign in to the Administration Portal using a web browser and the default `admin@ovirt` user.

1. Go to `https://`***`manager-fqdn`***`/ovirt-engine`. The **Welcome** page displays.
2. **(Optional)** Change the preferred language from the dropdown list on the **Welcome** page.

   You can view the **Administration Portal** in different languages. The default language is based on the locale of the web browser.
3. Select **Administration Portal**. The **Login** page displays.

4. Enter `admin@ovirt` for the **Username** and the password you specified when you configured the Manager.

> **Note**
>
> If you use the REST API, the admin user is identified as `admin@ovirt@internalsso` when Keycloak SSO is enabled.
>
> If you disabled Keycloak integration and use the legacy internal domain (AAA), the default administrative user is `admin@internal`.

5. If Keycloak SSO is disabled, from the **Profile** list select `internal`.

> **Note**
>
> This step doesn't apply when Keycloak SSO is enabled, because there is no profile list.

6. Select **Log In**.

> **Note**
>
> From the Welcome dashboard, you also have the option of signing in to two other portals:
>
> - The VM Portal
> - The Monitoring Portal
>
> For more information, see Access Portals in the Oracle Linux Virtualization Manager: Architecture and Planning Guide

To log out of the **Administration Portal**, select the person icon in the header bar and then select **Sign Out**. You're returned to the **Login** page.

> **Caution**
>
> Before you add Oracle Linux KVM hosts that run a later major version than the Manager OS, on the Manager host reinstall the release RPM that matches the Manager OS to refresh the repository definitions.
>
> On the Manager host, run:
>
> ```
> dnf reinstall oracle-ovirt-release-45-eln
> dnf clean all
> dnf repolist
> ```
>
> Proceed to add hosts only after confirming that the required repositories for Oracle Linux *n* are present.

Configure a KVM Host

To manage an Oracle Linux KVM host using Oracle Linux Virtualization Manager, prepare the KVM host by performing a fresh installation of Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) and enabling the required repositories, and then you add the host to a data center using the Administration Portal.

Before you begin, ensure you have satisfied the *KVM Host Requirements* as detailed in the Oracle Linux Virtualization Manager: Architecture and Planning Guide.

See the Oracle® Linux: KVM User's Guide for information on the supported guest OS.

Prepare a KVM Host

Before you can add an Oracle Linux KVM host, prepare it by performing a fresh installation of Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) and enabling the required repositories. You can download the installation ISO for Oracle Linux from the Oracle Software Delivery Cloud at https://edelivery.oracle.com.

> **Note**
>
> Oracle Linux KVM hosts are supported on the following kernel versions:
>
> - Oracle Linux 8: Unbreakable Enterprise Kernel Release 6, Unbreakable Enterprise Kernel Release 7, or Red Hat Compatible Kernel.
> - Oracle Linux 9: Unbreakable Enterprise Kernel Release 8.

> **Note**
>
> Oracle Linux 9 KVM hosts require a cluster compatibility version of 4.7 and an engine host running `ovirt-engine-4.5.5-1.65` or later release.
>
> For more information on cluster compatibility versions, see Changing Cluster Compatibility Versions in the *Oracle Linux Virtualization Manager Administrator's Guide*.

Configure the KVM host

Complete the following steps to for each KVM host in the environment.

1. Install Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on the host.

   > **Caution**
   >
   > KVM hosts running Oracle Linux 9 can't be added to a cluster that has GlusterFS or hyperconvergance configured.

   - Follow the installation instructions for the Oracle Linux release:

- Oracle® Linux 8: Installing Oracle Linux.
- Oracle® Linux 9: Installing Oracle Linux.

- Select **Minimal Install** as the base environment for the installation.

> **⚠ Caution**
>
> Do **NOT** select any other base environment than **Minimal Install** for the installation or the hosts will have incorrect qemu and libvirt versions, incorrect repositories configured, and no access to virtual machine consoles.

- Don't install any extra packages until after you have added the host to the Manager, because they might cause dependency issues.

2. **(Optional)** If you use a proxy server for Internet access, configure Yum with the proxy server settings. For more information, see the Oracle® Linux: Managing Software on Oracle Linux.
3. Complete one of the following sets of steps:
   - **For ULN registered hosts or using Oracle Linux Manager**

     Subscribe the system to the required channels and enable appstream modules.

     a. For ULN registered hosts, sign in to https://linux.oracle.com with a ULN username and password. For Oracle Linux Manager registered hosts, access the internal server URL.
     b. On the Systems tab, select the link named for the host in the list of registered machines.
     c. On the System Details page, select **Manage Subscriptions**.
     d. On the System Summary page, select each required channel from the list of available channels and select the right arrow to move the channel to the list of subscribed channels. Subscribe the system to the following channels, where *n* is the major Oracle Linux version (8 or 9):
        - `ol`*n*`_x86_64_baseos_latest`
        - `ol`*n*`_x86_64_appstream`
        - `ol8_x86_64_kvm_appstream` or `ol9_x86_64_kvm_utils`
        - `ol`*n*`_x86_64_ovirt45`
        - `ol`*n*`_x86_64_ovirt45_extras`
        - `ol`*n*`_x86_64_addons`
        - **(For Oracle Linux 8 only)** `ol8_x86_64_gluster_appstream`
        - **(For VDSM)**`ol`*n*`_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

     > **ⓘ Note**
     >
     > Gluster is only available on hosts running Oracle Linux 8.

     e. Select **Save Subscriptions**.

f. Install the Oracle Linux Virtualization Manager Release 4.5 package, which automatically enables/disables the required repositories.

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

- **For Oracle Linux yum server configured KVM hosts**

  Install the Oracle Linux Virtualization Manager Release 4.5 package and enable the required repositories. In the following instructions, *n* is the major Oracle Linux version (8 or 9):

  > **Note**
  >
  > Installing the Oracle Linux Virtualization Manager Release 4.5 package configures an Oracle Linux KVM host; it doesn't install the Manager.

  a. Enable the `oln_baseos_latest` repository.

  ```
  dnf config-manager --enable oln_baseos_latest
  ```

  b. Install the Oracle Linux Virtualization Manager Release 4.5 package, which automatically enables/disables the required repositories.

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

  c. Use the `dnf` command to verify that the required repositories are enabled.

  i. Clear the dnf cache.

  ```
  dnf clean all
  ```

  ii. List the configured repositories and verify that the required repositories are enabled.

  ```
  dnf repolist
  ```

  The following repositories must be enabled:

  - `oln_baseos_latest`
  - `oln_appstream`
  - `ol8_kvm_appstream` or `ol9_kvm_utils`
  - `ovirt-4.5`
  - `ovirt-4.5-extra`
  - **(For Oracle Linux 8 only)** `ol8_gluster_appstream`
  - **(For VDSM)**`ol8_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

> **Note**
>
> Gluster is only available on hosts running Oracle Linux 8.

iii. If a required repository isn't enabled, use the `dnf config-manager` command to enable it:

```
dnf config-manager --enable repository
```

4. If the host runs the Unbreakable Linux Kernel (UEK):

   a. Install the *Extra kernel modules* package.

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

   b. Reboot the host.

Check host configuration

To ensure that the KVM host is configured correctly, run the precheck script **BEFORE** you add it to the Manager.

> **Note**
>
> To run the script on several KVM hosts simultaneously, we recommend using an Ansible playbook.

1. Connect to the KVM host from a command line and run the precheck script:

   ```
   sudo olvm-pre-check.py
   ```

   A series of checks begins and you see something similar to

   ```
   ----------------------------------
   OLVM 4.5.5 PRE-CHECK SCRIPT
   ----------------------------------

   +++ Checking oracle-ovirt-release-45         [PASS]
   +++ Checking if Host is installed[WARN]
       The 'ovirt-host' package is already installed.
       This is a KVM Host, DO NOT install the engine on this host.

   +++ Checking if a Minimal Installation       [PASS]
   +++ Validating the 'Minimal Install' Group   [PASS]
   +++ Checking enabled repositories[WARN]

       Extra repositories are enabled:
       update-pcp

       Please run the command:
       dnf config-manager --set-disabled update-pcp
   ```

```
+++ Running 'dnf makecache'                  [PASS]
+++ Dry run 'dnf update --assumeno'          [PASS]
+++ Checking Linux Kernel                    [PASS]
+++ Checking kernel-uek-modules-extra        [WARN]

    Package kernel-uek-modules-extra is not installed.
    Please run: 'dnf install -y kernel-uek-modules-extra'
+++ Checking Firewalld status                [PASS]
+++ Checking SELinux status                  [PASS]
+++ Checking FIPS status                     [PASS]
FIPS is disabled.
+++ If installed, check ansible version      [PASS]
+++ If installed, check qemu-kvm version     [PASS]
+++ If installed, check libvirt version      [PASS]
+++ Checking Hostname/FQDN                   [PASS]
```

2. If any checks are marked **WARN** or **FAIL**, the script output provides information that can help you resolve the issues:

```
+++ Checking if Host is installed    [WARN]
    The 'ovirt-host' package is already installed.
    This is a KVM Host, DO NOT install the engine on this host.

+++ Checking enabled repositories             [WARN]

    Extra repositories are enabled:
    update-pcp

    Please run the command:
    dnf config-manager --set-disabled update-pcp

+++ Checking kernel-uek-modules-ext           [WARN]

    Package kernel-uek-modules-extra is not installed.
    Please run: 'dnf install -y kernel-uek-modules-extra'
```

3. If you had warnings or failures to address, rerun the script to ensure that the system passes all configuration checks. For example:

```
sudo olvm-pre-check.py

-----------------------------------
OLVM 4.5.5 PRE-CHECK SCRIPT
-----------------------------------

+++ Checking oracle-ovirt-release-45         [PASS]
+++ Checking if Host is installed[PASS]
+++ Checking if a Minimal Installation       [PASS]
+++ Validating the 'Minimal Install' Group   [PASS]
+++ Checking enabled repositories            [PASS]
+++ Running 'dnf makecache'                  [PASS]
+++ Dry run 'dnf update --assumeno'          [PASS]
+++ Checking Linux Kernel                    [PASS]
+++ Checking kernel-uek-modules-extra        [PASS]
+++ Checking Firewalld status                [PASS]
+++ Checking SELinux status                  [PASS]
```

```
+++ Checking FIPS status                    [PASS]
FIPS is disabled.
+++ If installed, check ansible version     [PASS]
+++ If installed, check qemu-kvm version    [PASS]
+++ If installed, check libvirt version     [PASS]
+++ Checking Hostname/FQDN                  [PASS]
```

The Oracle Linux KVM host is now ready to be added to the Manager using the Administration Portal.

Add a KVM Host

After you have configured an Oracle Linux KVM host, you use the Administration Portal to add the host to a data center so that it can be used to run virtual machines. Use the following steps to add KVM hosts installed with other supported guest OS.

> **⚠ Caution**
>
> Oracle Linux Virtualization Manager lets you overallocate a KVM host's memory and CPU resources. As the KVM host itself also needs memory and CPU to run, we recommend that you reserve some memory and CPU for the KVM host. To do this, go to **Administration** and set a memory quota and a vCPU quota.

To add an Oracle Linux KVM host:

1. Sign in to the Administration Portal.

   See Log in to the Administration Portal for details.
2. Go to **Compute** and then select **Hosts**.
3. On the **Hosts** pane, select **New**.

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

   The **Default** data center is automatically selected.

   When you install Oracle Linux Virtualization Manager, a data center and cluster named Default is created. You can rename and configure this data center and cluster, or you can add new data centers and clusters, as required. See the Data Centers or Clusters tasks in the Oracle Linux Virtualization Manager: Administration Guide.
5. In the **Name** field, enter a name for the host.
6. In the **Hostname** field, enter the fully-qualified domain name or IP address of the host.
7. In the **SSH Port** field, change the standard SSH port 22 if the SSH server on the host uses a different port.
8. Under **Authentication**, select the authentication method to use.

   We recommend that you select **SSH PublicKey** authentication. If you select this option, copy the key displayed in the **SSH PublicKey** field to the `/root/.ssh/authorized_keys` file on the host.

   Otherwise, enter the root user's password to use password authentication.
9. **(Optional)** Configure other settings for the host from the other tabs on the **New Host** sidebar.

> **Note**
>
> If you don't want to set any other configuration options now, you can always make changes later by selecting a host from the **Hosts** pane and then **Edit**.

10. Select **OK**.

    The **Power Management Configuration** screen is displayed.

11. Do one of the following:

    - If you don't want to configure power management, select **OK**.
    - Select **Configure Power Management** and then select OK. See Configure Power Management and Fencing for Host for more information.

    The host is added to the list of hosts in the Manager. While the Manager is installing the host agent (VDSM) and other required packages on the host, the status of the host is shown as **Installing**. You can view the progress of the installation in the Hosts details pane. When the installation is complete, the host status changes to **Up**.

12. **(Optional)** Complete the previous steps to add more KVM hosts to the Manager.

> **Note**
>
> After a KVM host is added to a cluster, it's also crucial to avoid any spontaneous changes to the network configuration in `/etc/sysconfig/network-scripts/` or through the NetworkManager (e.g. `nmcli`). Make all changes to the network configuration using the engine host/manager Administration Portal or REST API.

Now that the engine and host(s) are configured, see the Oracle Linux Virtualization Manager: Administration Guide for detailed configuration and administrative tasks.

4

FIPS Mode Deployment

To use Oracle Linux Virtualization Manager in Federal Information Processing Standard (FIPS) mode, you must install the OS with FIPS mode enabled before you install Oracle Linux Virtualization Manager.

You can create a FIPS-enabled bare metal machine by either installing the OS in FIPS mode or by switching the system into FIPS mode after installing the OS. For instructions, see either:

- *Configuring FIPS Mode in Oracle Linux 8* in Enhancing System Security
- *Configuring FIPS Mode in Oracle Linux 9* in Enhancing System Security

> **Important**
>
> Enabling FIPS mode while installing the OS ensures all the generated keys use the FIPS-approved algorithms and undertake continuous monitoring tests.

Deploy on a FIPS Enabled System

Whether you're using a standalone or self-hosted engine deployment, ensure you already have FIPS enabled on the system(s) you want use for the deployment. To check, we recommend you run the following command on the system(s):

```
fips-mode-setup --check


FIPS mode is enabled.
```

> **Important**
>
> Although it's possible to enable FIPS mode on any installed Oracle Linux server, Oracle doesn't support enabling it on an already deployed Engine or KVM host.

Standalone Engine and KVM hosts

After you have enabled the system for FIPS, follow the Installation and Configuration instructions.

Self-Hosted-Engine

After you have enabled the system for FIPS, follow the Deploy the Self-Hosted Engine instructions.

At the enable FIPS prompt, answer Yes.

```
Do you want to enable FIPS? (Yes/No) [No]: Yes
```

Encrypt VNC Console Connections

When you deploy Oracle Linux Virtualization Manager using FIPS enabled systems, you must ensure VNC console connections are encrypted.

Enable VNC Encryption at Cluster Level

When you have deployed Oracle Linux Virtualization Manager on FIPS enabled systems, you must enable VNC encryption to access virtual machine consoles. Do this at the cluster level:

1. From the Administration Portal, go to Compute > Clusters.
2. Edit the Cluster where you want to enable VNC Encryption.
3. Select on the Console tab on the left.
4. Check the Enable VNC Encryption checkbox and then select OK.

Reinstall KVM Host

After enabling VNC encryption, you're prompted to reinstall the KVM host, which applies all the required options to enable the VNC encrypted console connection.

> **Note**
>
> On all hosts assigned to the self-hosted engine, from the Reinstall dialog go to the Hosted-Engine tab and select Deploy.

1. From the Administration Portal, go to Compute > Hosts.
2. Select a host to configure, select Management, and then select Maintenance.
3. Select the Installation button.
4. Select Reinstall and clear the "Reboot host after installation" checkbox.
5. Select OK.

Run VNC SASL Ansible Playbook on KVM Hosts

To apply the playbook, the KVM host must be in Maintenance mode. You can run the playbook on more than one host at a time. Add all KVM hosts to be configured to the `/etc/hosts` file, one host per line. Ensure that those hosts are in Maintenance mode before applying the playbook.

1. From the Administration Portal, go to Compute > Hosts.
2. Select the host you want to configure, select Management, and then select Maintenance.
3. SSH into the Engine server.

4. Ansible tries to find the best Python interpreter to use. Set it to `/usr/bin/python3` to avoid errors, then run the `ovirt-vnc-sasl.yml` playbook:

```
cd /usr/share/ovirt-engine/ansible-runner-service-project/project/


sed -ri.orig '/defaults/ainterpreter_python = /usr/bin/python3' ansible.cfg


echo "IP-OR-HOSTNAME" > hosts


ansible-playbook --ask-pass --inventory=hosts ovirt-vnc-sasl.yml
```

Full output example:

```
cd /usr/share/ovirt-engine/ansible-runner-service-project/project/
echo "192.168.0.102" > hosts
sed -ri.orig '/defaults/ainterpreter_python = /usr/bin/python3' ansible.cfg
ansible-playbook --ask-pass --inventory=hosts ovirt-vnc-sasl.yml


SSH password:

PLAY [all]
*********************************************************************

TASK [Gathering Facts]
***********************************************************
ok: [192.168.0.102]

TASK [ovirt-host-setup-vnc-sasl : Create SASL QEMU config file]
****************
ok: [192.168.0.102]

TASK [ovirt-host-setup-vnc-sasl : Use saslpasswd2 to create file with
dummy user] ***
ok: [192.168.0.102]

TASK [ovirt-host-setup-vnc-sasl : Set ownership of the password db]
************
ok: [192.168.0.102]

TASK [ovirt-host-setup-vnc-sasl : Modify qemu config file - enable VNC
SASL authentication] ***
ok: [192.168.0.102]

PLAY RECAP
*********************************************************************

192.168.0.102 : ok=5 changed=5  unreachable=0  failed=0  skipped=0
rescued=0  ignored=0
```

Note

You might receive the following error message when running the `ovirt-vnc-sasl.yml` playbook:

```
ERROR! The requested handler 'populate service facts and restart
libvirtd' /
was not found in either the main handlers list nor in the listening
handlers list
```

This error is because of the automatically detected Python version being used. To fix it, add the `interpreter_python` option to the local `ansible.cfg` file and then rerun the playbook to complete all tasks.

```
cd /usr/share/ovirt-engine/ansible-runner-service-project/project/


sed -ri.orig '/defaults/ainterpreter_python = /usr/bin/python3'
ansible.cfg
```

5
Self-Hosted Engine Deployment

In Oracle Linux Virtualization Manager, a self-hosted engine is a virtualized environment where the engine runs inside a virtual machine on the hosts in the environment. The virtual machine for the engine is created as part of the host configuration process. And, the engine is installed and configured in parallel to the host configuration.

Because the engine runs as a virtual machine and not on physical hardware, a self-hosted engine requires less physical resources. Also, because the engine is configured to be highly available, if the host running the Engine virtual machine goes into maintenance mode or fails, the virtual machine is migrated automatically to another host in the environment. A minimum of two KVM hosts are required.

To review conceptual information, troubleshooting, and administration tasks, see the oVirt Self-Hosted Engine Guide in oVirt Documentation.

To deploy a self-hosted engine, you perform a fresh installation of Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on the host, install the Oracle Linux Virtualization Manager Release 4.5 package, and then run the hosted engine deployment tool to complete configuration.

> **Note**
>
> The self-hosted engine virtual machine is based on Oracle Linux 8, but you can add Oracle Linux 9 KVM hosts to the self-hosted engine cluster.

> **Caution**
>
> If you're deploying a self-hosted engine as a hyperconverged infrastructure with GlusterFS storage, you must deploy GlusterFS *before* you deploy the self-hosted engine. See Hyperconverged Infrastructure Deployment Using GlusterFS Storage.
>
> Gluster is only available on hosts running Oracle Linux 8.

You can also deploy a self-hosted engine using the command line or Cockpit portal. To use the command line, proceed to Use Command Line to Deploy Self-Hosted Engine. To use the Cockpit portal, proceed to Use Cockpit to Deploy Self-Hosted Engine.

> **Caution**
>
> If you're behind a proxy, you must use the command line option to deploy.

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 guide.

Self-Hosted Engine Prerequisites

In addition to the Requirements and Scalability Limits, you must satisfy the following prerequisites before deploying a self-hosted engine.

- A minimum of two (2) KVM hosts and no more than seven (7).
- A fully-qualified domain name for the engine and host with forward and reverse lookup records set in the DNS.
- A directory of at least 5 GB on the host for the oVirt Engine Appliance. During the deployment process the `/var/tmp` directory is checked to see if it has enough space to extract the appliance files. If the `/var/tmp` directory doesn't have enough space, you can specify a different directory or mount external storage.

  > **Note**
  >
  > The VDSM user and KVM group must have read, write, and execute permissions on the directory.

- Prepared storage of at least 74 GB to be used as a data storage domain dedicated to the engine virtual machine. The data storage domain is created during the self-hosted engine deployment.

  If you're using iSCSI storage, don't use the same iSCSI target for the self-hosted engine storage domain and any other storage domains.

  > **Caution**
  >
  > When you have a data center with only one active data storage domain and that domain gets corrupted, you're unable to add new data storage domains or remove the corrupted data storage domain. If you have deployed the self-hosted engine in such a data center and its data storage domain gets corrupted, you must redeploy the self-hosted engine.

- Ensure that the host you're using to deploy a self-hosted engine can access yum.oracle.com.

Deploy the Self-Hosted Engine

You must perform a fresh installation of Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on an Oracle Linux Virtualization Manager host before deploying a self-hosted engine. You can download the installation ISO from the Oracle Software Delivery Cloud at https://edelivery.oracle.com.

Configure the host

Complete the following steps to prepare the host for deployment.

1. Install Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on the host using the **Minimal Install** base environment.

> **⚠ Caution**
>
> Do **NOT** select any other base environment than **Minimal Install** for the installation or the hosts will have incorrect qemu and libvirt versions, incorrect repositories configured, and no access to virtual machine consoles.
>
> Don't install any extra packages until after you have installed the Manager packages, because they might cause dependency issues.

Follow the instructions in the appropriate guide:

- Oracle® Linux 8: Installing Oracle Linux
- Oracle® Linux 9: Installing Oracle Linux.

2. Ensure that the firewalld service is enabled and started.

   For more information about configuring `firewalld`, see *Configuring a Packet Filtering Firewall* in the appropriate guide:

   - Oracle® Linux 8: Configuring the Firewall
   - Oracle® Linux 9: Configuring the Firewall

3. Complete one of the following sets of steps:

   - **For ULN registered hosts or using Oracle Linux Manager**

     Subscribe the system to the required channels.

     a. For ULN registered hosts, sign in to https://linux.oracle.com with a ULN username and password. For Oracle Linux Manager registered hosts, access the internal server URL.

     b. On the Systems tab, select the link named for the host in the list of registered machines.

     c. On the System Details page, select **Manage Subscriptions**.

     d. On the System Summary page, select each required channel from the list of available channels and select the right arrow to move the channel to the list of subscribed channels. Subscribe the system to the following channels, where *n* is the major Oracle Linux version (8 or 9):

        - `ol`*n*`_x86_64_baseos_latest`
        - `ol`*n*`_x86_64_appstream`
        - `ol8_x86_64_kvm_appstream` or `ol9_x86_64_kvm_utils`
        - `ol`*n*`_x86_64_ovirt45`
        - `ol`*n*`_x86_64_ovirt45_extras`
        - **(Oracle Linux 8 only)** `ol8_x86_64_gluster_appstream`
        - **(For VDSM)**`ol8_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

        > **ⓘ Note**
        >
        > Gluster is only available on hosts running Oracle Linux 8.

     e. Select **Save Subscriptions**.

f. Install the Oracle Linux Virtualization Manager Release 4.5 package, which automatically enables/disables the required repositories.

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

- **For Oracle Linux yum server hosts**

  Install the Oracle Linux Virtualization Manager Release 4.5 package and enable the required repositories. In the following instructions, *n* is the major Oracle Linux version (8 or 9):

  a. Enable the `oln_baseos_latest` yum repository.

  ```
  dnf config-manager --enable oln_baseos_latest
  ```

  b. Install the Oracle Linux Virtualization Manager Release 4.5 package, which automatically enables/disables the required repositories.

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

  c. Use the `dnf` command to verify that the required repositories are enabled.

  i. Clear the yum cache.

  ```
  dnf clean all
  ```

  ii. List the configured repositories and verify that the required repositories are enabled.

  ```
  dnf repolist
  ```

  The following repositories must be enabled:

  - `oln_x86_64_baseos_latest`
  - `oln_x86_64_appstream`
  - `ol8_x86_64_kvm_appstream` or `ol9_x86_64_kvm_utils`
  - `oln_x86_64_ovirt45`
  - `oln_x86_64_ovirt45_extras`
  - `oln_x86_64_addons`
  - **(For Oracle Linux 8 only)** `ol8_x86_64_gluster_appstream`
  - **(For VDSM)**`ol8_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

  > **Note**
  >
  > Gluster is only available on hosts running Oracle Linux 8.

  iii. If a required repository isn't enabled, use the `dnf config-manager` command to enable it.

  ```
  dnf config-manager --enable repository
  ```

4. If the host runs the Unbreakable Linux Kernel (UEK):
   a. Install the *Extra kernel modules* package.

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

   b. Reboot the host.

Check host configuration

To ensure that the hosted engine host is configured correctly, run the precheck script **BEFORE** you deploy the hosted engine. You must also run the precheck script on all KVM hosts in the environment.

> **Note**
>
> To run the script on several hosts simultaneously, we recommend using an Ansible playbook.

1. Connect to the hosted engine host from a command line and run the precheck script:

   ```
   sudo olvm-pre-check.py
   ```

   A series of checks begins and you see something similar to

   ```
   ----------------------------------
              OLVM 4.5.5 PRE-CHECK SCRIPT
              ----------------------------------

              +++ Checking oracle-ovirt-release-45                    [PASS]
              +++ Checking if Host is installed                       [WARN]

              The 'ovirt-engine' package is already installed.
              DO NOT configure this Server as a KVM Host.

              +++ Checking if a Minimal Installation                  [PASS]
              +++ Validating the 'Minimal Install' Group              [PASS]
              +++ Checking enabled repositories                       [WARN]

              Extra repositories are enabled:
              update-pcp

              Please run the command:
              dnf config-manager --set-disabled update-pcp

              +++ Running 'dnf makecache'                             [PASS]
              +++ Dry run 'dnf update --assumeno'                     [PASS]
              +++ Checking Linux Kernel                               [PASS]
              +++ Checking kernel-uek-modules-extra                   [PASS]
              +++ Checking Firewalld status                           [PASS]
              +++ Checking SELinux status                             [PASS]
              +++ Checking FIPS status                                [PASS]
              FIPS is disabled.
              +++ If installed, check ansible version                 [PASS]
   ```

```
+++ If installed, check qemu-kvm version                [PASS]
+++ If installed, check libvirt version                 [PASS]
+++ Checking Hostname/FQDN                              [PASS]
```

2. If any checks are marked **WARN** or **FAIL**, the script output provides information that can help you resolve the issues:

```
+++ Checking if Host is installed                  [WARN]

          The 'ovirt-engine' package is already installed.
            DO NOT configure this Server as a KVM Host.

          +++ Checking enabled repositories                [WARN]

          Extra repositories are enabled:
            update-pcp

            Please run the command:
            dnf config-manager --set-disabled update-pcp
```

3. If you had warnings or failures to address, rerun the script to ensure that the system passes all configuration checks. For example:

```
sudo olvm-pre-check.py

          ----------------------------------
          OLVM 4.5.5 PRE-CHECK SCRIPT
          ----------------------------------

          +++ Checking oracle-ovirt-release-45                    [PASS]
          +++ Checking if Host is installed                       [PASS]
          +++ Checking if a Minimal Installation                  [PASS]
          +++ Validating the 'Minimal Install' Group              [PASS]
          +++ Checking enabled repositories                       [PASS]
          +++ Running 'dnf makecache'                             [PASS]
          +++ Dry run 'dnf update --assumeno'                     [PASS]
          +++ Checking Linux Kernel                               [PASS]
          +++ Checking kernel-uek-modules-extra                   [PASS]
          +++ Checking Firewalld status                           [PASS]
          +++ Checking SELinux status                             [PASS]
          +++ Checking FIPS status                                [PASS]
          FIPS is disabled.
          +++ If installed, check ansible version                 [PASS]
          +++ If installed, check qemu-kvm version                [PASS]
          +++ If installed, check libvirt version                 [PASS]
          +++ Checking Hostname/FQDN                              [PASS]
```

Install the engine

After you have successfully configured and verified the hosted engine host, install the hosted engine deployment tool and engine appliance:

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

Proceed to Use Command Line to Deploy Self-Hosted Engine or Use Cockpit to Deploy Self-Hosted Engine.

Use Command Line to Deploy Self-Hosted Engine

You can deploy the self-hosted engine from the command line. A script collects the details of the environment and uses them to configure the host and the engine.

1. Start the deployment. IPv6 is used by default. To use IPv4, specify the `--4` option:

   ```
   hosted-engine --deploy --4
   ```

   Optionally, use the `--ansible-extra-vars` option to define variables for the deployment. For example:

   ```
   hosted-engine --deploy --4 --ansible-extra-vars="@/root/extra-vars.yml"

   cat /root/extra-vars.yml
   ---
   he_pause_host: true
   he_proxy: "http://<host>:<port>"
   he_enable_keycloak: true
   ```

   See the oVirt Documentation for more information.

2. Enter Yes to begin deployment.

   ```
   Continuing will configure this host for serving as hypervisor and will
   create a local VM
   with a running engine. The locally running engine will be used to
   configure a new storage
   domain and create a VM there. At the end the disk of the local VM will be
   moved to the
   shared storage.
   Are you sure you want to continue? (Yes, No)[Yes]:
   ```

   > **Note**
   >
   > The hosted-engine script creates a virtual machine and uses cloud-init to configure it. The script also runs engine-setup and reboots the system so that the virtual machine can be managed by the high availability agent.

3. Enter the name of the data center or accept the default.

   ```
   Please enter the name of the data center where you want to deploy this
   hosted-engine
   host. Data center [Default]:
   ```

4. Enter a name for the cluster or accept the default.

```
Please enter the name of the cluster where you want to deploy this hosted-
engine host.
Cluster [Default]:
```

5. Keycloak integration provides internal Single Sign-On (SSO) for the Engine and deprecates legacy AAA authentication. Accept the default response of Yes.

```
Configure Keycloak integration on the engine (Yes, No) [Yes]:
```

6. Configure the network.
   a. If the gateway that displays is correct, press Enter to configure the network.
   b. Enter a pingable address on the same subnet so the script can check the host's connectivity.

```
Please indicate a pingable gateway IP address [X.X.X.X]:
```

   c. The script detects possible NICs to use as a management bridge for the environment. Select the default.

```
Please indicate a nic to set ovirtmgmt bridge on: (eth1, eth0) [eth1]:
```

7. Enter the path to an OVA archive to use a custom appliance for the virtual machine installation. Otherwise, leave this field empty to use the oVirt Engine Appliance.

```
If you want to deploy with a custom engine appliance image, please specify
the path to
the OVA archive you would like to use.
Entering no value will use the image from the ovirt-engine-appliance rpm,
installing it if needed.
Appliance image path []:
```

8. Specify the fully-qualified domain name for the engine virtual machine.

```
Please provide the FQDN you would like to use for the engine appliance.
 Note: This will be the FQDN of the engine VM you are now going to launch,
 it should not point to the base host or to any other existing machine.
 Engine VM FQDN:  manager.example.com
 Please provide the domain name you would like to use for the engine
appliance.
 Engine VM domain: [example.com]
```

9. Enter and confirm a root password for the engine.

```
Enter root password that will be used for the engine appliance:
Confirm appliance root password:
```

10. Optionally, enter an SSH public key to enable you to sign in to the engine as the root user and specify whether to enable SSH access for the root user.

```
Enter ssh public key for the root user that will be used for the engine
appliance (leave it empty to skip):
Do you want to enable ssh access for the root user (yes, no, without-
```

```
password)
[yes]:
You may provide an SSH public key, that will be added by the deployment
script to the
authorized_keys file of the root user in the engine appliance.
This should allow you passwordless login to the engine machine after
deployment.
If you provide no key, authorized_keys will not be touched.
SSH public key []:
[WARNING] Skipping appliance root ssh public key
Do you want to enable ssh access for the root user? (yes, no, without-
password) [yes]:
```

11. Enter the virtual machine's CPU and memory configuration.

```
Please specify the number of virtual CPUs for the VM (Defaults to
appliance
OVF value): [4]:
Please specify the memory size of the VM in MB. The default is the
appliance
OVF value [16384]:
```

12. Enter a MAC address for the engine virtual machine or accept a randomly-generated MAC address.

```
You may specify a unicast MAC address for the VM or accept a randomly
generated default [00:16:3e:3d:34:47]:
```

> **Note**
>
> To provide the engine virtual machine with an IP address using DHCP, ensure that you have a valid DHCP reservation for this MAC address. The deployment script doesn't configure the DHCP server for you.

13. Enter the virtual machine's networking details.

```
How should the engine VM network be configured (DHCP, Static)[DHCP]?
```

> **Note**
>
> If you specified Static, enter the IP address of the Engine. The static IP address must belong to the same subnet as the host. For example, if the host is in 10.1.1.0/24, the Engine virtual machine's IP must be in the same subnet range (10.1.1.1-254/24).
>
> ```
> Please enter the IP address to be used for the engine VM [x.x.x.x]:
> Please provide a comma-separated list (max 3) of IP addresses of
> domain
> name servers for the engine VM
> Engine VM DNS (leave it empty to skip):
> ```

14. Specify whether to add entries in the virtual machine's `/etc/hosts` file for the engine virtual machine and the base host. Ensure that the host names are resolvable.

    ```
    Add lines for the appliance itself and for this host to /etc/hosts on the
    engine VM?
    Note: ensuring that this host could resolve the engine VM hostname is
    still up to you.
    Add lines to /etc/hosts? (Yes, No)[Yes]:
    ```

15. Provide the name and TCP port number of the SMTP server, the email address used to send email notifications, and a comma-separated list of email addresses to receive these notifications. Or, press Enter to accept the defaults.

    ```
    Please provide the name of the SMTP server through which we will send
    notifications [localhost]:
    Please provide the TCP port number of the SMTP server [25]:
    Please provide the email address from which notifications will be sent
    [root@localhost]:
    Please provide a comma-separated list of email addresses which will get
    notifications [root@localhost]:
    ```

16. Enter and confirm a password for the `admin@ovirt` user to access the Administration Portal.

    ```
    Enter engine admin password:
    Confirm engine admin password:
    ```

    The script creates the virtual machine which can take time if it needs to install the oVirt Engine Appliance. After creating the virtual machine, the script continues gathering information.

17. Select the type of storage to use.

    ```
    Please specify the storage you would like to use (glusterfs, iscsi, fc,
    nfs)[nfs]:
    ```

    - If you selected NFS, enter the version, full address, and path to the storage, and any mount options.

      ```
      Please specify the nfs version you would like to use (auto, v3, v4,
      v4_1)[auto]:
      Please specify the full shared storage connection path to use (example:
      host:/path):
      storage.example.com:/hosted_engine/nfs
      If needed, specify additional mount options for the connection to the
      hosted-engine storage domain []:
      ```

    - If you selected iSCSI, enter the portal details and select a target and LUN from the automatically detected lists. You can only select one iSCSI target during the deployment, but multipathing is supported to connect all portals of the same portal group.

> **Note**
>
> To specify more than one iSCSI target, you must enable multipathing before deploying the self-hosted engine. There's also a Multipath Helper tool that generates a script to install and configure multipath with different options.

```
Please specify the iSCSI portal IP address:
  Please specify the iSCSI portal port [3260]:
  Please specify the iSCSI discover user:
  Please specify the iSCSI discover password:
  Please specify the iSCSI portal login user:
  Please specify the iSCSI portal login password:

  The following targets have been found:
      [1]    iqn.2017-10.com.redhat.example:he
          TPGT: 1, portals:
              192.168.1.xxx:3260
              192.168.2.xxx:3260
              192.168.3.xxx:3260

  Please select a target (1) [1]: 1

  The following luns have been found on the requested target:
    [1] 360003ff44dc75adcb5046390a16b4beb   199GiB  MSFT   Virtual HD
        status: free, paths: 1 active

  Please select the destination LUN (1) [1]:
```

- If you selected GlusterFS, enter the full address and path to the storage, and any mount options. Only replica 3 Gluster storage is supported.

```
* Configure the volume as follows as per [Gluster Volume Options for
Virtual
  Machine Image Store]
  (documentation/admin-guide/chap-Working_with_Gluster_Storage#Options
  set on Gluster Storage Volumes to Store Virtual Machine Images)

  Please specify the full shared storage connection path to use
(example: host:/path):
  storage.example.com:/hosted_engine/gluster_volume
  If needed, specify additional mount options for the connection to the
hosted-engine storage domain []:
```

- If you selected Fibre Channel, select a LUN from the automatically detected list. The host bus adapters must be configured and connected. The deployment script automatically detects the available LUNs, and the LUN must not contain any existing data.

```
The following luns have been found on the requested target:
  [1] 3514f0c5447600351   30GiB   XtremIO XtremApp
          status: used, paths: 2 active

  [2] 3514f0c5447600352   30GiB   XtremIO XtremApp
```

```
                    status: used, paths: 2 active

          Please select the destination LUN (1, 2) [1]:
```

18. Enter the engine disk size:

    ```
    Please specify the size of the VM disk in GB: [50]:
    ```

    If successful, one data center, cluster, host, storage domain, and the engine virtual machine are already running.

19. Optionally, sign in to the Oracle Linux Virtualization Manager Administration Portal to add any other resources.

    In the Administration Portal, the engine virtual machine, the host running it, and the self-hosted engine storage domain are flagged with a gold crown.

20. Enable the required repositories on the Engine virtual machine.

21. (Optional) Configure user federation in Keycloak (for example, integrate Microsoft Active Directory or LDAP) so that directory users and groups can sign in. For more information, see *Keycloak Integration and Management* in the Oracle Linux Virtualization Manager: Administrator's Guide.

Use Cockpit to Deploy Self-Hosted Engine

> ⚠ **Caution**
>
> - If the system is behind a proxy, you must use the command line option to deploy the self-hosted engine.
> - Cockpit deployment is available only for Oracle Linux 8 hosts.

To deploy the self-hosted engine using the Cockpit portal, complete the following steps.

1. Install the Cockpit dashboard.

   ```
   dnf install cockpit-ovirt-dashboard -y
   ```

2. Open the Cockpit port 9090 on firewalld.

   ```
   firewall-cmd --permanent --zone=public --add-port=9090/tcp


   firewall-cmd --reload
   ```

3. Enable and start the Cockpit service

   ```
   systemctl enable --now cockpit.socket
   ```

4. Sign in to the Cockpit portal at the following URL:

   `https://`***`host_IP_or_FQDN`***`:9090`

5. To start the self-hosted engine deployment, select **Virtualization** and select **Hosted Manager**.

6. Select **Start** under **Hosted Manager**.
7. Provide the following details for the Engine virtual machine.
    a. In the **Engine VM FQDN** field, enter the Engine virtual machine FQDN. Don't use the FQDN of the host.
    b. In the **MAC Address** field, enter a MAC address for the Engine virtual machine or leave blank and the system provides a randomy-generated address.
    c. From the **Network Configuration** dropdown list, select **DHCP** or **Static**.
        - To use **DHCP**, you must have a DHCP reservation (a preset IP address on the DHCP server) for the Engine virtual machine. In the **MAC Address** field, enter the MAC address.
        - To use **Static**, enter the virtual machine IP, the gateway address, and the DNS servers. The IP address must belong to the same subnet as the host.
    d. Select the **Bridge Interface** from the dropdown list.
    e. Enter and confirm the virtual machine's **Root Password**.
    f. Specify whether to enable **Root SSH Access**.
    g. Enter the **Number of Virtual CPUs** for the virtual machine.
    h. Enter the **Memory Size (MiB)**. The available memory is displayed next to the field.
8. Optionally, select **Advanced** to provide any of the following information.
    - Enter a **Root SSH Public Key** to use for root access to the Engine virtual machine.
    - Select the **Edit Hosts File** checkbox to add entries for the Engine virtual machine and the base host to the virtual machine's `/etc/hosts` file. You must ensure that the host names are resolvable.
    - Change the management **Bridge Name**, or accept the default of `ovirtmgmt`.
    - Enter the **Gateway Address** for the management bridge.
    - Enter the **Host FQDN** of the first host to add to the Engine. This is the FQDN of the host you're using for the deployment.
9. Select **Next**.
10. Enter and confirm the **Admin Portal Password** for the `admin@ovirt` user.
11. Optionally, configure event notifications.
    - Enter the **Server Name** and **Server Port Number** of the SMTP server.
    - Enter a **Sender E-Mail Address**.
    - Enter **Recipient E-Mail Addresses**.
12. Select **Next**.
13. Review the configuration of the Engine and its virtual machine. If the details are correct, select **Prepare VM**.
14. When the virtual machine installation is complete, select **Next**.
15. Select the **Storage Type** from the dropdown list and enter the details for the self-hosted engine storage domain.
    - For NFS:
        a. In the **Storage Connection** field, enter the full address and path to the storage.
        b. If required, enter any **Mount Options**.

c. Enter the **Disk Size (GiB)**.

d. Select the **NFS Version** from the dropdown list.

e. Enter the **Storage Domain Name**.

- For iSCSI:

  a. Enter the **Portal IP Address**, **Portal Port**, **Portal Username**, and **Portal Password**.

  b. Select **Retrieve Target List** and select a target. You can only select one iSCSI target during the deployment, but multipathing is supported to connect all portals of the same portal group.

  > **Note**
  >
  > To specify more than one iSCSI target, you must enable multipathing before deploying the self-hosted engine. You can use the Multipath Helper tool to generates a script that installs and configures multipath with different options.

  c. Enter the **Disk Size (GiB)**.

  d. Enter the **Discovery Username** and **Discovery Password**.

- For FibreChannel:

  a. Enter the **LUN ID**. The host bus adapters must be configured and connected and the LUN must not contain any existing data.

  b. Enter the **Disk Size (GiB)**.

- For Gluster Storage:

  a. In the **Storage Connection** field, enter the full address and path to the storage.

  b. If required, enter any **Mount Options**.

  c. Enter the **Disk Size (GiB)**.

16. Select **Next**.
17. Review the storage configuration. If the details are correct, select **Finish Deployment**.
18. When the deployment is complete, select **Close**.

    If successful, one data center, cluster, host, storage domain, and the engine virtual machine are already running.
19. Optionally, sign in to the Oracle Linux Virtualization Manager Administration Portal to add any other resources.

    In the Administration Portal, the engine virtual machine, the host running it, and the self-hosted engine storage domain are flagged with a gold crown.
20. Enable the required repositories on the Engine virtual machine.
21. (Optional) Configure user federation in Keycloak (for example, integrate Microsoft Active Directory or LDAP) so that directory users and groups can sign in. For more information, see *Keycloak Integration and Management* in the Oracle Linux Virtualization Manager: Administrator's Guide.
22. To view the self-hosted engine's status in Cockpit, under **Virtualization** select **Hosted Engine**.

Deploy Self-Hosted Engine Offline

You must perform a fresh installation of Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on a host to be configured as a KVM host *before* deploying a self-hosted engine. You can download the installation Oracle Linux ISO for from the Oracle Software Delivery Cloud at https://edelivery.oracle.com.

Prerequisites

1. Create a local mirror for the following repositories, where *n* is the major Oracle Linux version (8 or 9):
   - `ol`*n*`_x86_64_baseos_latest`
   - `ol`*n*`_x86_64_appstream`
   - `ol8_x86_64_kvm_appstream` or `ol9_x86_64_kvm_utils`
   - `ol`*n*`_x86_64_ovirt45`
   - `ol`*n*`_x86_64_ovirt45_extras`
   - `ol`*n*`_x86_64_addons`
   - **(Oracle Linux 8 only)**`ol8_x86_64_gluster_appstream`
   - **(For VDSM)**`ol8_x86_64_UEKR7`, or `ol9_x86_64_UEKR8` (Oracle Linux 9 only)

   > **Note**
   >
   > Gluster is only available on hosts running Oracle Linux 8.

   For information on creating local mirrors of Oracle Linux repositories, see Mirror a Yum Repository on Oracle Linux and Using Software Distribution Mirrors.
2. The local repositories can't mimic the original repository names from the Oracle Yum Public server or ULN. Add a prefix to each repository to identify it as local. For example, prefix all local repositories with by `local_`:
   - *`local`*`_ol`*n*`_baseos_latest`
   - *`local`*`_ol`*n*`_appstream`
   - *`local`*`_ol8_kvm_appstream` or `ol9_kvm_utils`
   - *`local`*`_ol`*n*`_ovirt45`
   - *`local`*`_ol`*n*`_ovirt45_extras`
   - *`local`*`_ol`*n*`_UEKR7`, or *`local`*`_ol9_UEKR8` (Oracle Linux 9 only)
   - **(Oracle Linux 8 only)***`local`*`_ol8_gluster_appstream`
3. To configure the Oracle Linux hosts in the network to use the local repository for updates and package installation, create a file called `/etc/yum.repos.d/local-ol`*n*`.repo` that lists all local repositories.

> **⚠ Caution**
>
> The file must be named `local-oln.repo` and must reside in the `/etc/yum.repos.d` directory. Ensure that *n* in the file name matches the Oracle Linux major release.

4. Follow the instructions in either Oracle® Linux 8: Installing Oracle Linux or Oracle® Linux 9: Installing Oracle Linux to install Oracle Linux 8.8 or later (8.*x*), or 9.6 or later (9.*x*) on the host using the **Minimal Install** base environment. See the Requirements and Scalability Limits section for an example of the partitioning schema.

   > **⚠ Caution**
   >
   > Do **NOT** select any other base environment than **Minimal Install** for the installation or the hosts will have incorrect qemu and libvirt versions, incorrect repositories configured, and no access to virtual machine consoles.
   >
   > Don't install any extra packages until after you have installed the Manager packages because they might cause dependency issues.

5. Reboot the host.

Offline Installation

Before you begin, ensure that you haven't made any changes to the **Minimal Install** of Oracle Linux.

1. List active repositories and disable all active **external** repositories. For example, using Oracle Linux 8:

   ```
   dnf repolist
   ```

   ```
   repo id           repo name
   ol8_UEKR7         Latest Unbreakable Enterprise Kernel Release 7 for
   Oracle Linux 8 (x86_64)
   ol8_appstream     Oracle Linux 8 Application Stream (x86_64)
   ol8_baseos_latest Oracle Linux 8 BaseOS Latest (x86_64)
   ```

   Disable the external repositories:

   ```
   dnf config-manager --disable ol8_UEKR7 ol8_appstream ol8_baseos_latest
   ```

   Or you can use a script, such as:

   ```
   for REPO in $( dnf repolist | awk '{print $1}' | tail -n +2 | grep -v
   local_ ); do dnf config-manager --disable ${REPO}; done
   ```

2. Rerun `dnf repolist` to confirm no external repositories are active.
3. So the installation can access the local repository, publish the `local-oln.repo` file to `/etc/yum.repos.d` directory.

4. Enable the local repositories. For example, using Oracle Linux 8:

   ```
   dnf config-manager --enable local_ol8_UEKR7 local_ol8_appstream
   local_ol8_baseos_latest local_ol8_gluster_appstream
   local_ol8_kvm_appstream local_ovirt-4.5 local_ovirt-4.5-extra
   ```

5. Rerun `dnf repolist` to confirm that the local repositories are active.
6. When you install the release packages, some external repositories (`ol8_gluster_appstream`, `ovirt-4.5`, and `ovirt-4.5-extra`) are enabled. To prevent the installation from failing because it can't access these external repositories, enable the `dnf skip_if_unavailable` option:

   ```
   dnf config-manager --save --setopt "skip_if_unavailable=True"
   ```

7. Install the Oracle Linux Virtualization Manager Release 4.5 package.
   a. (Optional) Without internet access, installing the `oracle-ovirt-release-45-eln` rpm package takes longer to complete. To speed up the installation, create an entry in the `/etc/hosts` file pointing `yum.oracle.com` to `localhost`, for example:

      ```
      echo '127.0.0.1  yum.oracle.com' >> /etc/hosts
      ```

   b. Install the release package:

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

      > **Note**
      >
      > Installing the release package automatically enables/disables the required external repositories. With the `skip_if_unavailable` configuration, alerts show on the screen that you can ignore.

8. Rerun `dnf repolist`. If the installation enabled any external repository, disable them.

   ```
   for REPO in $( dnf repolist | awk '{print $1}' | tail -n +2 | grep -v
   local_ ); do dnf config-manager --disable ${REPO}; done
   ```

9. If the host runs the Unbreakable Linux Kernel (UEK):
   a. Install the *Extra kernel modules* package.

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

   b. Install available updates for all installed packages:

      ```
      dnf update
      ```

   c. Reboot the host.
10. Install the `ovirt-hosted-engine-setup` and `ovirt-engine-appliance` rpm packages. Installing these packages together speeds up the deployment.

    ```
    dnf -y install ovirt-hosted-engine-setup ovirt-engine-appliance
    ```

11. To instruct the setup process to perform an offline installation and not enable KeyCloak, create a file called `/root/extra-vars.yml` that contains the following:

```
---
he_offline_deployment: true
he_enable_keycloak: true
```

> **Caution**
>
> The file must begin with three en dashes (`---`)

12. Deploy the self-hosted engine using the local repository:

```
hosted-engine --deploy --4 --ansible-extra-vars="@/root/extra-vars.yml"
```

13. Answer the on-screen questions to customize the deployment. Refer to the Engine Configuration Options.

14. When the installation completes and the self-hosted engine starts, connect to it using `ssh` and add the same entry to the `/etc/hosts` file pointing `yum.oracle.com` to `localhost`:

```
echo '127.0.0.1  yum.oracle.com' >> /etc/hosts
```

Enable High-Availability for Self-Hosted Engine Host

The host that houses the self-hosted engine isn't highly available by default. Because the self-hosted engine runs inside a virtual machine on a host, if you don't configure high-availability for the host, then virtual machine recovery after a host failure isn't possible.

If you want the self-hosted engine host to be responsive and available when unexpected failures happen, use fencing. Fencing lets the host react to unexpected failures and enforce power saving, load balancing, and virtual machine availability policies. Configure the fencing parameters for the hosts' power management device and test their correctness periodically.

A *Non Operational* host is different from a *Non Responsive* host. A *Non Operational* host can communicate with the Manager, but has incorrect configuration, for example a missing logical network. A *Non Responsive* host can't communicate with the Manager.

In a fencing operation, a non responsive host is rebooted, and if the host doesn't return to an active status within a prescribed time, it remains non responsive pending manual intervention and troubleshooting.

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

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

Configure Power Management and Fencing for Host

The Manager uses a proxy to send power management commands to a host power management device because the engine doesn't communicate directly with fence agents. The host agent (VDSM) runs 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 that 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.

   > ⚠ **Caution**
   >
   > 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 don't want the host's power management to be controlled by the scheduling policy of the host's cluster.
7. To configure a fence agent, select 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 dropdown list.
11. Enter the **Port** (SSH) number used by the power management device to communicate with the host.
12. Enter the **Slot** number used to identify the blade of the power management device.
13. Enter the **Options** for the power management device. Use a comma-separated list of key-value pairs.

- If you leave the **Options** field blank, you can 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 the power management device supports.

15. Select **Test** to ensure the settings are correct and then select **OK**.

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

> **Warning**
>
> Power management parameters (userid, password, options, and so on) are tested by the Manager only during setup and manually after that. If you ignore alerts about incorrect parameters, or if the parameters are changed on the power management hardware without being changed in the Manager, 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 the fence agent order in the **Agents by Sequential Order** field.

    b. To make two fence agents concurrent, next to one fence agent select the **Concurrent with** dropdown list and select the other fence agent.

       You can add more 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. Select 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 dropdown list and then select **OK**.

       The 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 can't find a fencing proxy within the cluster, it searches the data center.

19. Select **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.

Prevent Host Fencing During Boot

After you configure power management and fencing, when you start the Manager it automatically tries to fence non responsive hosts that have power management enabled *after* the quiet time (5 minutes by default) has elapsed. You can opt to extend the quiet time to prevent, for example, a scenario where the Manager 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
```

Check 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.

Install Additional Self-Hosted Engine Hosts

You add self-hosted engine hosts the same way as a regular host, with an extra step to deploy the host as a self-hosted engine host. The shared storage domain is automatically detected and the host can be used as a failover host to host the Engine virtual machine when required. You can also add regular hosts to a self-hosted engine environment, but they can't be used to host the Engine virtual machine.

> **Note**
>
> We recommend that all hosts in a self-hosted engine cluster run the same Oracle Linux KVM release.

Before you begin, see Prepare a KVM Host.

To install an extra self-hosted engine host, complete the following steps.

1. In the **Administration Portal**, go to **Compute** and select **Hosts**.
2. Select **New**.

   For information on other host settings, see the Admin Guide in the latest upstream oVirt Documentation.

3. Use the dropdown list to select the **Data Center** and **Host Cluster** for the new host.
4. Enter the **Name** and the **Address** of the new host. The standard SSH port, port 22, is automatically populated in the **SSH Port** field.
5. Select an authentication method to use for the engine to access the host.
   - Enter the root user's password to use password authentication.
   - Or, copy the key displayed in the **SSH PublicKey** field to `/root/.ssh/authorized_keys` on the host to use public key authentication.
6. Optionally, configure power management, where the host has a supported power management card. For information, see Configure Power Management and Fencing for Host.
7. Select the **Hosted Engine** sub tab.
8. Select the **Deploy** radio button.
9. Select **OK**.

Clean Up the Deployment

If the self-hosted engine deployment fails, you must perform a few cleanup tasks before retrying.

1. Run the hosted engine cleanup command:

   ```
   /usr/sbin/ovirt-hosted-engine-cleanup
   ```

2. Remove the storage:

   ```
   rm -rf <storage_repo>/*
   ```

3. If the deployment failed after the local, temporary hosted engine virtual machine is created, you might need to clean up the local virtual machine repository:

   ```
   rm -rf /var/tmp/localvm*
   ```

Upgrade or Update the Self-Hosted Engine

See *Upgrading Your Environment to 4.5* or *Updating the Self-Hosted Engine* in the Oracle Linux Virtualization Manager: Administration Guide.

6

Hyperconverged Infrastructure Deployment Using GlusterFS Storage

> **Note**
>
> Hyperconverged infrastructure deployment using GlusterFS is available only for Oracle Linux 8 hosts.

> **Caution**
>
> If you're deploying a self-hosted engine as hyperconverged infrastructure with GlusterFS storage, you must deploy GlusterFS *before* you deploy the self-hosted engine or any KVM hosts. For more information about using GlusterFS, including prerequisites, see the Oracle Linux GlusterFS documentation.

Oracle Linux Virtualization Manager is integrated with GlusterFS, an open source scale-out distributed file system, to provide a hyperconverged infrastructure (HCI) cluster where both compute and storage are provided from the same hosts. The HCI cluster with Gluster storage uses DAS disks to provide shared volumes and implements a KVM host in each node. The Gluster volumes are used as storage domains in the Manager to store the virtual machine images, and the Manager is run as a self-hosted engine within a virtual machine on these hosts.

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

> **Caution**
>
> You must deploy GlusterFS *before* you deploy the self-hosted engine or any KVM hosts. For more information about using GlusterFS, including prerequisites, see the Oracle Linux GlusterFS documentation.

To deploy Oracle Linux Virtualization Manager in a HCI architecture, you need three KVM hosts with local disks. These disks can be combined into a RAID array or used alone as JBOD. All KVM hosts must have the same number of disks and be the same size between hosts. If you want more than three KVM hosts, they must be added in factors of three.

For example, the minimum disk configuration for an HCI architecture is having two hard disks in each KVM host, where the first disk is used to install the OS and the second disk is used to deploy the Gluster volumes. For example:

```
Host 1          Host 2          Host 3
```

```
disk  1 - 250GB   disk  1 - 250GB   disk  1 - 250GB
disk  2 - 2TB     disk  2 - 2TB     disk  2 - 2TB


Host 1              Host 2              Host 3

disk  1 - 250GB     disk  1 - 250GB     disk  1 - 250GB
disks 2-8 - 4TB     disks 2-8 - 4TB     disks 2-8 - 4TB
```

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

Configure KVM Hosts for HCI Deployment

Before you can create Gluster volumes or deploy the Engine on the hyperconverged hosts, you must do a fresh installation of Oracle Linux 8.8 (8.*x*) and enabling the required repositories. For detailed instructions, see *Preparing a KVM host* in the Installation and Configuration section of Oracle Linux Virtualization Manager: Getting Started. (Do not proceed with *Adding a KVM host*.)

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

After installing the operating system on each host, prepare for deployment by completing the prerequisite tasks:

1. Cleanup host partitions/volumes
2. Configure KVM hosts and choose one as a deployment host
3. Install required packages

Ensure hosts have no partitions or LVM volumes on disks for Gluster use.

If you find any partitions or LVM volumes, remove them before continuing, for example:

```
[root@host1 ~]# lvscan | grep -i gluster

[root@host1 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0  250G  0 disk
|-sda1        8:1    0    1G  0 part /boot
+-sda2        8:2    0  249G  0 part
  |-ol-root 252:0    0  247G  0 lvm  /
  +-ol-swap 252:1    0  2.1G  0 lvm  [SWAP]
sdb           8:16   0  500G  0 disk
sr0          11:0    1 1024M  0 rom
```

Configure KVM hosts

1. Choose a *deployment host* referred to here as `kvmhost1`. The deployment host is used to start the Gluster and SHE deployment.

2. On the deployment host, use the `ssh-keygen` command to create an SSH keyring. This is used to configure Gluster nodes and volumes.

```
root@kvmhost1 ~]# ssh-keygen
```

3. Publish the SSH public key to the deployment host itself using its FQDN. For example:

```
root@kvmhost1 ~]# ssh-copy-id kvmhost1.example.com
```

4. Publish the SSH public key from the deployment host to all other hosts using their FQDNs. For example:

```
root@kvmhost1 ~]# ssh-copy-id kvmhost2.example.com
root@kvmhost1 ~]# ssh-copy-id kvmhost3.example.com
```

5. On the deployment host only, create a hard link to $HOME/.ssh/known_hosts for Gluster. For example:

```
[root@kvmhost1 ~]# ln $HOME/.ssh/known_hosts $HOME/.known_hosts
```

Install common rpm packages on all hosts and additional packages on the deployment host.

1. On all hosts
   a. Log in as root and install
      - `cockpit-ovirt-dashboard` to provide a web UI for installation
      - `vdsm-gluster` to manage Gluster services
      - `ovirt-host` to configure the host as a KVM hypervisor when added to the Engine console

      For example, run the following command on the `kvmhost1`, `kvmhost2`, and `kvmhost3`:

```
dnf install cockpit-ovirt-dashboard ovirt-host vdsm-gluster
```

   b. Run the following commands to ensure the `cockpit.socket` is enabled and started and to open the cockpit port in `firewalld`.
      For example, run the following commands on the `kvmhost1`, `kvmhost2`, and `kvmhost3`:

```
systemctl enable --now cockpit.socket
firewall-cmd --permanent --add-service cockpit
firewall-cmd --reload
```

2. On the deployment host only, install the `ovirt-engine-appliance` and `gluster-ansible-roles` packages.

```
[root@kvmhost1 ~]# dnf install ovirt-engine-appliance gluster-ansible-roles
```

Deploy GlusterFS Storage Using Cockpit

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

> **Important**
>
> Before you deploy Gluster, ensure you have read about deploying Oracle Linux Virtualization Manager in a HCI architecture and completed the required configuration for all KVM hosts.

1. From the deployment host, access the Cockpit web interface from `https://host_IP_or_FQDN:9090`, for example, `https://kvmhost1.example.com:9090`.
2. Log in using the user name and password of the root account.
3. From the Cockpit left navigation, click Virtualization.
4. From the Virtualization menu, click Hosted Manager.
5. On the Hosted Engine Setup page there are two Start buttons. Under the Hyperconverged statement *Configure Gluster storage and Oracle Linux Virtualization Manager*, click Start.
6. From the Gluster Configuration popup, click Run Gluster Wizard.
   The Gluster Deployment wizard displays.
7. On the Hosts screen, enter the FQDN for each Gluster host.
   - If the host has different network connections for the public network and the storage network, enter those different hostnames.
   - If hosts have only one network connection, check *Use same hostname for Storage and Public Network*.
8. Click the Next.
9. On the Packages screen, do not enter any information. Click Next.
10. On the Volumes screen, create the minimum required volumes of `engine` and `data`. You can also create `export` and `iso` volumes. Be sure to check the Arbiter box next to each volume you create.
    For example:
    - Name: `engine`
    - Volume Type: `Replicate` (default)
    - Arbiter: Ensure the check box is selected.
    - Brick Dirs: `/gluster_bricks/engine/engine` (default)
    - Name: `data`
    - Volume Type: `Replicate` (default)
    - Arbiter: Ensure the check box is selected.
    - Brick Dirs: `/gluster_bricks/data/data` (default)
11. Click Next.
12. On the Bricks screen:
    - Select the appropriate Raid Type. Use JBOD for internal disks or select the appropriate RAID level if internal disks are configured as RAID devices.
    - Under Multipath Configuration, ensure the Blacklist Gluster Devices checkbox is selected.
    - (Optional) Under Brick Configuration, adjust the LV size for each host's block device.

13. Click Next.
14. On the Review screen, review the configuration and then click Next to deploy the Gluster configuration and create volumes.
    This process takes some time to complete as the `gdeploy` tool installs required packages and configures Gluster volumes and their underlying storage.

    If successful, Cockpit displays the *Successfully deployed Gluster* message and your Gluster deployment is ready for use.
15. Click the Continue to Hosted Engine Deployment button.

> **Important**
>
> You can only continue with deploying the hosted engine with Cockpit if your hosts have a direct connection to the internet. If you do not have a direct internet connection, are behind a proxy, or click Close to continue deployment at a later date, you must use the command line to deploy the self- hosted engine.

Deploy Self-Hosted Engine Using Cockpit

If your hosts do not have a direct internet connection, are behind a proxy, or you clicked Close in Cockpit after deploying Gluster, you must use the command line to deploy the self- hosted engine.

To deploy the self-hosted engine using the Cockpit web interface *immediately* after deploying Gluster, you should have clicked Continue to Hosted Engine Deployment in the last step of the Gluster deployment instructions.

Complete the following steps using the Hosted Engine Deployment wizard.

1. On the VM screen, fill in the following VM settings information:
   - In the Engine VM FQDN field, enter the Engine virtual machine FQDN, which must be resolvable by a DNS search. Do not use the FQDN of the host.
   - In the MAC Address field, enter a MAC address for the Engine virtual machine only if you do not want to use the auto-generated address.
   - From the Network Configuration list, select either DHCP or Static.
     - To use DHCP, you must have a DHCP reservation (a pre-set IP address on the DHCP server) for the Engine virtual machine.
     - To use Static, enter the virtual machine IP, the netmask and gateway addresses, and DNS server. The IP address must belong to the same subnet as the host.
   - From the Bridge Interface list, select the physical network interface to configure the bridge on.
   - Enter and confirm the virtual machine's Root Password.
   - Specify whether to allow Root SSH Access.
   - Enter the Number of Virtual CPUs for the virtual machine.
   - Enter the Memory Size (MiB). The available memory is displayed next to the field.
2. (**Optional**) Click Advanced to provide any of the following information.
   - Enter a Root SSH Public Key to use for root access to the Engine virtual machine.

- Select the Edit Hosts File check box if you want to add entries for the Engine virtual machine and the base host to the virtual machine's `/etc/hosts` file. You must ensure that the host names are resolvable.
- Change the management Bridge Name, or accept the default of `ovirtmgmt`.
- Enter the Gateway Address for the management bridge.
- Enter the Host FQDN of the first host to add to the Engine. This is the FQDN of the host you are using for the deployment.

3. Click Next.
4. On the Engine screen, enter a password for the Admin user in the Admin Portal Password field. Do not change any other fields.
5. Click Next.
6. Review the options in the Prepare VM screen. Click Prepare VM to continue or the Back if you need to change any options.
7. When the Prepare VM completes successfully, click Next.
8. On the Storage screen, select Gluster as the Storage Type. The Storage Connection should have the deployment node as the primary connection and other nodes as backup mount servers.

   Do not change any other fields.
9. Click Next.
10. On the Finish screen, review the mount information and click Finish Deployment.

    This process

    - transfers the Hosted Engine virtual disk to the Gluster engine volume
    - creates a VM named `Hostedengine`
    - configures services to start this instance automatically when the hyperconverged hosts boots
    - configures the deployment host as a KVM host in the Administration Portal
11. Add the remaining hyperconverged nodes as KVM hosts. See Add Hyperconverged Hosts to Cluster for instructions.

Add Hyperconverged Hosts to Cluster

After deploying the self-hosted engine, you must add the remaining hyperconverged hosts to the virtualization cluster.

1. Log in to the Administration Portal.
2. Go to **Compute** and then select **Hosts**.
3. On the **Hosts** pane, select **New**.
   The **New Host** dialog box opens with the **General** tab selected on the sidebar.
4. From the **Host Cluster** dropdown list, select the data center and host cluster for the host. The **Default** data center is automatically selected.

   When you install Oracle Linux Virtualization Manager, a data center, and cluster named Default is created. You can rename and configure this data center and cluster, or you can add new data centers and clusters, to meet requirements. See the Data Centers or Clusters tasks in the Oracle Linux Virtualization Manager: Administration Guide.

5. In the **Name** field, enter a name for the host. This is the name you see in the UI.
6. In the **Hostname** field, enter the fully-qualified domain name or IP address of the host.
7. In the **SSH Port** field, change the standard SSH port 22 if the SSH server on the host uses a different port.
8. Under **Authentication**, select the authentication method to use.
   Oracle recommends that you select **SSH PublicKey** authentication. If you select this option, copy the key displayed in the **SSH PublicKey** field to the `/root/.ssh/authorized_keys` file on the host.

   Otherwise, enter the root user's password to use password authentication.
9. In the Power Management tab, check **Enable Power Management** and select + (plus) to configure an IPMI, iDRAC, ILO, or any other hardware management connection available.

> **Note**
>
> You should configure the KVM host Power Management to allow the Engine application and system administrators to manage (reboot or power off) hosts in NonResponsive or NonOperational states when recovering from a host failure.
>
> In NonResponsive or NonOperational states, `ssh` management might not be able to recover the host forcing manual intervention. See Configure Power Management and Fencing for Host for more information.

10. In the Hosted Engine tab, select **Deploy** from the **Choose hosted engine deployment action** dropdown.
11. Select **OK** to configure the host as a virtualization node.
12. Repeat this process for all remaining hyperconverged hosts.

> **Warning**
>
> Don't deploy the hosted engine on more than seven KVM hosts.