

Oracle Linux

Using the Cockpit Web Console



F51970-02
June 2022



Oracle Linux Using the Cockpit Web Console,
F51970-02
Copyright © 2022, Oracle and/or its affiliates.

Contents

Preface

| | |
|--|------|
| Conventions | vii |
| Documentation Accessibility | vii |
| Access to Oracle Support for Accessibility | vii |
| Diversity and Inclusion | viii |

1 Install and Log Into the Cockpit Web Console

| | |
|--------------------------------------|-----|
| Install the cockpit package | 1-1 |
| Enable and start the Cockpit service | 1-1 |
| Configure Firewall Rules (Optional) | 1-1 |
| Logging into Cockpit | 1-1 |

2 Use Cockpit to Configure Kdump

| | |
|---|-----|
| Access the Kernel Dump information | 2-1 |
| Enable kdump, configure memory, and specify the crash dump location | 2-1 |
| Test your kdump settings | 2-2 |

3 Use Cockpit to Configure System Settings and View System Information

| | |
|---|-----|
| Access the System Information | 3-1 |
| View information about resource usage | 3-1 |
| View more information about your hardware | 3-2 |
| Set the System Host Name | 3-2 |
| Join a Domain | 3-3 |
| Configure the System Time | 3-3 |

4 Use Cockpit to Set Up Performance Profiles

| | |
|--|-----|
| Performance profiles that are available in the web console | 4-1 |
| Change a performance profile | 4-2 |

| | | |
|----|---|------|
| 5 | Use Cockpit to Monitor System Logs | |
| | Access the system logs | 5-1 |
| | Customize the list of logs | 5-1 |
| | Use the text search functionality to filter the events list | 5-2 |
| | Using the available predefined filters | 5-2 |
| | Using available quantifiers | 5-2 |
| | Searching by log fields or free form text | 5-3 |
| | Using advanced search | 5-3 |
| 6 | Use Cockpit to Manage System Services | |
| | Access system service information | 6-1 |
| | Managing system services | 6-1 |
| 7 | Use Cockpit to Manage User Accounts | |
| | Access the Accounts page | 7-1 |
| | Create a user | 7-1 |
| | Specify additional user settings | 7-2 |
| 8 | Use Cockpit to Manage Software Updates | |
| | Manage manual software updates | 8-1 |
| | Manage automatic software updates | 8-1 |
| 9 | Use Cockpit to Manage Network Configuration | |
| | Access the Network configuration page | 9-1 |
| | Configure the Firewall | 9-2 |
| | Configure IP Addressing for a Network Interface | 9-2 |
| | Configure a Network Bond | 9-3 |
| | Configure a Network Team | 9-4 |
| 10 | Use Cockpit to Manage Physical Drives in Volume Groups | |
| | Access the storage information | 10-1 |
| | Add physical drives to volume groups | 10-1 |
| | Remove physical drives from volume groups | 10-1 |

11 Use Cockpit to Manage Partitions

| | |
|---|------|
| Access the storage information | 11-1 |
| Display partitions that are formatted with file systems | 11-1 |
| Create partitions | 11-1 |
| Delete partitions | 11-2 |

12 Use Cockpit to Manage Logical Volumes With LVM

| | |
|--------------------------------|------|
| Access the storage information | 12-1 |
| Create volume groups | 12-1 |
| Create logical volumes | 12-1 |
| Format logical volumes | 12-2 |
| Resize logical volumes | 12-2 |

13 Use Cockpit to Encrypt Block Devices With LUKS

| | |
|---------------------------------|------|
| Access the storage information | 13-1 |
| Lock data on a device with LUKS | 13-1 |
| Change the LUKS configuration | 13-1 |

14 Use Cockpit to Manage Virtual Data Optimizer Volumes

| | |
|--|------|
| Access the storage information | 14-1 |
| Create and configure VDO devices and volumes | 14-1 |
| Format the VDO volume | 14-2 |
| Extend a VDO volume | 14-2 |

15 Use Cockpit to Manage Redundant Arrays of Independent Disks

| | |
|-----------------------------------|------|
| Access the storage information | 15-1 |
| Create RAID storage | 15-1 |
| Format RAID storage | 15-1 |
| Create partitions on RAID storage | 15-2 |

16 Use Cockpit to Enable Network Bound Disk Encryption

| | |
|--|------|
| Access the storage information | 16-1 |
| Create a Tang key for the encrypted device | 16-1 |
| Confirm that the configuration is successful | 16-2 |

17 Use Cockpit to Manage NFS Mounts

| | |
|--------------------------------|------|
| Access the storage information | 17-1 |
| Connect NFS mounts | 17-1 |
| Customize mount options | 17-2 |

18 Use Cockpit to Manage Virtual Machines

| | |
|---|------|
| Install the Cockpit Virtual Machines Module | 18-1 |
| Enabling Virtualization | 18-1 |
| Review the Virtual Machines Page | 18-2 |
| Check the Storage Pools | 18-2 |
| Check the Networks | 18-3 |
| Create a Virtual Machine | 18-4 |
| Video Demonstration | 18-5 |

19 Use Cockpit to Manage Podman Containers

| | |
|-----------------------------------|------|
| Install the Cockpit Podman Module | 19-1 |
| Enable Podman Service | 19-1 |
| Managing Podman Images | 19-1 |
| Managing Podman Containers | 19-4 |

Preface

Oracle Linux includes a web console you can use for system administration. The web console is called Cockpit. For non-minimal installations, Cockpit is automatically installed, although not automatically enabled. Cockpit provides a web browser interface for performing system configuration and administration tasks, either locally or remotely on multiple servers. These tasks include system resource monitoring and log review, network and firewall configuration, and package management and updates. Cockpit uses the same APIs to access system services, so any changes you make using operating system command line tools are updated in real time in Cockpit.

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. |
| <i>italic</i> | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| <code>monospace</code> | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

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

Access to Oracle Support for Accessibility

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

Diversity and Inclusion

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

1

Install and Log Into the Cockpit Web Console

This chapter shows you how to install and set up the Cockpit web console on an Oracle Linux system to enable you to perform basic system configuration and administration by using a web-based user interface.

Install the `cockpit` package

On Oracle Linux systems with non-minimal installations, the `cockpit` package is included by default. Otherwise, you can manually install Cockpit. In either case, running the following command ensures that the package is installed and is up to date.

```
sudo dnf install cockpit
```

Enable and start the Cockpit service

To enable and start the Cockpit service, so that you can start accessing it immediately and so that it starts automatically after a reboot, run the following command:

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

The service starts and runs a web server that listens on TCP port 9090 by default. You can check the status of the service by running:

```
sudo systemctl status cockpit
```

Configure Firewall Rules (Optional)

If you are using a custom firewall profile, or an Oracle Cloud Infrastructure instance, open the firewall port for the web console (9090).

To enable the firewall port for the cockpit service and reload the default firewall service on Oracle Linux, run:

```
sudo firewall-cmd --add-service=cockpit --permanent  
sudo firewall-cmd --reload
```

Logging into Cockpit

Cockpit serves both HTTP and HTTPS requests on port 9090. By default, Cockpit creates self-signed certificates that are used to facilitate HTTPS. If you use the self-signed certificate, when you go to the web console, the browser displays a security exception warning. To avoid

having to grant a security exception, install a certificate signed by a certificate authority (CA) in the `/etc/cockpit/ws-certs.d` directory. The last file (in alphabetical order) with a `.cert` extension is used.

Cockpit uses a PAM stack located at `/etc/pam.d/cockpit` to handle authentication of users. Authentication with PAM allows you to log in with a username and password of any system account that has administrator privileges.

To log into Cockpit:

1. In a web browser, go to the Cockpit web console using the hostname or IP address of the system at port 9090 using HTTPS. For example:

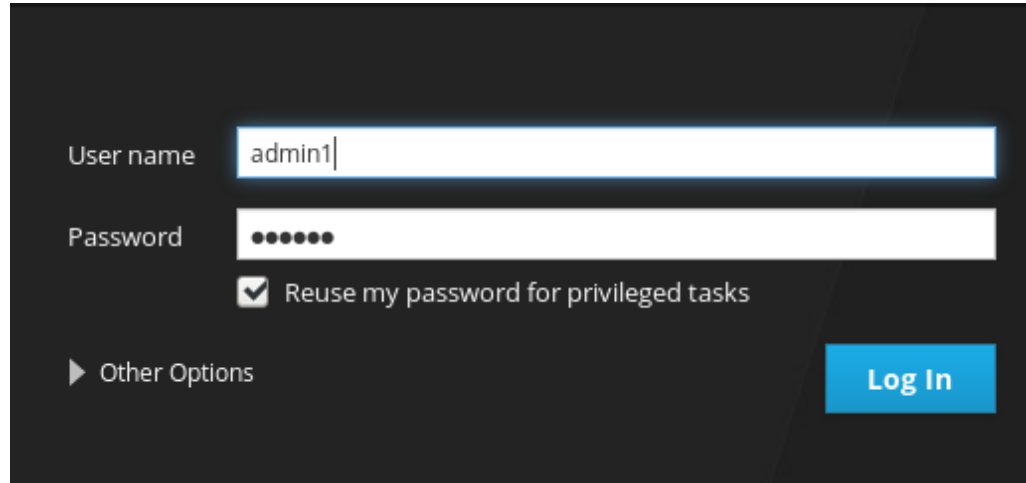
```
https://myserver.example.com:9090
```

If you are logging in on the local host, you can use:

```
https://localhost:9090
```

If you are not using a signed security certificate, a warning that the connection is not private is displayed. To continue, add an exception for the site in the browser.

2. Log into Cockpit using a system user account. If the user account has sudo privileges, you can run privileged tasks in the web console. To enable running sudo commands, check the **Reuse my password for privileged tasks** option. Click **Log In**. The Cockpit dashboard is displayed.



Tip:

To connect to a remote Oracle Linux server running Cockpit, use the **Connect to** field in **Other Options** and enter the URL for the remote host.

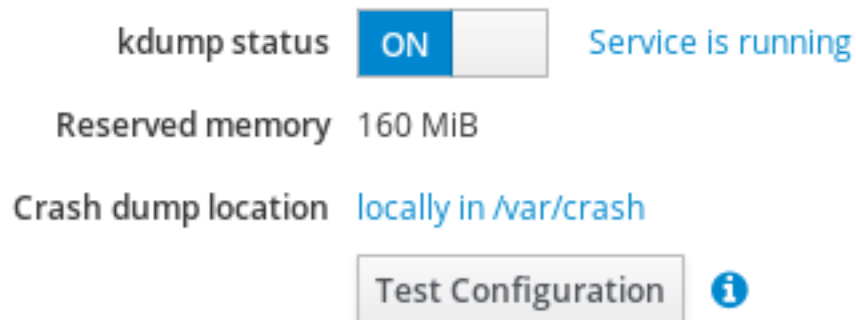
2

Use Cockpit to Configure Kdump

This chapter shows you how to enable and configure the kernel dump (kdump) feature on an Oracle Linux system by using the Cockpit web console. You can access the web console to view the current status of the `kdump` service and the amount of memory that is reserved for the kdump kernel, as well as specify the target location of the `vmcore` dump file and test your kdump settings..

Access the Kernel Dump information

After you log into Cockpit, the **Overview** page is displayed. Click **Kernel Dump** on the navigation panel on the left side of the screen to view the status of the `kdump` service, the reserved memory amount, and the current Crash dump location. The tab also provides an option for testing the kdump configuration. Note that selecting the **Test Configuration** option tests the current kdump configuration by crashing the kernel.



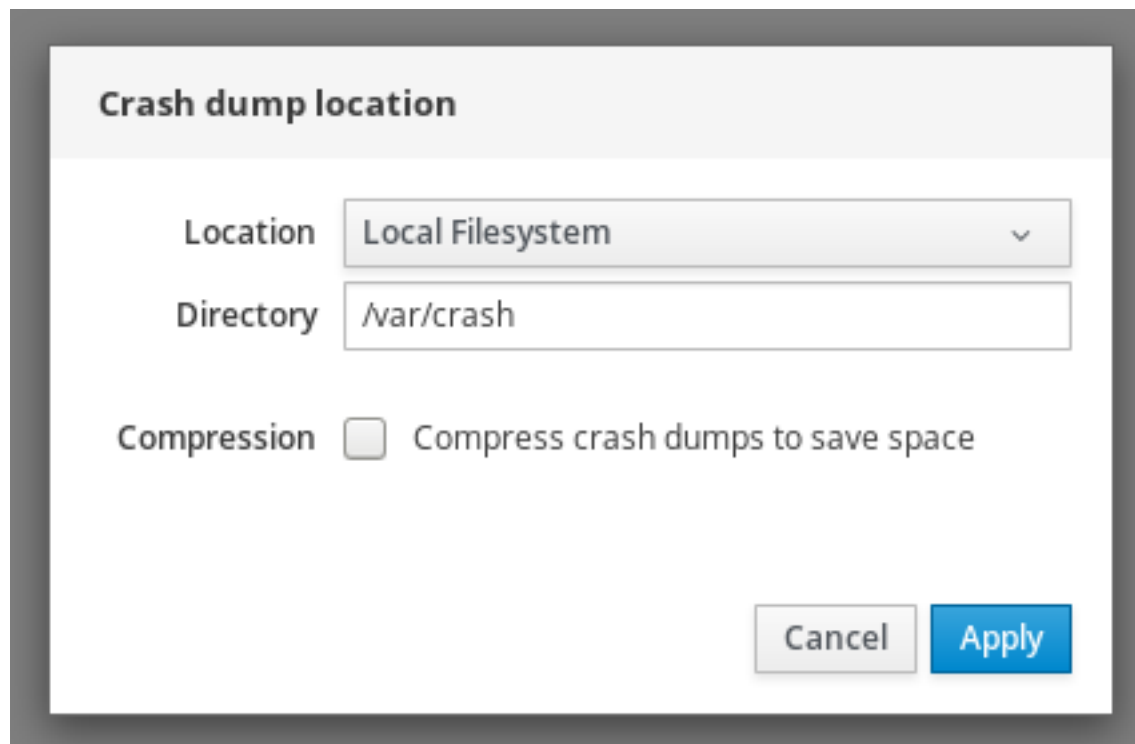
Enable kdump, configure memory, and specify the crash dump location

On the **Kernel Dump** tab, do the following:

1. Enable the `kdump` service by toggling the **kdump status** switch to **On**, as shown in the previous figure.
2. If necessary, configure the amount of memory to be reserved for kdump by using the command line.

For instructions, see *Working With Kernel Dumps* in [Oracle Linux 8: Monitoring and Tuning the System](#).

3. Click the link next to the **Crash dump location** option to open the Crash dump location window, as shown in the following figure.
4. For the **Location** where you want to save the crash dump, select from the options provided in the drop-down list. The default location is **Local Filesystem**. Other locations include the following:
 - **Remote over SSH:** This option sends the `vmcore` to a remote system by using SSH. To use this option, you must provide the information for the Server, ssh key, and Directory fields with the remote machine address, ssh key location, and a target directory.
 - **Remote over NFS:** This option sends the `vmcore` to a remote system by using the NFS protocol. To use this option, provide the required information in the Mount field.
5. (Optional) To specify whether to compress the crash dump to save space, select the **Compression** check box.
6. Click **Apply** to save the changes.

A screenshot of a 'Crash dump location' configuration window. The window has a title bar 'Crash dump location'. Inside, there are three main sections: 'Location' with a dropdown menu showing 'Local Filesystem', 'Directory' with a text input field containing '/var/crash', and 'Compression' with an unchecked checkbox and the text 'Compress crash dumps to save space'. At the bottom right, there are two buttons: 'Cancel' and 'Apply'.

Test your kdump settings

To test your kdump settings:

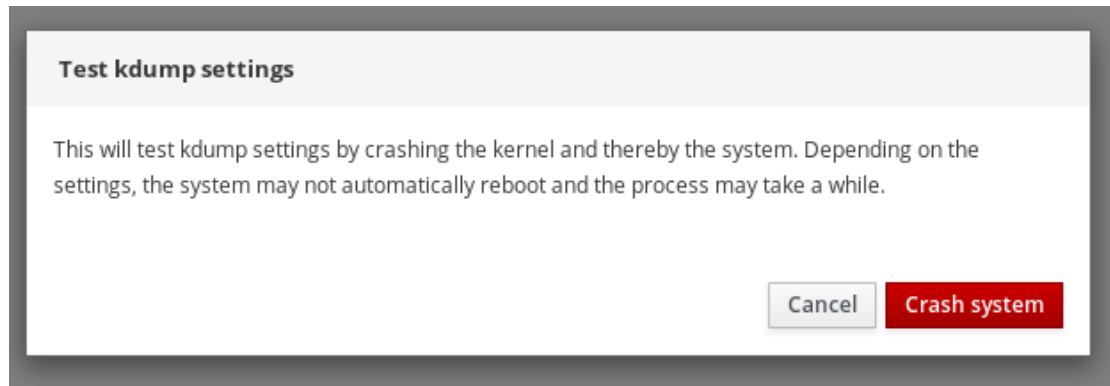
1. Access the **Kernel Dump** tab from the web console's navigation panel located on the left side of the screen.
2. On the **Kernel Dump** tab, select the **Test Configuration** option to open the **Test kdump settings** window.
3. To test your kdump configuration, select **Crash system**.



WARNING:

Selecting this option causes a system crash and loss of data.

Clicking **Cancel** cancels the operation.



3

Use Cockpit to Configure System Settings and View System Information

This chapter shows you how to configure basic system settings, such as the system host name, date, and time for an Oracle Linux system by using the Cockpit web console.

Access the System Information

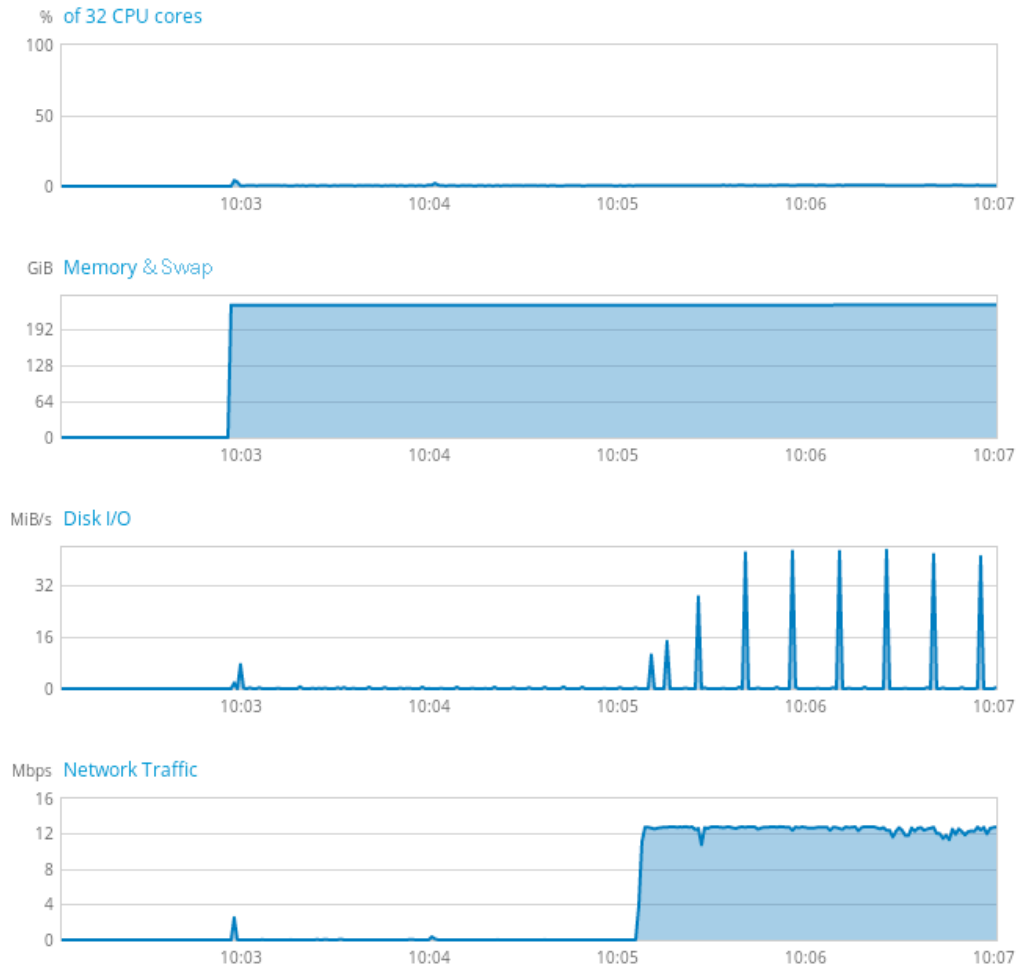
After you log into Cockpit, the **Overview** page is displayed by default. This page displays information about the system in the following sections:

- **Health:** Provides a general assessment of the system's health, including an indication of any failed services.
- **Usage:** Provides graphical presentations of how system resources are used. A link is provided to view more detailed graphs.
- **System information:** Provides information about the system hardware. A link is provided to view more details.
- **Configuration:** Provides information about the system's base settings, such as the host name, system time and date, and so on.

On this same page, you can edit current settings in their respective sections to reconfigure parameters according to your preference.

View information about resource usage

In the **Usage** section, click **View graphs**.



A page displays realtime graphical representations of system resource usage. Each graph is titled to indicate the resources being displayed. The graph titles are links that enable you to view more information or to configure resources. For example, CPU and Memory & Swap show more detailed graphs of the usage of these resources while the Disk I/O and Network Traffic provide access to these areas within Cockpit, where you can view more information or configure these resources.

View more information about your hardware

The **System information** section displays a table of useful information about the system. Click **View hardware details** to open a page that lists additional hardware information that Cockpit can detect on your system. Typically, much of this information is extracted from the SMBIOS data structures. A listing of hardware components using the PCI bus is displayed in a similar format to that returned by the **lspci** command on the command line.

Set the System Host Name

In the **Configuration** section, click **edit** next to the existing host name. On the dialog that opens, set the **Pretty Host Name** and the **Real Host Name**.

The **Pretty Host Name** is a friendly free-form system name that is displayed in user interface environments. If it is not set, the **Real Host Name** is used instead.

The **Real Host Name** is equivalent to the static host name set in the `/etc/hostname` file and the transient host name that is used at run time by the system and which can be reset automatically by services like DHCP or mDNS. Setting this value takes immediate effect and does not require a reboot.

This feature is equivalent to using the **hostnamectl** command to set these values on the command line.

Join a Domain

If your system is not already configured as part of an Active Directory or IPA domain, you can use Cockpit to enroll the system and join a domain. This process is handled using the **realmd** DBus API. The **Join Domain** link in the **Overview's Configuration** section opens a dialog where you can enter the domain details required to enroll the system. Note that this process is similar to using the **realm join** command from the command line. More information is available on the `realm(8)` manual page.

Configure the System Time

You can set the system time in the same **Configuration** section. The current date and time are displayed, both of which are relative to the configured time zone for the system. Click the displayed date and time to open a dialog where you can configure the time zone and the system time.

The **Time Zone** field is a searchable drop-down selector where time zones are listed by global region and city. Type the first few letters of the closest city to quickly navigate through the list.

The **Set Time** field is a drop-down list of the following options:

- **Manually:** You must set the values for the exact time and date specific to the time zone that you have selected.
- **Automatically using NTP:** The system uses any available NTP service to obtain the correct time. On Oracle Linux, the **chrony** NTP service is available and is typically configured to use the `pool.ntp.org` servers by default.
- **Automatically using specific NTP servers:** The system uses the NTP servers that you specify in the provided fields. This option is greyed out to indicate a non-functioning option because the system does not have `systemd-timesyncd`, a package that is not provided by Oracle Linux.

Use Cockpit to Set Up Performance Profiles

This chapter describes how to set up Tuned performance profiles for an Oracle Linux system by using the Cockpit web console. The web console configures the `tuned` service for the selected profile. For more information about working with Tuned by using the command line, see *Working With Tuned* in [Oracle Linux 8: Monitoring and Tuning the System](#).

Performance profiles that are available in the web console

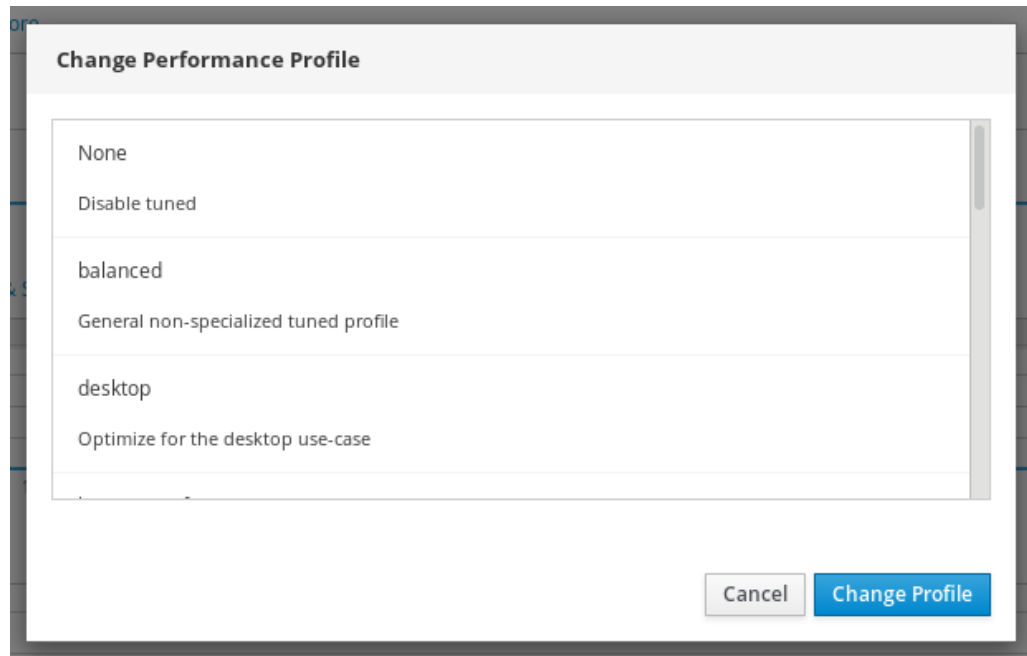
After you log into Cockpit, the **Overview** page is displayed. Scroll to the **Configuration** section and click the current performance profile to open the **Change Performance Profile** page. A list of available profiles is displayed from which you can select to replace the current profile. Oracle Linux provides the following tuned performance profiles:

- **None**: Disables the `tuned` service.
- **accelerator performance**: Throughput performance based tuning with disabled higher latency STOP states
- **balanced**: Is a general non-specialized tuned profile.
- **desktop**: Optimizes for a desktop environment.
- **hpc-compute**: Optimize for HPC compute workloads
- **intel-sst**: Configure for Intel Speed Select Space Base Frequency
- **latency-performance**: Optimizes for deterministic performance at the cost of increased power consumption.
- **network-latency**: Optimizes for deterministic performance at the cost of increased power consumption and focuses on low latency network performance.
- **network-throughput**: Optimizes for streaming network throughput. This profile is generally only necessary on older CPUs or 40G+ networks.
- **power-save**: Optimizes for low power consumption.
- **throughput-performance**: Applies broadly to tuning that provides excellent performance across a variety of common server workloads.
- **virtual-guest**: Optimizes for running inside a virtual host.
- **virtual-host**: Optimizes for running KVM guests.



Note:

The "Recommended" profile is usually indicated as such.



Change a performance profile

To change to a different performance profile, do the following:

1. Select a profile from the list of available performance profiles.
2. Click **Change Profile** to save the changes.

The new performance profile is now reflected on the **Performance profile** field of the **Configuration** section.

5

Use Cockpit to Monitor System Logs

This chapter shows you how to use Cockpit to monitor processes through the system logs to check their status and identify faulty operations that require further investigation.

Access the system logs

After logging in, the Overview page is displayed by default. Click **Logs** on the left navigation panel. Process or events logs are then displayed similar to the following example:

```
PAM adding faulty module: /usr/lib64/security/pam_sss.so      sudo
Failed to start dnf makecache                                systemd
...
```

Each item in the list has a corresponding time stamp as well as the source of that event log, such as `sudo`, `systemd`, `kernel`, and so on. Because monitoring is ongoing, logs are added to the list over time. To halt the continuing display, click **Pause**.

By selecting a specific event log, you can obtain further details about the event. The report displays the event's priority, syslog facility, syslog identifier, the audit login UID and session, and other details.

Customize the list of logs

You can limit the list to specific event logs by applying a filter or a combination of filters. At the top of the page, predefined filters are provided as drop down lists. You can filter by time, priority, and identifier.

For the **Time** filter, the following options are available:

- Current boot (default)
- Previous boot
- Last 24 hours
- Last 7 days

For the **Priority** filter, the following options are available:

- Only emergency
- Alert and above
- Critical and above
- Error and above (default)
- Warning and above
- Notice and above
- Info and above

- Debug and above

The priorities are listed in descending order and the lowest priority (Debug and above) provides the most expansive list of events.

For the `Identifier` filter, the following options are available:

- All (default)
- cockpit-session
- kernel
- password
- sshd
- sudo
- systemd

Use the text search functionality to filter the events list

The `Text` field provides you with greater flexibility to further customize the event logs list.

Each filter or combination of filters that is applied to the list has a corresponding `journalctl` command syntax with specific parameters. For example, the log page, by default, displays all event logs with the `Error` and above priority. The underlying command that applies this filter is as follows:

```
sudo journalctl --priority=err
```

To display the command syntax, click the question mark icon next to the `Text` field.

You can perform a text search of the logs in several ways:

Using the available predefined filters

You can specify the predefined filters `priority` and `identifier` in the `Text` field to perform a text search, for example:

```
priority:emerg identifier:kernel
```

When you press Enter, a list of logs that match the search criteria is generated.

Using available quantifiers

Some quantifiers, such as `priority` and `identifier`, are already available as predefined filters. In addition, the following quantifiers can be used for a text search:

- `follow`: Show only the most recent journal entries. This quantifier generates a "live" display, which means that more entries are added as these events are appended to the journal.
- `service`: Show messages that apply to a specific `systemd` unit.

- **boot:** Show messages that apply to a specific boot. This quantifier is partially used by the predefined filter **Time**.
- **since:** Show messages that apply to a specified date or time. The format must be YYYY-MM-DD HH:MM:SS. The time uses the 24-hour format.

For example, you might search the logs with the following quantifiers:

```
priority:err identifier:systemd since:2021-05-03 follow
```

Searching by log fields or free form text

Searching by using log fields requires that you also provide the content of those fields, for example:

```
priority:err exe:/usr/bin/sudo hostname:mssystem audit_loginuid:1000
```

However, you can also type free form text to search the logs. The following search lists logs that refer to the `makecache` process:

```
makecache
```

Using advanced search

Advanced search is simply the combination of the preceding methods for performing text searches, as shown in the following example:

```
identifier:systemd since:2021-03-15 JOB_TYPE=start,restart
```

6

Use Cockpit to Manage System Services

This chapter shows you how to use Cockpit to manage Oracle Linux system services.

Access system service information

After logging in, the Overview page is displayed by default. Click **Services** to display the **System Services** tab.

The services page displays all the services available and their current configuration for your system or instance. The information provided includes all the service names, descriptions, their current states, and whether or not they start automatically on boot.

Select any of the services that are listed to open a new screen, where you can review any system log entries for that service, as well as examine details about any related services. You can also use this screen to activate, deactivate, and restart the selected service.

Managing system services

1. Select the service that you want to manage.
2. In the screen for that service, click the on-off switch to set service behavior.
3. (Optional) Click the **Additional actions** menu for further options, such as restarting the service or adding a mask to prevent the service from running.

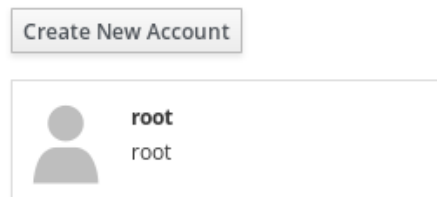
7

Use Cockpit to Manage User Accounts

This chapter shows you how to create and manage user accounts on an Oracle Linux system using the Cockpit web console. The chapter also describes how to view individual user account settings so you can modify the settings as necessary

Access the Accounts page

After you log into Cockpit, the **Overview** page is displayed. Click **Accounts** on the navigation panel on the left side of the screen to access the page for creating users.



The page displays the current users on the system. At a minimum, the root account is displayed. If you created a user when you installed Oracle Linux, then that user is also displayed.

Create a user

Click **Create New Account** to create a new user. A new window opens that prompts you for the following information about the user:

- Full name
- User name
- Password
- Confirm

Note:

The password must be at least 8 characters in length. Otherwise, you cannot proceed to create the account.

Select **Lock Account**, if needed.

After you have provided the required information, click **Create**. The new user is added to the list of users on the page.

Specify additional user settings

Click the user's name to open a new window where you can configure additional user settings, such as the following:

- Grant an administrator role to the user.
If you grant this role to the user, the user must log out and back in for the change to become effective.
- Determine whether the user account should be locked.
If you did not select **Lock Account** in the previous step, you can set the option on this page.
- Set an expiration date for the account.
This option is useful if you are creating a temporary user account. Click **Never lock account** to open the **Account Expiration** window. Select the **Lock account on** option and click the field to display a calendar where you can choose the account's expiration date.
- Set a new password or force a password change.
If you select to force a password change, the user is forced to change the password the next time the user logs in. This option is useful if you create the password for the new user, but you want the user to set a personal password at login.
- Set an expiration date for a password.
By default, the password of the newly created user is set to never expire. To limit the validity of a password, select **Never expire password**, which opens the **Password Expiration** window. Then, select the second option, where you can specify the number of days that the password remains valid.
- Add the authorized public SSH keys.



Note:

You can also use this page to delete a user account.

8

Use Cockpit to Manage Software Updates

This chapter describes how to use Cockpit to manage manual software updates, as well how to automate software updates. Note that to apply software updates by using either of the following methods, you must meet the same requirements and you must be logged in to the web console.

The Software Updates module in the web console is based on the yum utility.

Manage manual software updates

Follow these steps to manually apply software updates by using the web console:

1. Click **Software Updates**.
The list of available updates refreshes automatically if the last check happened longer than 24 hours.
2. To trigger a refresh, click **Check for Updates**.
3. Apply updates.
 - a. To install all available updates, click **Install all updates**.
 - b. If there are security updates available, you can install them separately by clicking **Install Security Updates**.

You can observe the update log while the update is running.

4. Restart the system.
After the system applies updates, you are presented with the option to restart your system. This step is recommended, especially if the update included a new kernel or system services that you do not want to restart individually.
5. Click **Ignore** to cancel the restart of the system; or, click **Restart Now** to restart the system.
6. After the system restarts, log in to Cockpit, then go to the **Software Updates** page to verify that the update was successful.

Manage automatic software updates

Using the web console, you can choose to apply all updates, security updates, as well as manage the periodicity and timing of automatic updates for your system.

Follow these steps to apply automatic software updates by using the web console:

1. Click **Software Updates**.
2. To automatically apply only security updates, from the drop-down list, click **Apply all updates**, and then select **Apply security updates**.
3. (Optional) To modify the day of the automatic update, from the drop-down list, click **every day**, and then select a specific day, for example, **Monday**.

4. (Optional) To modify the time of the automatic update, from the drop-down list, click the time and then select a specific time, for example, **6:00**.

If you want to disable automatic software updates, switch the **Automatic Updates** button to the disabled position.

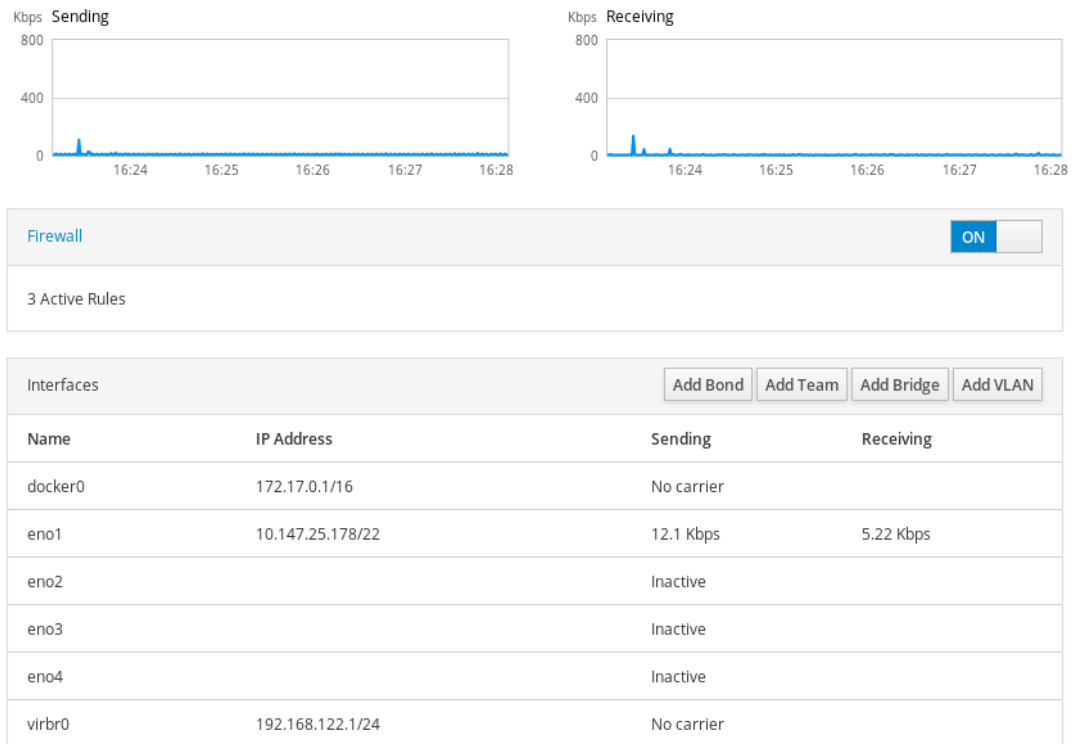
9

Use Cockpit to Manage Network Configuration

This chapter shows you how to configure network interfaces and firewall rules on an Oracle Linux system using the Cockpit web console.

Access the Network configuration page

After you log into Cockpit, the **Overview** page is displayed by default. Click **Networking** on the navigation panel on the left side of the screen to view information about the network.



The Network configuration page displays realtime graphical representations of incoming and outgoing network traffic. The graphs show aggregated traffic across all network interfaces. To see graphs specific to an interface, click on the specific interface in the **Interfaces** section.

The page also has a **Firewall** section. A switch indicates whether or not the `firewalld` service is enabled. The section also indicates the number of firewall rules that are currently configured. Click the **Firewall** heading to view more information or to add or remove services.

The **Interfaces** section consists of a table of configured network interfaces and their corresponding IP addresses in CIDR format as well as their current traffic throughput for send and receive operations. On the right side of the table heading are options for adding different networking features, such as bonding, teams, bridges, and VLANs. Click on any of the

interface names to view more information about the interface or to configure IP addressing or other related settings.

The **Unmanaged Interfaces** section provides a list of network interfaces that have been created outside of the Network Manager. These interfaces are typically virtual devices created by products such as Oracle VM VirtualBox. The interfaces are not listed as active links to indicate that they are not configurable in Cockpit.

Finally, the page lists logs that are specific to the Network Manager service. By default only the 10 most recent log entries are displayed. The list is identical to the output of the following command:

```
sudo journalctl -u NetworkManager -n 10
```

Configure the Firewall

Cockpit provides a rudimentary interface to configure the `firewalld` service. Click the **Firewall** heading to open a configuration page where you can enable or disable the `firewalld` service through the **On/Off** toggle switch.

When the `firewalld` service is enabled, all incoming network traffic is dropped by default. To allow traffic for particular services, click **Add Services**. A dialog opens where you can select different service types, for example, SSH, to add as an allowed service. Note that you can select only from the services listed on the page. You cannot add your own customized services.

All of the enabled services and corresponding ports for specific zones on the system are displayed in table format. Click on any of the listed services to view additional detail about the service, including an option to delete that service.

Note that custom rules for specific ports that are unattached to a predefined service and that have been opened by an alternative utility, such as the `firewall-cmd` tool, are not shown in this table.

Configure IP Addressing for a Network Interface

To configure and enable IP Addressing for a network interface, click on the device name in the **Interfaces** table.

A page opens to display the network traffic graphs for the specific interface. Below the graphs are available details about the interface, such as the device type and the MAC address, if available. An **On/Off** toggle switch enables or disables the network interface device.

In addition, the page displays the following configuration status and options:

- **Status:** Displays configured IP addressing for the device if it is active. Otherwise this row indicates an Inactive status.
- **Carrier:** Indicates the strength of the carrier signal, if detected.
- **General:** Provides a checkbox option to `Connect automatically`, which enables the device at boot and as soon as a carrier signal is available.
- **IPv4:** Provides a link to a page of IPv4 related configuration options. By default, **Automatic (DHCP)** is set and the device is automatically configured by a DHCP server. However, you can select **Manual** if you wish to set a static IP address,

netmask, and gateway for the interface. Other configurable options include adding DNS servers, DNS search domains and configuring static routes.

- **IPv6:** Provides a link to a page of IPv6 related configuration options. By default, **Automatic** is set and the device is automatically configured by the IPv6 auto-configuration mechanism. However, you can select **Automatic (DHCP)** if you prefer to rely on DHCPv6 instead. Or, you can select **Manual** if you want to set a static IP address, netmask, and gateway for the interface. Other configurable options include adding DNS servers, DNS search domains and configuring static routes.
- **MTU:** A link indicating the currently configured MTU setting for the network interface, which is typically **Automatic**. However, you can provide a fixed MTU value to resolve network issues such as lag or disconnection issues caused by inappropriate packet sizes.

Configure a Network Bond

The Network configuration page provides options to add different networking features such as network bonds and bridges. To configure any of these options, click the appropriate button on the **Interfaces** table header row.

The **Add Bond** button opens a dialog where you can create the bond name, select the interfaces to bond together, and the MAC address that the bond should use.

Bond mode options are available and the rest of the configuration options depend on which bond mode you select. Bond mode options include:

- **Active Backup:** A single interface is configured as the primary interface and other interfaces in the bond act as backups if the primary interface fails.
- **Round Robin:** Network traffic is balanced by transmitting in sequential order beginning with the first available interface. If an interface fails, it is skipped in the round-robin selection.
- **XOR:** Network traffic is balanced based on a hash policy derived from interface MAC addresses. This mode ensures that network traffic destined for specific peers always comes from the same physical interface.
- **Broadcast:** All network traffic is sent on all network interfaces. This provides fault tolerance, but no load balancing.
- **802.3ad:** Uses the IEEE 802.3ad dynamic link aggregation policy and requires an 802.3ad capable switch. Traffic is broadcast in aggregation groups to maximize fault tolerance and to provide load balancing functionality.
- **Adaptive transmit load balancing:** Outgoing traffic is balanced across interfaces within the bond based on each interface's current load. Incoming traffic is delivered to the current active interface.
- **Adaptive load balancing:** Similar to dynamic link aggregation, but does not require an 802.3ad capable switch. Outgoing traffic is handled in the same manner as adaptive transmit load balancing. Incoming traffic is balanced based on ARP negotiation.

Select the appropriate bond mode for your requirements. You need to configure a bond link monitor to handle how the bond determines whether an interface is available and to identify the appropriate balancing mechanism to apply, if supported.

- **MII (Recommended):** This default option uses the MII (Media Independent Interface) monitor, which detects carrier signal for each interface using the local device driver or MII registers to determine carrier state. You can set the `Monitoring Interval`, which

determines how often the carrier state is checked in milliseconds; the `Link up delay` to determine how long to wait, in milliseconds, until using an interface that is up; and the `Link down delay` which determines how long to wait before switching to another interface if the interface is marked as down.

- **ARP:** The ARP monitor sends ARP queries to peer systems on the network and uses the response to indicate whether an interface is up. The ARP monitor relies on the device driver to keep track of the last transmit and receive times. If the information is not updated by the device driver, the interface is marked as down. If you select this monitor, you need to configure the **Monitoring Interval**, which determines how often ARP requests are sent, in milliseconds; and the **Monitoring Targets**, a comma-separated list of IP addresses for peers on the network that can be used for ARP monitoring.

Click **Apply** to create the new bond.

Configure a Network Team

The **Add Team** button opens the **Team Settings** window where you can specify the new team's name and select the interfaces to combine into the team.

Other options that you can configure for the network team are the following:

- **Runner:** Runners are load balancing and failover schemes that are implemented on the team. Select from one of the following:
 - **Round Robin:** Transmits packets over the available ports in a round-robin fashion.
 - **Active Backup (Default):** Monitors the link for changes and selects the active port that is used to send packets.
 - **Load Balancing:** In passive mode, uses the BPF hash function to select the port that is used to send packets.
 - **Broadcast:** Sends packets on all member ports.
 - **802.3ad LACP:** Provides load balancing by implementing the Link Aggregation Control Protocol 802.3ad on the member ports.
- **Link Match:** Tool to use to monitor the state of the interfaces that comprise the team. Select from one of the following:
 - **Ethtool (Default):** Uses the ethtool utility.
 - **ARP Ping:** Uses the arp_ping utility and Address Resolution Protocol (ARP).
 - **NSNA Ping:** For IPv6 connections. Uses the Neighbor Advertisement and Neighbor Solicitation features of the IPv6 Neighbor Discovery protocol.
- **Link up delay** and **Link down delay:** Specify time delays in milliseconds. Delays enable the different devices of the team to synchronize. In the case of `Link up delay`, you might want to specify a time between when a device link is reestablished and when the device can actually be used to service network traffic. In this way, switch initialization and other device processes can be completed before these are actually used. In the case of `Link down delay`, some devices and switches might take some time before their backup mode becomes activated. Specifying a delay prevents a failover to immediately occur before those backup devices are ready to be used.

Click **Apply** to create the new team.

Use Cockpit to Manage Physical Drives in Volume Groups

This chapter shows you how to use Cockpit to configure physical drives in volume groups.

For detailed information about LVM, see Working With Logical Volume Manager in [Oracle Linux 8: Managing Storage Devices](#).

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Add physical drives to volume groups

1. If you have not already done so, create a volume group.
2. In the **Volume Groups** box, select the volume group in which you will add a physical volume.
3. In the **Physical Volumes** box, click the **+** icon.
4. In the **Add Disks** dialog box, select the drive and click **Add**.

Remove physical drives from volume groups

1. In the **Physical Volumes** section, locate the drive you want to remove.
2. Click the **-** icon next to that physical drive.

11

Use Cockpit to Manage Partitions

This chapter describes how to use Cockpit to manage file systems.

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Display partitions that are formatted with file systems

The **Storage** section of the web console displays all of the available file systems in the **Filesystems** table. In this section, you can navigate to the list of partitions that are formatted with file systems, and which are displayed in the web console.

1. To display the partitions that are formatted with file systems, click the **Storage** tab.
The **Filesystems** table is displayed.
2. View the following information about all of the available partitions that are formatted with file systems:
 - Partition name
 - Partition size
 - Space that is available on each partition

Create partitions

To create partitions in the web console, you must meet all of the previously mentioned requirements, as well as have an unformatted volume that is connected to the system visible in the **Other Devices** table of the **Storage** tab.

1. In the **Other Devices** table, click the volume in which you want to create the partition.
The **Create partition** dialog box opens.
2. In the **Create partition** dialog box, select the size of the new partition.
3. In the **Erase** drop-down list, select from the following:

- **Don't overwrite existing data:** The web console rewrites only the disk header. Select this option for speedier formatting.
 - **Overwrite existing data with zeros:** The web console rewrites the entire disk with zeros. Note that this option is slower, but more secure, because the program has to go through the whole disk. Select this option if the disk includes any data that needs to be overwritten.
4. Select a file system type from the **Type** drop-down list:
 - **XFS** (selected by default): This file system type supports the following: large logical volumes, switching physical drives online without outage, and growing an existing file system.
 - **ext4:** This file system type supports the following: logical volumens, switching physical drives online without outage, and growing and shrinking a file system.

Note that an option for enabling encryption of the partition through LUKS (Linux Unified Key Setup) is also available. This option enables you to encrypt the volume with a passphrase.
 5. In the **Name** field, type the logical volume name.
 6. In the **Mounting** drop down menu, select **Custom**.

The **Default** option does not ensure that the file system is mounted on the next boot.
 7. In the **Mount Point** field, add the mount path, then select **Mount at Boot**.
 8. Click **Create partition**.

Formatting can take several minutes and is dependent on volume size and the formatting options you selected.
 9. Verify that the partition was successfully added by switching to the **Storage** tab and then checking for the partition in the **Filesystems** table.

Delete partitions

Follow these steps to delete partitions by using the web console:

1. In the **Filesystems** table, select the volume in which you want to delete the partition.
2. In the **Content** section, click on the partition that you want to delete.
3. When the partition information is exposed, click **Delete** to delete the partition.

Note:

To perform this operation, the partition must **not** be currently mounted and used.

4. Verify that the partition was successfully deleted by switching to the **Storage** tab and then checking the **Content** section.

Use Cockpit to Manage Logical Volumes With LVM

This chapter shows you how to use Cockpit to configure volumes that are managed with Logical Volume Management (LVM).

For detailed information about LVM, see Working With Logical Volume Manager in [Oracle Linux 8: Managing Storage Devices](#).

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Create volume groups

1. Click the **+** icon in the **Volume Groups** box.
 - a. If the **Volume Groups** box is not visible, select **Create volume group** in the **Devices** box.
2. In the **Name** field, enter a name for the volume group.
3. Assign drives to your volume group in the **Disks** area.
4. Click **Create**.

Create logical volumes

1. Select the volume group for which you want to create logical volumes.
2. Click **Create new Logical Volume**.
3. In the **Name** field, enter a name for the logical volume.
4. In the **Purpose** drop-down list, select **Block device for filesystems**.
5. Adjust the slider or enter the size of your logical volume, then click **Create**.

Format logical volumes

1. Select the volume group in which you want to format logical volumes, then select the logical volume that you will format.
2. Select the **Unrecognized Data** tab, then click **Format**.
3. In the **Erase** drop-down list, you can decide whether to ignore or overwrite existing data.
4. In the **Type** drop-down list, select your chosen file system, then set a **Name** for your file system.
5. Select the **Custom** mounting option to ensure the file system is mounted on next boot.
6. In the **Mount Point** field, add the directory path, for example, **/mnt/example**.
7. Select **Mount at boot**, then click **Format**.
8. To use the logical volume, click **Mount**.

Resize logical volumes



Note:

You cannot reduce volumes that contain the GFS2 or XFS file systems.

1. Select the volume group in which you want to create logical volumes, then select the logical volume that you will resize.
2. On the **Volume** tab, click **Grow**.
3. In the **Grow Logical Volume** dialog box, adjust the volume space, then click **Grow**.

13

Use Cockpit to Encrypt Block Devices With LUKS

This chapter shows you how to use Cockpit to configure encrypted block devices with the Linux Unified Key Setup (LUKS).

For detailed information about LUKS encryption on block devices, see Using Encrypted Block Devices in [Oracle Linux 8: Managing Storage Devices](#).

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Lock data on a device with LUKS

1. On the right panel, select a device that you want to encrypt.
The device's information page is displayed.
2. Under Content, expand the device's displayed information.
3. Click the menu icon and select **Format**.
4. On the Format page that is displayed, select the **Encrypt data** checkbox.
5. Specify a passphrase and then confirm it as prompted.
6. Configure other encryption options for the device as necessary.
7. Click **Format** to start the operation.

Note the warning that indicates that formatting the device erases all data on that device.

Change the LUKS configuration

The Cockpit console also enables you to change the LUKS passphrase. Do these steps:

1. On the Storage page, select the disk in the Drives table that has encrypted data.

2. Under Content, expand the encrypted partition.
3. Select **Encryption**.
4. In the Keys table and to the right of `Passphrase`, click the edit icon.
5. Follow the prompts to change the passphrase.
6. Click **Save**.

Use Cockpit to Manage Virtual Data Optimizer Volumes

This chapter shows you how to use Cockpit to manage Virtual Data Optimization (VDO) on system volumes.

VDO is a technology that enables you to use your storage efficiently through its compression, deduplication, and thin provisioning features. Thus you are able to allot greater virtual disk space that would otherwise be available from the physical or logical storage.

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Create and configure VDO devices and volumes

1. On the Storage page and next to Devices, click menu icon and then select **Create VDO device**.
2. On the VDO creation page that opens, specify appropriate configuration settings to be applied to the new VDO device.
 - VDO name (no spaces)
 - Disk on which to configure the VDO
 - Logical size
 - Index memory
 - Other options to apply (optional)
3. Click **Create**.

At the end of the process, the new VDO device is added to the VDO Device box.

Format the VDO volume

1. On the Storage page, select the VDO volume.
2. Click the **Unrecognized Data** tab.
3. Click **Format** to display the Format page.
4. From the **Erase** drop down menu, select one of the following:
 - Don't overwrite existing data: Only the disk header is rewritten, and therefore the format operation is faster.
 - Overwrite existing data with zeros: The whole disk is rewritten with zeros and therefore formatting takes longer to finish.
5. From the **Type** drop down menu, select the type of file system:
 - XFS (Default)
 - ext4
6. From the **Mounting** drop down menu, select Custom to ensure that the file system is mounted at the next system boot.
7. Specify the mount point for the file system.
8. Configure other mount options as needed.
9. Click **Format**.
10. After formatting is completed, click the **Filesystem** tab to see detailed information about the VDO volume.
11. Click **Mount** to use the VDO volume.

Extend a VDO volume

1. On the Storage page, select the VDO volume in the **VDO Devices** box.
2. On the page that shows VDO volume details, click **Grow**.
3. On the dialog box that opens, type the new logical size of the VDO volume.
4. Click **Grow**.

At the end of the operation, the new logical size is reflected in the information details about the VDO volume.

Use Cockpit to Manage Redundant Arrays of Independent Disks

This chapter shows you how to use Cockpit to configure redundant arrays of independent disks.

For detailed information about RAID, see Working With Software RAID in [Oracle Linux 8: Managing Storage Devices](#).

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Create RAID storage

1. Click the **+** icon in the **RAID Devices** box.
 - a. If the **RAID Devices** box is not visible, select **Create RAID device** in the **Devices** box.
2. Enter a name for a new RAID device in the **Create RAID Device** dialog.
3. Select your chosen **RAID Level** from the drop-down list.
4. Set the **Chunk Size** from the drop down list. The default value is **512 KiB**.
5. Assign hard disks to your RAID array in the **Disks** area, then click **Create**.

Format RAID storage

1. Select the RAID you want to format in the **RAID Devices** box.
2. In the RAID details screen, scroll down to **Content**.
3. (Optional) Select **Create partition table**, if there are no existing volumes in your RAID:
 - a. Click the **Create Partition Table** button.

- b. In the **Erase** drop-down list, you can decide whether to ignore or overwrite existing data.
 - c. In the **Partitioning** drop-down list, select **GPT** for volumes larger than 2TB, or **MBR** for smaller volumes, then click **Format**.
4. Click the **Format** button next to the RAID you would like to format.
 5. In the **Erase** drop-down list, you can decide whether to ignore or overwrite existing data.
 6. In the **Type** drop-down list, select the file system for the new RAID device, then set a **Name** for your formatted volume.
 7. Select the **Custom** mounting option to ensure the file system is mounted on next boot.
 8. In the **Mount Point** field add the directory path, for example, **/mnt/raid**.
 9. Select **Mount at boot**, then click **Format**.
 10. When the format is complete, click **Mount**.

Create partitions on RAID storage

1. Select the RAID device that you want to format in the **RAID Devices** box.
2. In the RAID details screen, scroll down to **Content**, then select your RAID and click **Create Partition**.
3. In the **Create Partition** dialog, adjust the slider or enter the size of your first partition.
4. In the **Erase** drop-down list, decide whether to ignore or overwrite existing data.
5. In the **Type** drop-down list, select the file system for the new partition, then set a **Name** for your file system.
6. Select the **Custom** mounting option to ensure the file system is mounted on next boot.
7. In the **Mount Point** field, add the directory path, for example, **/mnt/raid/example**.
8. Select **Mount at boot**, then click **Create Partition**.

When the format is complete, you can continue creating more partitions.

Use Cockpit to Enable Network Bound Disk Encryption

This chapter shows you how to use Cockpit to automatically unlock an encrypted storage device through the use of a key from a Tang server. The steps in this chapter are part of a wider task of implementing Policy-Based Decryption (PBD) by configuring Network-Bound Disk Encryption (NBDE) that features Tang and Clevis server and client components. For more information, see [Oracle Linux: Enabling Network-Bound Disk Encryption](#).

For a tutorial in installing and configuring a Tang server, see [Use Network Bound Disk Encryption on Oracle Linux 8](#).

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Create a Tang key for the encrypted device

1. On the Storage page, select an encrypted device in the Devices box.
2. Under Content, expand the encrypted device.
3. Click the **Encryption** tab.
4. Next to the Keys heading, click the + icon to create a Tang key.

The Add Key page is displayed.

5. Provide the following information:
 - **Key source:** Make sure that Tang `keyserver` is selected.
 - **Keyserver address:** Specify the fully qualified domain name (FQDN) or IP address of the Tang server including the port number that the server uses, for example, `tangserver.example.com:7500`.

Note: By default, the Tang server uses port 80. However, you can configure the server to use a different port number. For instructions, see [Oracle Linux: Enabling Network-Bound Disk Encryption](#).

- Disk passphrase: Specify the LUKS passphrase for the encrypted device.

6. Click **Add**.

The Verify key window opens and displays the generated key hash. The window also provides instructions for verifying the key.

7. Verify the hash key by doing the following steps:

- a. On the left navigation panel, select **Terminal** to open a terminal window.
- b. Obtain the key hash that the Tang server provided by typing the following command.

```
sudo curl -s tangserver.example.com:7500/adv | jose fmt -j- -g  
payload -y -o- | jose jwk use -i- -r -u verify -o- | jose jwk  
thp -i-
```

Check that the key hash that is generated matches the key that is displayed in the Verify key window.

8. On the Verify key window, click **Trust key**.
9. Return to the terminal window and enable early boot decryption.

```
sudo dracut -fv --regenerate-all
```

Confirm that the configuration is successful

1. On the Storage page and under the encrypted device's Content page, click the **Encryption** tab.
2. Verify that the Tang server is now included in the Keys list, for example:

```
Keys  
Passphrase                               Slot 0  
Keyserver: tangserver.example.com:7500   Slot 1
```

3. On the left navigation panel, click **Terminal** to open a terminal window.
4. Verify that the bindings are available for early boot, for example:

```
sudo lsinitrd | grep clevis  
clevis  
clevis-pin-sss  
clevis-pin-tang  
clevis-pin-tpm2  
-rwxr-xr-x 1 root root 1600 May 3 16:30 usr/bin/clevis  
-rwxr-xr-x 1 root root 1654 May 3 16:30 usr/bin/clevis-decrypt  
...  
-rwxr-xr-x 2 root root 45 May 3 16:30 usr/lib/dracut/hooks/  
initqueue/settled/60-clevis-hook.sh  
-rwxr-xr-x 1 root root 2257 May 3 16:30 usr/libexec/clevis-  
luks-askpass
```

Use Cockpit to Manage NFS Mounts

This chapter describes how to use Cockpit mount remote directories by using the Network File System (NFS) protocol.

Access the storage information

You must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

After logging in, the Overview page is displayed by default. Click **Storage** in the navigation panel.

The storage page displays information about the storage devices and their configuration on your system or instance. The information includes usage, file system configuration, mount points, and event logs specific to storage. On this page, you can also create new storage configurations, such as new NFS mounts, RAID or VDO devices.

Note: The appearance of the storage page differs depending on your initial system configuration, as well as the range of devices that are connected and visible to the `cockpitd` service.

Connect NFS mounts

NFS enables you to mount remote directories that are located on the network and then work with those files as though the directory was located on your physical system.

To connect NFS mounts by using the web console, click the **Storage** tab.

1. In the **NFS mounts** section, click the plus sign (+) to add a new NFS mount.
The **New NFS Mount** dialog box is displayed.
2. In the **New NFS Mount** dialog box, enter the server or IP address of the remote server.
3. In the **Path on Server** field, type the path to the directory that you want to mount.
4. In the **Local Mount Point** field, type the path to the directory location on your local system.
5. Select **Mount at boot**.
This step ensures that the directory is reachable after restarting the local system.
6. (Optional) If you do not want to change the content, select **Mount read only**.
7. Click **Add** to add a new NFS mount.
8. Verify that the content for the new NFS mount is accessible by opening the mounted directory.

If necessary, you can troubleshoot the connection by customizing the mount options, as described in the next section.

Customize mount options

Customizing mount options entails editing an existing NFS mount and adding custom mount options. Custom mount options are helpful in troubleshooting a connection or changing parameters of the NFS mount, such as timeout limits or configuring authentication. Note that to customize mount options by using the web console, the NFS mount must already be added.

1. Click the **Storage** tab.

Note that you must have previously installed the `cockpit-storaged` package to access the **Storage** tab in the web console.

2. Select the NFS mount that you want to customize. If the remote directory is currently mounted, click **Unmount**.

Note: The directory cannot be currently mounted when customizing mount options; otherwise, the web console does not save the configuration and an error is reported.

3. Click **Edit** to open the **NFS Mount** dialog box, then select the **Custom mount option**.

4. Type the new mount options, separated by a comma:

- `nfsvers=4`: NFS protocol version number.
- `soft`: Type of recovery when an NFS request times out.
- `sec=krb5`: Files on the NFS server can be secured by Kerberos authentication. To specify this customization, both the NFS client and server must support Kerberos authentication.

To display all of the options that are available, run the `man nfs` command.

5. Click **Apply**, then click **Mount**.
6. Verify that the updated content is accessible by opening the mounted directory.

Use Cockpit to Manage Virtual Machines

This chapter shows you how to create and manage KVM-based virtual machines on an Oracle Linux system by using the Cockpit web console.

Install the Cockpit Virtual Machines Module

To allow Cockpit to create and manage virtual machines (VMs), you must install the `cockpit-machines` package.

```
sudo dnf install cockpit-machines
```

Enabling Virtualization

You must enable the virtualization module, install virtualization packages, and ensure that your system is configured to host VMs. The virtualization module provides the `virt-install` package used to install VMs from a CLI and a `virt-viewer` package used to view VMs.

To enable virtualization from within the Cockpit web console, follow these steps:

1. Click **Terminal** in the left panel to open a terminal.
2. Enable the virtualization module:

```
sudo dnf module install virt
```

3. Install the virtualization packages:

```
sudo dnf install virt-install virt-viewer
```

4. Start the `libvirtd` daemon:

```
sudo systemctl start libvirtd.service
```

5. Configure `libvirtd` to start automatically on each boot:

```
sudo systemctl enable libvirtd.service
```

6. Check the status of `libvirt`:

```
sudo systemctl status libvirtd.service
```

7. Verify that your system is prepared to be a virtualization host:

```
virt-host-validate
```

Output similar to the following is displayed:

```
QEMU: Checking for hardware virtualization           : PASS
QEMU: Checking if device /dev/kvm exists             : PASS
QEMU: Checking if device /dev/kvm is accessible      : PASS
QEMU: Checking if device /dev/vhost-net exists       : PASS
QEMU: Checking if device /dev/net/tun exists        : PASS
QEMU: Checking for cgroup 'cpu' controller support   : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup 'blkio' controller support : PASS
QEMU: Checking for device assignment IOMMU support   : WARN
(No ACPI DMAR table found, IOMMU either disabled in BIOS or not
supported by this hardware platform)
QEMU: Checking for secure guest support             : WARN
(Unknown if this platform has Secure Guest support)
```

For more information, see [Installing Virtualization Packages](#).

Review the Virtual Machines Page

Once you have logged into the Cockpit web console and installed the Virtualization Module, review the Virtual Machines page. Click **Virtual Machines** on the navigation panel on the left side of the screen to access the page for creating and managing KVM-based VMs. The Virtual Machines pane lists the VMs for the host and any storage or networks configured.

Check the Storage Pools

You must have a storage pool to be able to create a VM. However, if you do not have a storage pool set up, you have the option of creating a pool now or when you create the VM.

To create a storage pool by using the Cockpit web console, follow these steps:

1. Navigate to the **Virtual Machines** page.
2. Click **Storage Pools**.
3. In the **Storage Pools** page, click **Create Storage Pool**.
4. Enter a unique name for the new storage pool.
5. Select a type and enter the appropriate information.
 - **Filesystem Directory**
Requires path on host's filesystem.
 - **Network File System**
Requires path on host's filesystem, host name, and the directory on the server being exported.
 - **iSCSI Target**
Requires path on host's filesystem, host name, and iSCSI target IQN.

- **Physical Disk Device**
Requires path on host's filesystem, physical device disk on host, and format.
 - **LVM Volume Group**
Requires volume group name.
 - **iSCSI direct Target**
Requires host name, iSCSI target IQN, and iSCSI initiator IQN.
6. Check **Start pool when host boots**.
 7. Click **Create**.
 8. In **Storage Pool** list, click the > next to the newly created pool to see the its details.
 9. Click **Activate**.

Note that you can also create storage pools by using virsh commands from within the Cockpit web console's terminal. You can access the terminal by clicking Terminal in the left navigation pane. As long as you are logged in as a user with Administrative rights, you can run any of the virsh (or other) commands to work with storage pools.

For more information about working with storage pools, see [Working With Storage for KVM Guests](#).

Check the Networks

Before you can create a VM from within the Cockpit web console at least one virtual network must exist. You have the option of using the default network for your VMs or creating a new one.

To create a virtual network by using the Cockpit web console, follow these steps:

1. Navigate to the **Virtual Machines** page.
2. Click **Networks**.
3. In the **Networks** page, click **Create Virtual Network**.
4. Enter a unique name for the new virtual network.
5. Select a forward mode and enter the appropriate information.
 - **NAT**
Select a device or use Automatic, choose an IP configuration, and optionally set the DHCP range.
 - **Open or None (isolated network)**
Choose an IP configuration and optionally set the DHCP range.
6. Click **Create**.
7. In **Networks** list, click the > next to the newly created virtual network to see the its details.
8. Click **Activate**.

For more information, see [Setting Up Networking for KVM Guests](#).

Create a Virtual Machine

1. Navigate to the **Virtual Machines** page.
2. Click **Create VM**.
The **Create New Virtual Machine** window opens.
3. Enter a unique name for the new VM.
4. Select an installation type and enter the appropriate information.
 - **Download an OS**
Choose an operating system.
 - **Local Install Media**
Enter the path to the ISO on the host's filesystem and choose an operating system.
 - **URL**
Enter a remote URL and choose an operating system.
 - **Network Boot (PXE)**
Select an installation source and choose an operating system.
5. For **Storage**, do one of the following:
 - Select **Create a New Volume** and enter the size in MiBs or GiBs.
 - Select **No Storage**.
 - Select an existing storage pool and choose a volume.
6. (Optional) Check **Run unattended installation**.
7. (Optional) Adjust the memory requirements.
8. (Optional) Check **Immediately Start VM**.
9. If you opted to immediately start the VM, the installation process begins using an integrated console, such as VNC. Otherwise, in the **Virtual Machines** list, click the > next to the newly created virtual machine to see the its details.
 - a. If you want to autostart the VM, check **Run when host boots**.
 - b. Click **Install**.
If you did not check **Run unattended installation**, the Oracle Linux installation process begins. Follow the prompts and if you need help see [Oracle Linux 8: Installing Oracle Linux](#).
10. Once the operating system installation completes, ensure that you can log in.
11. From the VM's details page, review its information. You can add or delete a disk, edit or add a network interface, pause, shutdown, restart and more.

Note that you can also create a VM by using the terminal in the Cockpit web console. For more information, see [Creating a New Virtual Machine](#).

Video Demonstration

The video demonstration and tutorial provided at <https://www.youtube.com/watch?v=daHQeCY13s8> may also be useful if you need more information on using Cockpit to create a new virtual machine.

You may also refer to the video demonstration at <https://www.youtube.com/watch?v=-Z3AwP2HPa4> for more information on setting up Cockpit to manage your virtual machines:

Use Cockpit to Manage Podman Containers

This chapter shows you how to manage podman containers on an Oracle Linux system by using the Cockpit web console.

Install the Cockpit Podman Module

To allow Cockpit to pull or download Podman images and to manage Podman containers, you must install the `cockpit-podman` package.

```
sudo dnf install -y cockpit-podman
```

Podman and any other dependencies are installed automatically if they are not on the system already.

Enable Podman Service

Although Podman is not a daemon, a Systemd service is available to provide access to the Podman API so that Cockpit can interact directly with Podman.

If it is not already running, the **Podman Containers** page displays a warning that the Podman service is not active. You can click on the **Start podman** button to start the service. Make sure that the **Automatically start podman on boot checkbox** is checked to restart the service across subsequent boots.

If the Podman Service is running, the **Podman Containers** page displays the containers and images that are currently available on the system. These can be filtered by type or by a matching string.

Managing Podman Images

Images are listed under the **Images** heading.

Images

[Get new image](#)

| Name | Created | Size | Owner | |
|--|---------------------------|---------|--------|--|
| > container-registry.oracle.com/mysql/community-server:latest | 01/20/2021 | 412 MiB | admin | |
| ▼ docker.io/library/nginx:latest | Last Wednesday at 7:20 PM | 131 MiB | system | |
| <div>Details Used By</div> <div><div>ID 35c43ace9216</div><div>Tags docker.io/library/nginx:latest</div><div>Entrypoint</div><div>Command nginx -g "daemon off;"</div><div>Created Last Wednesday at 7:20 PM</div><div>Author</div><div>Ports 80/tcp</div></div> <div></div> | | | | |
| > docker.io/library/oraclelinux:8-slim | 02/01/2021 | 109 MiB | system | |

You can use Cockpit to pull any images that you want to use by clicking on the **Get new image** button.

A search dialog is presented to allow you to search any configured registries for matching images. You can select the owner of the image to either download the image as a `system` user or as the currently logged in user. Images and containers that are owned by the `system` user are run in root mode. The search dialog also includes an optional drop down selector that you can use to limit your search to a particular registry. In the search field, type the name of an image that you are interested in using. For example, you can type `oraclelinux`. The search area populates automatically with images that you can pull from any of the configured registries. Select an image and registry, for example you could select the `docker.io/library/oraclelinux` image. In the tag field, type the tag of the image that you wish to download. For example, you can type `8-slim`. Click on the **Download** button to pull the image.

Each image that is available on the host system is listed to provide details including the image name, image creation date, image size and image owner. Additional information about each image is available by clicking on the image in the listing. The dropdown display includes information such as the entry point, runtime command and exposed ports and also provides an option to delete an image from the system. A play icon on the right of each image in the listing allows you to run the image within a new container. When you click on the icon to run an image, a dialog is displayed to allow you to configure runtime options.

Run Image ✕

Image `docker.io/library/nginx:latest`

Name

Command

Memory Limit ☐ ⬆ ⬇ ⬆
 ▼

CPU Shares ☐ ⬆ ⬇ ⬆

With terminal ☒

Ports ⬆ ⬇ ⬆ ⬆ ⬇ ⬆ ✕ +
 ▼

Volumes ▼ +
 ▼

Environment +

Container image runtime options include:

1. **Name:** the name of the container when it runs. This value is automatically populated, but you can change the value if you wish.
2. **Command:** the command that should be run when the container starts. This value is automatically populated, but you can change the value if you wish. The command must be facilitated within the container image. If the command cannot run within the image, the container fails to run.
3. **Memory limit:** an optional setting that allows you to restrict memory availability for a container.
4. **CPU shares:** an option setting that allows you to restrict the number of CPU shares that are available to the container.
5. **With terminal:** enables an interactive terminal for the container. If the runtime command does not facilitate terminal access, this option may not be particularly useful.

6. **Ports:** allows you to configure port mappings between the container and the host system. This can be useful to expose services running inside a container to the host or to other systems in your infrastructure.
7. **Volumes:** allows you to configure volume mappings to share file system space on the host system with the container and to determine access modes for these volume shares.
8. **Environment:** allows you to specify environment variables as key and value entries. These are commonly used by container images to set configuration options for container-based applications.

Managing Podman Containers

Stopped and running containers are listed under the **Containers** heading.

Containers

| Name | Image | Command | CPU | Memory | Owner | State |
|---|--------------------------------|--|--|--------------------|--------|---------|
| <div><div>▼</div><div>xenodochial_mayer</div></div> | docker.io/library/nginx:latest | /docker-entrypoint.sh nginx -g "daemon off;" | 0% | 0.00232 / 7.49 GiB | system | running |
| <div><div>Details</div><div>Logs</div><div>Console</div></div> | | | <div><div></div><div>Commit</div><div>Restart</div><div>▼</div><div>Stop</div><div>▼</div></div> | | | |
| ID 7d59b339ad2719b965ea42ae5e1c11cdc5ba2a4330928b3f27def8e1f980b6 | | | | | | |
| Created Today at 4:03 AM | | | | | | |
| Image docker.io/library/nginx:latest | | | | | | |
| Command /docker-entrypoint.sh nginx -g "daemon off;" | | | | | | |
| State Up since Today at 4:03 AM | | | | | | |
| Ports 0.0.0.0:8080 → 80/tcp | | | | | | |

Each listed container provides the runtime container name, the image used to instantiate the container, the runtime command, CPU and memory usage, the owner of the container and the container runtime state. You can click on each container to view more information and to access container management controls.

Each container information dropdown displays tabs to allow you to view container information, container logs and to access a web-based terminal into the container. Controls are also provided to delete or destroy the container; commit changes in the container as a new image; restart or force restart the container; and stop or force stop the container. If you opt to commit changes, a dialog prompts you for the image name and tag, runtime command and other variables that should be used when the container is saved as an image.