

How to Install Cockpit on Rocky Linux 9

Learn how to install and configure Cockpit web console on Rocky Linux 9 for easy server management. Step-by-step guide with commands and screenshots.

Contents

01	Introduction	3
02	Prerequisites	3
03	Install Cockpit	3
04	Secure Cockpit with Trusted SSL Certificates	5
05	Access Cockpit	9
06	Manage a Rocky Linux 9 Server Using Cockpit	10
07	Install Cockpit-Podman to Deploy Containerized Applications	14
08	Conclusion	17

Introduction

Cockpit is an open-source web-based control panel used to perform system administration tasks, including service management, monitoring applications, devices, and resources on Linux. Cockpit supports multiple addon modules and applications that extend its functionality to run or manage multiple system components.

This article explains how to install Cockpit on Rocky Linux 9 and manage the server.

Prerequisites

Before you begin, you need to:

- Have access to a [Rocky Linux 9 instance](#).
- Create a [domain A record pointing to the instance](#) such as `cockpit.example.com`.

Install Cockpit

Cockpit is installed but not active on most Rocky Linux 9 instances by default. You can install Cockpit using the DNF package manager if it's not available and activate it using systemd. Follow the steps below to install Cockpit on Rocky Linux 9.

1. Update the server's package index.

CONSOLE

```
$ sudo dnf update
```

2. Install Cockpit.

CONSOLE

```
$ sudo dnf install cockpit -y
```

3. Enable Cockpit to automatically start at boot.

CONSOLE

```
$ sudo systemctl enable cockpit.socket
```

Output:

```
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket → /usr/lib/systemd/system/cockpit.socket.
```

4. Start the Cockpit system service.

CONSOLE

```
$ sudo systemctl start cockpit
```

5. Verify that the Cockpit service is active and running.

CONSOLE

```
$ sudo systemctl status cockpit
```

Output:

```
● cockpit.service - Cockpit Web Service
   Loaded: loaded (/usr/lib/systemd/system/cockpit.service; static)
   Active: active (running) since Wed 2025-02-12 10:11:04 UTC; 4s ago
     TriggeredBy: ● cockpit.socket
       Docs: man:cockpit-ws(8)
    Process: 62303 ExecStartPre=/usr/libexec/cockpit-certificate-ensure --for-cockpit-tls (code=exited, status=0/SUCCESS)
   Main PID: 62322 (cockpit-tls)
      Tasks: 1 (limit: 4424)
```

```
Memory: 1.9M
CPU: 1.328s
CGroup: /system.slice/cockpit.service
└─62322 /usr/libexec/cockpit-tls
```

```
Feb 12 10:11:03 cockpit-test systemd[1]: Starting Cockpit Web Service...
```

```
Feb 12 10:11:04 cockpit-test systemd[1]: Started Cockpit Web Service.
```

6. Create a dedicated non-root user such as `cockpit-admin` to manage Cockpit.

CONSOLE

```
$ sudo adduser cockpit-admin
```

7. Assign the user a strong password.

CONSOLE

```
$ sudo passwd cockpit-admin
```

8. Add the user to the `wheel` sudo users group to perform administrative tasks using Cockpit.

CONSOLE

```
$ sudo usermod -aG wheel cockpit-admin
```

Secure Cockpit with Trusted SSL Certificates

Cockpit uses the TCP port `9090` by default to handle incoming network connections without end-to-end encryption using plain HTTP. Encrypting connections to the Cockpit port enables secure data transfer and authentication using HTTPS on all client web browsers. Follow the steps below to generate trusted SSL certificates using your domain to encrypt connections to the Cockpit control panel interface.

1. Confirm the Firewall status and verify that it's active on your server.

CONSOLE

```
$ sudo systemctl status firewalld
```

- Install Firewalld and allow SSH connections if its not available.

CONSOLE

```
$ sudo dnf install firewalld -y && sudo systemctl start firewalld && sudo firewall-cmd --permanent --add-service=ssh
```

2. Temporarily allow HTTP network connections to enable Let's Encrypt validations.

CONSOLE

```
$ sudo firewall-cmd --add-service=http --permanent
```

3. Restart Firewalld to apply the firewall changes.

CONSOLE

```
$ sudo firewall-cmd --reload
```

4. Enable the EPEL repository.

CONSOLE

```
$ sudo dnf install epel-release -y
```

5. Install the Certbot Let's Encrypt client.

CONSOLE

```
$ sudo dnf install certbot -y
```

- Request for a new SSL certificate from Let's Encrypt using your domain. Replace `cockpit.example.com` with your domain and `admin@example.com` with your email.

CONSOLE

```
$ sudo certbot certonly --standalone -d cockpit.example.com -m admin@example.com --agree-tos
```

Your output should appear like the one below when the certificate is generated.

```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/cockpit.example.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/cockpit.example.com/privkey.pem
This certificate expires on 2025-05-13.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in
the background.

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
```

- Test the Certbot automatic renewal process.

CONSOLE

```
$ sudo certbot renew --dry-run
```

- Link the `fullchain.pem` SSL certificate to the `/etc/cockpit/ws-certs.d` Cockpit directory.

CONSOLE

```
$ sudo ln -sf /etc/letsencrypt/live/cockpit.example.com/fullchain.pem /etc/cockpit/ws-certs.d/certificate.cert
```

The above command links the SSL certificate file to the Cockpit SSL certificates directory as `cockpit.cert`. Linking the certificate ensures that it's auto-renewed upon expiry.

9. Link the `privkey.pem` private key file to the `/etc/cockpit/ws-certs.d` Cockpit directory.

CONSOLE

```
$ sudo ln -sf /etc/letsencrypt/live/cockpit.example.com/privkey.pem /etc/cockpit/ws-certs.d/certificate.key
```

10. Restart Cockpit to apply the SSL configuration changes.

CONSOLE

```
$ sudo systemctl restart cockpit
```

11. Allow network connections to the Cockpit port `9090` through the firewall.

CONSOLE

```
$ sudo firewall-cmd --permanent --add-port=9090/tcp
```

12. Restart Firewalld to apply the firewall configuration changes.

CONSOLE

```
$ sudo firewall-cmd --reload
```

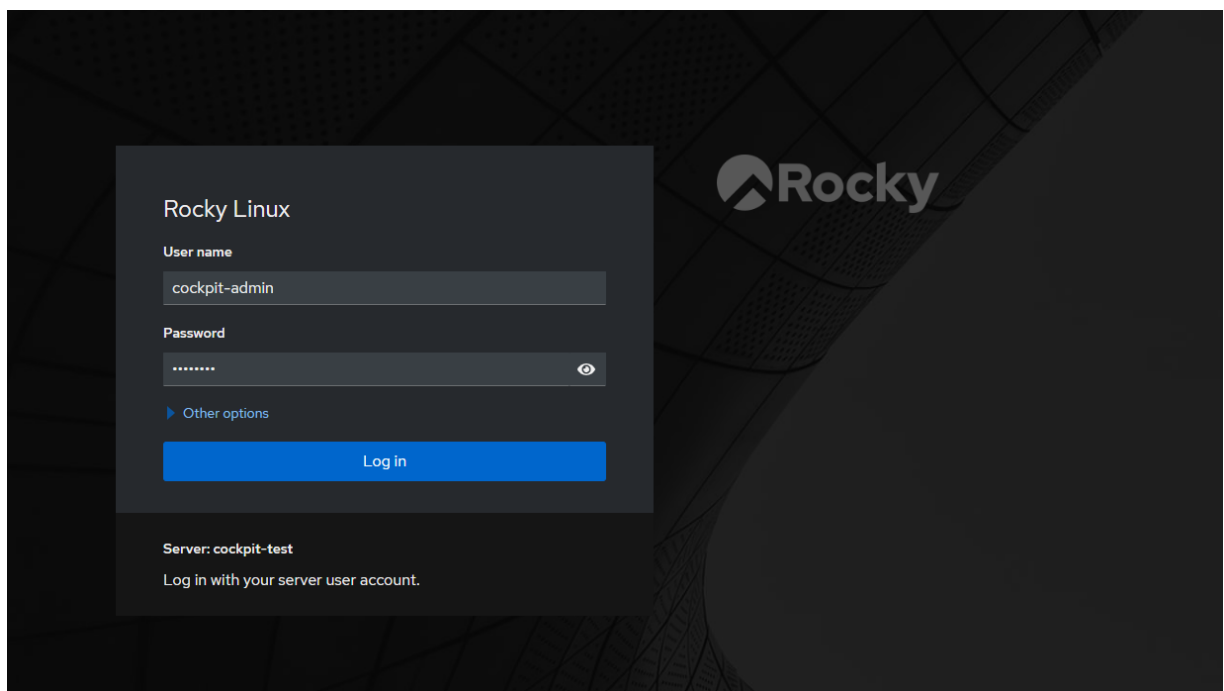
Access Cockpit

You can access the Cockpit web console using its default port `9090`. To define a different port, modify the `cockpit.conf` file with a new port directive. Follow the steps below to access Cockpit and manage your Rocky Linux 9 server.

1. Access port `9090` using your domain in a new web browser window to open the Cockpit login page.

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

2. Log in to the Cockpit web console using the `cockpit-admin` non-root sudo user details you created earlier.

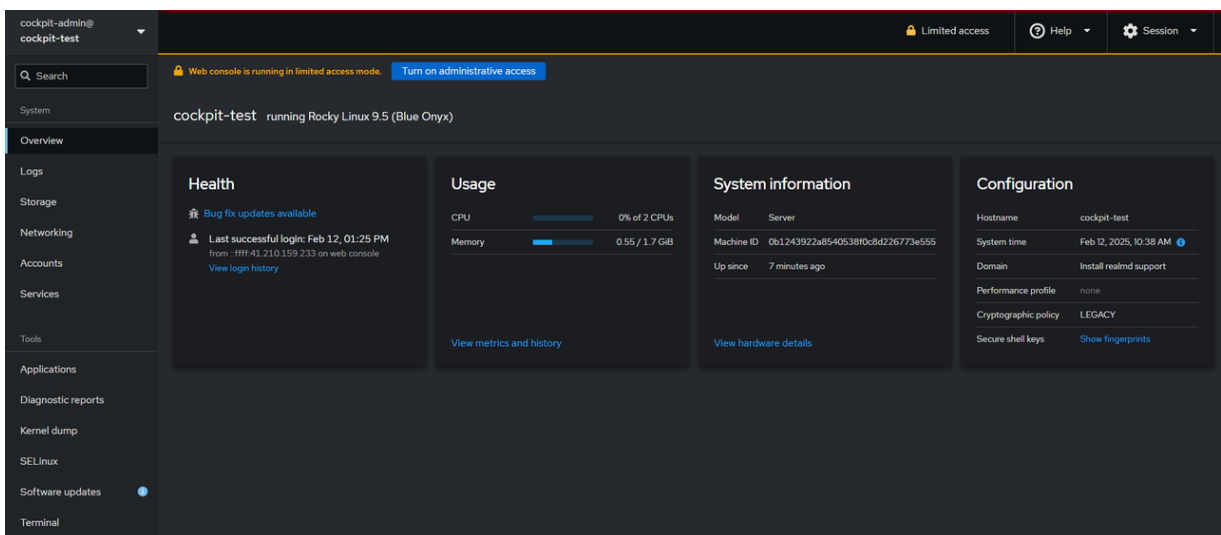


The `root` user login is disabled, and you can only use non-root users to access Cockpit. Sudo users can perform administrative tasks using Cockpit, while regular non-root users can view active processes, system information, and resources.

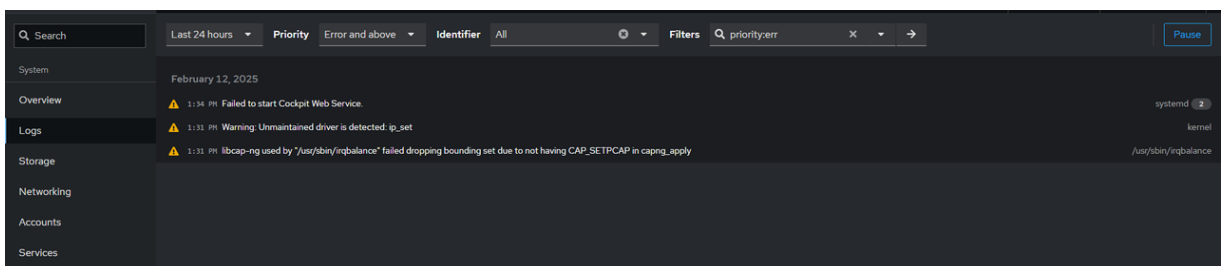
Manage a Rocky Linux 9 Server Using Cockpit

You can manage a Rocky Linux 9 server using Cockpit to install packages, monitor system processes, manage users, and enable or disable specific features. Follow the steps below to manage your Rocky Linux 9 server using the Cockpit web console.

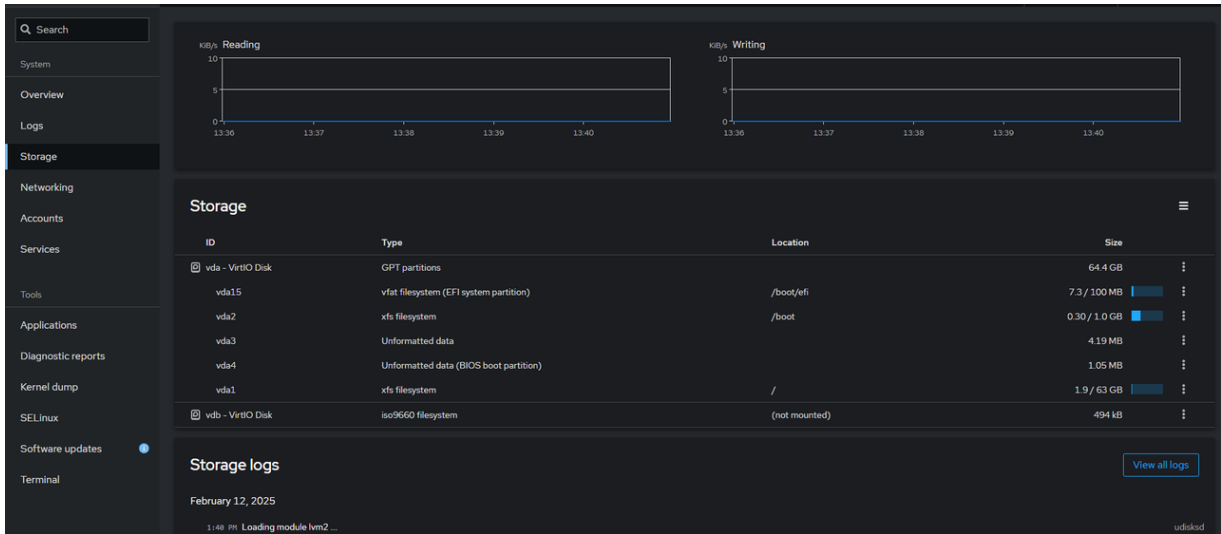
1. Click **Turn on administrative access** within the Cockpit interface and enter your sudo user password to enable administrative privileges.



2. Monitor the server status and basic information within the **Overview** tab, including **Health**, **Usage**, **System information**, and **Configuration**.
3. Click **Logs** on the main navigation menu to monitor the server logs. Click **last 24 hours** to change the duration, **Priority** to set the log type, **Identifier**, and **Filters** to modify your log results.



4. Click **Storage** to manage the server's storage devices, view the status and read-write information.

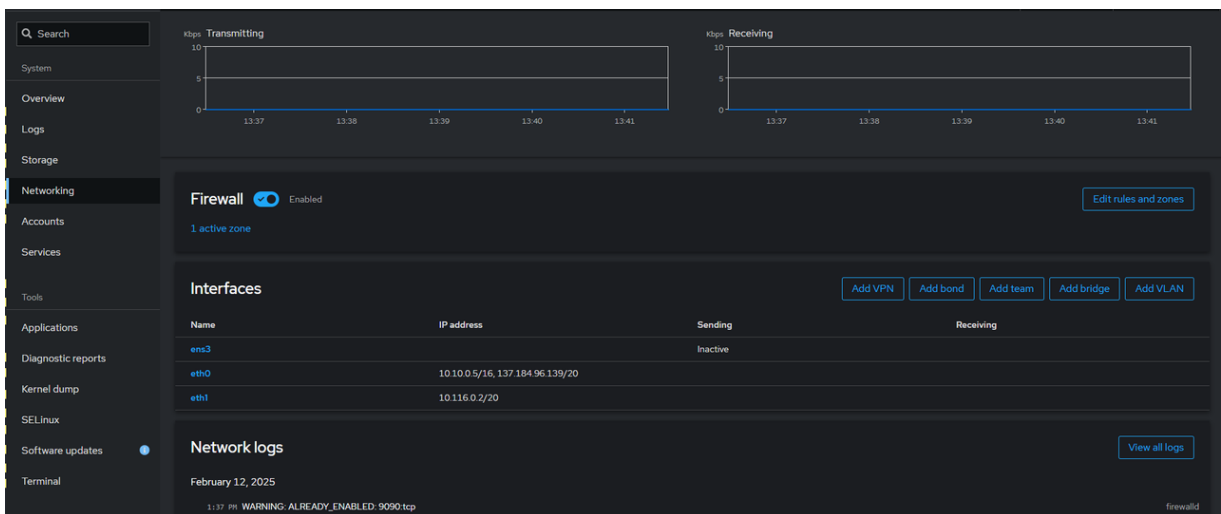


The screenshot displays the Cockpit Storage interface. At the top, there are two line graphs: 'kB/s Reading' and 'kB/s Writing', both showing zero activity over a time period from 13:36 to 13:40. Below the graphs is a table of storage devices:

ID	Type	Location	Size
vda - VirtIO Disk	GPT partitions		64.4 GB
vda15	vfat filesystem (EFI system partition)	/boot/efi	7.3 / 100 MB
vda2	xfs filesystem	/boot	0.30 / 1.0 GB
vda3	Unformatted data		4.19 MB
vda4	Unformatted data (BIOS boot partition)		1.05 MB
vda1	xfs filesystem	/	1.9 / 63 GB
vdb - VirtIO Disk	iso9660 filesystem	(not mounted)	494 kB

Below the table is a 'Storage logs' section with a 'View all logs' button. The terminal at the bottom shows the command '1:48 [m] Loading module lvm2 ...'.

5. Click **Networking** to monitor the incoming and outgoing network traffic statistics. Monitor your server's network interfaces within the **Interfaces** section and view all traffic logs within the **Network logs** section.

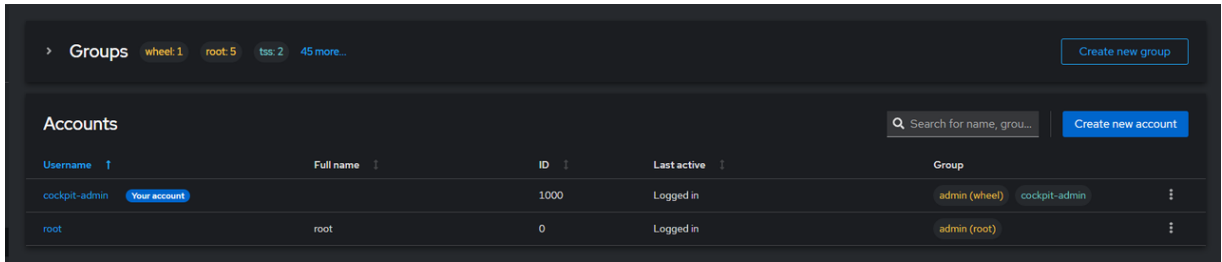


The screenshot displays the Cockpit Networking interface. At the top, there are two line graphs: 'Kbps Transmitting' and 'Kbps Receiving', both showing zero activity over a time period from 13:37 to 13:41. Below the graphs is a 'Firewall' section with a status of 'Enabled' and a button 'Edit rules and zones'. Below that is an 'Interfaces' section with buttons 'Add VPN', 'Add bond', 'Add team', 'Add bridge', and 'Add VLAN'. A table lists network interfaces:

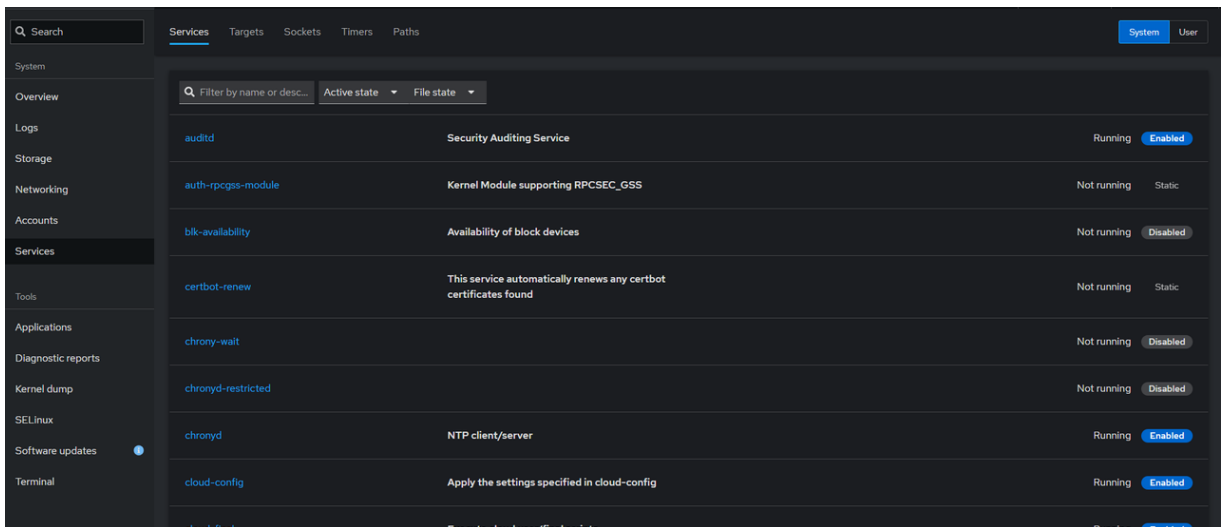
Name	IP address	Sending	Receiving
ens3		Inactive	
eth0	10.10.0.5/16, 137.184.96.139/20		
eth1	10.116.0.2/20		

Below the table is a 'Network logs' section with a 'View all logs' button. The terminal at the bottom shows the command '1:17 [m] WARNING: ALREADY_ENABLED: 3090 tcp'.

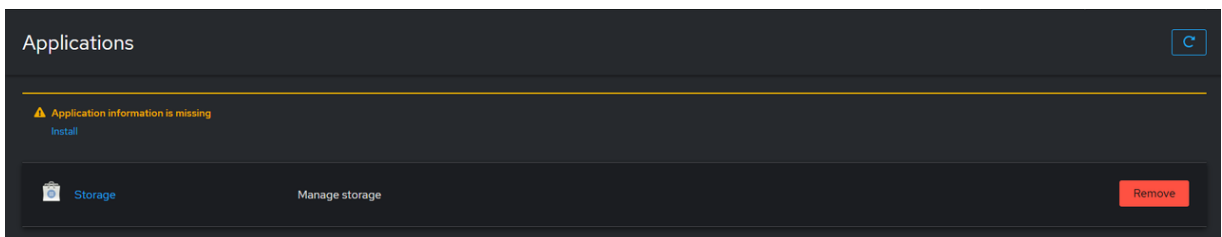
6. Click **Accounts** to manage user accounts and groups. Click **Create new group** to add a new group on your Rocky Linux 9 server and click **Create new account** to enter a new user's details, then click **Create** a non-root user account.



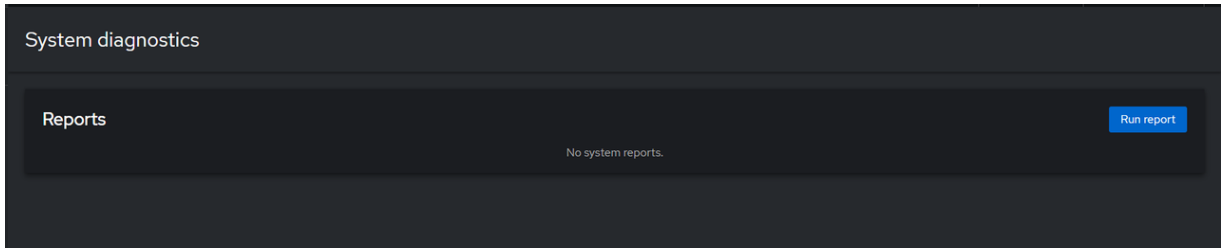
7. Click **Services** to manage all system and user services. Click the **Targets, Sockets, Timers,** and **Paths** tabs to filter the additional service types.



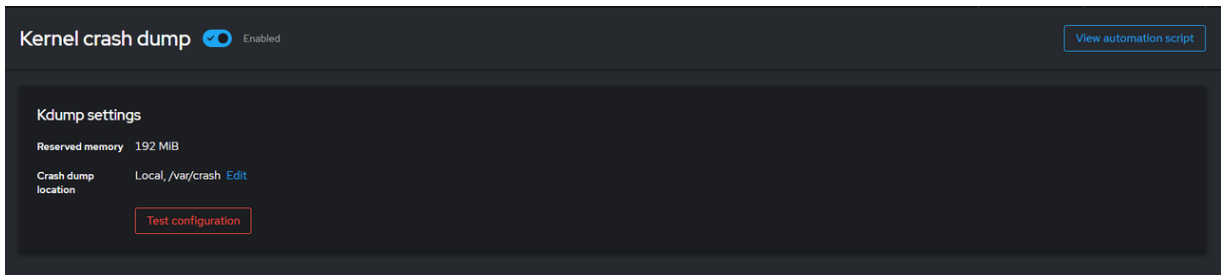
8. Click **Applications** to manage the Cockpit addon applications on your server. Click **Install** if you receive an `application information is missing` prompt to update the application information.



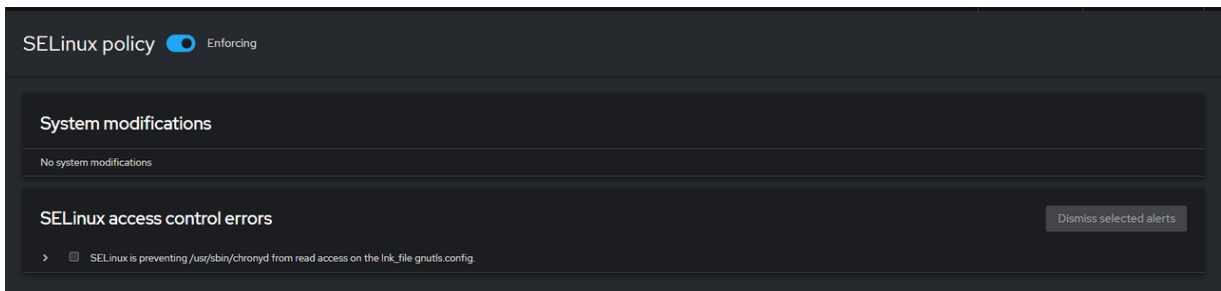
9. Click **Diagnostic reports** to manage the system reports and click **Run report** to generate an SOS report to diagnose problems on your Rocky Linux 9 server.



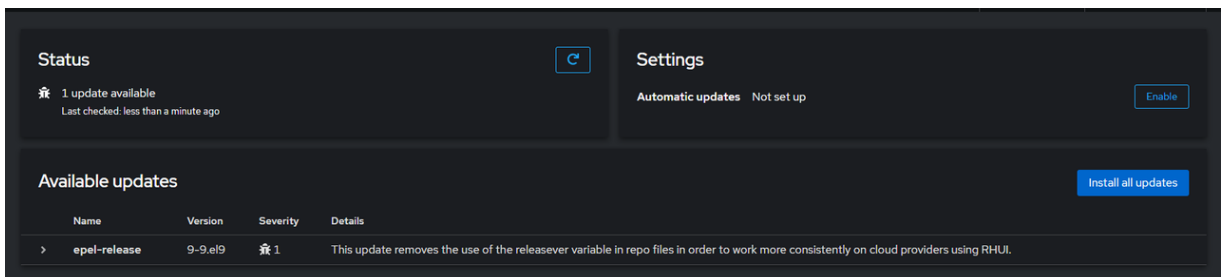
10. Click **Kernel dump** to manage the `kdump` system service to recover your Kernel configurations in case of errors.



11. Click **SELinux** to manage the SELinux policy and monitor any access control errors on your system.



12. Click **Software updates** to monitor all install applications, update, restart, and update packages. Monitor your general packages status within the **Status** tab and click **Restart services** to restart installed packages. Monitor the **Available updates** section and click **Install all updates** or **Install security updates** to update your Rocky Linux 9 server.



13. Click **Terminal** to open a new terminal session and manage your server using Cockpit. Click **Font size** and **Appearance** to modify the terminal display.

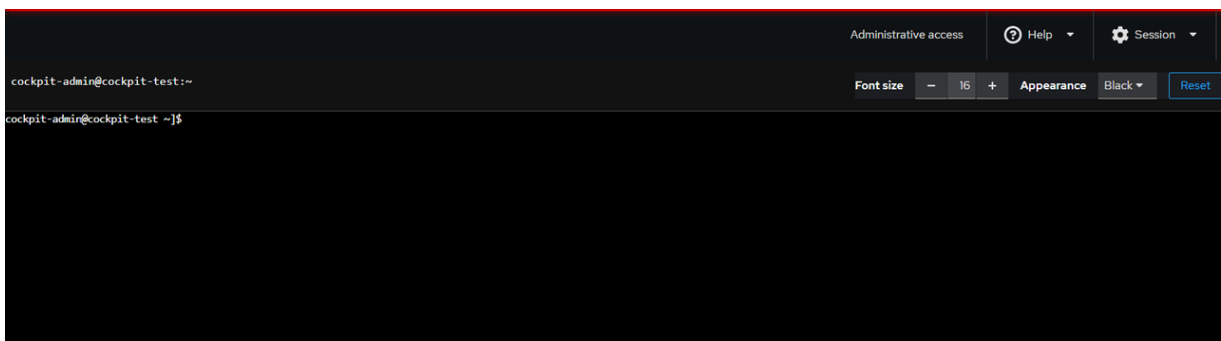
Note

Continuously monitor the Cockpit interface for error and warning icons to manage the specific component on your Rocky Linux 9 server.

Install Cockpit-Podman to Deploy Containerized Applications

Cockpit-Podman is an addon application for Cockpit that allows you to deploy containerized applications on your Rocky Linux 9 server. Follow the steps below to install the Cockpit-Podman application in your terminal session.

1. Click **Terminal** to open a new session.



2. Update the server.

CONSOLE

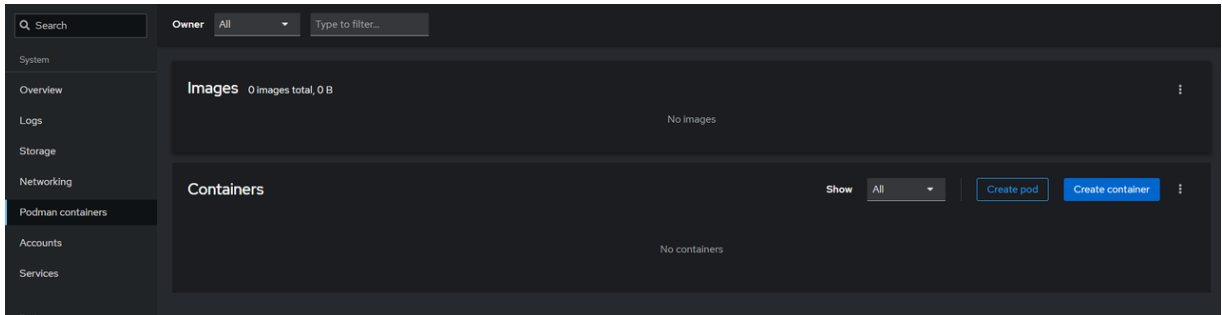
```
$ sudo dnf update
```

3. Install the Cockpit module.

CONSOLE

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

4. Refresh the Cockpit interface to apply the Cockpit-Podman changes.
5. Click **Podman Containers** to manage container images and containers on your Rocky Linux 9 server.
6. Click **Start Podman** to start the Podman service on your server.
7. Click **Create Container** to open the container setup page.

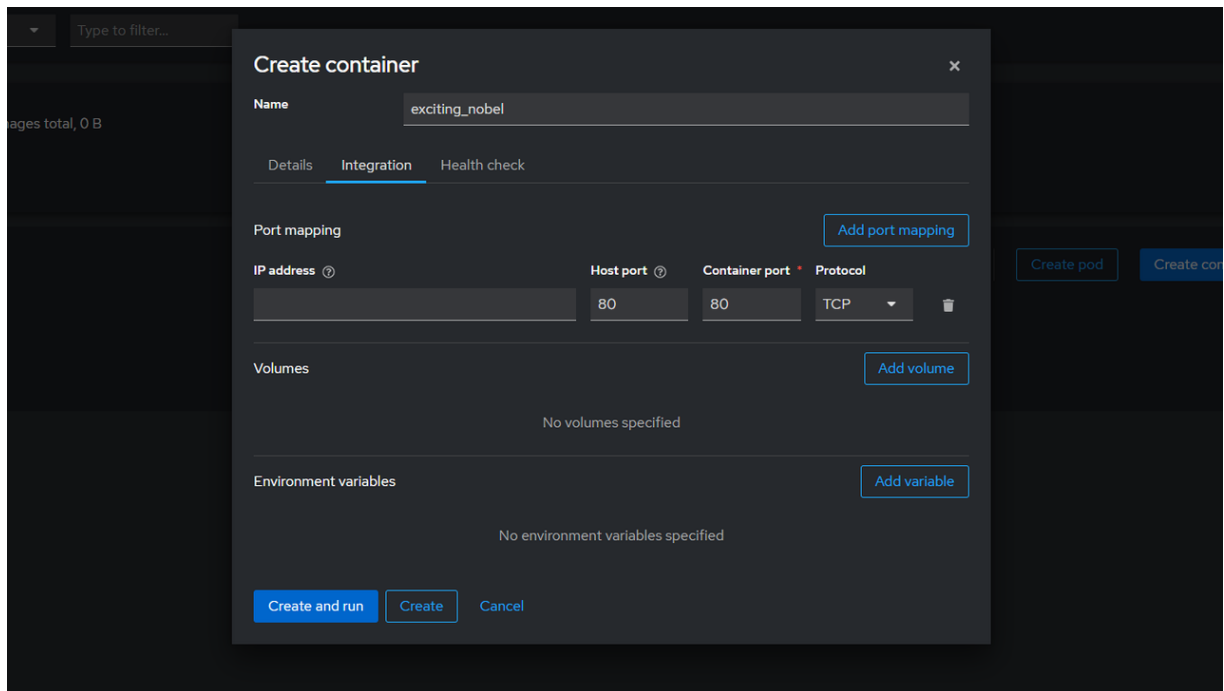


8. Click **Image** within details to search for your target container image in all available registries. For example, enter `docker.io/nginx` and select the official build image for Nginx.
9. Enter the following command to replace existing contents in the **Command** field.

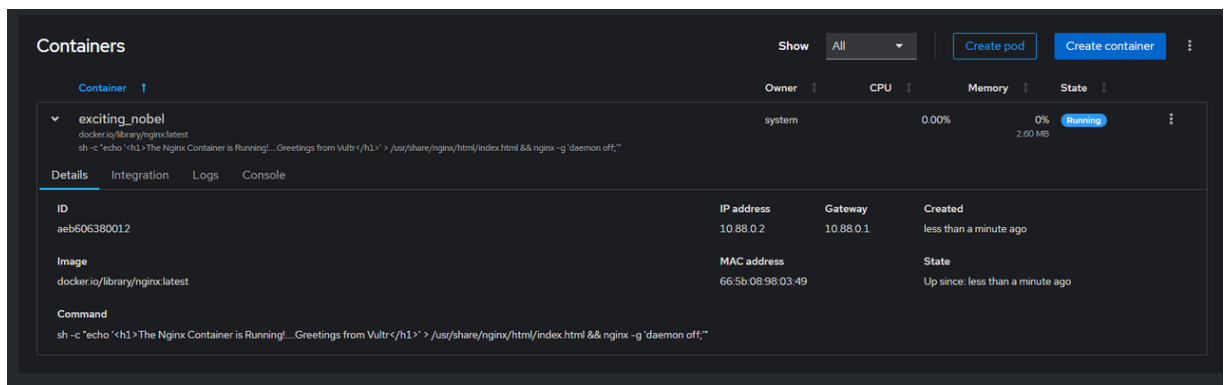
CONSOLE

```
sh -c "echo '<h1>The Nginx Container is Running!...Greetings from Vultr</h1>' > /usr/share/nginx/html/index.html && nginx -g 'daemon off;'"
```

10. Click **Integration**, and select **Add port mapping** to map the container ports to the host. Maintain **IP address** as an empty field, enter `80` in the **Host port** and **Container port** fields, respectively.



11. Click **Create and run** to deploy the container application using Podman on your server.
12. Monitor the container deployment within **Containers** and click the container to manage it.



13. Access your Rocky Linux 9 server's IP address in a new web browser window and confirm that your custom web application page displays well.

The Nginx Container is Running!....Greetings from Vultr

Conclusion

You have successfully installed Cockpit on your Rocky Linux 9 server and installed Cockpit-Podman as an additional module to manage the server. Cockpit is a powerful web-based control panel you can use to manage all aspects of your server. For more details, visit the [Cockpit documentation](#).



VULTR

