Red Hat Enterprise Linux 7
Getting Started with Cockpit

Getting Started with Cockpit

Red Hat Enterprise Linux Documentation
Team

Red Hat Enterprise Linux 7 Getting Started with Cockpit

Getting Started with Cockpit

Legal Notice

Abstract

This guide demonstrates how to use the Cockpit web-based interface to manage Red Hat Enterprise Linux and Red Hat Enterprise Linux Atomic Host servers.

Table of Contents

CHAPTER 1. OVERVIEW

Cockpit is a user-friendly web-based interface for administering servers. It allows monitoring system resources and adjusting configuration with ease.

1.1. WHAT MAKES COCKPIT UNIQUE?

- Cockpit builds upon existing functionality.
- There is no lock-in. Feel free to use other tools alongside Cockpit. Switch back and forth with ease.
- Cockpit does not need special infrastructure or configuration. Once installed, it is ready to use.
- When not in use, Cockpit uses no memory or CPU on the server.
- Cockpit always updates its data to reflect the current state of the server, within seconds.
- Cockpit stores no data or policy. People keep their system-wide permissions and use the system credentials.
- Optionally take advantage of single sign-on with Kerberos.
- Cockpit itself is not used for configuration management. However, Cockpit can interact with configuration management and custom server tools.

1.2. THE ROLE OF THIS GUIDE

This document helps you get started with Cockpit. It walks through installation, explains typical server configuration, and demonstrates the Cockpit interface in detail.

CHAPTER 2. INSTALLING AND ENABLING COCKPIT

A primary Cockpit server is the machine that runs a Cockpit service with the user interface. A secondary server is a machine that is administered using Cockpit. It is possible to add one or more secondary hosts to the primary server.

Setting up a primary Cockpit server involves:

1. Installing the *cockpit* packages.
2. Opening the port for Cockpit.
3. Starting the cockpit service.

After setting up, you can connect to Cockpit in a browser by typing the hostname and port of the server. For example, from the primary host you can connect using **`localhost:9090`**.

For setting up a primary server on Red Hat Enterprise Linux Atomic Host, see Installing Cockpit on Atomic Host.

2.1. PREREQUISITES FOR A COCKPIT SERVER

Before setting up Cockpit, ensure that you have:

1. Installed Red Hat Enterprise Linux. If required, see the Installation Guide.
2. Enabled networking. If required, see the Networking Guide.
3. Registered the system and attached subscription. If required, see the Registering the System and Attaching Subscriptions section of the *System Administrator's Guide*.

2.2. SETTING UP THE PRIMARY COCKPIT SERVER

To install and enable Cockpit:

1. Enable the Extras and Optional repositories:

   ```
   # subscription-manager repos --enable=rhel-7-server-extras-rpms
   # subscription-manager repos --enable=rhel-7-server-optional-rpms
   ```

   This gives you access to supplementary Cockpit packages such as *cockpit-dashboard*.

2. Install the *cockpit* and *cockpit-dashboard* packages:

   ```
   $ sudo yum install cockpit cockpit-dashboard
   ```

   The *cockpit-dashboard* package provides the "Dashboard" tab in the interface. This package is optional, but is assumed to be installed in this guide.

3. Allow external connections to port 9090 through the firewall:

   ```
   #  firewall-cmd --add-port=9090/tcp
   #  firewall-cmd --permanent --add-port=9090/tcp
   ```

4. Enable and start the **`cockpit.socket`** service:

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

5. Cockpit is now installed and running.

If you are installing Cockpit on a Red Hat Enterprise Linux Atomic Host system, see Installing Cockpit on Atomic Host.

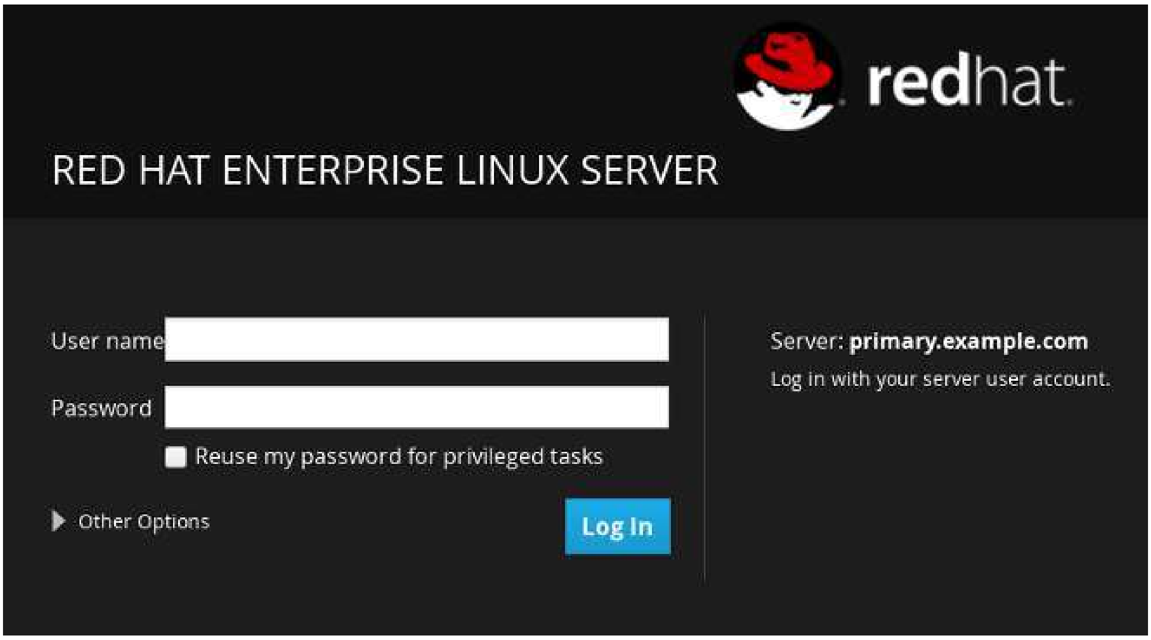
2.3. OPENING THE INTERFACE

1. Open a web browser and enter the server’s IP address with port 9090 in the address bar. If the web browser is on the Cockpit server, open **`localhost:9090`** or ***`hostname`*`:9090`**.

Note

If you use a self-signed certificate, the browser issues a warning. Carefully check the certificate before accepting the warning. Consider using a certificate signed by a certificate authority (CA). For more information on certificates, see the An Overview of Certificates and Security section of the RHEL System Administrator’s Guide.

If you are sure you want to use self-signed certificates, then add this connection to the security exceptions. Click **Advanced** → **Add Exception** → **Confirm Security Exception**. After that, you will see the login screen.


redhat
RED HAT ENTERPRISE LINUX SERVER
User name
Password
Reuse my password for privileged tasks
Other Options
Log In

Server: **primary.example.com**

Log in with your server user account.

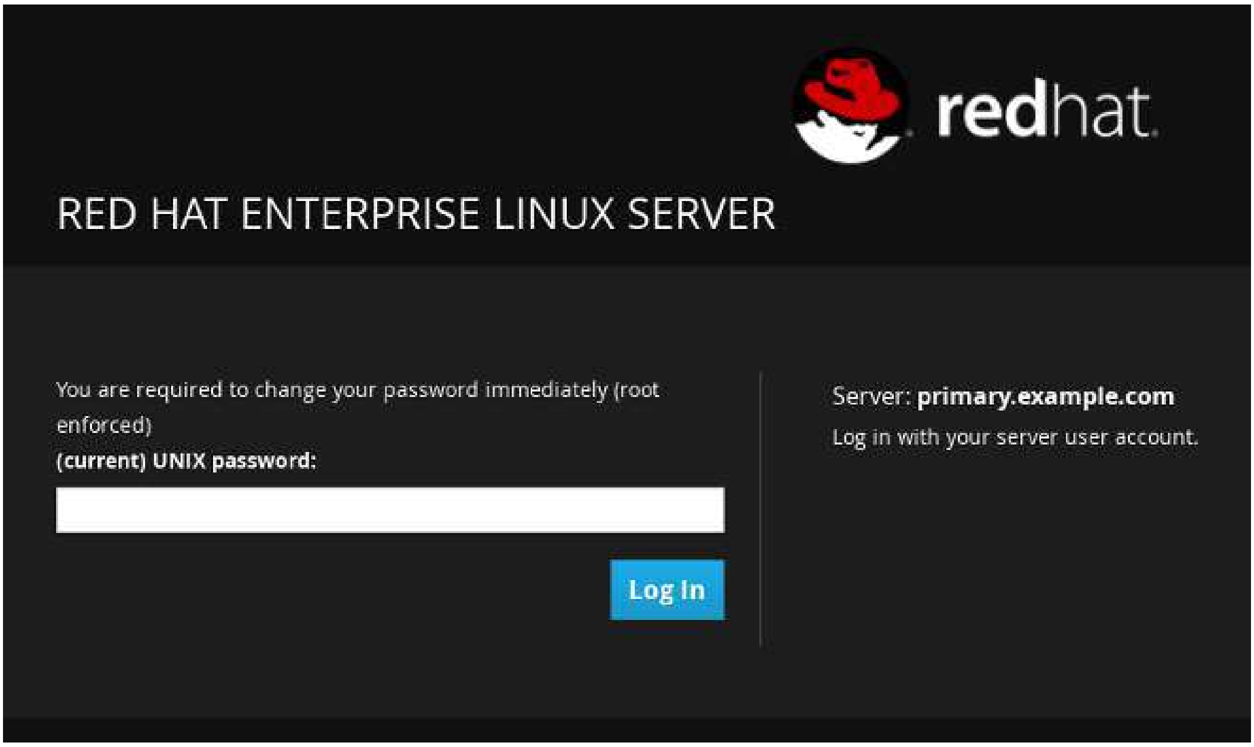
2. Log into the Cockpit interface with the same user name and password that you would normally use to log into the system.

2.4. CHANGING EXPIRED PASSWORDS

Cockpit supports changing expired passwords.

A fresh system installation with an expired password will prompt a password change during the first login. System administrators often use this feature to make sure users change their pre-assigned passwords to a custom password.

When logging in with an expired password, Cockpit prompts you to enter the current password a second time. Enter your current password and click **Log In**.



RED HAT ENTERPRISE LINUX SERVER

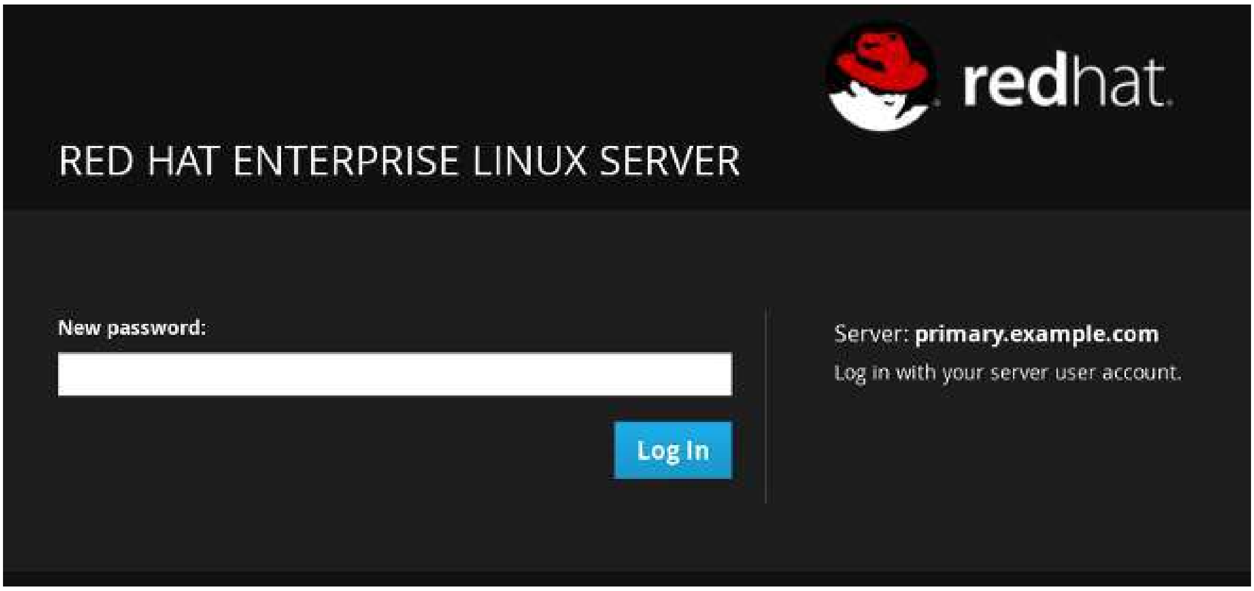
You are required to change your password immediately (root enforced)

(current) UNIX password:

Server: primary.example.com

Log in with your server user account.

Choose a new password and click **Login**.



RED HAT ENTERPRISE LINUX SERVER

New password:

Server: primary.example.com

Log in with your server user account.

Note

If you have issues logging in to Cockpit and the prompt for changing the password is not shown, check the **/etc/ssh/sshd_config** file on the Cockpit Server. Make sure **ChallengeResponseAuthentication** is set to **yes** and restart **sshd** with the **systemctl restart sshd** command.

2.5. SSH TWO-FACTOR AUTHENTICATION WITH COCKPIT

Cockpit supports two-factor authentication. If you have protected your SSH server with two-factor authentication, the login screen will prompt you to enter your password and PIN pair.

Setting up SSH for two-factor authentication requires two components:

1. A company’s authenticator application that provides one-time passwords or PIN numbers. An example application is **`Google Authenticator`**, which also has its own Pluggable Authentication Module (PAM).

2. A server that validates the PINs from a dongle.

These two components are often implemented differently for different companies.

After setting up the authenticator application and the validation server, enable SSH two-factor authentication in Cockpit:

1. In the **`/etc/pam.d/sshd`** file, right after the last **`auth`** line, add this line:

   ```
   auth       required         <your_PAM_module>
   ```

   Substitute **`<your_PAM_module>`** with the PAM module used by your application.

2. In the **`/etc/ssh/sshd_config`** file, set **`ChallengeResponseAuthentication`** to **`yes`**.

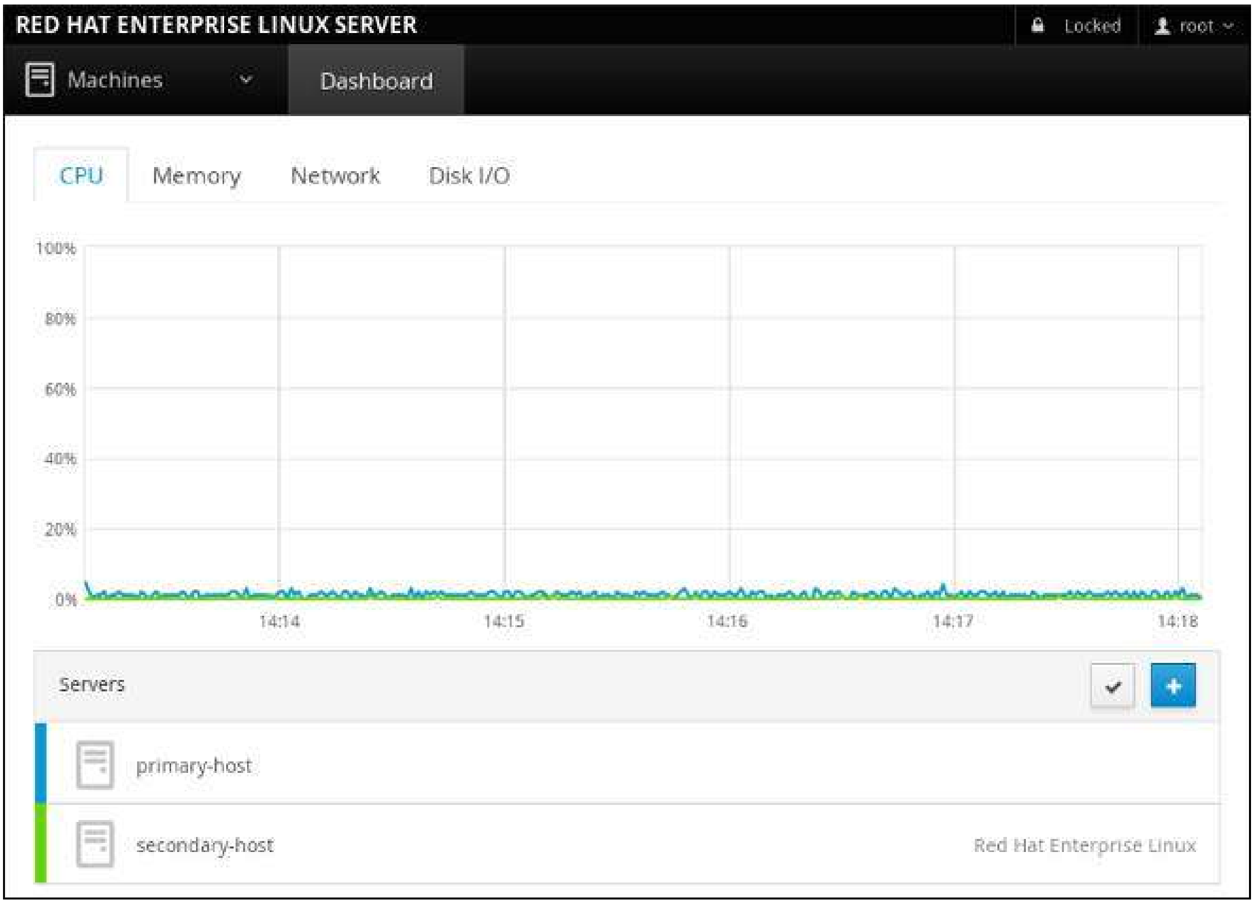
3. Restart the **`sshd`** service with the **`systemctl restart shhd`** command.

Cockpit will ask for your verification code the next time you log in.

CHAPTER 3. USING COCKPIT

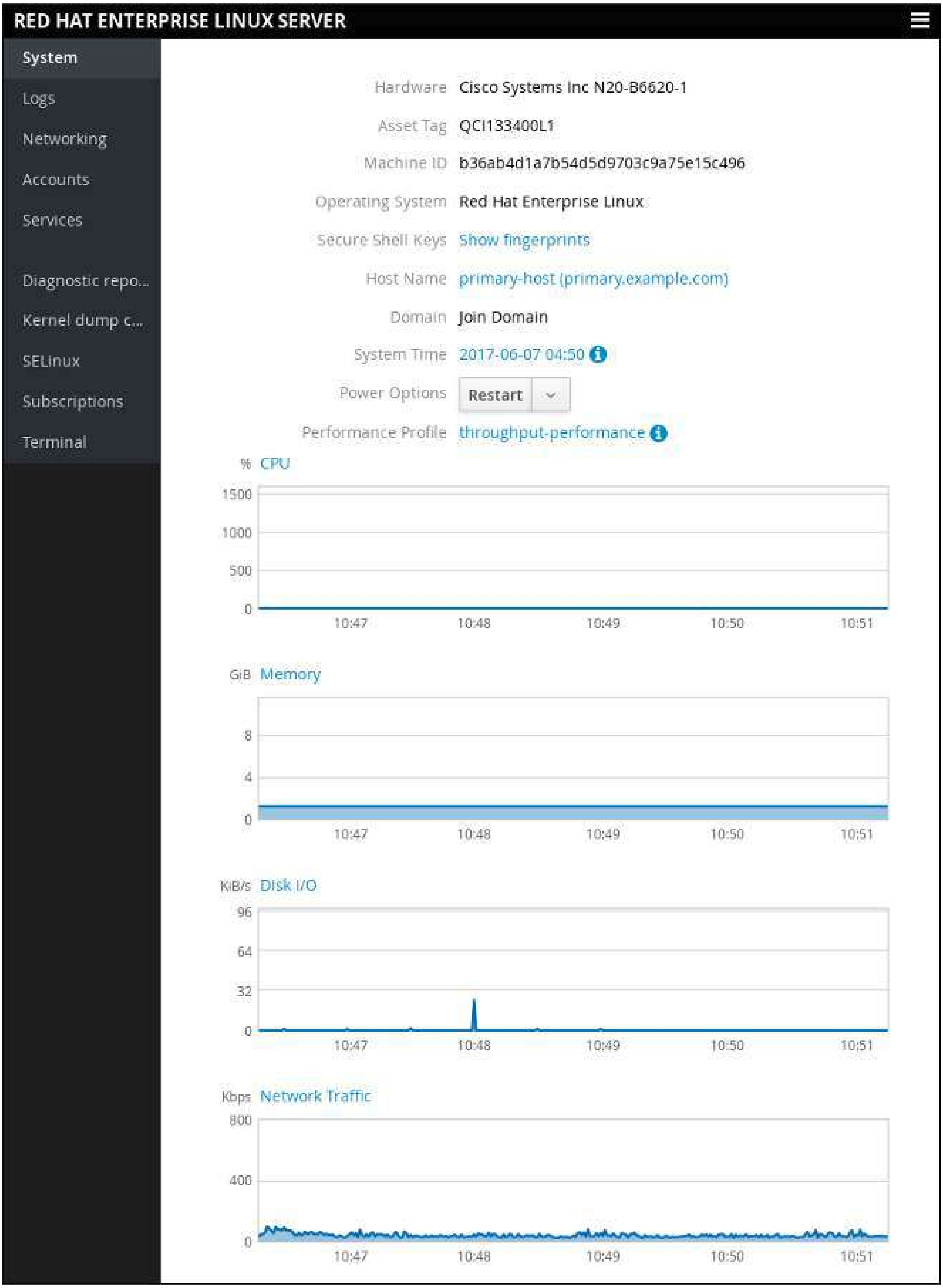
3.1. GETTING TO KNOW THE COCKPIT INTERFACE

Once you have logged in, you will see the main Cockpit interface. It has the **Dashboard** tab on the top and a side menu with details for the selected system on the left. The Dashboard shows a list of all systems added to the Cockpit server with graphs for their CPU usage, memory usage, disk I/O, and network traffic.



From Dashboard, you can select a system name, in this case `primary-host`, and have a look at the side menu:

System: Shows information about the system that Cockpit is running on. This includes CPU usage, memory usage, disk I/O, and network traffic, as well as hardware and operating system details.



Hardware Cisco Systems Inc N20-B6620-1

Asset Tag QCI133400L1

Machine ID b36ab4d1a7b54d5d9703c9a75e15c496

Operating System Red Hat Enterprise Linux

Secure Shell Keys Show fingerprints

Host Name primary-host (primary.example.com)

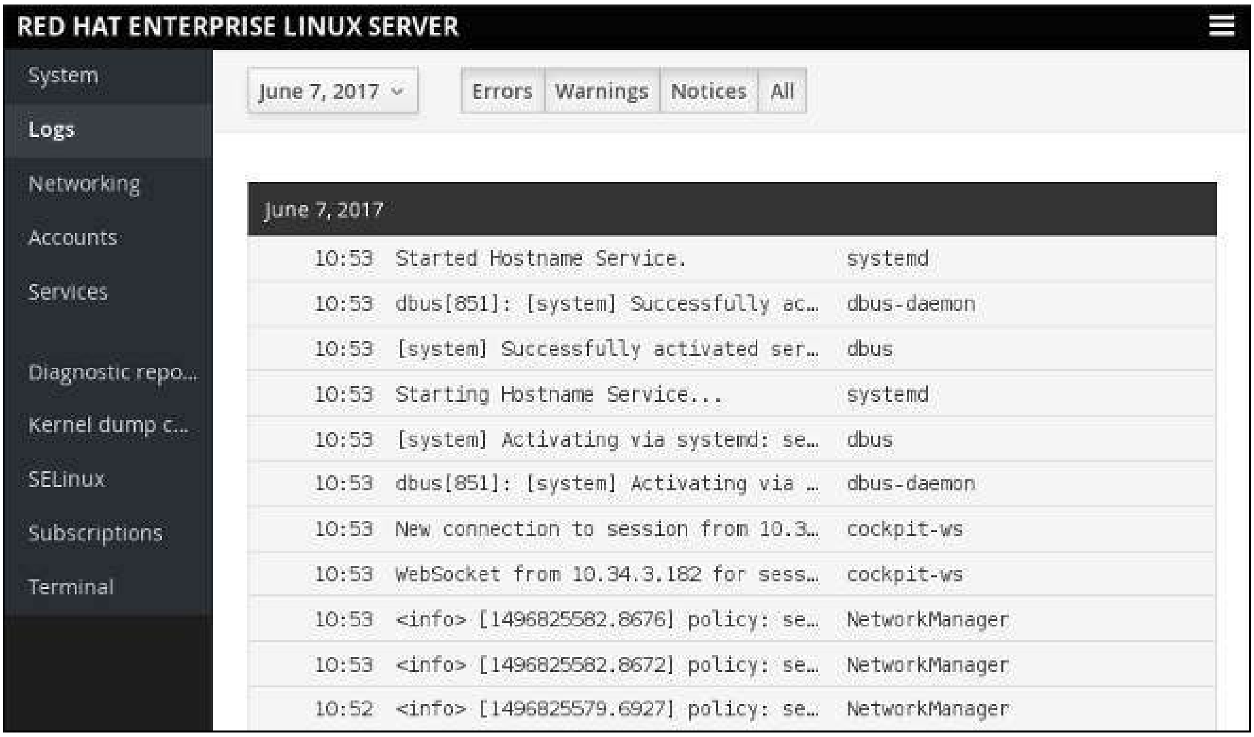
Domain Join Domain

System Time 2017-06-07 04:50

Power Options Restart

Performance Profile throughput-performance

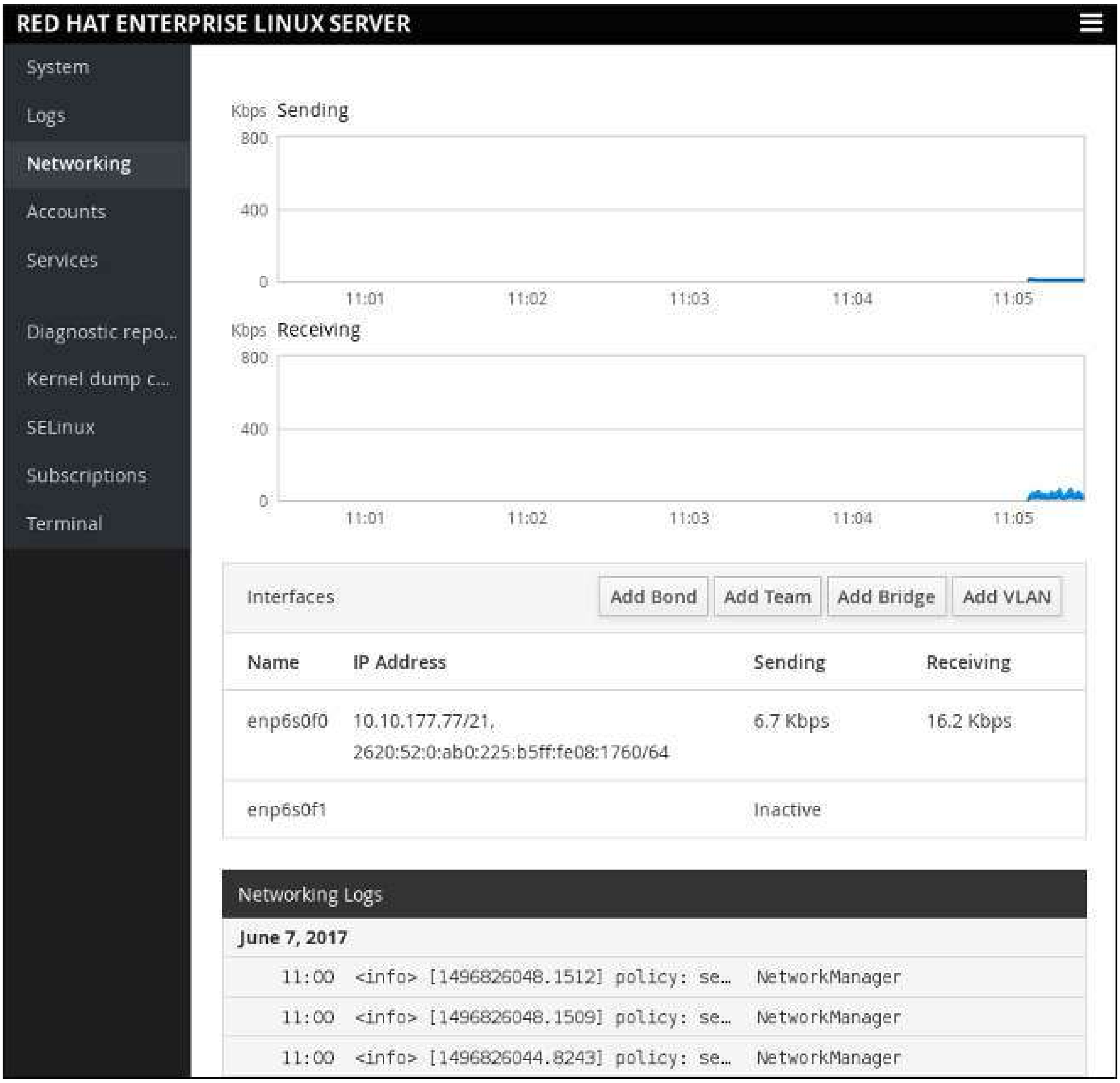
Logs: See messages produced by the systemd journal, including errors, warnings, and notices. The log is similar to the output of the **`journalctl`** command. The log displays newest entries first, with options to filter by type.



June 7, 2017

```
10:53  Started Hostname Service.                systemd
10:53  dbus[851]: [system] Successfully ac…     dbus-daemon
10:53  [system] Successfully activated ser…     dbus
10:53  Starting Hostname Service...             systemd
10:53  [system] Activating via systemd: se…     dbus
10:53  dbus[851]: [system] Activating via …     dbus-daemon
10:53  New connection to session from 10.3…     cockpit-ws
10:53  WebSocket from 10.34.3.182 for sess…     cockpit-ws
10:53  <info> [1496825582.8676] policy: se…     NetworkManager
10:53  <info> [1496825582.8672] policy: se…     NetworkManager
10:52  <info> [1496825579.6927] policy: se…     NetworkManager
```

Networking: See networking interfaces (for example, eth0) and active graphs of sent and received data.



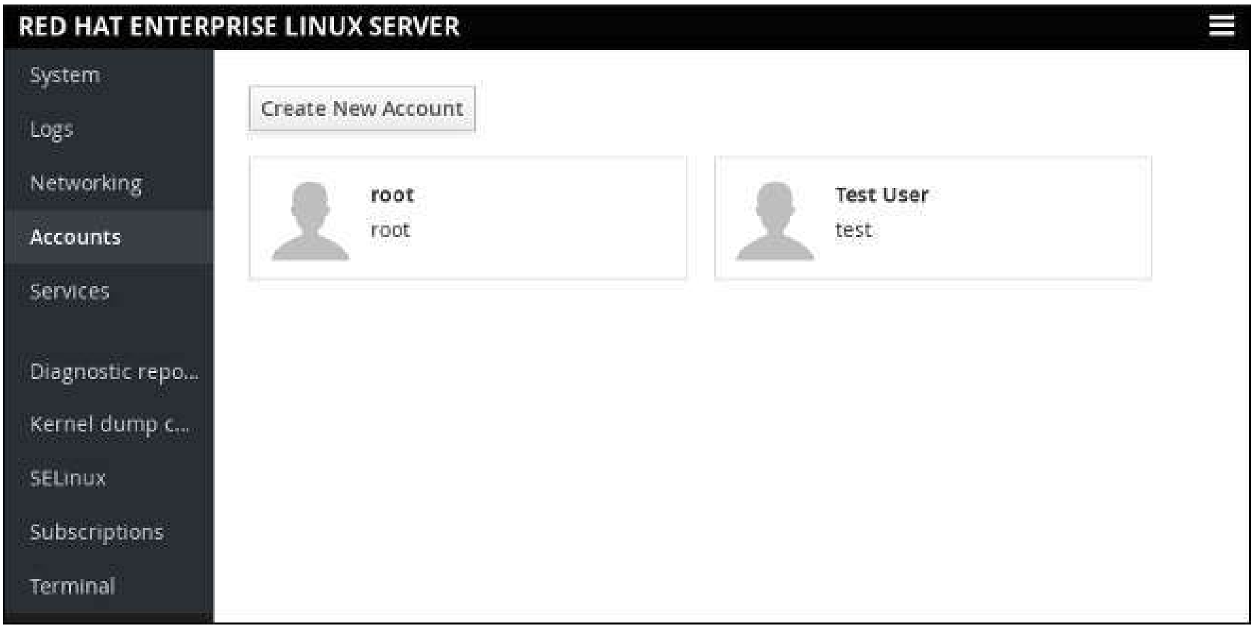
Name	IP Address	Sending	Receiving
enp6s0f0	10.10.177.77/21, 2620:52:0:ab0:225:b5ff:fe08:1760/64	6.7 Kbps	16.2 Kbps
enp6s0f1		Inactive	

Networking Logs

June 7, 2017

```
11:00  <info> [1496826048.1512] policy: se…  NetworkManager
11:00  <info> [1496826048.1509] policy: se…  NetworkManager
11:00  <info> [1496826044.8243] policy: se…  NetworkManager
```

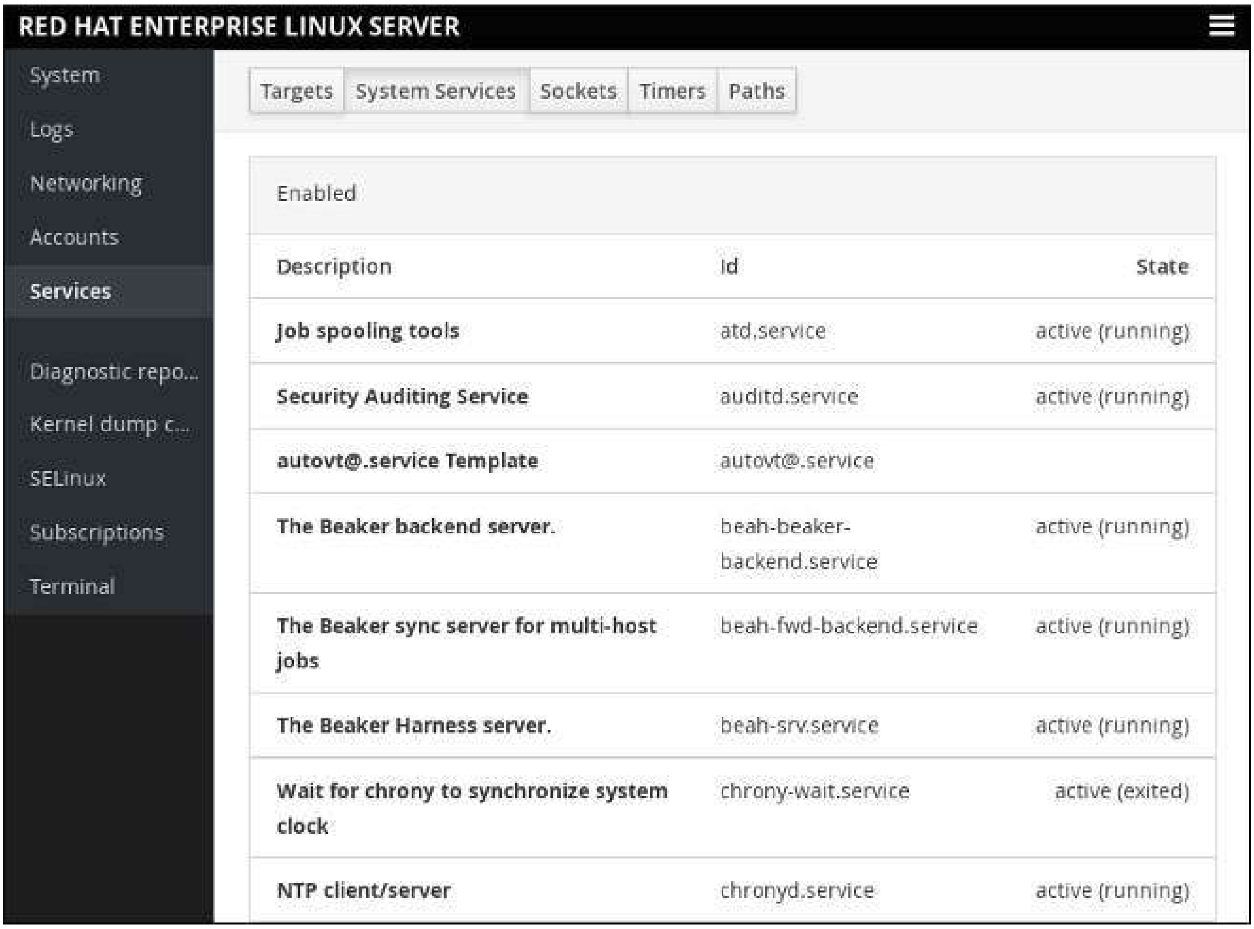
Accounts: Shows which administrative (root) and other users (for example, alan, djohnson) have accounts on the system.



root
root

Test User
test

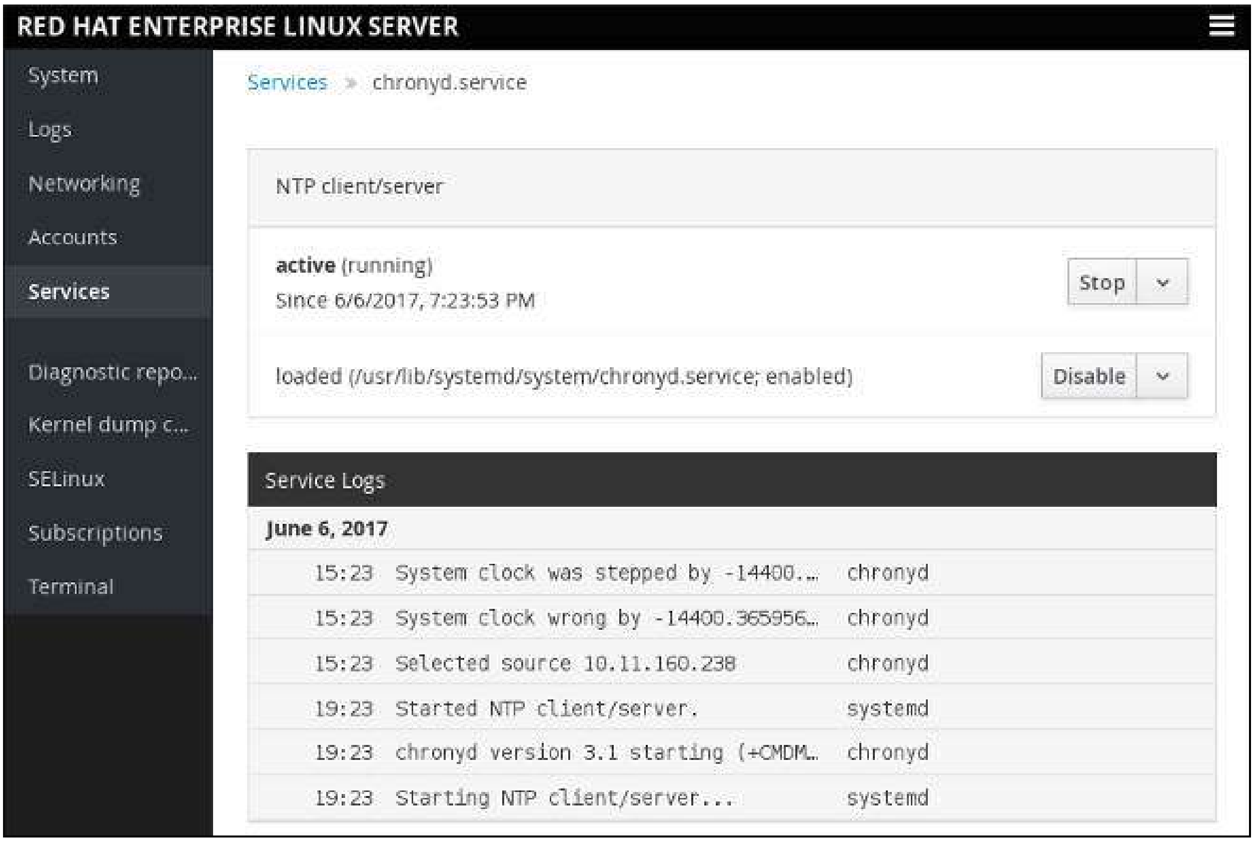
Services: Shows the systemd services running on the Cockpit server. You can see which are active/enabled or inactive. You can also see other systemd features: Targets, sockets, timers, and paths.



Enabled

Description	Id	State
Job spooling tools	atd.service	active (running)
Security Auditing Service	auditd.service	active (running)
autovt@.service Template	autovt@.service	
The Beaker backend server.	beah-beaker-backend.service	active (running)
The Beaker sync server for multi-host jobs	beah-fwd-backend.service	active (running)
The Beaker Harness server.	beah-srv.service	active (running)
Wait for chrony to synchronize system clock	chrony-wait.service	active (exited)
NTP client/server	chronyd.service	active (running)

Select a service to view its details:



NTP client/server

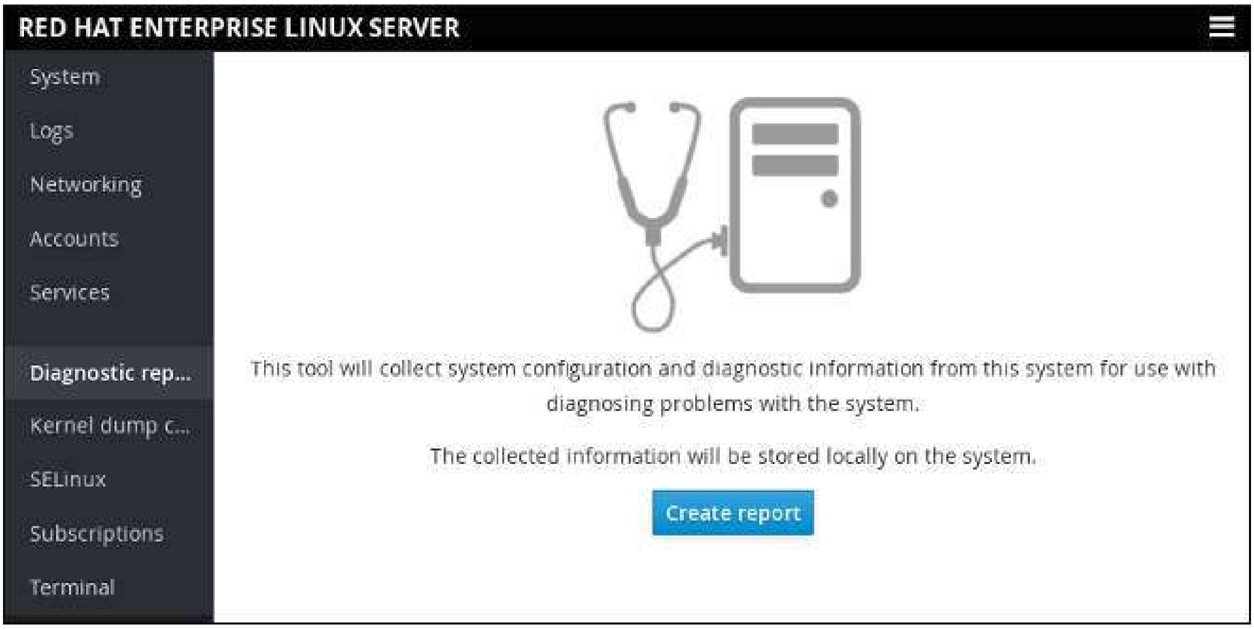
active (running)
Since 6/6/2017, 7:23:53 PM

loaded (/usr/lib/systemd/system/chronyd.service; enabled)

June 6, 2017

```
15:23  System clock was stepped by -14400...   chronyd
15:23  System clock wrong by -14400.365956...  chronyd
15:23  Selected source 10.11.160.238           chronyd
19:23  Started NTP client/server.              systemd
19:23  chronyd version 3.1 starting (+CMDM...  chronyd
19:23  Starting NTP client/server...           systemd
```

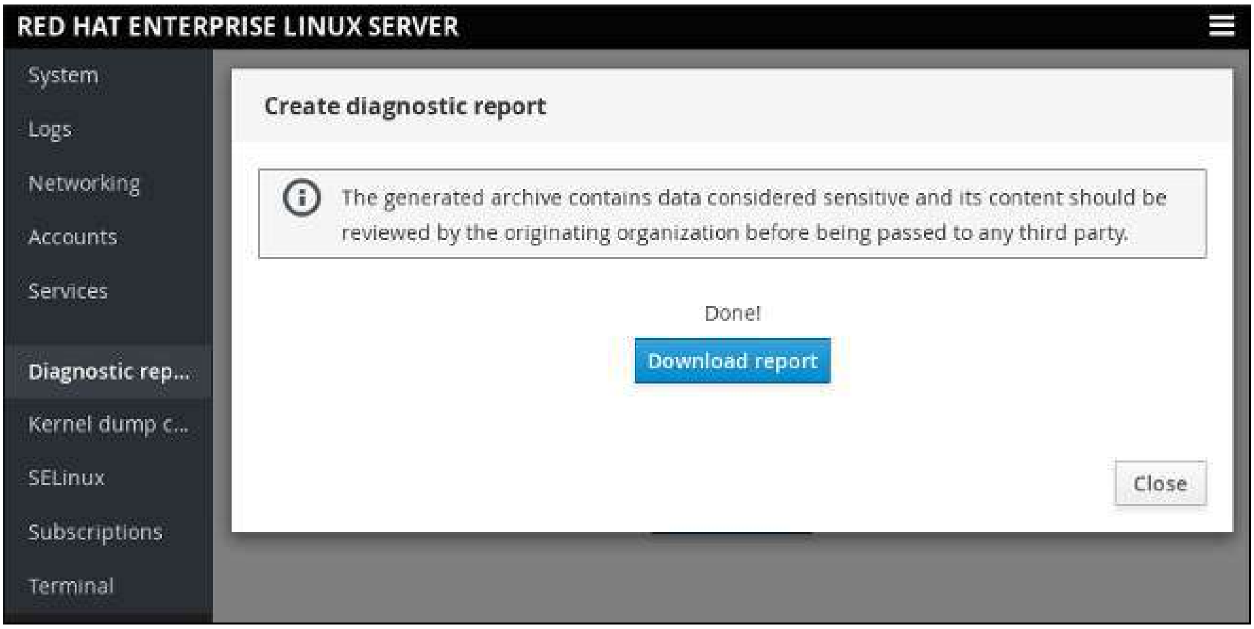
Diagnostic reports: Collects system configuration and diagnostics information and prepares a report in the .xz compressed format.



This tool will collect system configuration and diagnostic information from this system for use with diagnosing problems with the system.

The collected information will be stored locally on the system.

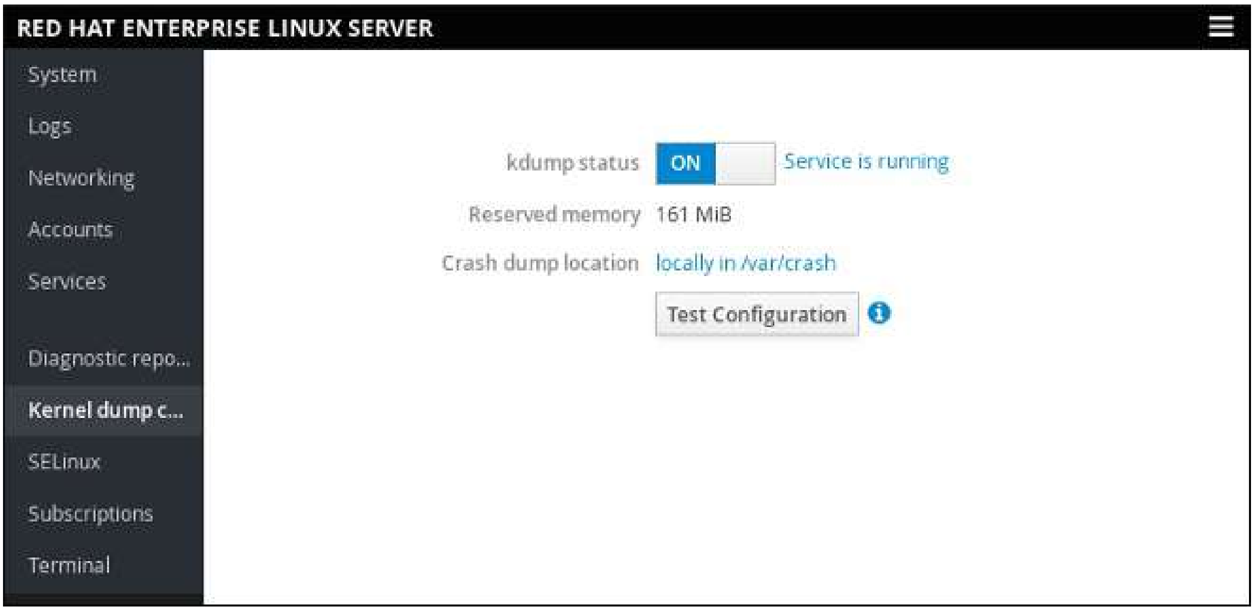
You can then download the report locally to your system:



The generated archive contains data considered sensitive and its content should be reviewed by the originating organization before being passed to any third party.

Done!

Kernel dump configuration: Shows kdump status and configuration and allows to crash the kernel to test kdump.

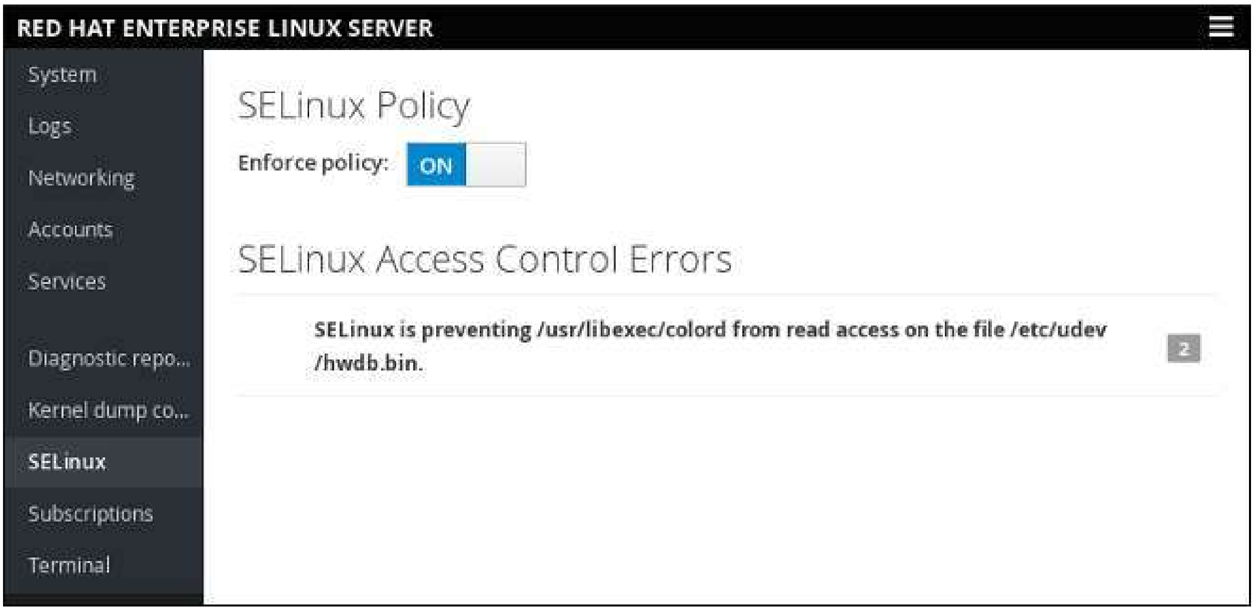


kdump status ON Service is running

Reserved memory 161 MiB

Crash dump location locally in /var/crash

SELinux: Shows whether SELinux is enabled and lists access control errors.



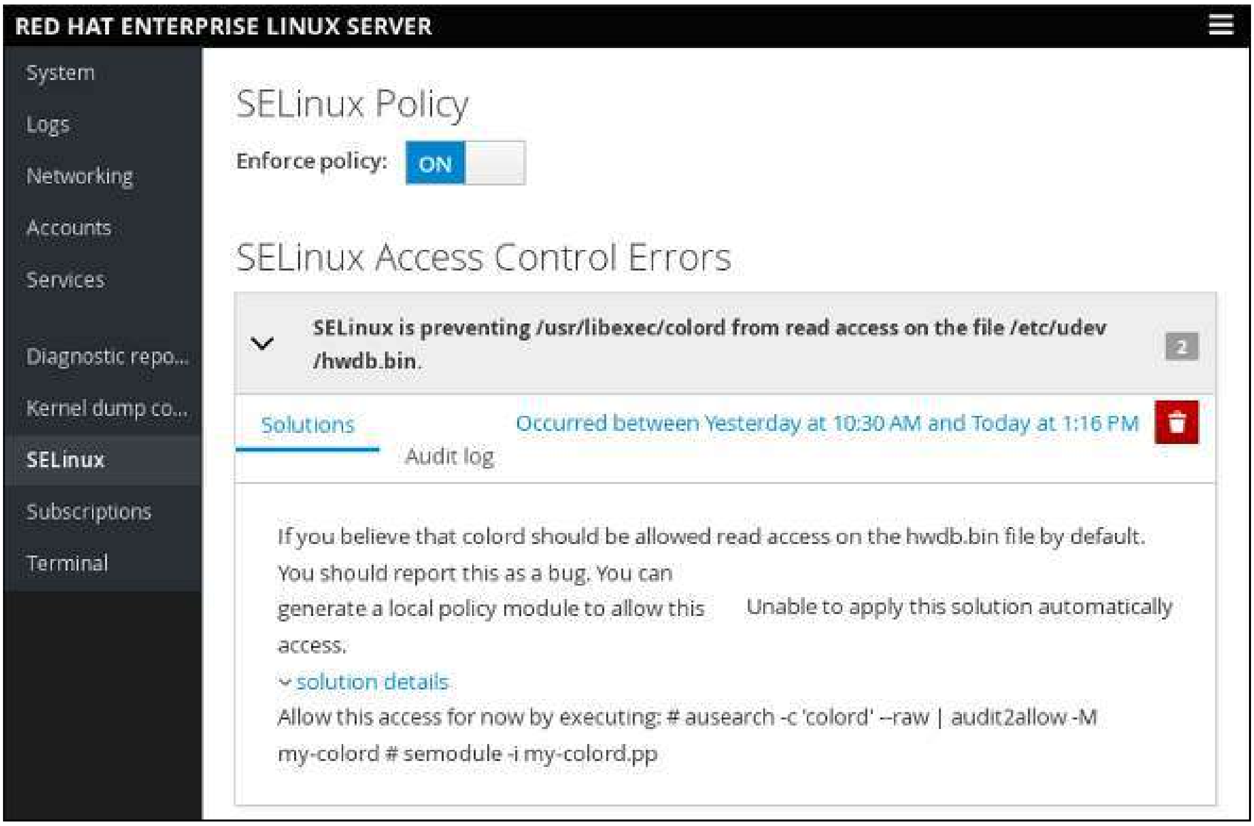
SELinux Policy

Enforce policy: ON

SELinux Access Control Errors

SELinux is preventing /usr/libexec/colord from read access on the file /etc/udev/hwdb.bin.

Click on an error to see detailed information about it, proposed solution, and audit log:



SELinux Policy

Enforce policy: ON

SELinux Access Control Errors

SELinux is preventing /usr/libexec/colord from read access on the file /etc/udev /hwdb.bin.

Occurred between Yesterday at 10:30 AM and Today at 1:16 PM

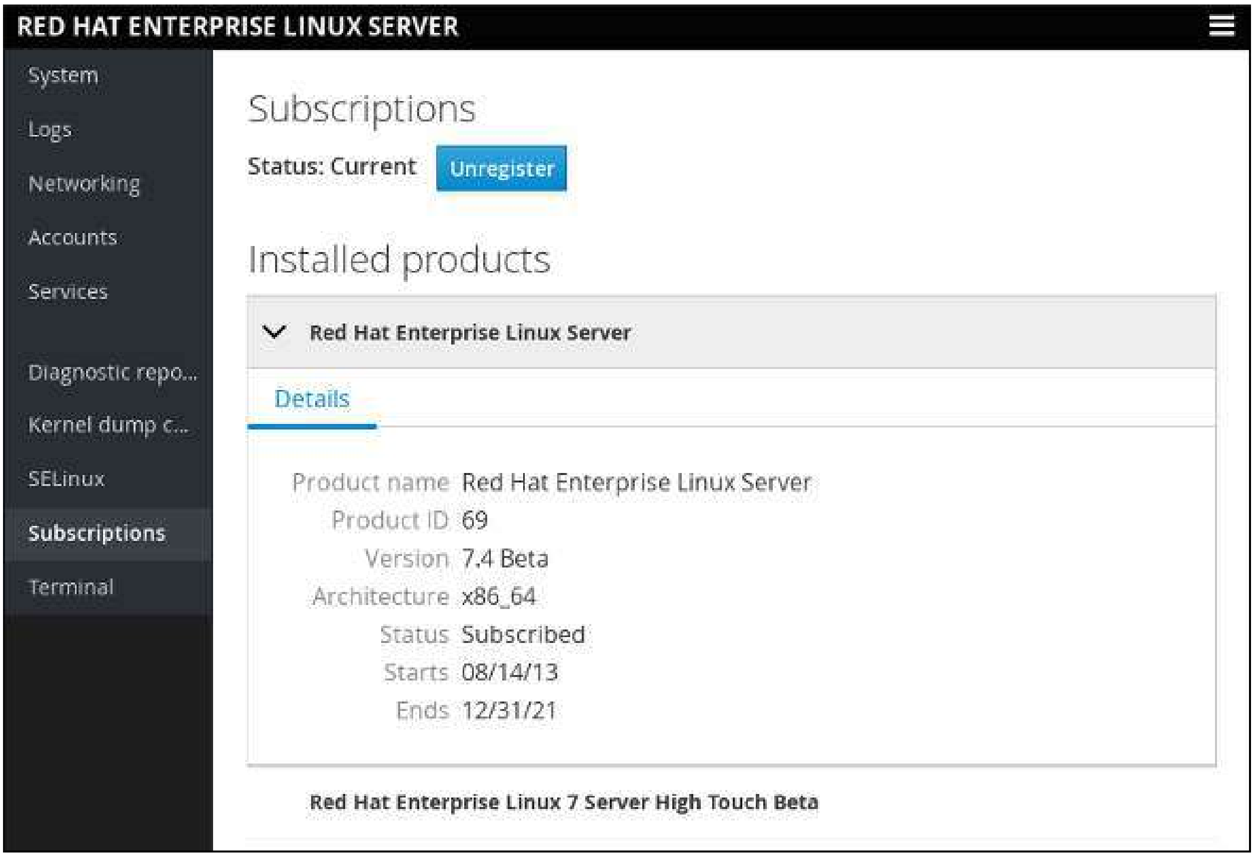
If you believe that colord should be allowed read access on the hwdb.bin file by default. You should report this as a bug. You can generate a local policy module to allow this access.

Unable to apply this solution automatically

solution details

Allow this access for now by executing: # ausearch -c 'colord' --raw | audit2allow -M my-colord # semodule -i my-colord.pp

Subscriptions: Displays installed Red Hat products and subscriptions.



Subscriptions

Status: Current

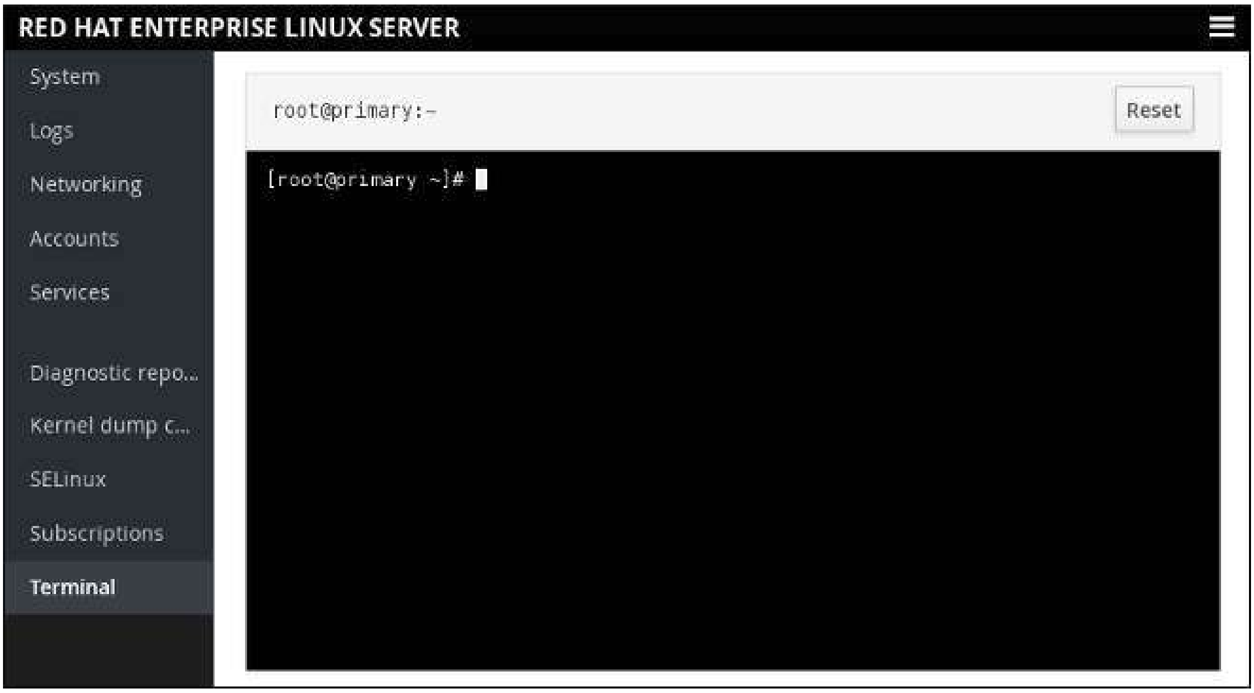
Installed products

Red Hat Enterprise Linux Server

Product name	Red Hat Enterprise Linux Server
Product ID	69
Version	7.4 Beta
Architecture	x86_64
Status	Subscribed
Starts	08/14/13
Ends	12/31/21

Red Hat Enterprise Linux 7 Server High Touch Beta

Terminal: Opens an in-browser terminal with a command line session to the Cockpit system. In this terminal, you can run commands from your signed-in user account. For example, as root, you could run the **`systemctl start`** or **`dnf install`** commands.



```
[root@primary ~]#
```

For Red Hat Enterprise Linux Atomic Host systems, there are additional features in the Cockpit interface. See Cockpit Interface Specific to Atomic Host.

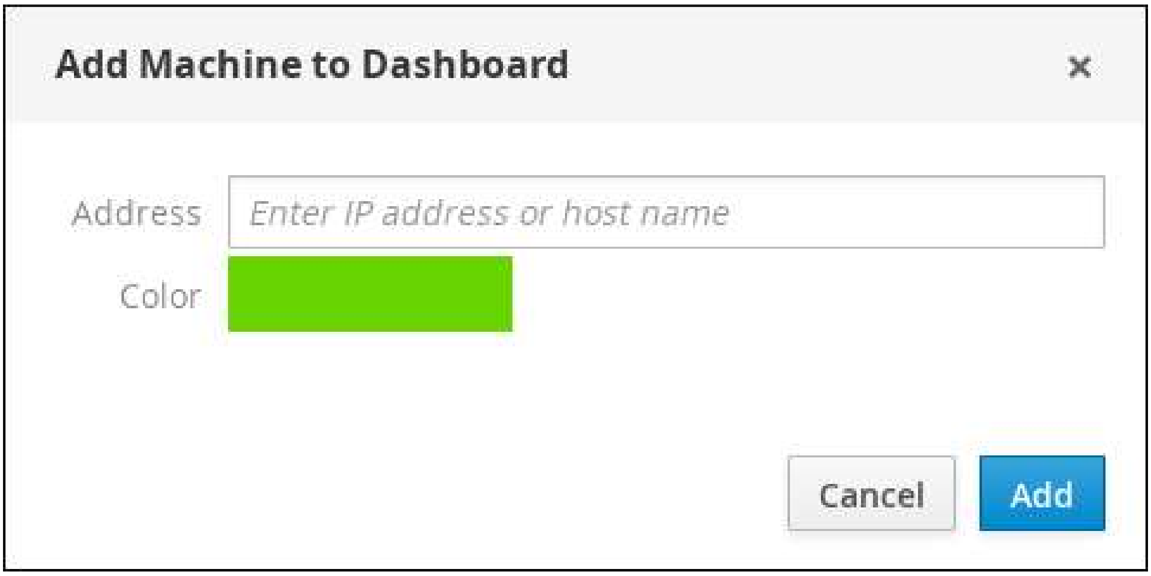
3.1.1. Adding secondary systems

Once you log in to the primary server, you will be able to connect to secondary servers. These secondary systems need to have:

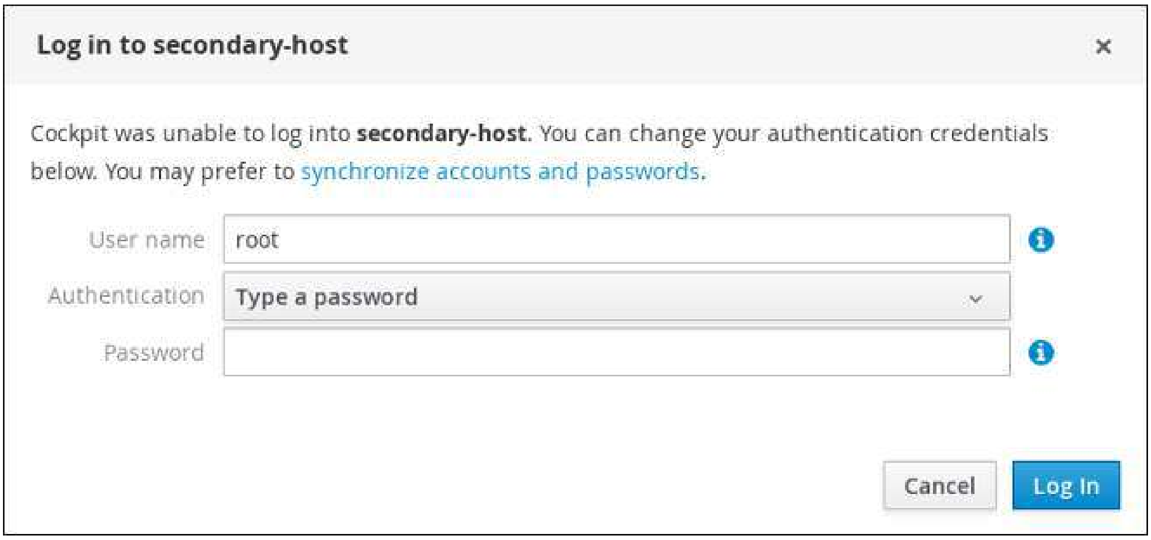
- The **cockpit** packages installed.
- An SSH server running and available on port 22 that supports password or key-based authentication.

To add a new secondary server:

1. From the "Dashboard" tab next to the system name, click the plus button.
2. Enter IP of the server you are adding and choose a color label for it.
3. Click the "Add" button.



4. Log in to the system with a user name and password:



Cockpit was unable to log into **secondary-host**. You can change your authentication credentials below. You may prefer to synchronize accounts and passwords.

User name: root

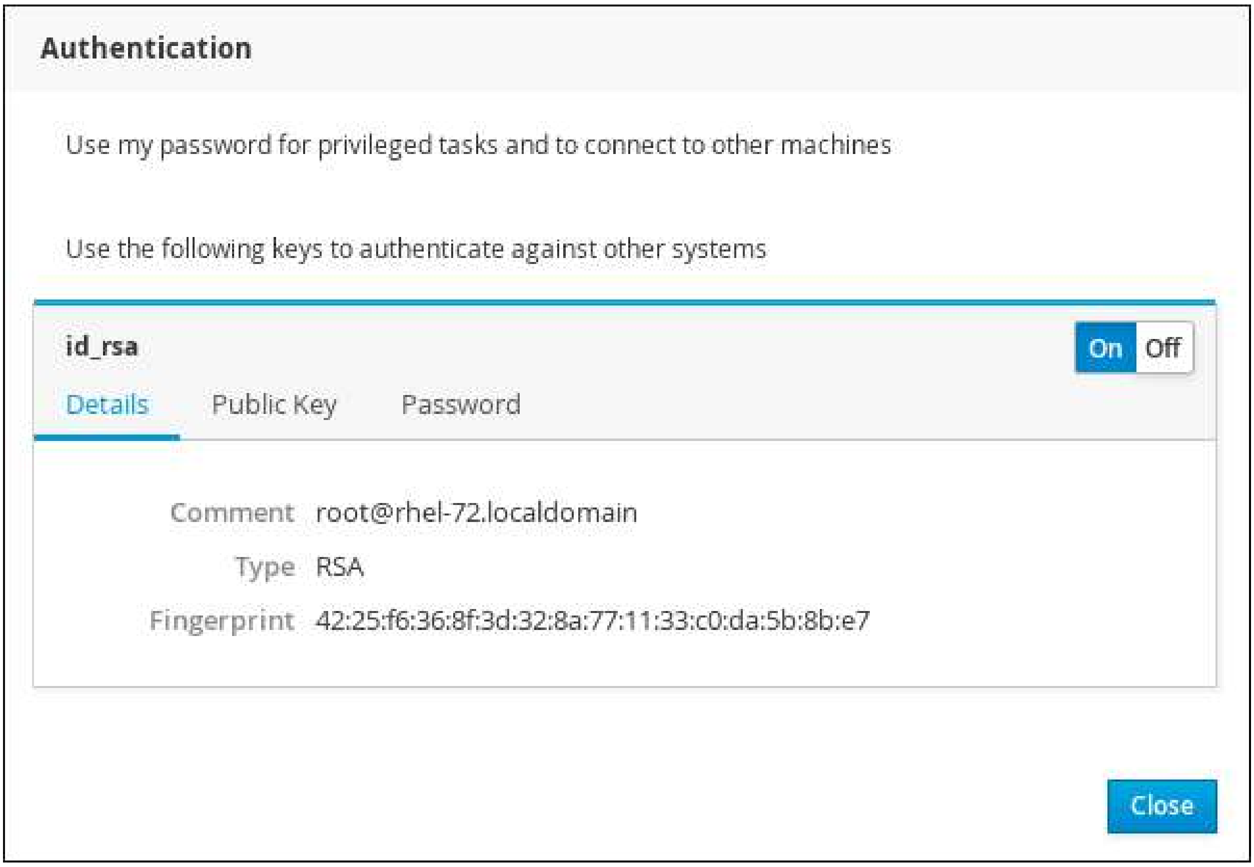
Authentication: Type a password

Configuring Key-Based Authentication

If you have keys generated on the primary server, you need to add them to the target server, in the *~/.ssh/authorized_keys* file. If you do not have keys, use the following command:

```
$ ssh-keygen
```

Next, copy the contents of the *~/.ssh/id_rsa.pub* file to the *~/.ssh/authorized_keys* file **on the target server**. Then, return to the user interface on the primary server, click the top right corner menu with the user name on it, choose **Authentication**, and enable the preloaded key.



Authentication

Use my password for privileged tasks and to connect to other machines

Use the following keys to authenticate against other systems

id_rsa

Comment root@rhel-72.localdomain

Type RSA

Fingerprint 42:25:f6:36:8f:3d:32:8a:77:11:33:c0:da:5b:8b:e7

After you type in the IP when adding the new system to the Dashboard, change the **Authentication** type to **Use available credentials**.

3.1.2. Logging into other systems through Cockpit

On the login screen, you can also choose an alternate host to connect to.

The alternate host needs to have:

- SSH listening on port 443
- the **`cockpit-bridge`** package and all relevant subpackages to interact with the system, such as **`cockpit-system`**, installed. The packages should be the same version as in the Cockpit server.

To connect to an alternate host:

1. Type in your username and password from that alternate host and click **Other Options**.
2. In the entry field type the IP address of the new host and click **Log In**.
3. Provide the SSH fingerprint and click **Log In** again.

Now you are able to browse the new system. Cockpit uses SSH to authenticate you against that host, so you do not need to configure anything else on the new system.

Note

If the new machine is not known to Cockpit, and you get the **Refusing to connect. Host is unknown** error, use the following command to allow connections from unknown hosts:

```
ssh-keyscan -H [ip_address] >> /var/lib/cockpit/known_hosts
```

3.1.3. Logging into a system via a Bastion Host

On the Cockpit login screen you can choose an alternate host to connect to. Cockpit uses SSH to authenticate you against that host and to display the admin interface for that host.

Although browsers cannot use SSH directly to connect to machines or authenticate against them, Cockpit can make this happen. Only one host needs to have Cockpit listen on port 9090 available to browsers over TLS. Other hosts only need to have SSH accessible on the usual port 22.

3.2. CHANGING THE COCKPIT PORT

To change the Cockpit port:

1. If required, create the */etc/systemd/system/websocket.cockpit.d/* directory and its parent directories:

   ```
   # mkdir -p /etc/systemd/system/websocket.cockpit.d/
   ```

2. Create the */etc/systemd/system/websocket.cockpit.d/listen.conf* file with these contents:

   ```
   [Socket]
   ListenStream=9898
   ```

3. Allow the new port through the firewall:

   ```
   # firewall-cmd --add-port=9898/tcp
   # firewall-cmd --permanent --add-port=9898/tcp
   ```

4. If you have SELinux enabled, change the default SELinux policy to allow the **websm_port_t** domain to listen on the TCP 9898 port:

   ```
   $ sudo semanage port -a -t websm_port_t -p tcp 9898
   ```

   If the port is already defined by some other part of the SELinux policy, use the *-m* argument instead of *-a* to modify the definition:

   ```
   $ sudo semanage port -m -t websm_port_t -p tcp 9898
   ```

   1. To make the changes take effect, run the following commands:

      ```
      $ sudo systemctl daemon-reload
      $ sudo systemctl restart cockpit.socket
      ```

You can now use the address with the newly assigned port in the web browser.

For changing port on a Red Hat Enterprise Linux Atomic Host system, see Changing the Cockpit port on Atomic Host.

3.3. ENABLING MORE COCKPIT FEATURES

You can add more Cockpit features by installing additional **`cockpit-*`** packages using **`yum`**.

CHAPTER 4. COCKPIT ON RED HAT ENTERPRISE LINUX ATOMIC HOST

A Cockpit server can run on Red Hat Enterprise Linux Atomic Host, and Atomic Host servers can be monitored and administered using Cockpit. Additionally, Cockpit can control life cycle of container instances and manipulate container images.

Note

Cockpit does not yet support Kubernetes on Red Hat Enterprise Linux or Red Hat Enterprise Linux Atomic Host servers.

This chapter describes Cockpit features specific to Atomic Host.

4.1. INSTALLING COCKPIT ON ATOMIC HOST

To install Cockpit on Atomic Host:

1. Pull the **cockpit-ws** image:

```
# atomic install rhel7/cockpit-ws
```

2. Run the **cockpit-ws** image:

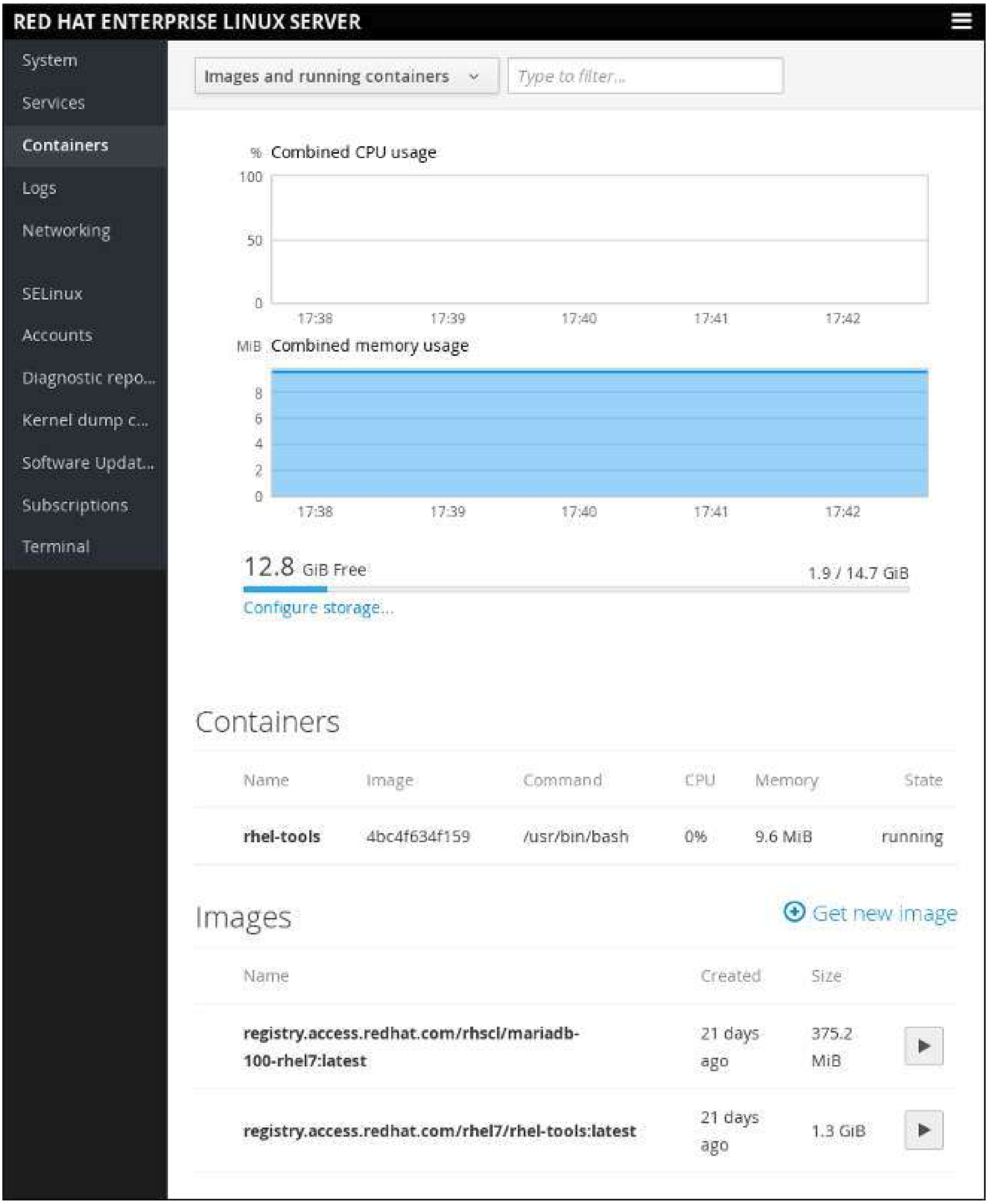
```
# atomic run rhel7/cockpit-ws
```

Now you can log into Cockpit. See Opening the Interface for instructions.

4.2. COCKPIT INTERFACE SPECIFIC TO ATOMIC HOST

In addition to information about systems presented in Getting to know the Cockpit interface, extra tabs appear on Atomic Host systems:

Containers: Lists all images available on the system, all running and non-running containers, combined CPU & memory usage graphs, and a storage usage bar. See Working with Containers for more information on using this tab.



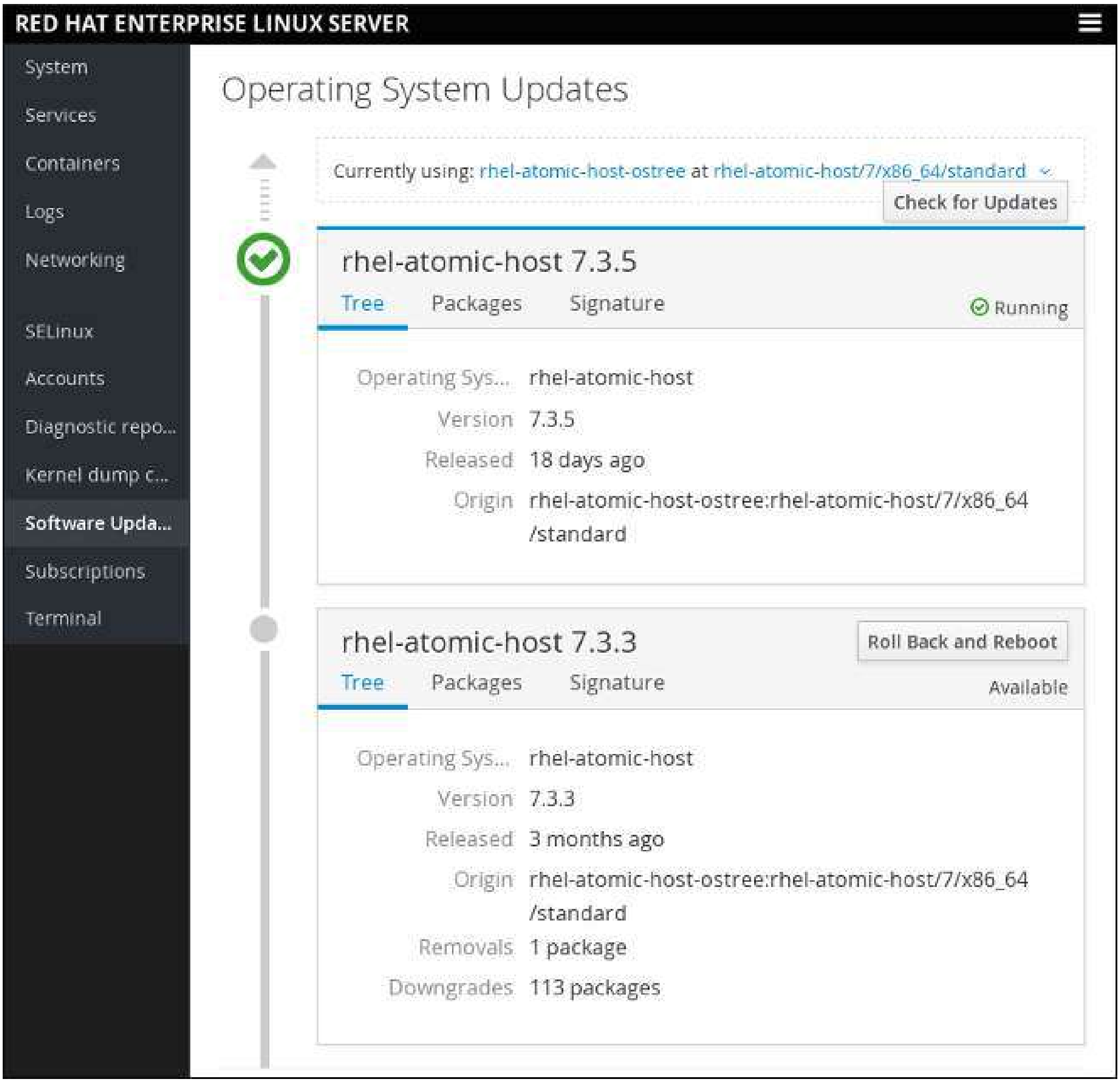
Containers

Name	Image	Command	CPU	Memory	State
rhel-tools	4bc4f634f159	/usr/bin/bash	0%	9.6 MiB	running

Images

Name	Created	Size	
registry.access.redhat.com/rhscl/mariadb-100-rhel7:latest	21 days ago	375.2 MiB	▶
registry.access.redhat.com/rhel7/rhel-tools:latest	21 days ago	1.3 GiB	▶

Software Updates: Shows the available OSTrees on the system. You can also check for a newer tree, or roll back to a previous version.



Operating System Updates

Currently using: rhel-atomic-host-ostree at rhel-atomic-host/7/x86_64/standard

rhel-atomic-host 7.3.5

Operating Sys... rhel-atomic-host
Version 7.3.5
Released 18 days ago
Origin rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard

rhel-atomic-host 7.3.3

Operating Sys... rhel-atomic-host
Version 7.3.3
Released 3 months ago
Origin rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
Removals 1 package
Downgrades 113 packages

4.2.1. Working with Containers

The **Containers** tab presents you with a UI to interact with your Atomic Host images and containers. Apart from the system resources graphs, there are lists of all images you have locally on the system as well as all running and non-running containers.

Download an image. Click the "Get new image" button from the images list to the right and enter an image name or a keyword. Choose an image and click "Download".

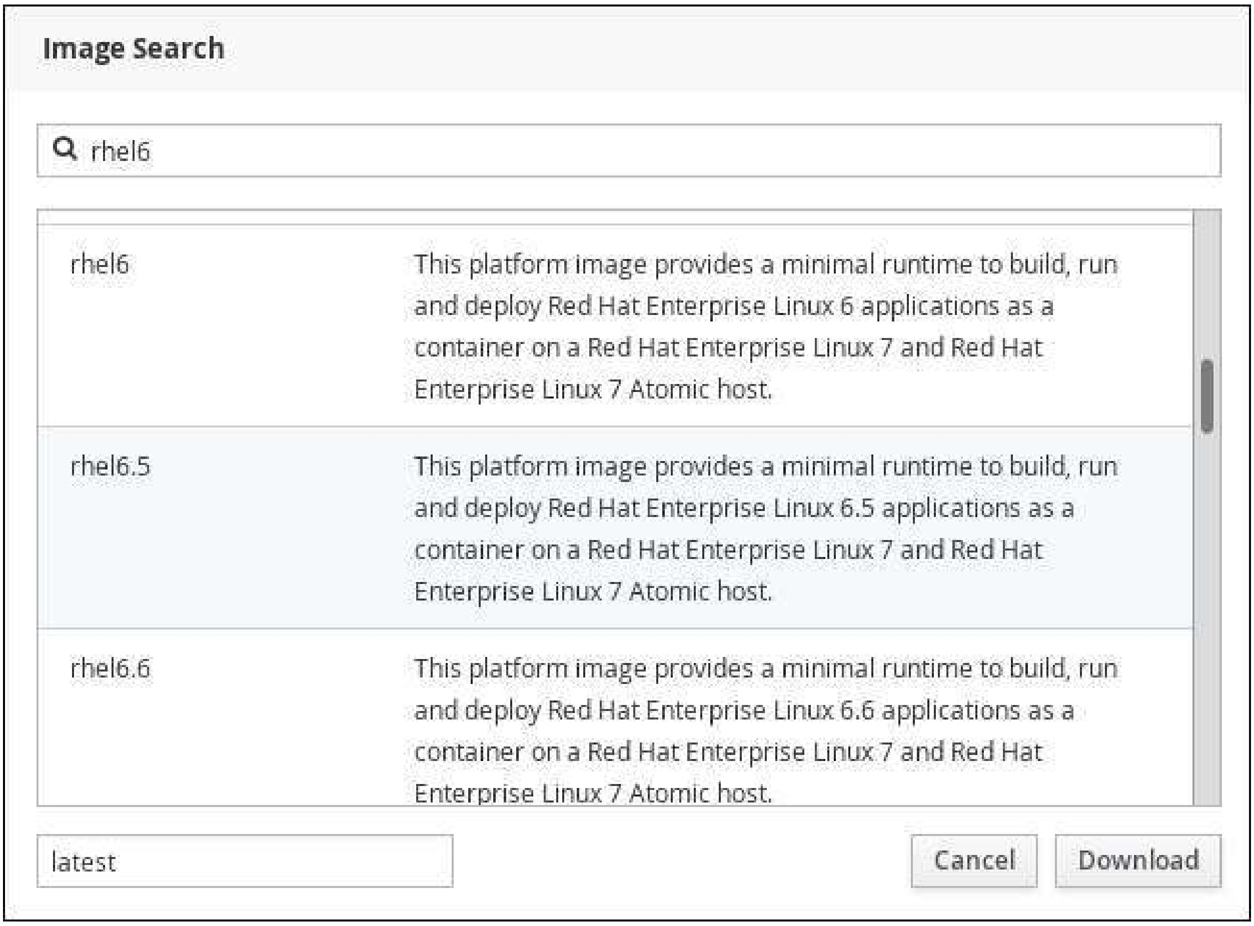


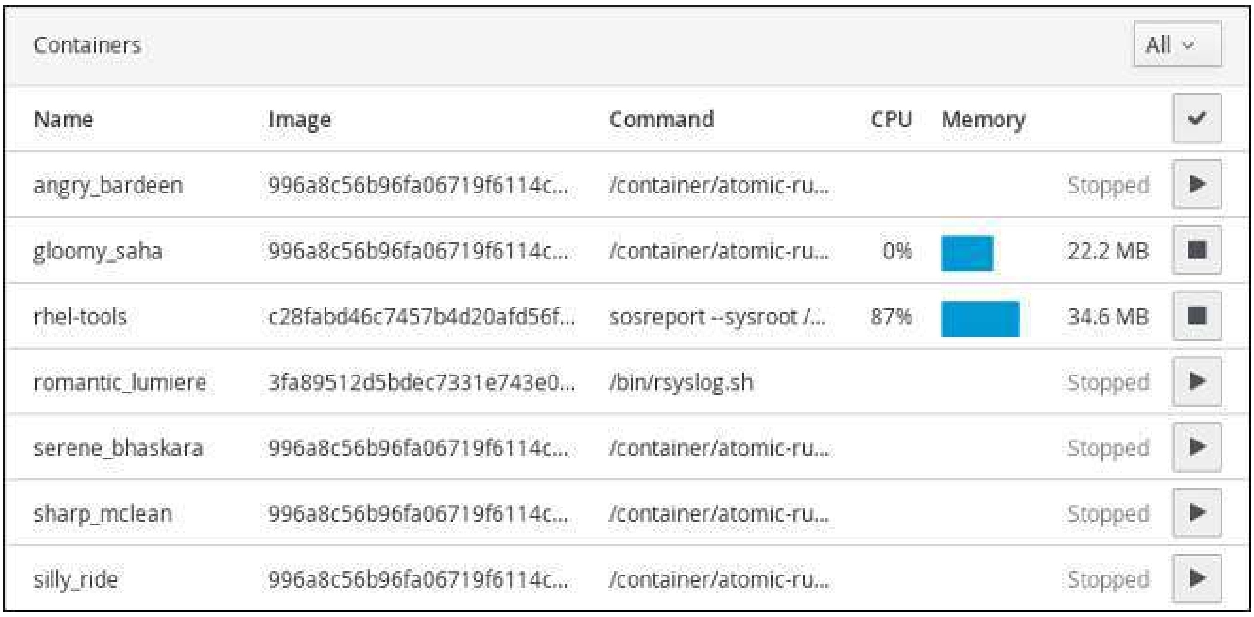
Image Search

rhel6

rhel6	This platform image provides a minimal runtime to build, run and deploy Red Hat Enterprise Linux 6 applications as a container on a Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 7 Atomic host.
rhel6.5	This platform image provides a minimal runtime to build, run and deploy Red Hat Enterprise Linux 6.5 applications as a container on a Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 7 Atomic host.
rhel6.6	This platform image provides a minimal runtime to build, run and deploy Red Hat Enterprise Linux 6.6 applications as a container on a Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 7 Atomic host.

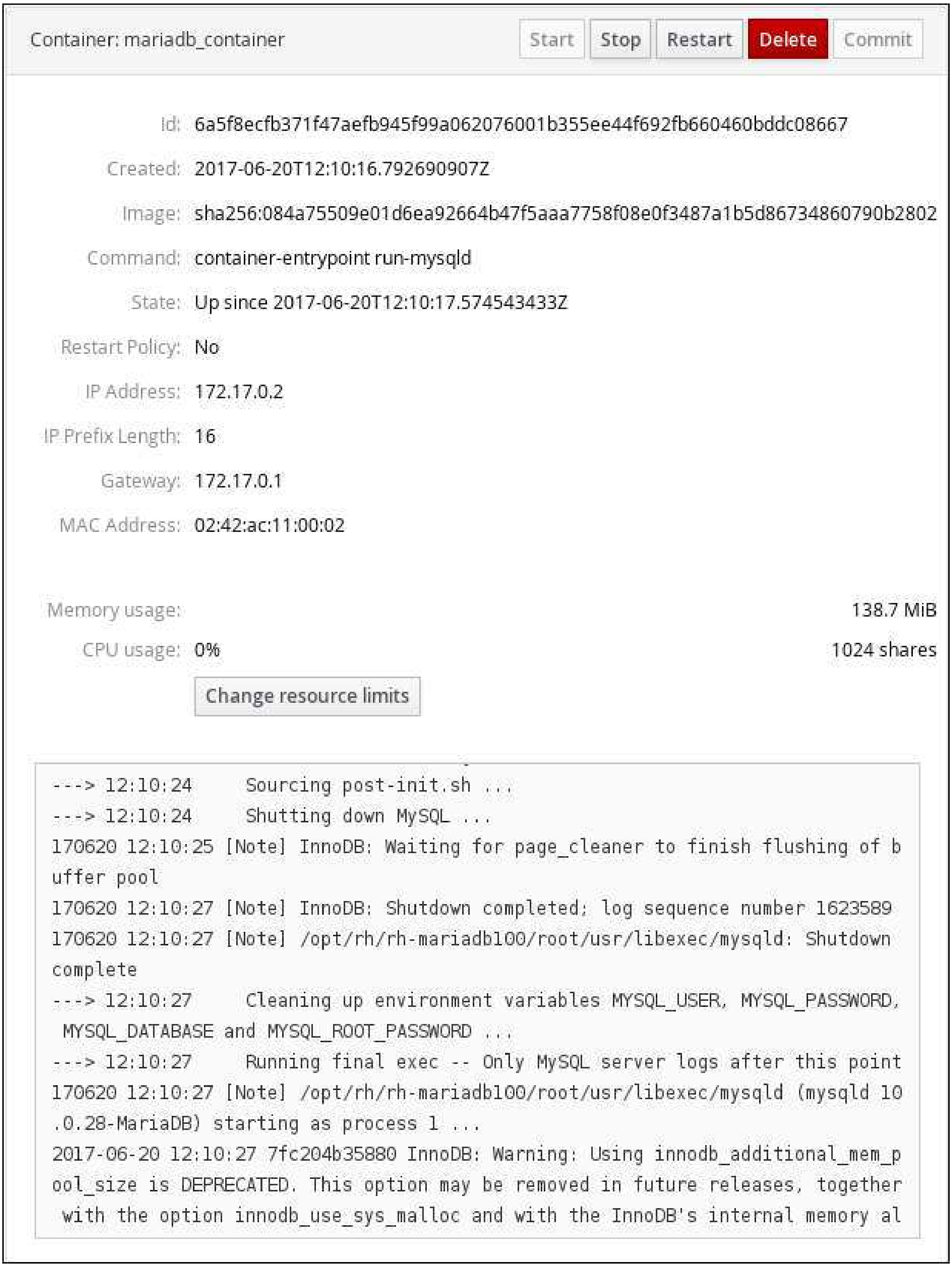
latest

Starting and stopping containers. From the "Containers" list, you can start and stop containers using the buttons on the right side. Use the drop-down menu to see all or filter out the non-running containers.



Name	Image	Command	CPU	Memory		✓
angry_bardeen	996a8c56b96fa06719f6114c...	/container/atomic-ru...			Stopped	▶
gloomy_saha	996a8c56b96fa06719f6114c...	/container/atomic-ru...	0%		22.2 MB	■
rhel-tools	c28fabd46c7457b4d20afd56f...	sosreport --sysroot /...	87%		34.6 MB	■
romantic_lumiere	3fa89512d5bdec7331e743e0...	/bin/rsyslog.sh			Stopped	▶
serene_bhaskara	996a8c56b96fa06719f6114c...	/container/atomic-ru...			Stopped	▶
sharp_mclean	996a8c56b96fa06719f6114c...	/container/atomic-ru...			Stopped	▶
silly_ride	996a8c56b96fa06719f6114c...	/container/atomic-ru...			Stopped	▶

Click on a container to inspect it. Shows the state, the command executed, the container’s and image’s IDs, a timestamp, as well as the container’s own terminal:



Container: mariadb_container

Id: 6a5f8ecfb371f47aefb945f99a062076001b355ee44f692fb660460bddc08667

Created: 2017-06-20T12:10:16.792690907Z

Image: sha256:084a75509e01d6ea92664b47f5aaa7758f08e0f3487a1b5d86734860790b2802

Command: container-entrypoint run-mysqld

State: Up since 2017-06-20T12:10:17.574543433Z

Restart Policy: No

IP Address: 172.17.0.2

IP Prefix Length: 16

Gateway: 172.17.0.1

MAC Address: 02:42:ac:11:00:02

Memory usage: 138.7 MiB

CPU usage: 0% 1024 shares

```
---> 12:10:24     Sourcing post-init.sh ...
---> 12:10:24     Shutting down MySQL ...
170620 12:10:25 [Note] InnoDB: Waiting for page_cleaner to finish flushing of b
uffer pool
170620 12:10:27 [Note] InnoDB: Shutdown completed; log sequence number 1623589
170620 12:10:27 [Note] /opt/rh/rh-mariadb100/root/usr/libexec/mysqld: Shutdown
complete
---> 12:10:27     Cleaning up environment variables MYSQL_USER, MYSQL_PASSWORD,
 MYSQL_DATABASE and MYSQL_ROOT_PASSWORD ...
---> 12:10:27     Running final exec -- Only MySQL server logs after this point
170620 12:10:27 [Note] /opt/rh/rh-mariadb100/root/usr/libexec/mysqld (mysqld 10
.0.28-MariaDB) starting as process 1 ...
2017-06-20 12:10:27 7fc204b35880 InnoDB: Warning: Using innodb_additional_mem_p
ool_size is DEPRECATED. This option may be removed in future releases, together
 with the option innodb_use_sys_malloc and with the InnoDB's internal memory al
```

Click on an image to inspect it. Shows the image’s ID, entrypoint and command, and a list of containers based on that image. You can also delete the image from here or run a container from it.

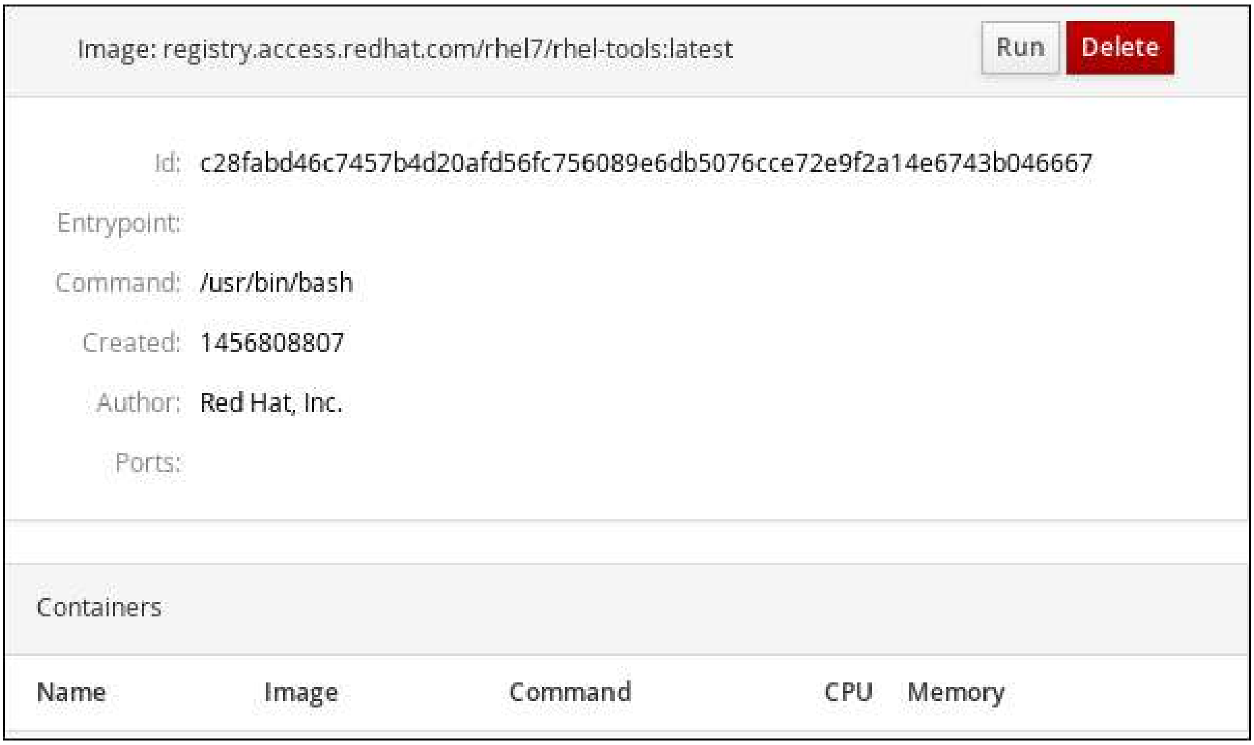


Image: registry.access.redhat.com/rhel7/rhel-tools:latest

Id: c28fabd46c7457b4d20afd56fc756089e6db5076cce72e9f2a14e6743b046667

Entrypoint:

Command: /usr/bin/bash

Created: 1456808807

Author: Red Hat, Inc.

Ports:

Name	Image	Command	CPU	Memory

Run a container. To run a container from an image, either click the triangle button from the right side of the list or choose the image first and then click "Run" from the top right corner. You can then enter the required data for the new container in the following dialog:

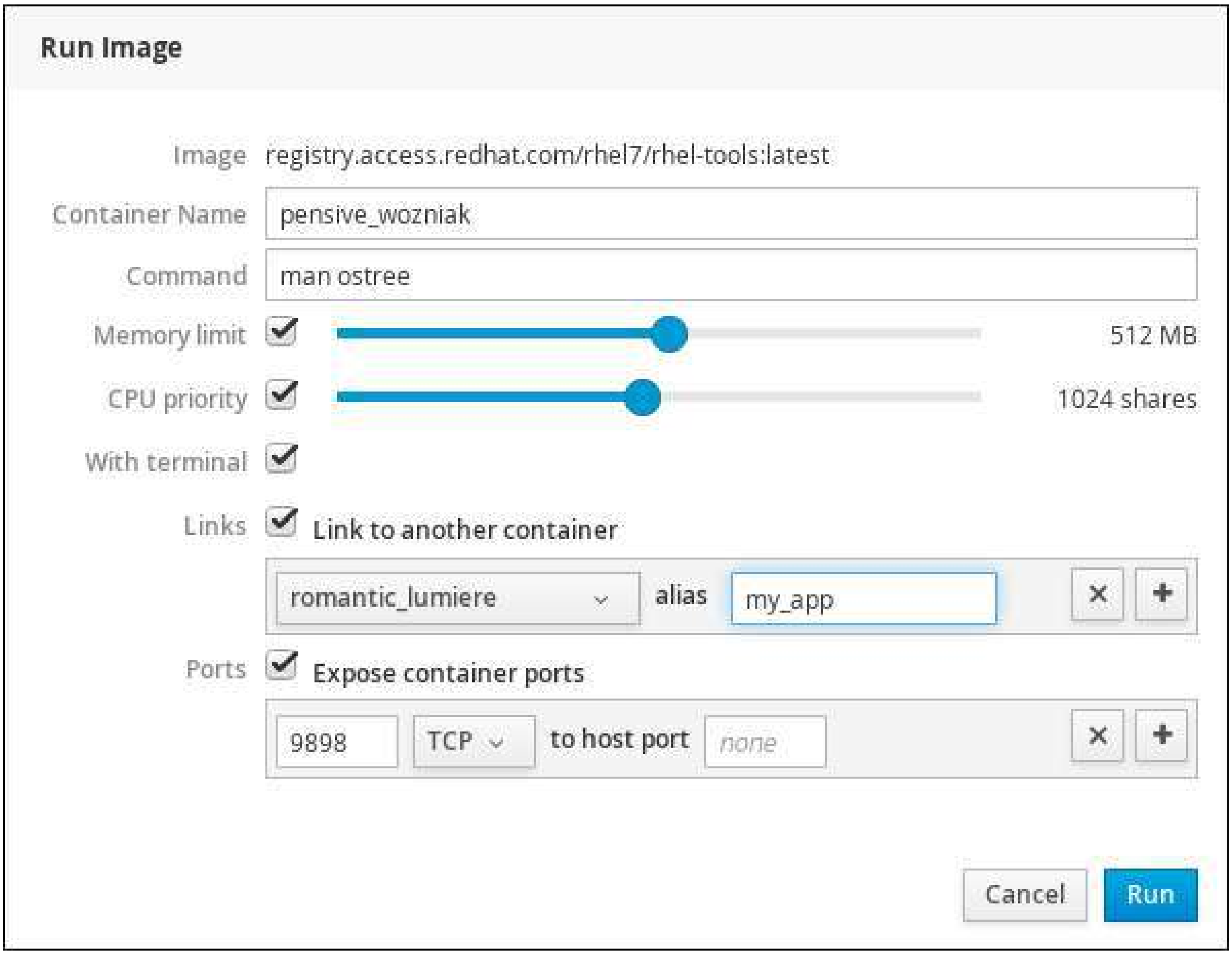


Image registry.access.redhat.com/rhel7/rhel-tools:latest

Container Name pensive_wozniak

Command man ostree

Memory limit ☑ 512 MB

CPU priority ☑ 1024 shares

With terminal ☑

Links ☑ Link to another container

romantic_lumiere alias my_app

Ports ☑ Expose container ports

9898 TCP to host port none

You can select which command the container should run, and you can also link that container to other containers, which will allow them to interact. Exposing ports for specific services to be visible from the host is also possible.

4.3. CHANGING THE COCKPIT PORT ON ATOMIC HOST

To change the Cockpit port on Atomic Host:

```
atomic run rhel7/cockpit-ws --port 9898
```

4.4. ENABLING MORE COCKPIT FEATURES ON ATOMIC HOST

You can add more Cockpit features by installing additional **`cockpit-*`** packages using package layering.