Red Hat Enterprise Linux Atomic Host 7
Getting Started with Cockpit

Getting Started with Cockpit

Red Hat Atomic Host Documentation Team

Getting Started with Cockpit

Legal Notice

Abstract

Get started using the Cockpit system administration tool for your servers

Table of Contents

CHAPTER 1. OVERVIEW

Cockpit is a system administration tool that provides a user interface for monitoring and administering servers through a web browser. It allows you to monitor current values and adjust limits on system resources, control life cycle on container instances, and manipulate container images. Here are a few important facts about Cockpit:

- Cockpit does not add a layer of other functionalities that are not present on your systems. It exposes user interface elements that enable you to interact with the system.
- Cockpit does not take control over your servers, in a way that when you configure something from Cockpit, you can only manage it from there. You can effectively move away from Cockpit to the command-line and come back to it at any point.
- Cockpit does not require configuration or infrastructure, and once you install it, it is ready for use. You could, however, configure it to make use of the authentication infrastructure that is available to you, for example a single sign-on system like Kerberos.
- Cockpit has zero memory and process footprint on the server when not in use.
- Cockpit does not store data or policy. This also means it does not have its own users. The users from the systems can authenticate in Cockpit using their system credentials and they keep the same permissions.
- Cockpit dynamically updates itself to reflect the current state of the server, within a time frame of a few seconds.
- Cockpit is not intended for configuration management. This means that Cockpit itself does not have a predefined template or state for the server that it then imposes on the server. Cockpit can interact with other configuration management systems or custom tools that are manipulating server configuration.

This document provides instructions on how to install and enable Cockpit so you can monitor your servers, describes basic configuration, and walks you through the interface.

Both Red Hat Enterprise Linux and Red Hat Enterprise Linux Atomic Host can be used for the role of a Cockpit server and that of a secondary server. In this document, all monitored systems are Atomic, but the instructions also cover how to set up Red Hat Enterprise Linux as a primary server.

Note

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

CHAPTER 2. INSTALLING AND ENABLING COCKPIT

2.1. SETTING UP A COCKPIT SERVER

A Cockpit server is the machine that is running the cockpit service and exposes the user interface. Depending on the operating system, you need to install the *cockpit* packages or the *cockpit-ws* container. You can then open the interface in a browser by typing *localhost:9090*, or use any other machine and type in the IP address of the Cockpit server. Through Cockpit, you can also add more secondary hosts to this primary server. They need to have the cockpit packages installed on them. This document refers to the Cockpit server as the primary server and the added hosts as secondary.

2.1.1. Installing Cockpit

A. On Red Hat Enterprise Linux Atomic Host

1. Run the **cockpit-ws** image. Use this command:

```
-bash-4.2# atomic run rhel7/cockpit-ws
```

Afterwards, you can log into Cockpit. Go to Opening The Interface

B. On Red Hat Enterprise Linux

1. Once you have Red Hat Enterprise Linux installed and with enabled networking, you need to register the system and enable the Extras and Optional repositories:

```
# subscription-manager register --auto-attach --username=
<rhnuser> --password=<rhnpasswd>
# subscription-manager repos --enable=rhel-7-server-extras-rpms
# subscription-manager repos --enable=rhel-7-server-optional-rpms
```

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

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

3. Install the *cockpit* packages:

```
$ sudo yum install cockpit
```

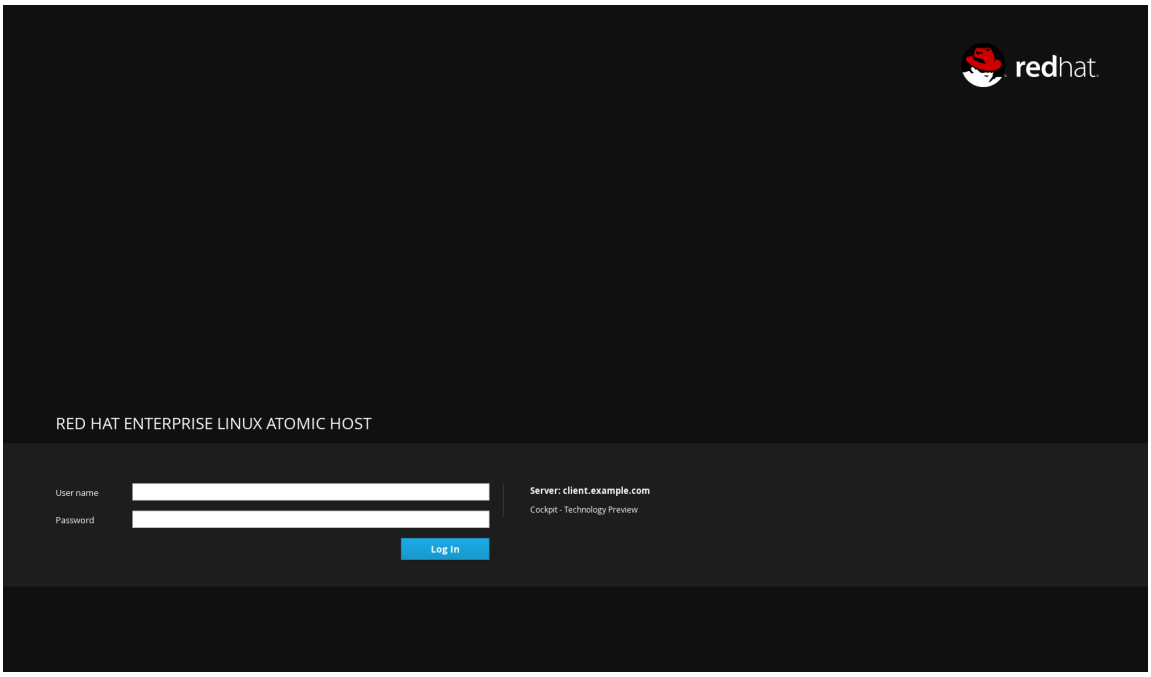
4. Enable and start the cockpit.socket service:

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

2.1.2. 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*. If you get

a security warning by the browser, you will need to add this connection to the security exceptions. Click **Advanced → Add Exception → Confirm Security Exception**. After that, you will see the login screen:

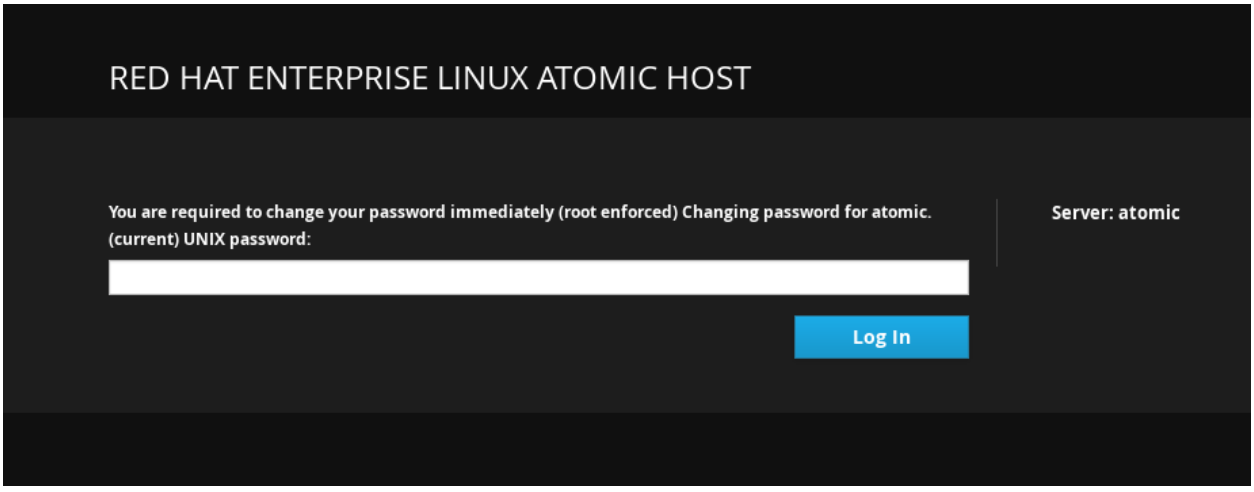


2. Log into the Cockpit interface with the same user name and password that you would use to log into the Atomic system.

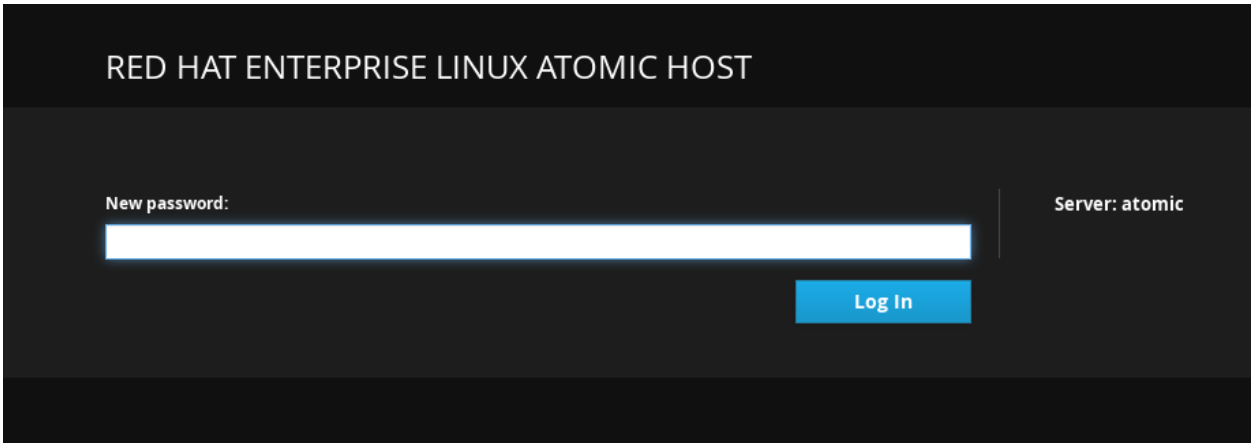
2.1.3. Changing Expired Passwords

If there is an account on your Atomic system that has an expired password, you can change it from Cockpit. For example, if you have provisioned your system using **cloud-init** to set up an expired password, you will be prompted to change it the first time you log into the system. It can also be used by system administrators who want to make sure the user changes his password on the first login.

When you try to log in with the usual password and that password has expired, Cockpit will prompt you to enter the current password again. Enter your current password and click **Login**.



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



Note

If you can't log into Cockpit and you are not redirected to the changing password screen, check the **`/etc/ssh/sshd_config`** file on the Cockpit Server and make sure the **`ChallengeResponseAuthentication`** line is set to **`yes`**. After that, restart **`sshd`** with the **`systemctl restart sshd`** command.

2.1.4. SSH two-factor authentication with Cockpit

Cockpit now supports two-factor authentication so if you have protected your SSH server with such configuration, the Cockpit login screen will prompt you to enter your password and PIN pair. To set up SSH for two-factor authentication you need two components:

- Your company's authenticator application that provides one-time passwords or PIN numbers. An example application is the **`Google Authenticator`**, which also has its own PAM (Pluggable Authentication Module).
- A server that validates the PINs from your dongle.

These two components can be built in many different ways depending on the infrastructure of your particular company. When you have these two set up, you will need to do the following things:

1. Enter the following line in the **`/etc/pam.d/sshd`** file as the last **`auth`** line:

```
auth       required         <your_PAM_module>
```

2. Edit the **`/etc/ssh/sshd_config`** file so that the **`ChallengeResponseAuthentication`** line is set to **`yes`**.
3. Restart the **`sshd`** service with the **`systemctl restart shhd`** command.

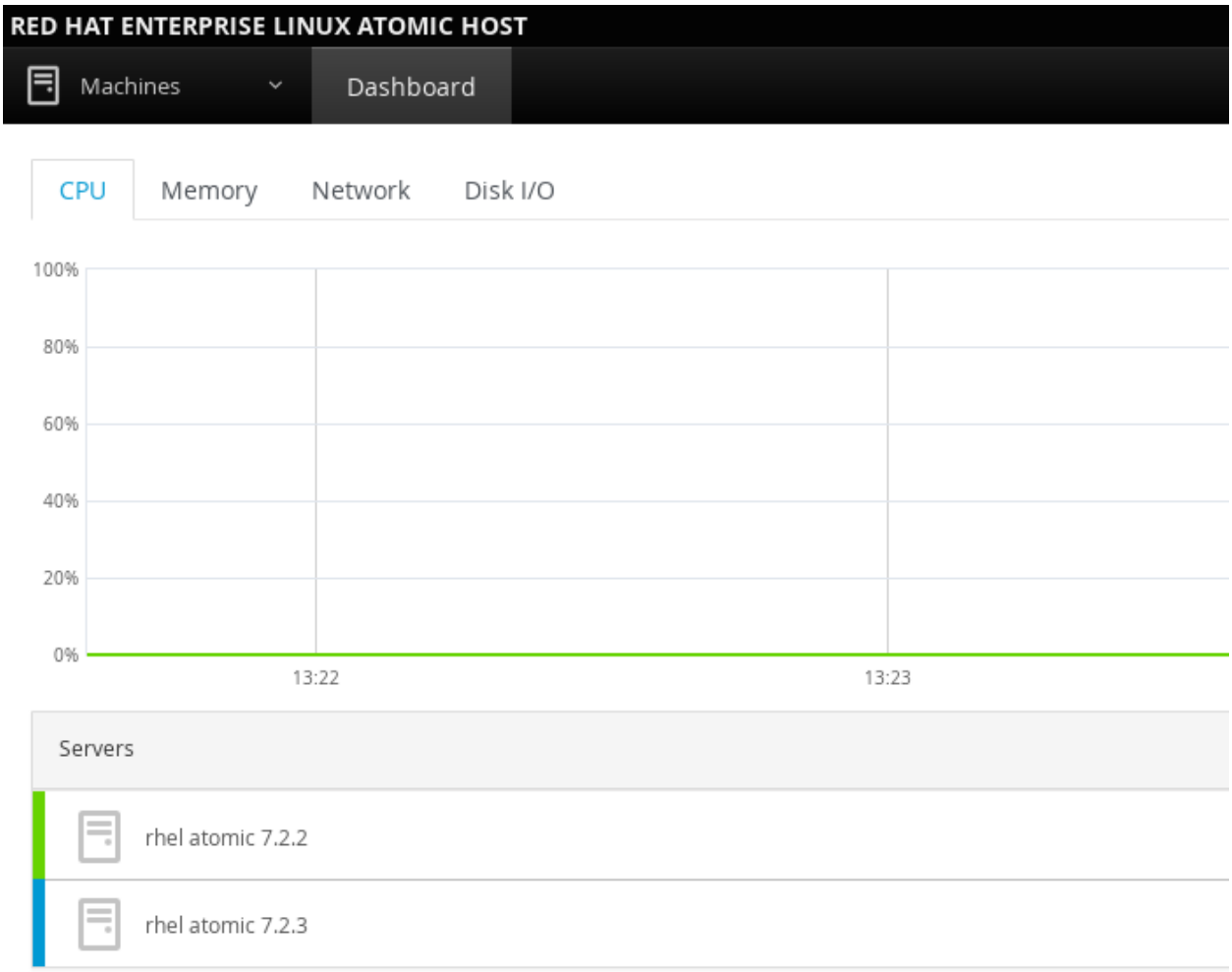
When you open Cockpit's interface, and enter your password, you will then be prompted to enter your Verification code:

CHAPTER 3. USING COCKPIT

3.1. GETTING TO KNOW THE COCKPIT INTERFACE

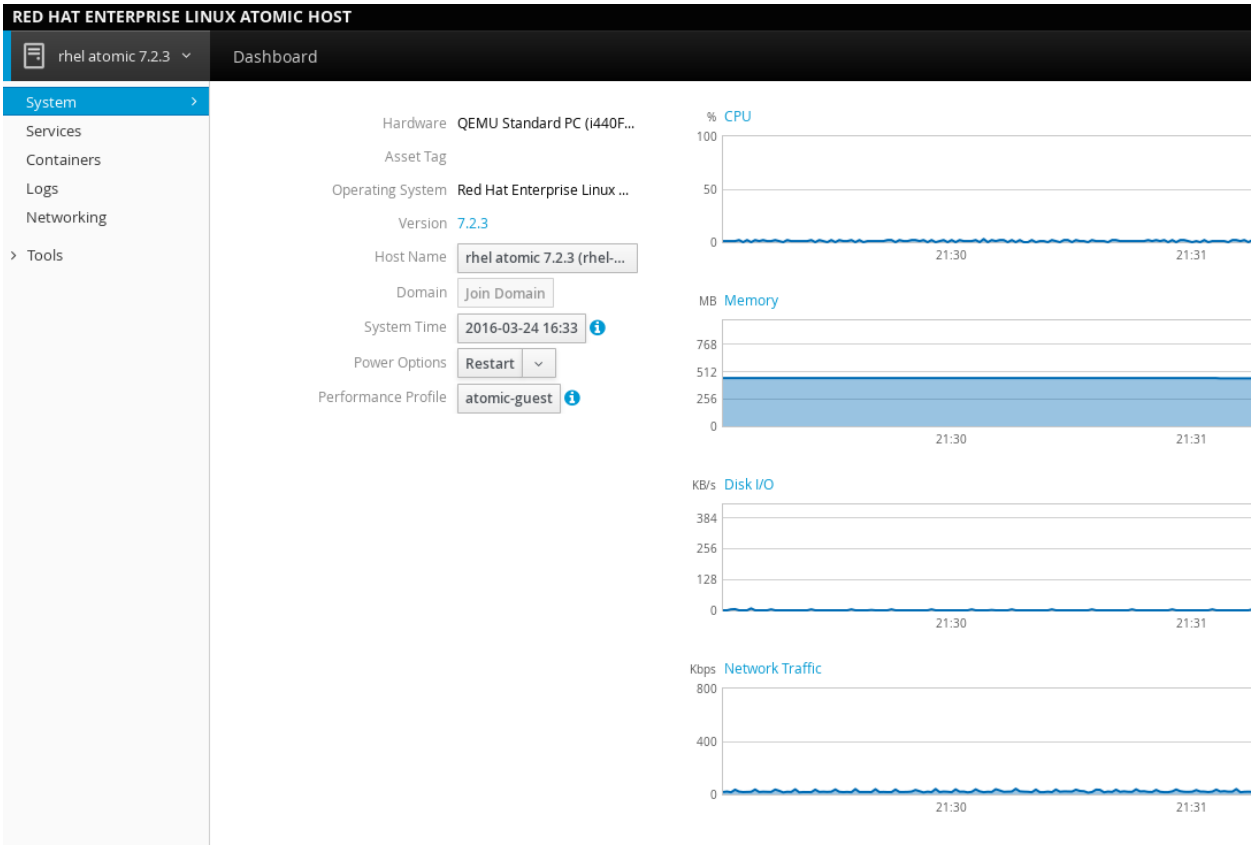
Once you have logged in, you will see the tabs for the Dashboard and the individual machines added to Cockpit.

Dashboard: Shows a list of all systems added to the Cockpit server with graphs for CPU usage, memory usage, disk I/O, and network traffic.

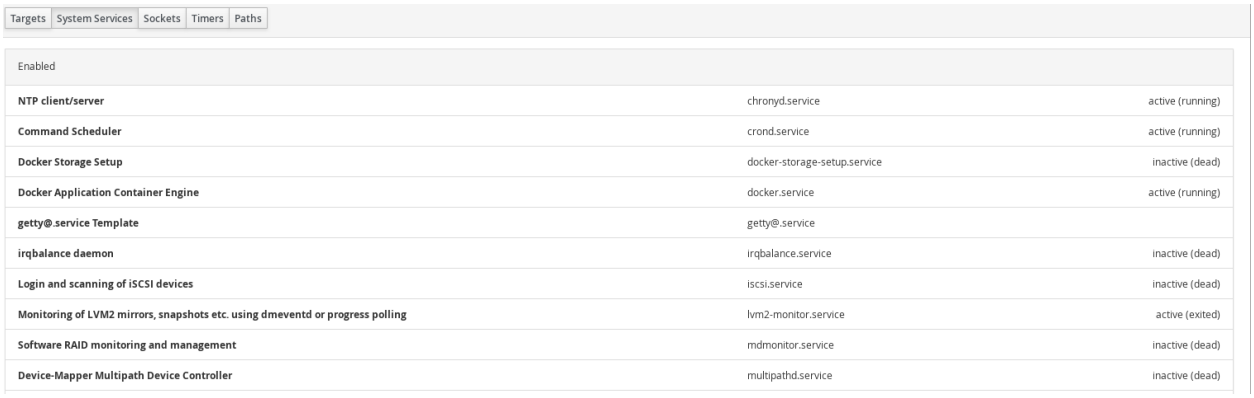


You can then select a system name, in this case "rhel atomic 7.2.3", and have a look at the side menu:

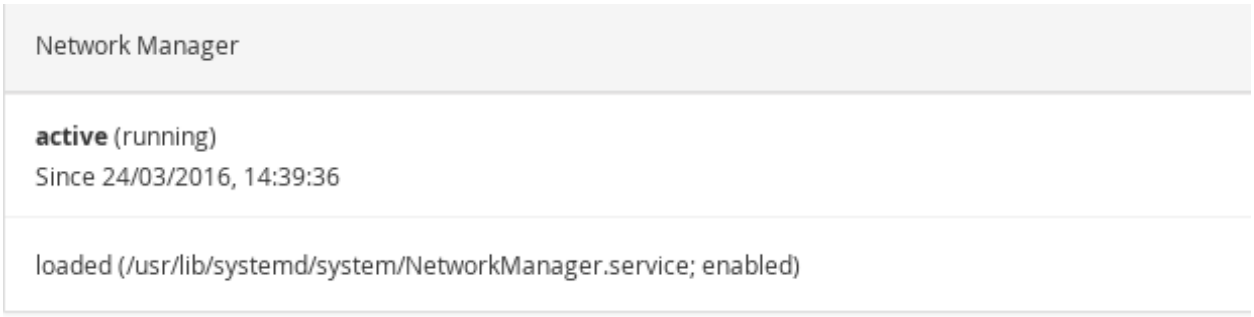
System: Shows information about the system that Cockpit is running on. This includes CPU usage, memory usage, disk I/O, and network traffic, as well as hardware and operating system details.

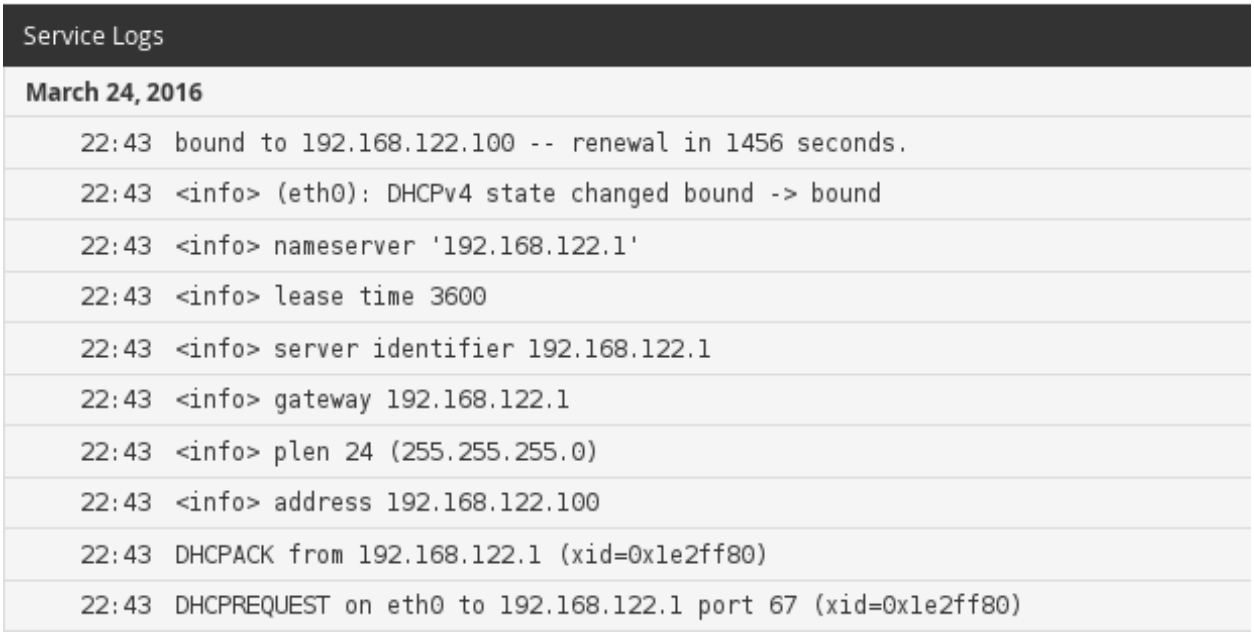


Services: Shows the systemd services running on the Cockpit server. You can see which are active/enabled or inactive. You can also see other systemd features: Targets, sockets, timers, and paths.

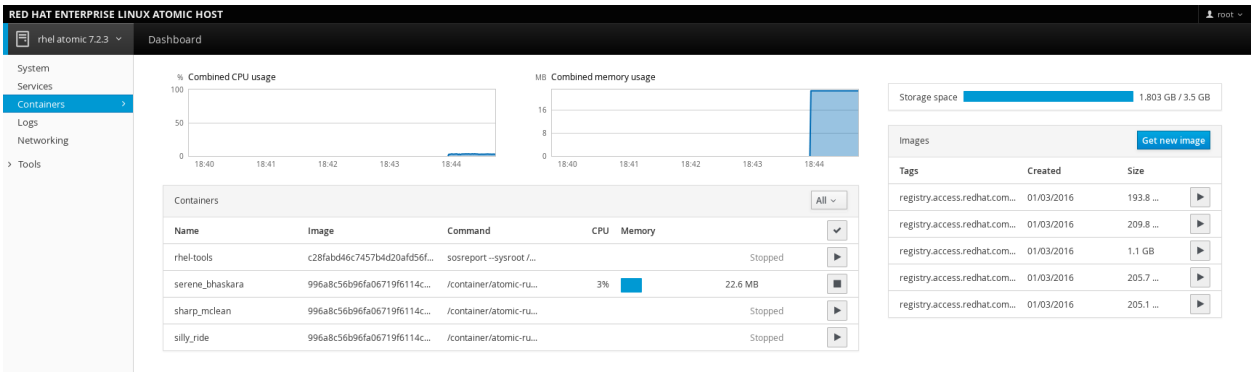


Select a service to view its details:

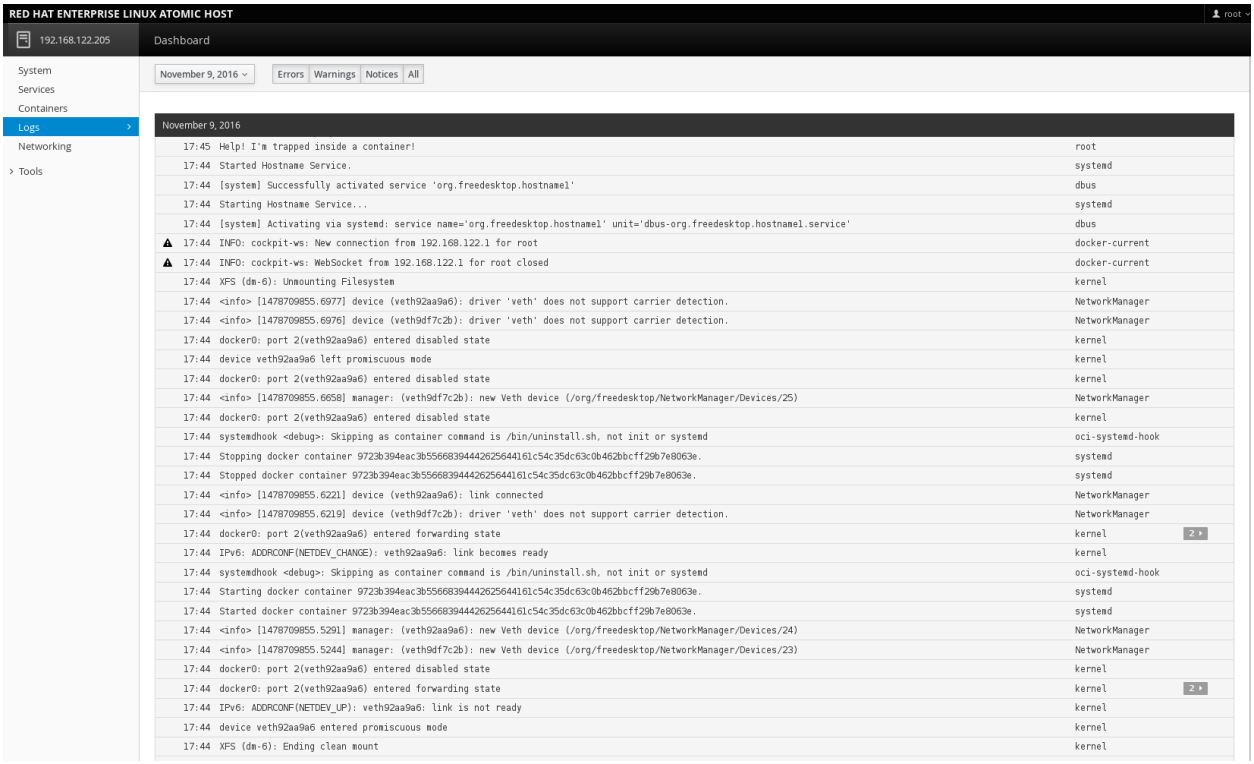




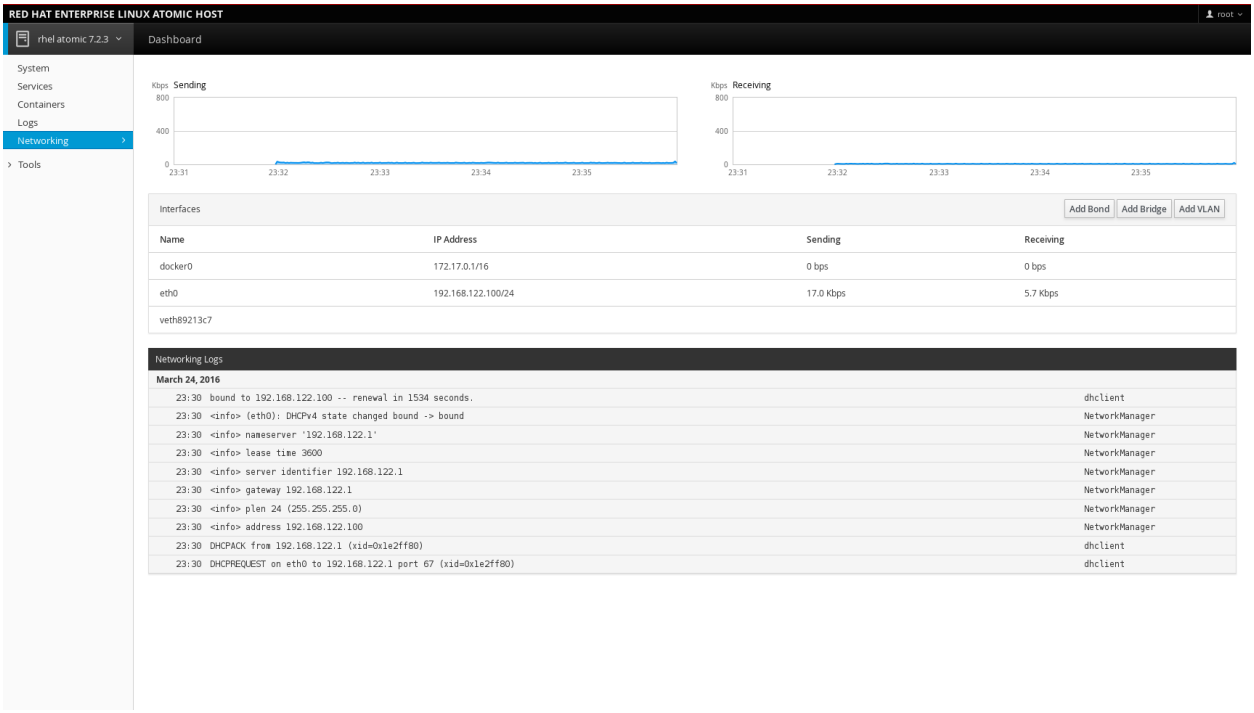
Containers: Lists all images available on the system, all running and non-running containers, combined CPU & memory usage graphs, and a storage usage bar.



Logs: See messages produced by the systemd journal. These are errors, warnings, and notices that are generated by systemd services and gathered by the journal (like the output of the journalctl command). Errors are listed by the date they occurred. You can also view warnings, notices, or all messages.

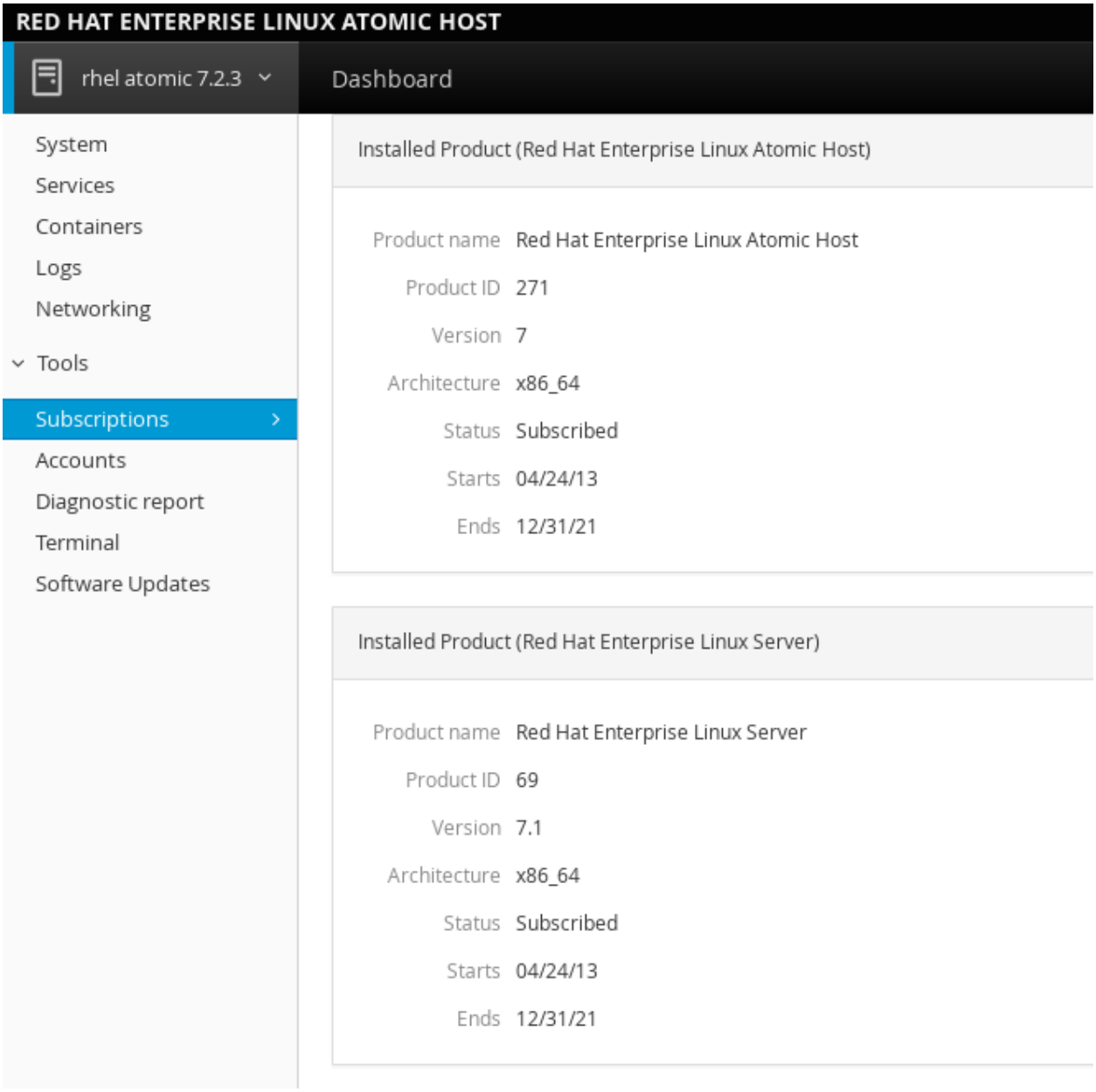


Networking: See networking interfaces (eth0, docker0, etc.) as well as the amount of data being sent and received.

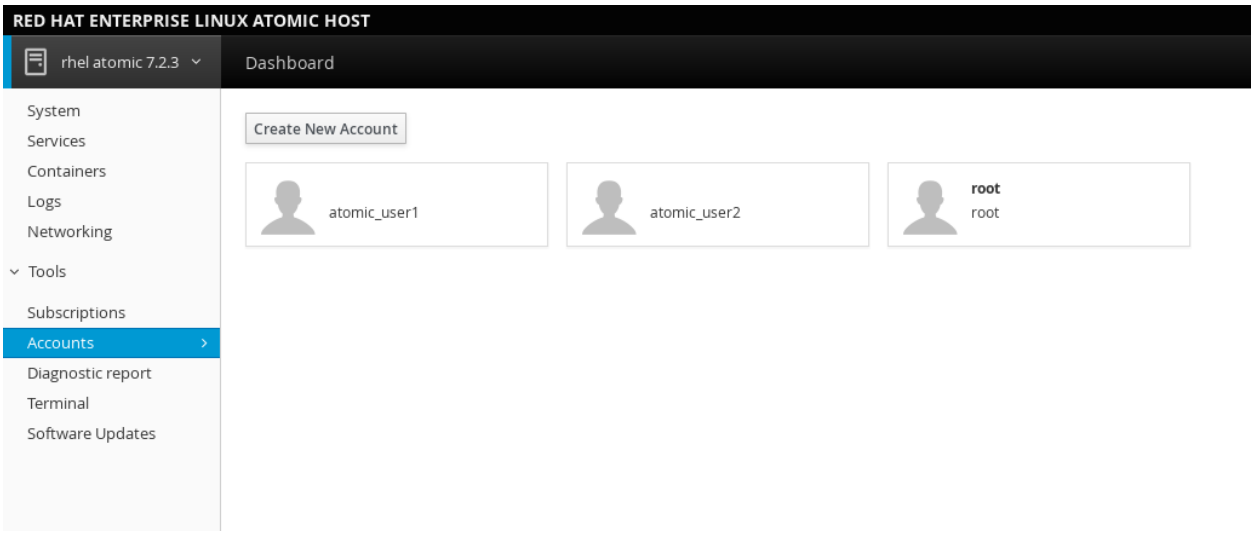


Tools: View other system information:

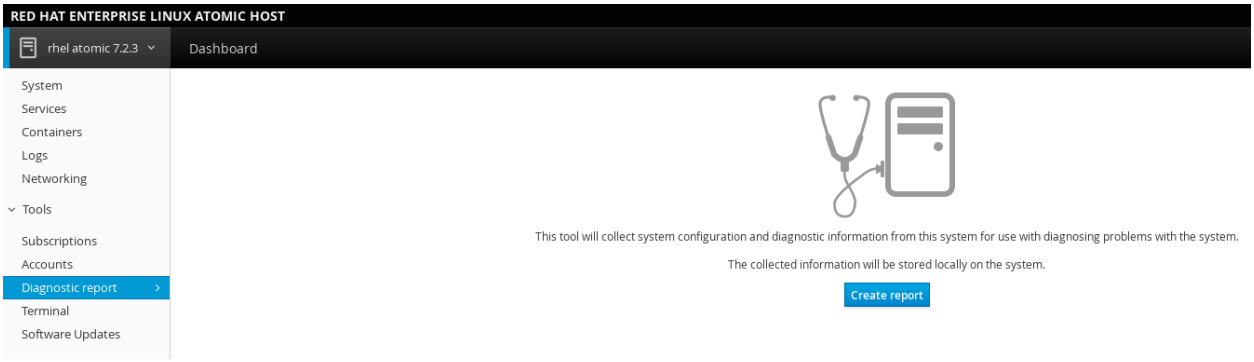
- **Subscriptions**: Displays what Red Hat products are installed and subscribed.



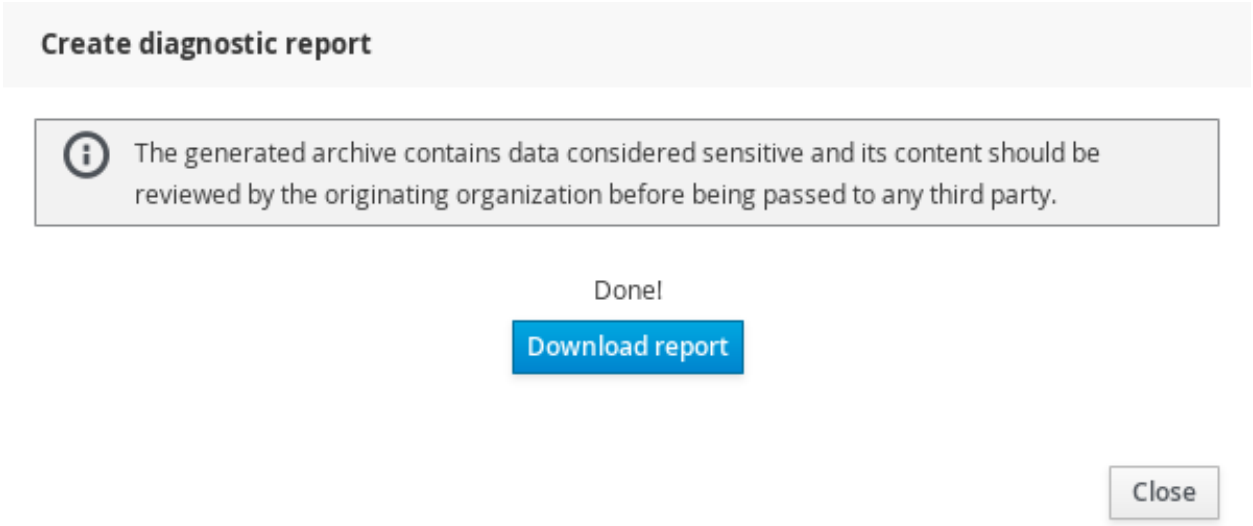
- **Accounts**: Shows which administrative (root) and other users (atomic_user1, atomic_user2) have accounts on the system.



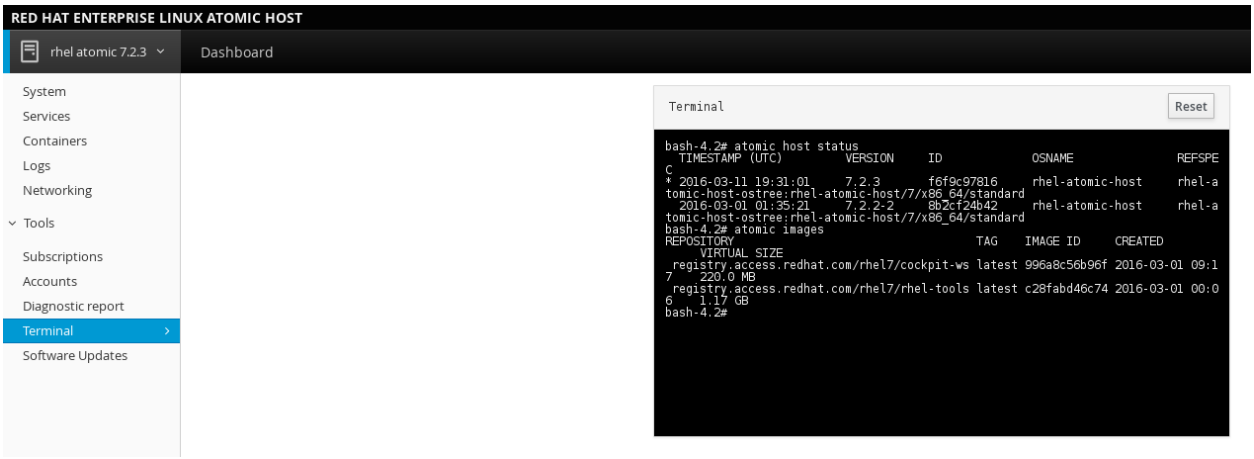
- **Diagnostic report**: Collects system configuration and diagnostics information and prepares a report in an XZ format.



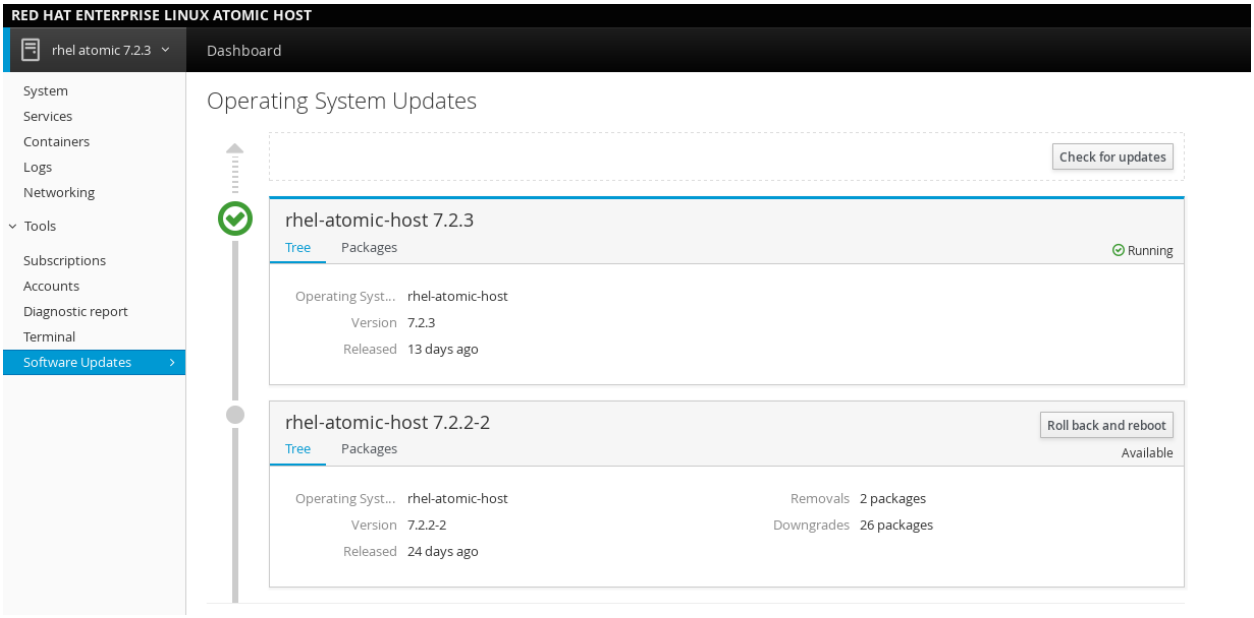
You can then download the report locally on your system:



- **Terminal**: Opens a Terminal (command line) session to the Cockpit system. From there, you can run any commands available to the user you are logged in as. For example, as root, you could run docker or kubectl commands.



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



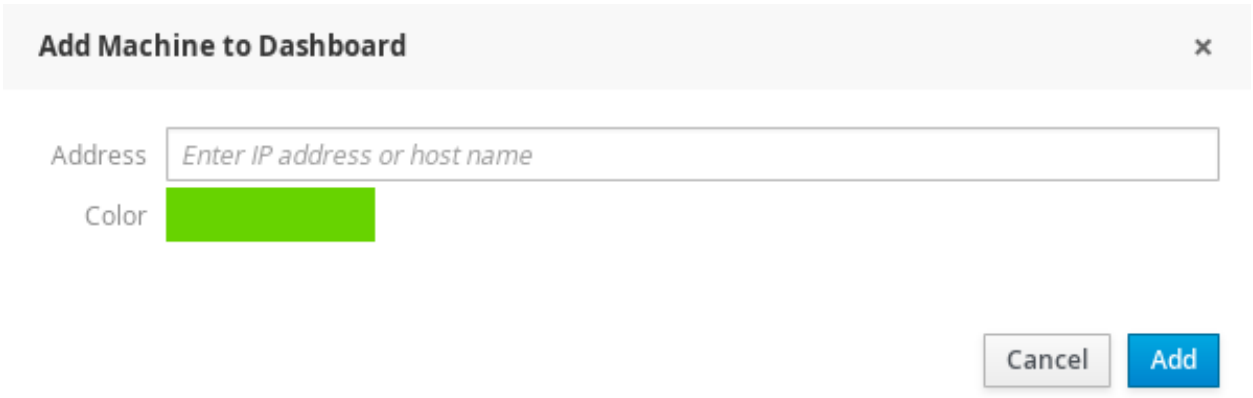
3.1.1. Adding another system to monitor

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

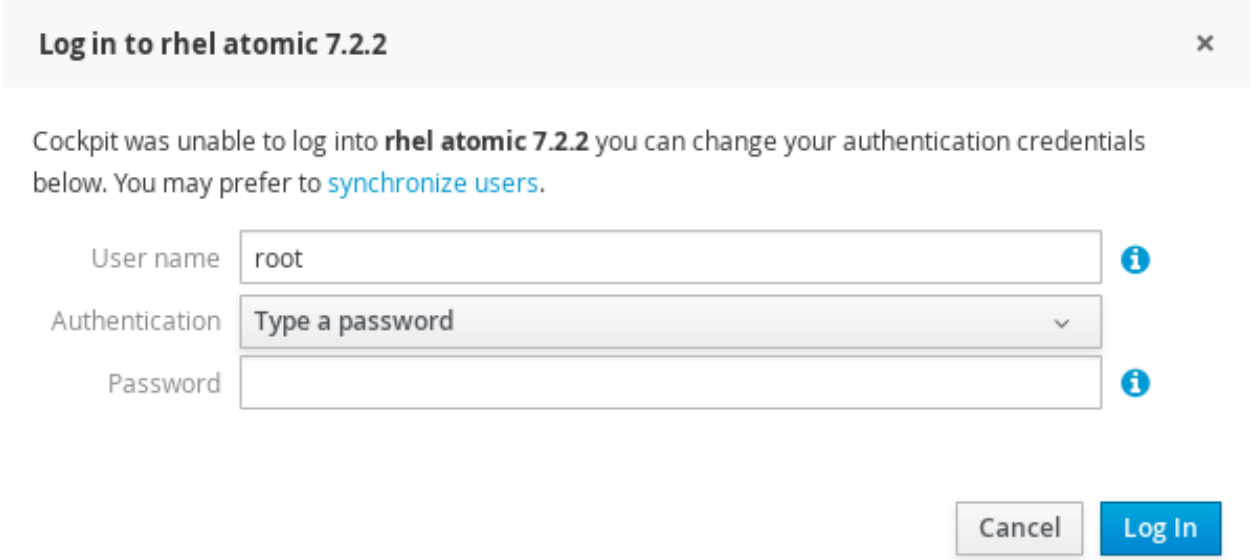
- The Cockpit packages installed.
- An SSH server running and available on port 22 that supports password or key-based authentication.

The cockpit-ws component is not necessary on these additional systems.

From the "Dashboard" tab next to the system name, choose the "plus" button to add a new host. You can then add the IP of the secondary machine and choose which color will represent it in the user interface.



Select the user name and type in the password:

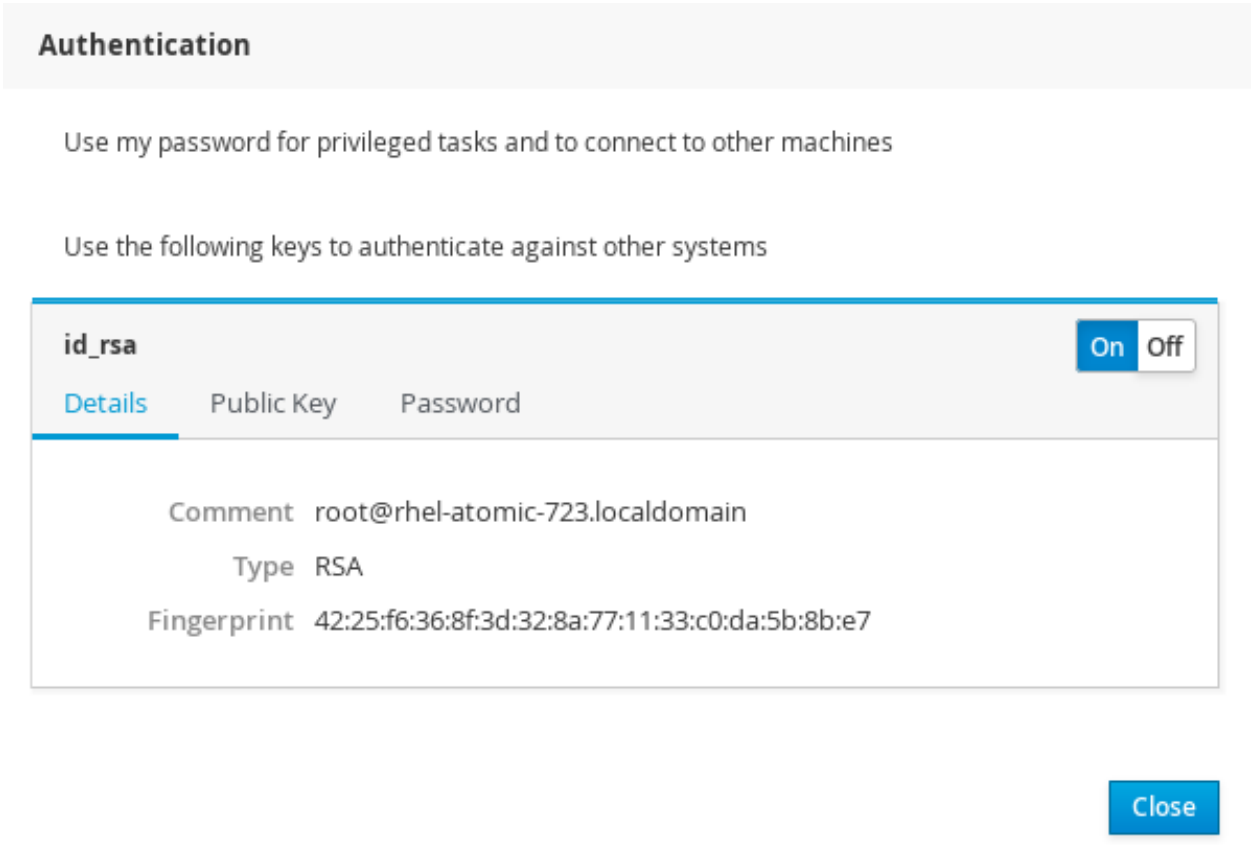


Configuring Key-Based Authentication

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.



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 to other systems through Cockpit

On the login screen, you can also choose an alternate host to connect to. Type in your username and password from that alternate host, then click **Other Options**, in the entry field type the IP address of the new host, and click **Log In**. You will be prompted for the SSH fingerprint, click **Log In** again, and you will be able to browse the new system. Cockpit uses SSH to authenticate you against that host, and you do not need to configure anything additionally on the new system. As a prerequisite, it will need to have SSH listening on port 443, and the **`cockpit-bridge`** package installed and the same version as in the Cockpit server.

Note

As a prerequisite, it will need to have SSH listening on port 443, and the **`cockpit-bridge`** package installed and the same version as in the Cockpit server. If the new machine is not known to Cockpit, and you get the **Refusing to connect. Host is unknown** use the following command to allow connections from unknown hosts:

+

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

3.2. LOGGING INTO A SYSTEM VIA A BASTION HOST

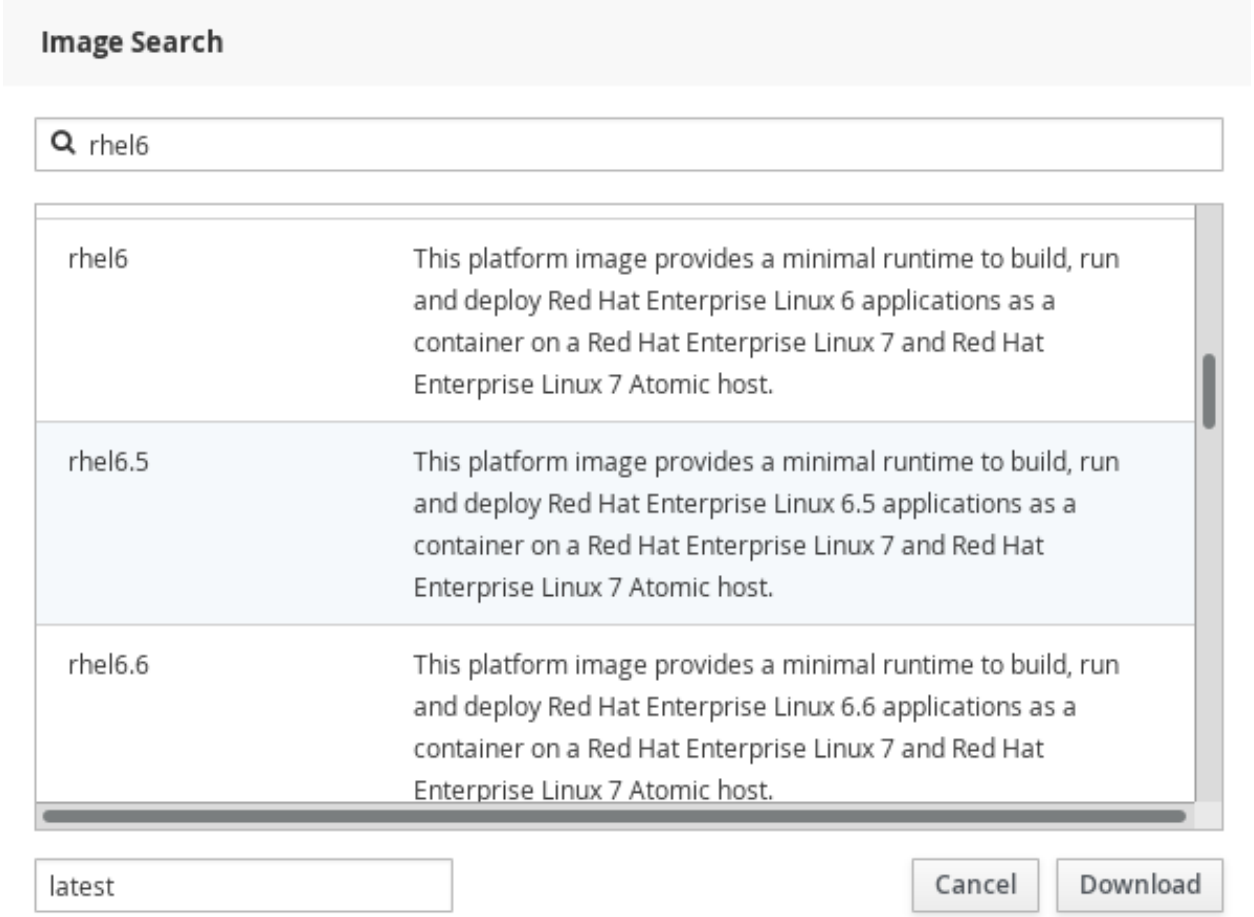
On the Cockpit login screen you can now choose an alternate host to connect to. Cockpit with use SSH to authenticate you against that host, and display the admin interface for that host.

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, and other hosts can only have SSH accessible on the usual port 22.

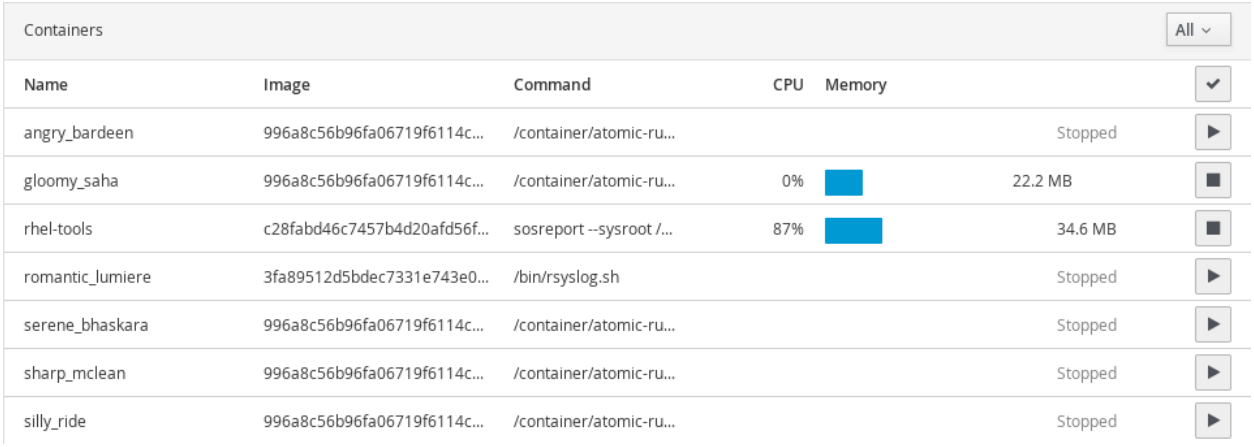
3.2.1. Working with containers

The **Containers** tab presents you with a UI to interact with your 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.

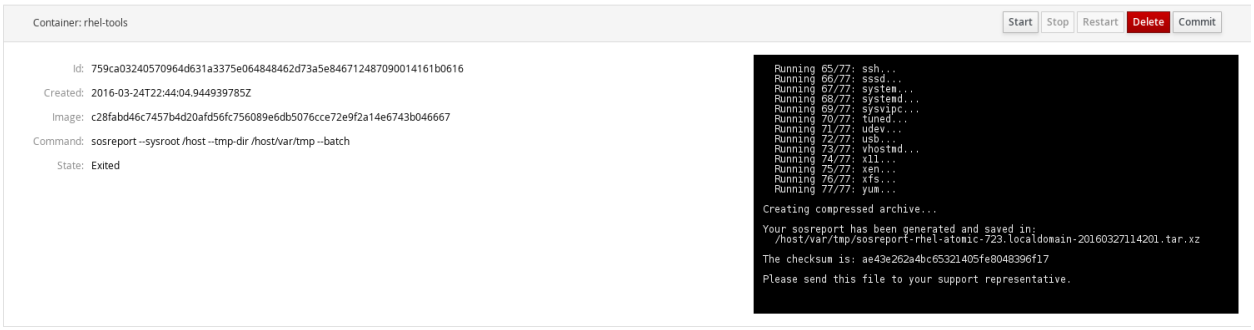
- **Download an image.** Click the "Get new image" button from the images list to the right and enter an image name or a keyword. Choose an image and click "Download".



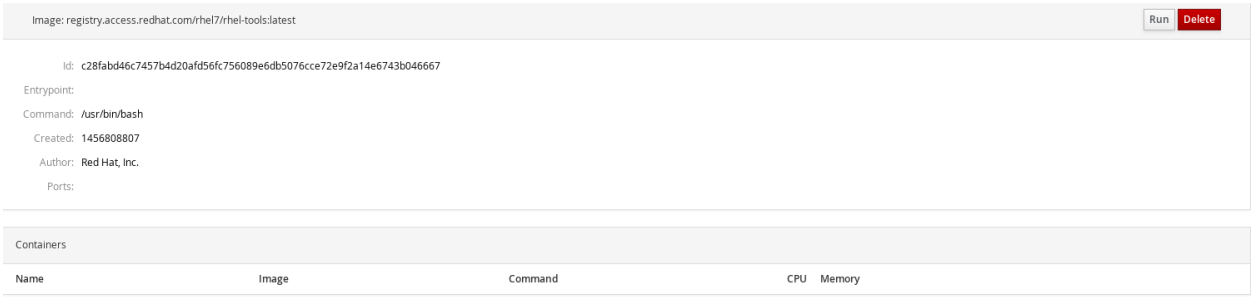
- **Starting and stopping containers.** From the "Containers" list, you can start and stop containers using the buttons on the right-hand side. Use the drop-down menu to see all or filter out the non-running containers.



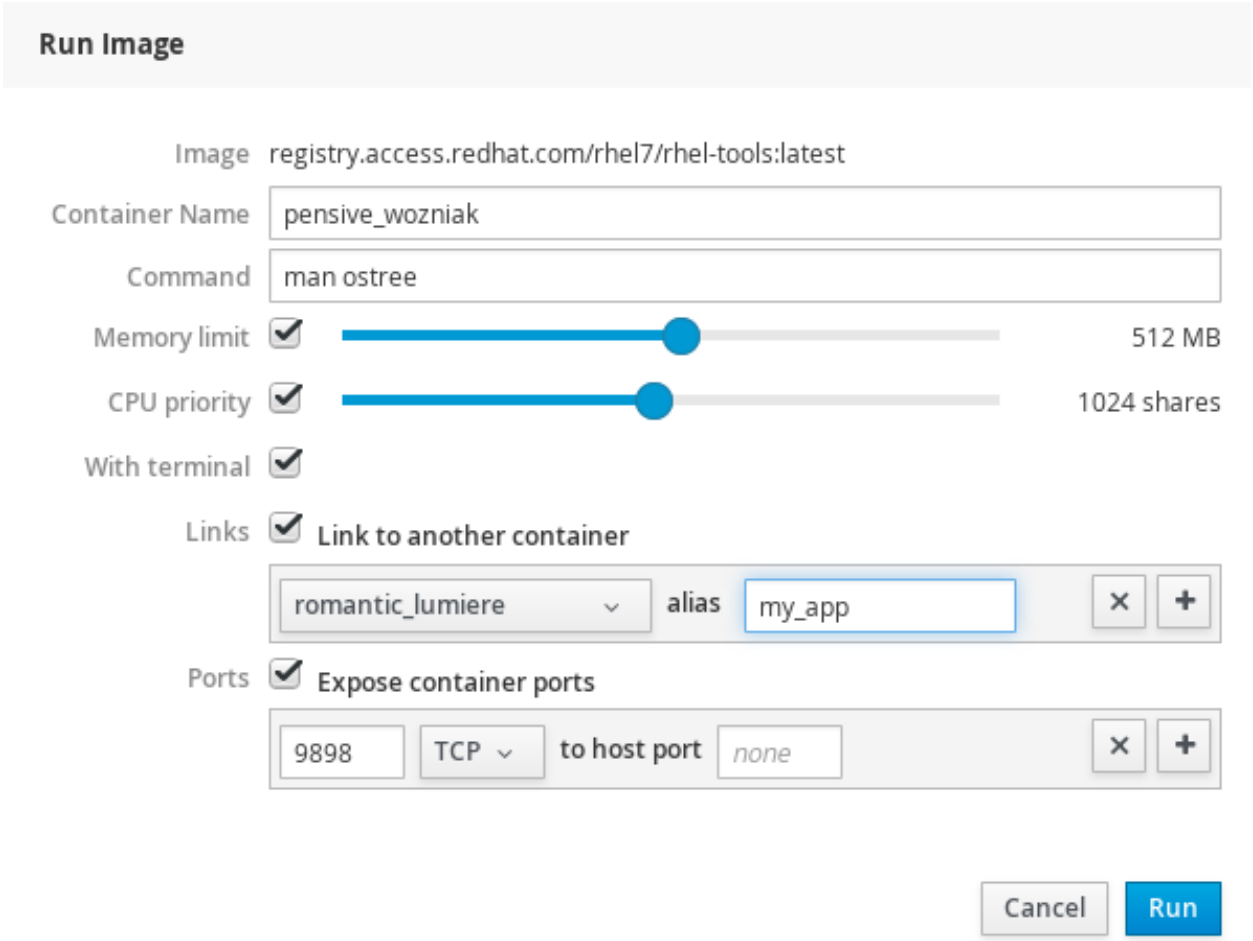
- **Click on a container to inspect it.** Shows the state, the command executed, the container's and image's IDs, a timestamp, as well as the container's own terminal:



- **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.



- **Run a container**. To run a container from an image, either click the triangle button from the right-hand side of the list or choose the image first and then click "Run" from the top right corner. A dialog is displayed where you can enter the required data for the new container:



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. In addition, you can expose a port when you want a specific service to be visible from the host.

3.2.2. Changing the port

A. On Red Hat Enterprise Linux Atomic Host:

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

B: On Red Hat Enterprise Linux:

Create the */etc/systemd/system/websocket.cockpit.d/listen.conf* file and, if needed, the preceding directories.

```
$ mkdir /etc/systemd/system/websocket.cockpit.d/
$ touch /etc/systemd/system/websocket.cockpit.d/listen.conf
```

The file should have the following content:

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

Next, allow the new port through the firewall:

```
$ sudo firewall-cmd --add-port=9898/tcp
$ sudo firewall-cmd --permanent --add-port=9898/tcp
```

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
```

In order for the changes to 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.