



# Red Hat Enterprise Linux 9

## Managing systems using the RHEL 9 web console

Server management with a graphical web-based interface



# Red Hat Enterprise Linux 9 Managing systems using the RHEL 9 web console

---

Server management with a graphical web-based interface

## Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

The RHEL web console is a web-based graphical interface, which is based on the upstream Cockpit project. By using it, you can perform system administration tasks, such as inspecting and controlling systemd services, managing storage, configuring networks, analyzing network issues, and inspecting logs.

# Table of Contents

<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b>	<b>4</b>
<b>CHAPTER 1. GETTING STARTED USING THE RHEL WEB CONSOLE</b>	<b>5</b>
1.1. WHAT IS THE RHEL WEB CONSOLE	5
1.2. INSTALLING AND ENABLING THE WEB CONSOLE	6
1.3. LOGGING IN TO THE WEB CONSOLE	6
1.4. DISABLING BASIC AUTHENTICATION IN THE WEB CONSOLE	7
1.5. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE	8
1.6. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE AS A ROOT USER	9
1.7. LOGGING IN TO THE WEB CONSOLE USING A ONE-TIME PASSWORD	9
1.8. ADDING A BANNER TO THE LOGIN PAGE	10
1.9. CONFIGURING AUTOMATIC IDLE LOCK IN THE WEB CONSOLE	11
1.10. CHANGING THE WEB CONSOLE LISTENING PORT	12
<b>CHAPTER 2. INSTALLING AND CONFIGURING WEB CONSOLE BY USING RHEL SYSTEM ROLES</b>	<b>14</b>
2.1. INSTALLING THE WEB CONSOLE BY USING THE COCKPIT RHEL SYSTEM ROLE	14
<b>CHAPTER 3. INSTALLING WEB CONSOLE ADD-ONS AND CREATING CUSTOM PAGES</b>	<b>16</b>
3.1. ADD-ONS FOR THE RHEL WEB CONSOLE	16
3.2. CREATING NEW PAGES IN THE WEB CONSOLE	16
3.3. OVERRIDING THE MANIFEST SETTINGS IN THE WEB CONSOLE	17
<b>CHAPTER 4. MANAGING SOFTWARE UPDATES IN THE WEB CONSOLE</b>	<b>19</b>
4.1. MANAGING MANUAL SOFTWARE UPDATES IN THE WEB CONSOLE	19
4.2. MANAGING AUTOMATIC SOFTWARE UPDATES IN THE WEB CONSOLE	19
4.3. MANAGING ON-DEMAND RESTARTING AFTER APPLYING SOFTWARE UPDATES IN THE WEB CONSOLE	20
4.4. APPLYING PATCHES WITH KERNEL LIVE PATCHING IN THE WEB CONSOLE	21
<b>CHAPTER 5. MANAGING SUBSCRIPTIONS IN THE WEB CONSOLE</b>	<b>23</b>
5.1. SUBSCRIPTION MANAGEMENT IN THE WEB CONSOLE	23
5.2. REGISTERING A SYSTEM BY USING THE WEB CONSOLE	23
5.3. REGISTERING SUBSCRIPTIONS WITH ACTIVATION KEYS IN THE WEB CONSOLE	24
<b>CHAPTER 6. MANAGING REMOTE SYSTEMS IN THE WEB CONSOLE</b>	<b>26</b>
6.1. REMOTE SYSTEM MANAGER IN THE WEB CONSOLE	26
6.2. ADDING REMOTE HOSTS TO THE WEB CONSOLE	27
6.3. ENABLING SSH LOGIN FOR A NEW HOST	28
6.4. CONFIGURING THE WEB CONSOLE TO ALLOW A USER AUTHENTICATED WITH A SMART CARD TO SSH TO A REMOTE HOST WITHOUT BEING ASKED TO AUTHENTICATE AGAIN	31
6.5. USING ANSIBLE TO CONFIGURE THE WEB CONSOLE TO ALLOW A USER AUTHENTICATED WITH A SMART CARD TO SSH TO A REMOTE HOST WITHOUT BEING ASKED TO AUTHENTICATE AGAIN	33
<b>CHAPTER 7. CONFIGURING SINGLE SIGN-ON FOR LOGGING IN TO THE RHEL WEB CONSOLE SERVICE RUNNING ON AN IDM CLIENT AND AN IDM SERVER</b>	<b>36</b>
7.1. JOINING A RHEL 9 SYSTEM TO AN IDM DOMAIN USING THE WEB CONSOLE	36
7.2. LOGGING IN TO THE WEB CONSOLE USING KERBEROS AUTHENTICATION	37
7.3. ENABLING THE RHEL WEB CONSOLE SINGLE SIGN-ON WITH GSSAPI ON IDM SERVERS	38
7.4. ENABLING SUDO ACCESS FOR IDM ADMINISTRATORS ON IDM HOSTS	39
<b>CHAPTER 8. CONFIGURING SMART CARD AUTHENTICATION WITH THE WEB CONSOLE FOR CENTRALLY MANAGED USERS</b>	<b>40</b>
8.1. SMART CARD AUTHENTICATION FOR CENTRALLY MANAGED USERS	40
8.2. INSTALLING TOOLS FOR MANAGING AND USING SMART CARDS	40

8.3. PREPARING YOUR SMART CARD AND UPLOADING YOUR CERTIFICATES AND KEYS TO YOUR SMART CARD	41
8.4. ENABLING SMART CARD AUTHENTICATION FOR THE WEB CONSOLE	43
8.5. LOGGING IN TO THE WEB CONSOLE WITH SMART CARDS	43
8.6. ENABLING PASSWORDLESS SUDO AUTHENTICATION FOR SMART CARD USERS	44
8.7. LIMITING USER SESSIONS AND MEMORY TO PREVENT A DOS ATTACK	46
8.8. ADDITIONAL RESOURCES	46
<b>CHAPTER 9. SATELLITE HOST MANAGEMENT AND MONITORING IN THE WEB CONSOLE .....</b>	<b>47</b>



## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

### Submitting feedback through Jira (account required)

1. Log in to the [Jira](#) website.
2. Click **Create** in the top navigation bar
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

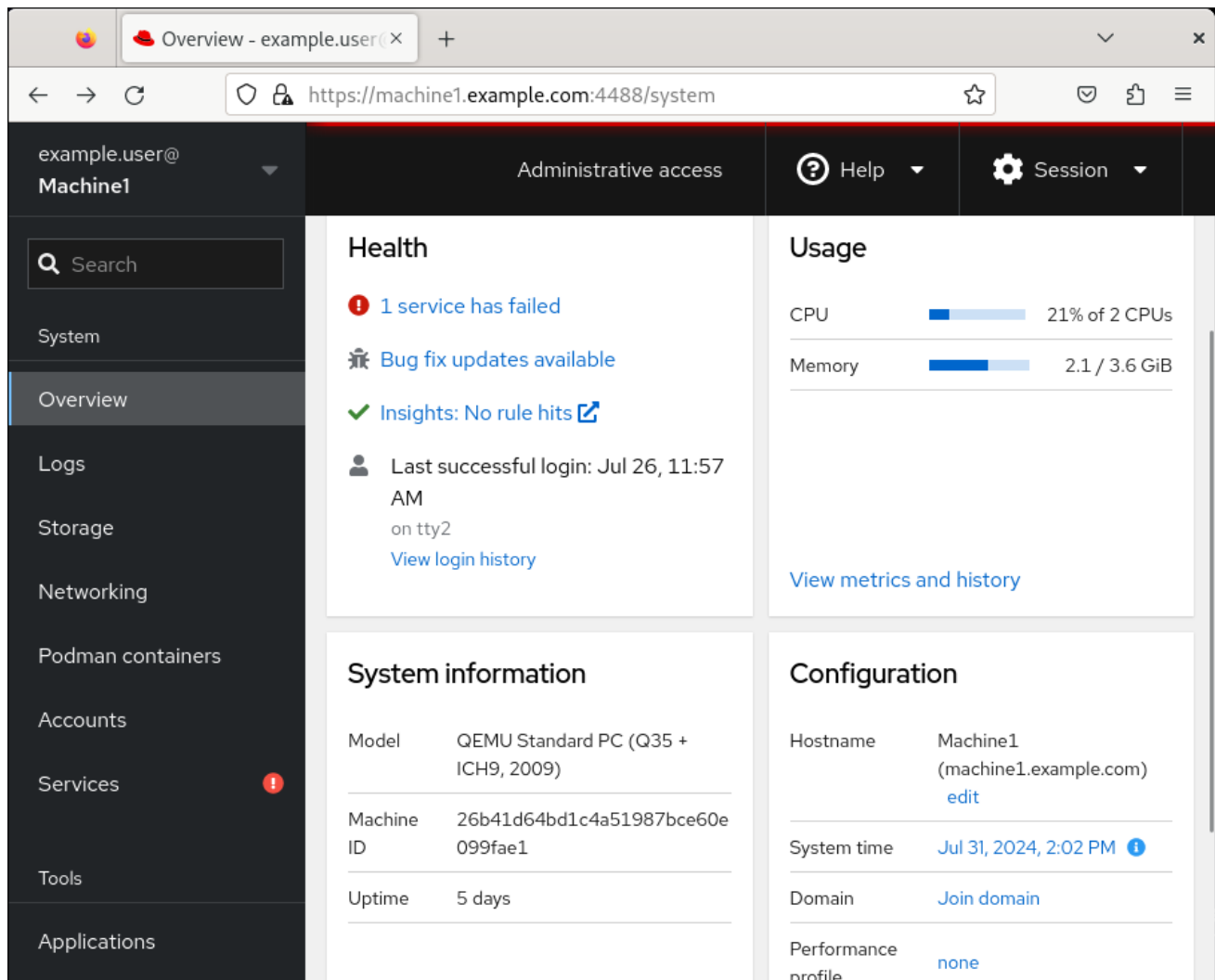


# CHAPTER 1. GETTING STARTED USING THE RHEL WEB CONSOLE

Learn how to install the Red Hat Enterprise Linux 9 web console, how to [add and manage remote hosts](#) through its convenient graphical interface, and how to monitor the systems managed by the web console.

## 1.1. WHAT IS THE RHEL WEB CONSOLE

The RHEL web console is a web-based interface designed for managing and monitoring your local system, as well as Linux servers located in your network environment.



The RHEL web console enables you to perform a wide range of administration tasks, including:

- Managing services
- Managing user accounts
- Managing and monitoring system services
- Configuring network interfaces and firewall
- Reviewing system logs
- Managing virtual machines

- Creating diagnostic reports
- Setting kernel dump configuration
- Configuring SELinux
- Updating software
- Managing system subscriptions

The RHEL web console uses the same system APIs as you would use in a terminal, and actions performed in a terminal are immediately reflected in the RHEL web console.

You can monitor the logs of systems in the network environment, as well as their performance, displayed as graphs. In addition, you can change the settings directly in the web console or through the terminal.

## 1.2. INSTALLING AND ENABLING THE WEB CONSOLE

To access the RHEL web console, first enable the **cockpit.socket** service.

Red Hat Enterprise Linux 9 includes the web console installed by default in many installation variants. If this is not the case on your system, install the **cockpit** package before enabling the **cockpit.socket** service.

### Procedure

1. If the web console is not installed by default on your installation variant, manually install the **cockpit** package:

```
# dnf install cockpit
```

2. Enable and start the **cockpit.socket** service, which runs a web server:

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

3. If the web console was not installed by default on your installation variant and you are using a custom firewall profile, add the **cockpit** service to **firewalld** to open port 9090 in the firewall:

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

### Verification

- To verify the previous installation and configuration, [open the web console](#).

## 1.3. LOGGING IN TO THE WEB CONSOLE

When the **cockpit.socket** service is running and the corresponding firewall port is open, you can log in to the web console in your browser for the first time.

### Prerequisites

- Use one of the following browsers to open the web console:

- Mozilla Firefox 52 and later
- Google Chrome 57 and later
- Microsoft Edge 16 and later
- System user account credentials  
The RHEL web console uses a specific pluggable authentication modules (PAM) stack at **/etc/pam.d/cockpit**. The default configuration allows logging in with the user name and password of any local account on the system.
- Port 9090 is open in your firewall.

## Procedure

1. In your web browser, enter the following address to access the web console:

**https://localhost:9090**



### NOTE

This provides a web-console login on your local machine. If you want to log in to the web console of a remote system, see [Section 1.5, “Connecting to the web console from a remote machine”](#)

If you use a self-signed certificate, the browser displays a warning. Check the certificate, and accept the security exception to proceed with the login.

The console loads a certificate from the **/etc/cockpit/ws-certs.d** directory and uses the last file with a **.cert** extension in alphabetical order. To avoid having to grant security exceptions, install a certificate signed by a certificate authority (CA).

2. In the login screen, enter your system user name and password.
3. Click **Log In**.

After successful authentication, the RHEL web console interface opens.



### IMPORTANT

To switch between limited and administrative access, click **Administrative access** or **Limited access** in the top panel of the web console page. You must provide your user password to gain administrative access.

## 1.4. DISABLING BASIC AUTHENTICATION IN THE WEB CONSOLE

You can modify the behavior of an authentication scheme by modifying the **cockpit.conf** file. Use the **none** action to disable an authentication scheme and only allow authentication through GSSAPI and forms.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

- You have **root** privileges or permissions to enter administrative commands with **sudo**.

### Procedure

1. Open or create the **cockpit.conf** file in the **/etc/cockpit/** directory in a text editor of your preference, for example:

```
# vi cockpit.conf
```

2. Add the following text:

```
[basic]
action = none
```

3. Save the file.
4. Restart the web console for changes to take effect.

```
# systemctl try-restart cockpit
```

## 1.5. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE

You can connect to your web console interface from any client operating system and also from mobile phones or tablets.

### Prerequisites

- A device with a supported internet browser, such as:
  - Mozilla Firefox 52 and later
  - Google Chrome 57 and later
  - Microsoft Edge 16 and later
- The RHEL 9 you want to access with an installed and accessible web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Open your web browser.
2. Type the remote server's address in one of the following formats:

- a. With the server's host name:

```
https://<server.hostname.example.com>:<port-number>
```

For example:

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

- b. With the server's IP address:

–

```
https://<server.IP_address>:<port-number>
```

For example:

```
https://192.0.2.2:9090
```

3. After the login interface opens, log in with your RHEL system credentials.

## 1.6. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE AS A ROOT USER

On new installations of RHEL 9.2 or later, the RHEL web console disallows root account logins by default for security reasons. You can allow the **root** login in the **/etc/cockpit/disallowed-users** file.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Open the **disallowed-users** file in the **/etc/cockpit/** directory in a text editor of your preference, for example:

```
# vi /etc/cockpit/disallowed-users
```

2. Edit the file and remove the line for the **root** user:

```
# List of users which are not allowed to login to Cockpit root
```

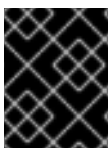
3. Save the changes and quit the editor.

### Verification

- Log in to the web console as a **root** user.  
For details, see [Logging in to the web console](#).

## 1.7. LOGGING IN TO THE WEB CONSOLE USING A ONE-TIME PASSWORD

If your system is part of an Identity Management (IdM) domain with enabled one-time password (OTP) configuration, you can use an OTP to log in to the RHEL web console.



### IMPORTANT

It is possible to log in using a one-time password only if your system is part of an Identity Management (IdM) domain with enabled OTP configuration.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).
- An Identity Management server with enabled OTP configuration.
- A configured hardware or software device generating OTP tokens.

### Procedure

1. Open the RHEL web console in your browser:

- Locally: **`https://localhost:PORT_NUMBER`**
- Remotely with the server hostname: **`https://example.com:PORT_NUMBER`**
- Remotely with the server IP address:  
**`https://EXAMPLE.SERVER.IP.ADDR:PORT_NUMBER`**

If you use a self-signed certificate, the browser issues a warning. Check the certificate and accept the security exception to proceed with the login.

The console loads a certificate from the `/etc/cockpit/ws-certs.d` directory and uses the last file with a `.cert` extension in alphabetical order. To avoid having to grant security exceptions, install a certificate signed by a certificate authority (CA).

2. The Login window opens. In the Login window, enter your system user name and password.
3. Generate a one-time password on your device.
4. Enter the one-time password into a new field that appears in the web console interface after you confirm your password.
5. Click **Log in**.
6. Successful login takes you to the **Overview** page of the web console interface.

## 1.8. ADDING A BANNER TO THE LOGIN PAGE

You can set the web console to show a content of a banner file on the login screen.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).
- You have **root** privileges or permissions to enter administrative commands with **sudo**.

### Procedure

1. Open the `/etc/issue.cockpit` file in a text editor of your preference:

```
# vi /etc/issue.cockpit
```

2. Add the content you want to display as the banner to the file, for example:

```
This is an example banner for the RHEL web console login page.
```

You cannot include any macros in the file, but you can use line breaks and ASCII art.

3. Save the file.
4. Open the **cockpit.conf** file in the **/etc/cockpit/** directory in a text editor of your preference, for example:

```
# vi /etc/cockpit/cockpit.conf
```

5. Add the following text to the file:

```
[Session]
Banner=/etc/issue.cockpit
```

6. Save the file.
7. Restart the web console for changes to take effect.

```
# systemctl try-restart cockpit
```

## Verification

- Open the web console login screen again to verify that the banner is now visible:

This is an example banner for the RHEL web console login page.

Red Hat Enterprise Linux

User name

Password

☒ Reuse my password for remote connections

[Other Options](#)

Log In

Server: mymachine.idm.example.com  
Log in with your server user account.

## 1.9. CONFIGURING AUTOMATIC IDLE LOCK IN THE WEB CONSOLE

You can enable the automatic idle lock and set the idle timeout for your system through the web console interface.

## Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).
- You have **root** privileges or permissions to enter administrative commands with **sudo**.

## Procedure

1. Open the **cockpit.conf** file in the **/etc/cockpit/** directory in a text editor of your preference, for example:

```
# vi /etc/cockpit/cockpit.conf
```

2. Add the following text to the file:

```
[Session]
IdleTimeout=<X>
```

Substitute **<X>** with a number for a time period of your choice in minutes.

3. Save the file.
4. Restart the web console for changes to take effect.

```
# systemctl try-restart cockpit
```

## Verification

- Check if the session logs you out after a set period of time.

## 1.10. CHANGING THE WEB CONSOLE LISTENING PORT

By default, the RHEL web console communicates through TCP port 9090. You can change the port number by overriding the default socket settings.

## Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).
- You have **root** privileges or permissions to enter administrative commands with **sudo**.
- The **firewalld** service is running.

## Procedure

1. Pick an unoccupied port, for example, **<4488/tcp>**, and instruct SELinux to allow the **cockpit** service to bind to that port:

```
# semanage port -a -t websm_port_t -p tcp <4488>
```

Note that a port can be used only by one service at a time, and thus an attempt to use an already occupied port implies the **ValueError: Port already defined** error message.



2. Open the new port and close the former one in the firewall:

```
# firewall-cmd --service cockpit --permanent --add-port=<4488>/tcp
# firewall-cmd --service cockpit --permanent --remove-port=9090/tcp
```

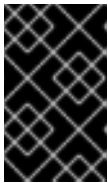
3. Create an override file for the **cockpit.socket** service:

```
# systemctl edit cockpit.socket
```

4. In the following editor screen, which opens an empty **override.conf** file located in the **/etc/systemd/system/cockpit.socket.d/** directory, change the default port for the web console from 9090 to the previously picked number by adding the following lines:

```
[Socket]
ListenStream=
ListenStream=<4488>
```

Note that the first **ListenStream=** directive with an empty value is intentional. You can declare multiple **ListenStream** directives in a single socket unit and the empty value in the drop-in file resets the list and disables the default port 9090 from the original unit.



### IMPORTANT

Insert the previous code snippet between the lines starting with **# Anything between here** and **# Lines below this**. Otherwise, the system discards your changes.

5. Save the changes by pressing **Ctrl+O** and **Enter**. Exit the editor by pressing **Ctrl+X**.
6. Reload the changed configuration:

```
# systemctl daemon-reload
```

7. Check that your configuration is working:

```
# systemctl show cockpit.socket -p Listen
Listen=[::]:4488 (Stream)
```

8. Restart **cockpit.socket**:

```
# systemctl restart cockpit.socket
```

### Verification

- Open your web browser, and access the web console on the updated port, for example:

```
https://machine1.example.com:4488
```

### Additional resources

- **firewall-cmd(1)**, **semanage(8)**, **systemd.unit(5)**, and **systemd.socket(5)** man pages on your system

## CHAPTER 2. INSTALLING AND CONFIGURING WEB CONSOLE BY USING RHEL SYSTEM ROLES

With the **cockpit** RHEL system role, you can automatically deploy and enable the web console on multiple RHEL systems.

### 2.1. INSTALLING THE WEB CONSOLE BY USING THE **cockpit** RHEL SYSTEM ROLE

You can use the **cockpit** system role to automate installing and enabling the RHEL web console on multiple systems.

In this example, you use the **cockpit** system role to:

- Install the RHEL web console.
- Allow the **firewalld** and **selinux** system roles to configure the system for opening new ports.
- Set the web console to use a certificate from the **ipa** trusted certificate authority instead of using a self-signed certificate.



#### NOTE

You do not have to call the **firewall** or **certificate** system roles in the playbook to manage the firewall or create the certificate. The **cockpit** system role calls them automatically as needed.

#### Prerequisites

- [You have prepared the control node and the managed nodes](#)
- You are logged in to the control node as a user who can run playbooks on the managed nodes.
- The account you use to connect to the managed nodes has **sudo** permissions on them.

#### Procedure

1. Create a playbook file, for example, `~/playbook.yml`, with the following content:

```
---
- name: Manage the RHEL web console
  hosts: managed-node-01.example.com
  tasks:
    - name: Install RHEL web console
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.cockpit
      vars:
        cockpit_packages: default
        cockpit_manage_selinux: true
        cockpit_manage_firewall: true
        cockpit_certificates:
          - name: /etc/cockpit/ws-certs.d/01-certificate
            dns: ['localhost', 'www.example.com']
            ca: ipa
```

■

The settings specified in the example playbook include the following:

**cockpit\_manage\_selinux: true**

Allow using the **selinux** system role to configure SELinux for setting up the correct port permissions on the **websm\_port\_t** SELinux type.

**cockpit\_manage\_firewall: true**

Allow the **cockpit** system role to use the **firewalld** system role for adding ports.

**cockpit\_certificates: <YAML\_dictionary>**

By default, the RHEL web console uses a self-signed certificate. Alternatively, you can add the **cockpit\_certificates** variable to the playbook and configure the role to request certificates from an IdM certificate authority (CA) or to use an existing certificate and private key that is available on the managed node.

For details about all variables used in the playbook, see the **/usr/share/ansible/roles/rhel-system-roles.cockpit/README.md** file on the control node.

2. Validate the playbook syntax:

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

Note that this command only validates the syntax and does not protect against a wrong but valid configuration.

3. Run the playbook:

```
$ ansible-playbook ~/playbook.yml
```

### Additional resources

- **/usr/share/ansible/roles/rhel-system-roles.cockpit/README.md** file
- **/usr/share/doc/rhel-system-roles/cockpit** directory
- [Requesting certificates using RHEL system roles](#)

## CHAPTER 3. INSTALLING WEB CONSOLE ADD-ONS AND CREATING CUSTOM PAGES

Depending on how you want to use your Red Hat Enterprise Linux system, you can add additional **available** applications to the web console or create custom pages based on your use case.

### 3.1. ADD-ONS FOR THE RHEL WEB CONSOLE

While the **cockpit** package is a part of Red Hat Enterprise Linux by default, you can install add-on applications on demand by using the following command:

```
# dnf install <add-on>
```

In the previous command, replace *<add-on>* by a package name from the list of available add-on applications for the RHEL web console.

Feature name	Package name	Usage
Composer	<b>cockpit-composer</b>	Building custom OS images
File manager	<b>cockpit-files</b>	Managing files and directories in the standard web-console interface
Machines	<b>cockpit-machines</b>	Managing <b>libvirt</b> virtual machines
PackageKit	<b>cockpit-packagekit</b>	Software updates and application installation (usually installed by default)
PCP	<b>cockpit-pcp</b>	Persistent and more fine-grained performance data (installed on demand from the UI)
Podman	<b>cockpit-podman</b>	<a href="#">Managing containers</a> and <a href="#">managing container images</a>
Session Recording	<b>cockpit-session-recording</b>	Recording and managing user sessions
Storage	<b>cockpit-storaged</b>	Managing storage through <b>udisks</b>

### 3.2. CREATING NEW PAGES IN THE WEB CONSOLE

If you want to add customized functions to your Red Hat Enterprise Linux web console, you must add the package directory that contains the HTML and JavaScript files for the page that runs the required function.

For detailed information about adding custom pages, see [Creating Plugins for the Cockpit User Interface](#) on the [Cockpit Project](#) website.

#### Additional resources

- [Cockpit Packages](#) section in the [Cockpit Project Developer Guide](#)

### 3.3. OVERRIDING THE MANIFEST SETTINGS IN THE WEB CONSOLE

You can modify the menu of the web console for a particular user and all users of the system. In the **cockpit** project, a package name is a directory name. A package contains the **manifest.json** file along with other files. Default settings are present in the **manifest.json** file. You can override the default **cockpit** menu settings by creating a **<package-name>.override.json** file at a specific location for the specified user.

#### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

#### Procedure

1. Override manifest settings in the **<systemd>.override.json** file in a text editor of your choice, for example:

- a. To edit for all users, enter:

```
# vi /etc/cockpit/<systemd>.override.json
```

- b. To edit for a single user, enter:

```
# vi ~/.config/cockpit/<systemd>.override.json
```

2. Edit the required file with the following details:

```
{
  "menu": {
    "services": null,
    "logs": {
      "order": -1
    }
  }
}
```

- The **null** value hides the **services** tab
- The **-1** value moves the **logs** tab to the first place.

3. Restart the **cockpit** service:

```
# systemctl restart cockpit.service
```

#### Additional resources

- **cockpit(1)** man page on your system
- [Manifest overrides](#)

## CHAPTER 4. MANAGING SOFTWARE UPDATES IN THE WEB CONSOLE

Learn how to manage software updates in the RHEL 9 web console and ways to automate them.

The Software Updates module in the web console is based on the **dnf** utility. For more information about updating software with **dnf**, see the [Updating packages](#) section.

### 4.1. MANAGING MANUAL SOFTWARE UPDATES IN THE WEB CONSOLE

You can manually update your software by using the web console.

#### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

#### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. Click **Software Updates**.  
The list of available updates refreshes automatically after 24 hours. To trigger a refresh, click the **Check for Updates** button.
3. Apply updates. You can watch the update log while the update is running.
  - a. To install all available updates, click the **Install all updates** button.
  - b. If you have security updates available, you can install them separately by clicking the **Install Security Updates** button.
  - c. If you have **kpatch** updates available, you can install them separately by clicking the **Install kpatch updates** button.
4. Optional: You can turn on the **Reboot after completion** switch for an automatic restart of your system.  
If you perform this step, you can skip the remaining steps of this procedure.
5. After the system applies updates, you get a recommendation to restart your system. Restart the system if the update included a new kernel or system services that you do not want to restart individually.
6. Click **Ignore** to cancel the restart, or **Restart Now** to proceed with restarting your system.  
After the system restart, log in to the web console and go to the **Software Updates** page to verify that the update is successful.

### 4.2. MANAGING AUTOMATIC SOFTWARE UPDATES IN THE WEB CONSOLE

In the web console, you can choose to apply all updates, or security updates and also manage periodicity and time of your automatic updates.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. Click **Software Updates**.
3. In the **Settings** table, click the **Edit** button.
4. Pick one of the types of automatic updates. You can select from **Security updates only**, or **All updates**.
5. To modify the day of the automatic update, click on the **every day** drop-down menu and select a specific day.
6. To modify the time of the automatic update, click into the **6:00** field and select or type a specific time.
7. If you want to disable automatic software updates, select the **No updates** type.

## 4.3. MANAGING ON-DEMAND RESTARTING AFTER APPLYING SOFTWARE UPDATES IN THE WEB CONSOLE

The intelligent restarting feature informs the users whether it is necessary to reboot the whole system after you apply a software update or if it is sufficient to only restart certain services.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. Click **Software Updates**.
3. Apply an update of your system.
4. After a successful update, click **Reboot system...**, **Restart services...**, or **Ignore**
5. If you decide to ignore, you can return to the restart or reboot menu by doing one of the following:
  - a. Rebooting:



- i. Click the **Reboot system** button in the **Status** field of the **Software Updates** page.
  - ii. Optional: Write a message to the logged in users.
  - iii. Select a delay from the **Delay** drop-down menu.
  - iv. Click **Reboot**.
- b. Restarting services:
- i. Click the **Restart services** button in the **Status** field of the **Software Updates** page.  
You will see a list of all the services that require a restart.
  - ii. Click **Restart services**.  
Depending on your choice, the system will reboot or your services will restart.

## 4.4. APPLYING PATCHES WITH KERNEL LIVE PATCHING IN THE WEB CONSOLE

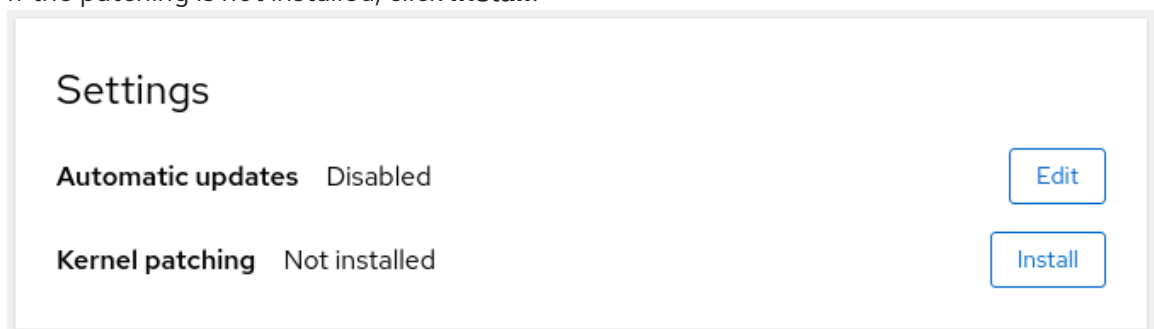
You can configure the **kpatch** framework, which applies kernel security patches without forcing reboots, in the RHEL web console.

### Prerequisites

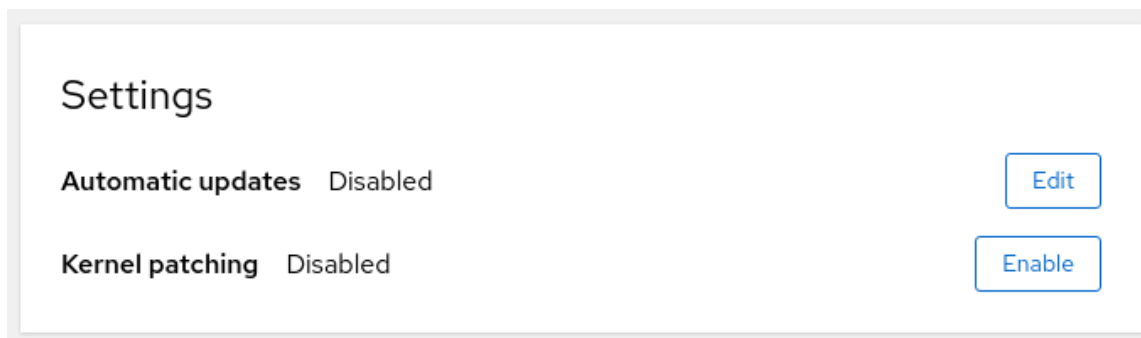
- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. Click **Software Updates**.
3. Check the status of your kernel patching settings.
  - a. If the patching is not installed, click **Install**.



- b. To enable kernel patching, click **Enable**.

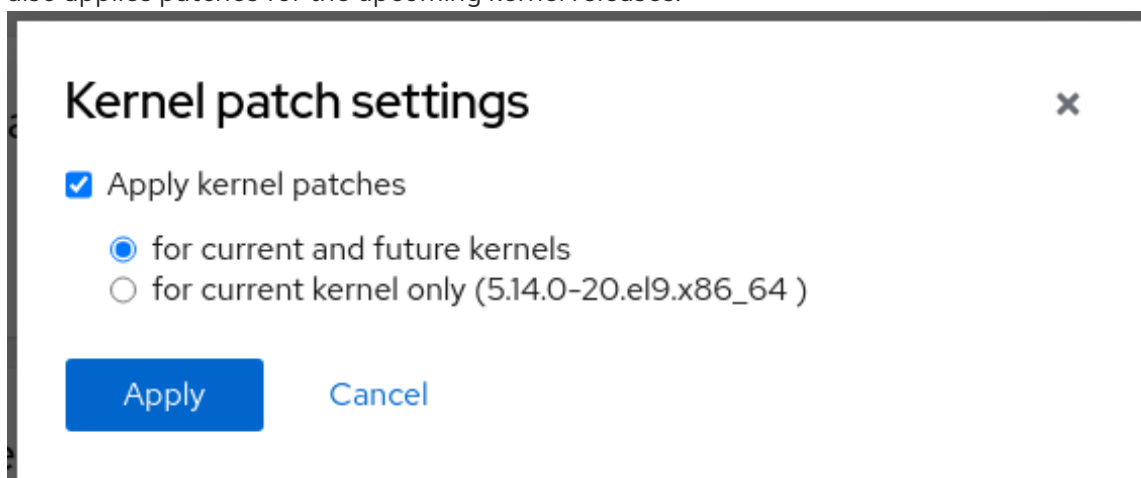


**Settings**

**Automatic updates** Disabled Edit

**Kernel patching** Disabled Enable

- c. Check the check box for applying kernel patches.
- d. Select whether you want to apply patches for current and future kernels or the current kernel only. If you decide to subscribe to applying patches for future kernels, the system also applies patches for the upcoming kernel releases.



**Kernel patch settings** ×

☒ **Apply kernel patches**

☒ for current and future kernels

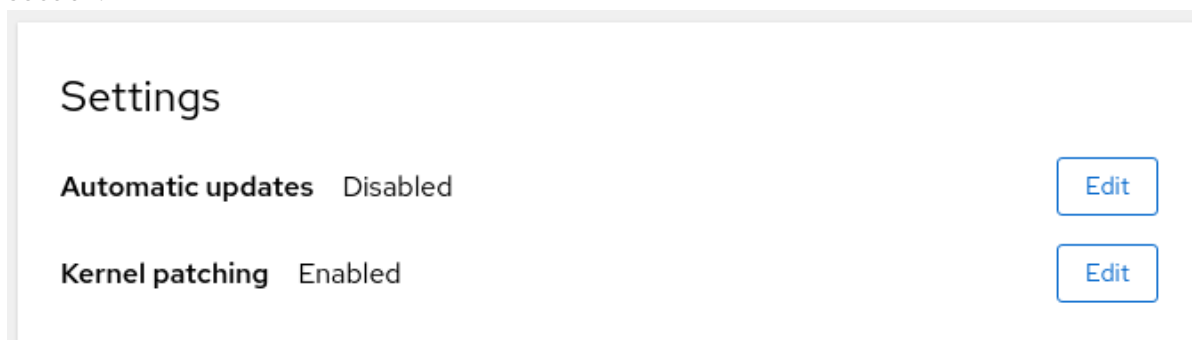
☐ for current kernel only (5.14.0-20.el9.x86\_64 )

Apply Cancel

- e. Click **Apply**.

## Verification

- Check that the kernel patching is now **Enabled** in the **Settings** table of the **Software updates** section.



**Settings**

**Automatic updates** Disabled Edit

**Kernel patching** Enabled Edit

## Additional resources

- [Applying patches with kernel live patching](#)

## CHAPTER 5. MANAGING SUBSCRIPTIONS IN THE WEB CONSOLE

You can manage your Red Hat product subscriptions in the Red Hat Enterprise Linux 9 web console.

### Prerequisites

- Your [Red Hat Customer Portal](#) or a subscription activation key.

## 5.1. SUBSCRIPTION MANAGEMENT IN THE WEB CONSOLE

The RHEL 9 web console provides an interface for using Red Hat Subscription Manager installed on your local system.

The Subscription Manager connects to the Red Hat Customer Portal and verifies available:

- Active subscriptions
- Expired subscriptions
- Renewed subscriptions

If you want to renew the subscription or get a different one on the Red Hat Customer Portal, you do not have to update the Subscription Manager data manually.

The Subscription Manager synchronizes data with the Red Hat Customer Portal automatically.

## 5.2. REGISTERING A SYSTEM BY USING THE WEB CONSOLE

Subscriptions cover products installed on Red Hat Enterprise Linux, including the operating system itself. If you have not registered the system, you have no access to the RHEL repositories. You cannot install software updates such as security, bug fixes. Even if you have a self-support subscription, it grants access to the knowledge base while more resources remain unavailable in the lack of subscriptions. You can register a newly installed Red Hat Enterprise Linux with account credentials in the Red Hat web console.

### Prerequisites

- You have an active subscription of the RHEL system.
- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Open `https://<ip_address_or_hostname>:9090` in a browser, and log in to the web console.
2. In the **Health** field on the **Overview** page, click the **Not registered** warning, or click **Subscriptions** in the main menu to move to page with your subscription information.
3. In the **Overview** field, click **Register**.
4. In the **Register system** dialog, select the registration method.

Optional: Enter your organization's name or ID. If your account belongs to more than one organization on the Red Hat Customer Portal, you must add the organization name or ID. To get the organization ID, check with your Technical Account Manager at Red Hat.

5. If you do not want to connect your system to Red Hat Insights, clear the **Insights** checkbox.
6. Click **Register**.

### Verification

- Check details of your subscription in [the Hybrid Cloud Console](#).

## 5.3. REGISTERING SUBSCRIPTIONS WITH ACTIVATION KEYS IN THE WEB CONSOLE

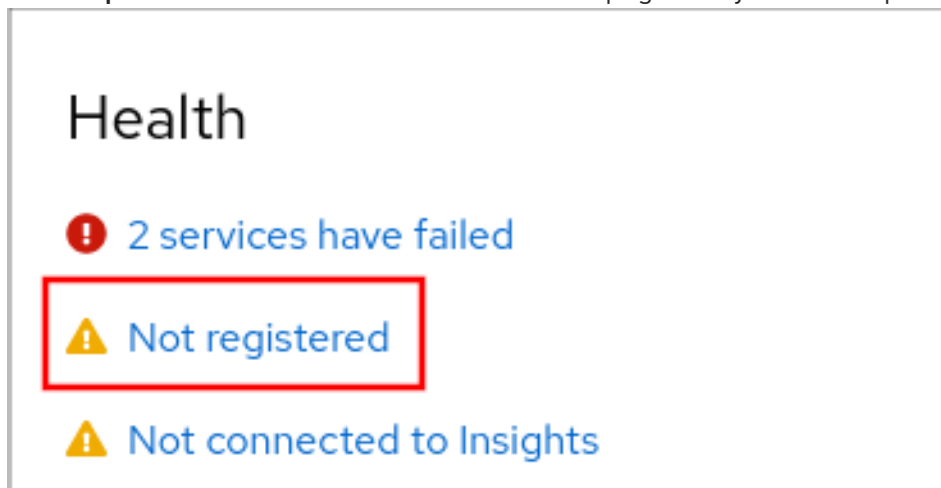
You can register a newly installed Red Hat Enterprise Linux with an activation key in the RHEL web console.

### Prerequisites

- An activation key of your Red Hat product subscription.
- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. In the **Health** field on the **Overview** page, click the **Not registered** warning, or click **Subscriptions** in the main menu to move to the page with your subscription information.



3. In the **Overview** field, click **Register**.
4. In the **Register system** dialog box, select **Activation key** to register using an activation key.

## Register System

URL

Default

☐ Use proxy server

Method

☐ Account ☒ Activation key

Activation Key

key\_one,key\_two

Organization

Subscriptions

☒ Attach automatically

Insights

☒ Connect this system to [Red Hat Insights](#)

Register

Cancel

5. Enter your key or keys.
6. Enter your organization's name or ID.  
To get the organization ID, go to your Red Hat contact point.
  - If you do not want to connect your system to Red Hat Insights, clear the **Insights** check box.
7. Click **Register**.

## CHAPTER 6. MANAGING REMOTE SYSTEMS IN THE WEB CONSOLE

You can connect to the remote systems and manage them in the RHEL 9 web console.

You learn:

- The optimal topology of connected systems.
- How to add and remove remote systems.
- When, why, and how to use SSH keys for remote system authentication.
- How to configure a web console client to allow a user authenticated with a smart card to **SSH** to a remote host and access services on it.

### Prerequisites

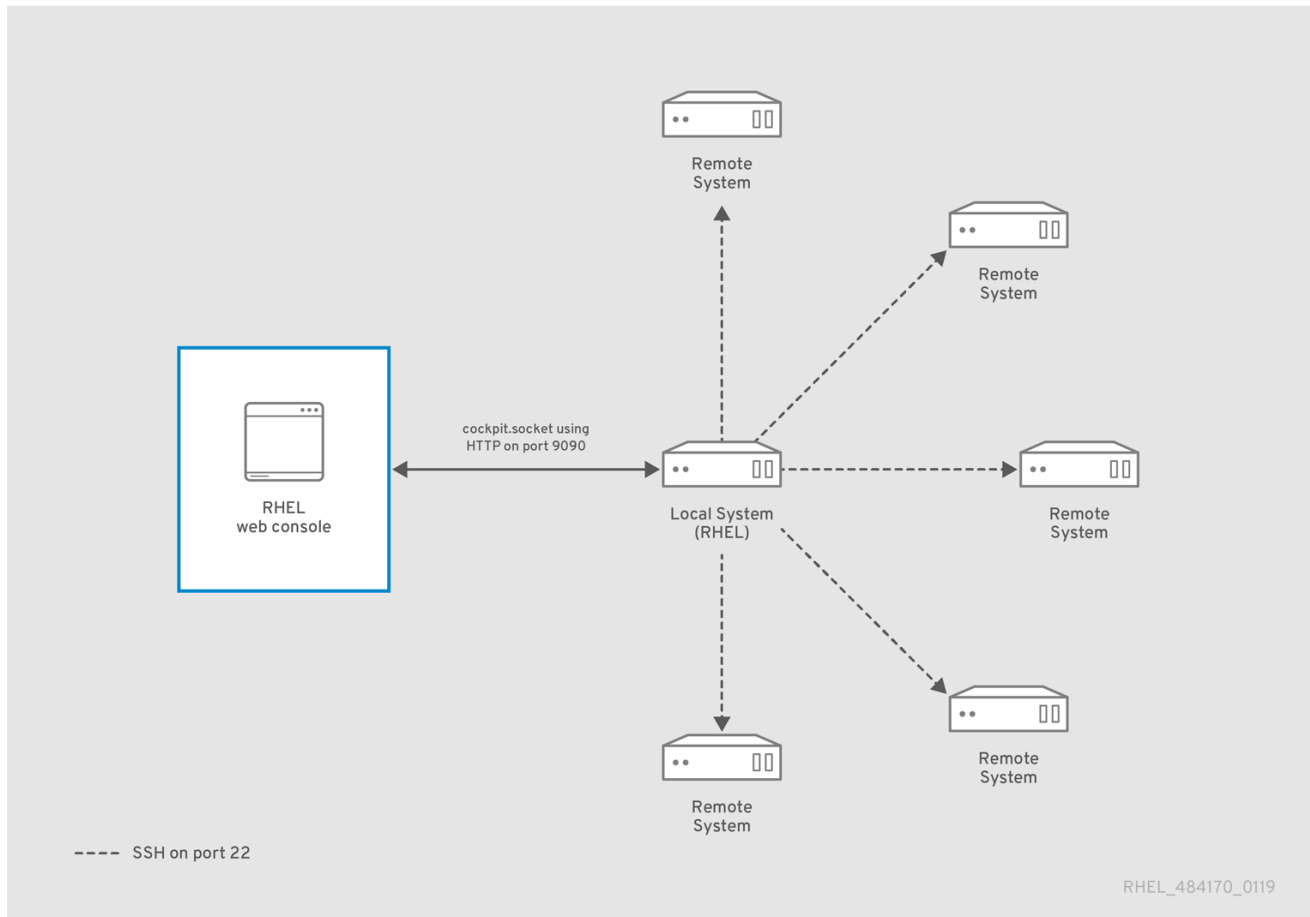
- The SSH service is running on remote systems.

### 6.1. REMOTE SYSTEM MANAGER IN THE WEB CONSOLE

For security reasons, use the following network setup of remote systems managed by the the RHEL 9 web console:

- Configure one system with the web console as a bastion host. The bastion host is a system with opened HTTPS port.
- All other systems communicate through SSH.

With the web interface running on the bastion host, you can reach all other systems through the SSH protocol using port 22 in the default configuration.



## 6.2. ADDING REMOTE HOSTS TO THE WEB CONSOLE

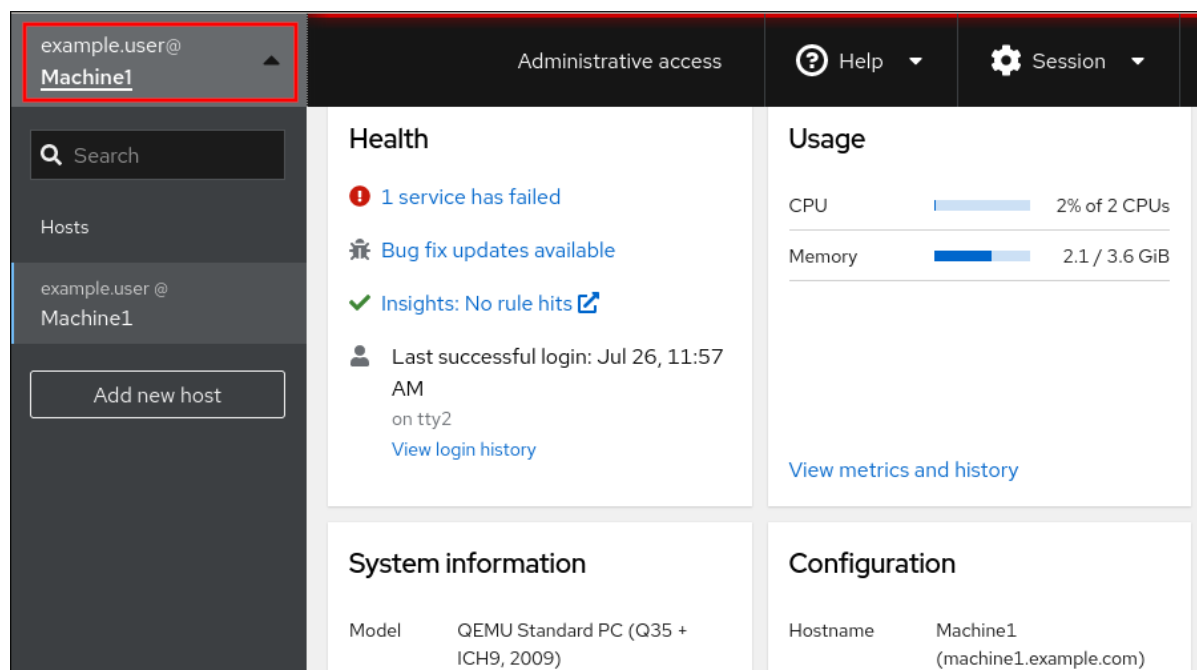
In the RHEL web console, you can manage remote systems after you add them with the corresponding credentials.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

- Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
- In the RHEL 9 web console, click your **<username>@<hostname>** in the top left corner of the **Overview** page.



3. From the drop-down menu, click **Add new host**.
4. In the **Add new host** dialog box, specify the host you want to add.
5. Optional: Add the user name for the account to which you want to connect.  
You can use any user account of the remote system. However, if you use the credentials of a user account without administration privileges, you cannot perform administration tasks.

If you use the same credentials as on your local system, the web console authenticates remote systems automatically every time you log in. Note that using the same credentials on more systems weakens the security.

6. Optional: Click the **Color** field to change the color of the system.
7. Click **Add**.



### IMPORTANT

The web console does not save passwords used to log in to remote systems, which means that you must log in again after each system restart. Next time you log in, click **Log in** placed on the main screen of the disconnected remote system to open the login dialog.

### Verification

- The new host is listed in the **<username>@<hostname>** drop-down menu.

## 6.3. ENABLING SSH LOGIN FOR A NEW HOST

When you add a new host to the web console, you can also log in to the host with an SSH key. If you already have an SSH key on your system, the web console uses the existing one; otherwise, the web console can create a key.

### Prerequisites

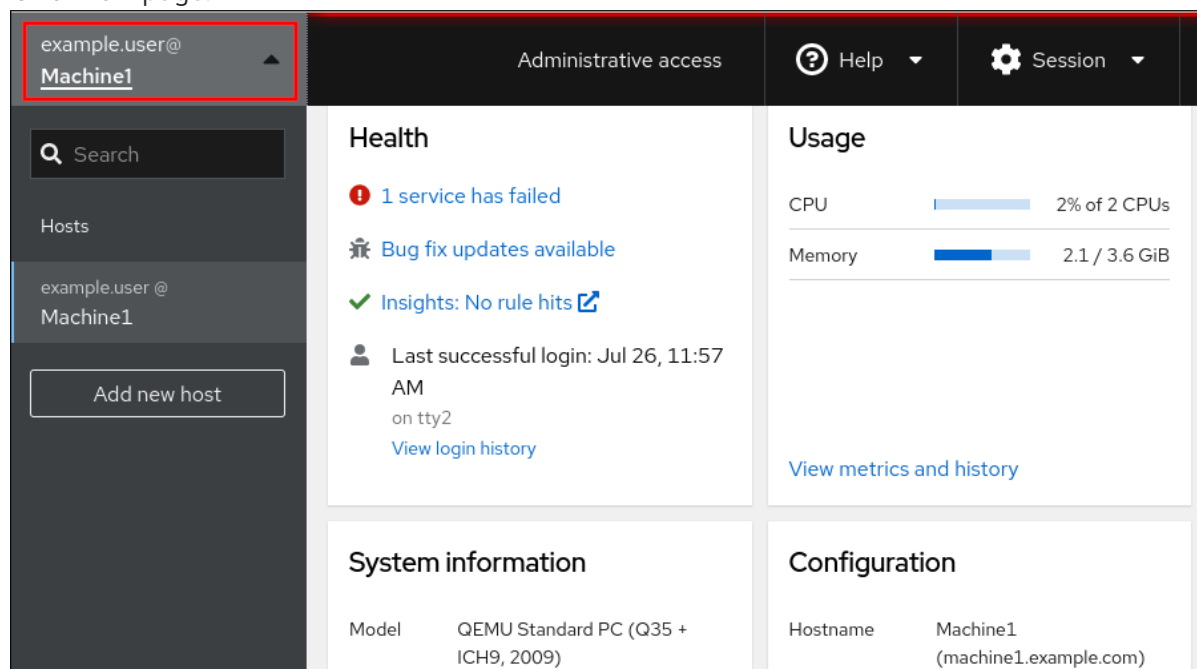
- You have installed the RHEL 9 web console.



For instructions, see [Installing and enabling the web console](#).

## Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
2. In the RHEL 9 web console, click your **<username>@<hostname>** in the top left corner of the **Overview** page.



3. From the drop-down menu, click **Add new host**.
4. In the **Add new host** dialog box, specify the host you want to add.
5. Add the user name for the account to which you want to connect.  
You can use any user account of the remote system. However, if you use a user account without administration privileges, you cannot perform administration tasks.
6. Optional: Click the **Color** field to change the color of the system.
7. Click **Add**.  
A new dialog window appears asking for a password.
8. Enter the user account password.
9. Check **Authorize SSH key** if you already have an SSH key.

## Log in to mymachine

Unable to log in to **euser@mymachine** using SSH key authentication. Please provide the password. You may want to set up your SSH keys for automatic login.

**Password**

**Automatic login**
☒ Authorize SSH key.

The SSH key `/home/euser/.ssh/id_rsa` of **euser** on **localhost** will be added to the `~/.ssh/authorized_keys` file of **euser** on **mymachine**.

This will allow you to log in without password in the future.

10. Check **Create a new SSH key and authorize it** if you do not have an SSH key. The web console creates the key.

## Log in to mymachine

Unable to log in to **euser@mymachine** using SSH key authentication. Please provide the password. You may want to set up your SSH keys for automatic login.

**Password**

**Automatic login**
☒ Create a new SSH key and authorize it.

A new SSH key at `/home/euser/.ssh/id_rsa` will be created for **euser** on **localhost** and it will be added to the `~/.ssh/authorized_keys` file of **euser** on **mymachine**.

**Key password**

**Confirm key password**

In order to allow log in to **mymachine** as **euser** without password in the future, use the login password of **euser** on **localhost** as the key password, or leave the key password blank.

- a. Add a password for the SSH key.
- b. Confirm the password.

11. Click **Log in**.

### Verification

1. Log out.
2. Log back in.
3. Click **Log in** in the **Not connected to host** screen.
4. Select **SSH key** as your authentication option.

**Log in to mymachine** ×

The SSH key for logging in to **euser@mymachine** is protected. You can log in with either your login password or by providing the password of the key at **/home/euser/.ssh/id\_rsa**. You may want to change the password of the key for automatic login.

**Authentication** ☐ Password ☒ **SSH key**

**Key password**

The SSH key **/home/euser/.ssh/id\_rsa** will be made available for the remainder of the session and will be available for login to other hosts as well.

**Automatic login** ☐ Change the password of **/home/euser/.ssh/id\_rsa**.

**Log in** **Cancel**

5. Enter your key password.
6. Click **Log in**.

### Additional resources

- [Using secure communications between two systems with OpenSSH](#)

## 6.4. CONFIGURING THE WEB CONSOLE TO ALLOW A USER AUTHENTICATED WITH A SMART CARD TO SSH TO A REMOTE HOST WITHOUT BEING ASKED TO AUTHENTICATE AGAIN

After logging in to a user account on the RHEL web console, you can connect to remote machines by using the SSH protocol. You can use the constrained delegation feature to use **SSH** without being asked to authenticate again.

In the example procedure, the web console session runs on the **myhost.idm.example.com** host, and you configure the console to access the **remote.idm.example.com** host by using SSH on behalf of the authenticated user.

## Prerequisites

- You have obtained an IdM **admin** ticket-granting ticket (TGT).
- You have **root** access to **remote.idm.example.com**.
- The **cockpit** service is running in IdM.
- The web console has created an **S4U2Proxy** Kerberos ticket in the user session. To verify it, log in to the web console as an IdM user, open the **Terminal** page, and enter:

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
Default principal: user@IDM.EXAMPLE.COM

Valid starting    Expires          Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06 07/31/21 09:19:06 krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
                    for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

## Procedure

1. Create a list of the target hosts that the delegation rule can access:

- a. Create a service delegation target:

```
$ ipa servicedelegationtarget-add cockpit-target
```

- b. Add the target host to the delegation target:

```
$ ipa servicedelegationtarget-add-member cockpit-target \
--principals=host/remote.idm.example.com@IDM.EXAMPLE.COM
```

2. Allow **cockpit** sessions to access the target host list by creating a service delegation rule and adding the HTTP service Kerberos principal to it:

- a. Create a service delegation rule:

```
$ ipa servicedelegationrule-add cockpit-delegation
```

- b. Add the web console client to the delegation rule:

```
$ ipa servicedelegationrule-add-member cockpit-delegation \
--principals=HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

- c. Add the delegation target to the delegation rule:

```
$ ipa servicedelegationrule-add-target cockpit-delegation \
--servicedelegationtargets=cockpit-target
```

3. Enable Kerberos authentication on the **remote.idm.example.com** host:

- a. Connect through SSH to **remote.idm.example.com** as **root**.

- b. Open the `/etc/ssh/sshd_config` file for editing.
- c. Enable **GSSAPIAuthentication** by uncommenting the **GSSAPIAuthentication no** line and replacing it with **GSSAPIAuthentication yes**.
4. Restart the **sshd** service on **remote.idm.example.com** so that the changes take effect immediately:

```
$ systemctl try-restart sshd.service
```

#### Additional resources

- [Logging in to the web console with smart cards](#)
- [Constrained delegation in Identity Management](#)

## 6.5. USING ANSIBLE TO CONFIGURE THE WEB CONSOLE TO ALLOW A USER AUTHENTICATED WITH A SMART CARD TO SSH TO A REMOTE HOST WITHOUT BEING ASKED TO AUTHENTICATE AGAIN

After logging in to a user account on the RHEL web console, you can connect to remote machines by using the SSH protocol. You can use the **servicedelegationrule** and **servicedelegationtarget** modules to configure the web console for the constrained delegation feature, which enables SSH connections without being asked to authenticate again.

In the example procedure, the web console session runs on the **myhost.idm.example.com** host and you configure it to access the **remote.idm.example.com** host by using SSH on behalf of the authenticated user.

#### Prerequisites

- The IdM **admin** password.
- **root** access to **remote.idm.example.com**.
- The web console service runs in IdM.
- The web console has created an **S4U2Proxy** Kerberos ticket in the user session. To verify it, log in to the web console as an IdM user, open the **Terminal** page, and enter:

```
$ klist
Ticket cache: FILE:/run/user/1894000001/cockpit-session-3692.ccache
Default principal: user@IDM.EXAMPLE.COM

Valid starting    Expires          Service principal
07/30/21 09:19:06 07/31/21 09:19:06
HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
07/30/21 09:19:06 07/31/21 09:19:06 krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
for client HTTP/myhost.idm.example.com@IDM.EXAMPLE.COM
```

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.14 or later.

- You have installed the **ansible-freeipa** package.
- The example assumes that in the `~/MyPlaybooks/` directory, you have created an **Ansible inventory file** with the fully-qualified domain name (FQDN) of the IdM server.
- The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin\_password**.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

## Procedure

1. Navigate to your `~/MyPlaybooks/` directory:

```
$ cd ~/MyPlaybooks/
```

2. Create a **web-console-smart-card-ssh.yml** playbook with the following content:

- a. Create a task that ensures the presence of a delegation target:

```
---
- name: Playbook to create a constrained delegation target
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure servicedelegationtarget web-console-delegation-target is present
    ipaservicedelegationtarget:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web-console-delegation-target
```

- b. Add a task that adds the target host to the delegation target:

```
- name: Ensure servicedelegationtarget web-console-delegation-target member
principal host/remote.idm.example.com@IDM.EXAMPLE.COM is present
ipaservicedelegationtarget:
  ipaadmin_password: "{{ ipaadmin_password }}"
  name: web-console-delegation-target
  principal: host/remote.idm.example.com@IDM.EXAMPLE.COM
  action: member
```

- c. Add a task that ensures the presence of a delegation rule:

```
- name: Ensure servicedelegationrule delegation-rule is present
ipaservicedelegationrule:
  ipaadmin_password: "{{ ipaadmin_password }}"
  name: web-console-delegation-rule
```

- d. Add a task that ensures that the Kerberos principal of the web console client service is a member of the constrained delegation rule:

```
- name: Ensure the Kerberos principal of the web console client service is added to the
servicedelegationrule web-console-delegation-rule
```

```

ipaservicedelegationrule:
  ipadmin_password: "{{ ipadmin_password }}"
  name: web-console-delegation-rule
  principal: HTTP/myhost.idm.example.com
  action: member

```

- e. Add a task that ensures that the constrained delegation rule is associated with the web-console-delegation-target delegation target:

```

- name: Ensure a constrained delegation rule is associated with a specific delegation
  target
  ipaservicedelegationrule:
    ipadmin_password: "{{ ipadmin_password }}"
    name: web-console-delegation-rule
    target: web-console-delegation-target
    action: member

```

3. Save the file.
4. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```

$ ansible-playbook --vault-password-file=password_file -v -i inventory web-console-smart-
card-ssh.yml

```

5. Enable Kerberos authentication on **remote.idm.example.com**:
  - a. Connect through SSH to **remote.idm.example.com** as **root**.
  - b. Open the **/etc/ssh/sshd\_config** file for editing.
  - c. Enable **GSSAPIAuthentication** by uncommenting the **GSSAPIAuthentication no** line and replacing it with **GSSAPIAuthentication yes**.
6. Restart the **sshd** service on **remote.idm.example.com** so that the changes take effect immediately:

```

$ systemctl try-restart sshd.service

```

### Additional resources

- [Logging in to the web console with smart cards](#)
- [Constrained delegation in Identity Management](#)
- **README-servicedelegationrule.md** and **README-servicedelegationtarget.md** in the **/usr/share/doc/ansible-freeipa/** directory
- Sample playbooks in the **/usr/share/doc/ansible-freeipa/playbooks/servicedelegationtarget** and **/usr/share/doc/ansible-freeipa/playbooks/servicedelegationrule** directories

## CHAPTER 7. CONFIGURING SINGLE SIGN-ON FOR LOGGING IN TO THE RHEL WEB CONSOLE SERVICE RUNNING ON AN IDM CLIENT AND AN IDM SERVER

Using Single Sign-on (SSO) authentication provided by Identity Management (IdM) in the RHEL 9 web console has the following advantages:

- Users with a Kerberos ticket in the IdM domain do not need to provide login credentials to access the web console.
- Users with a certificate issued by the IdM certificate authority (CA) do not need to provide login credentials to access the web console. The web console server automatically switches to a certificate issued by the IdM certificate authority and accepted by browsers. Certificate configuration is not necessary.
- IdM domain administrators can use the RHEL 9 web console to manage the systems in the domain. With the proper permissions configured, IdM administrators can use their web console client, that is their browser, to run any command on any IdM host.

This chapter describes:

- How to configure SSO for logging in to the RHEL web console service running on an IdM client.
- How to configure SSO for logging in to the RHEL web console service running on an IdM server.
- How to configure **sudo** access to IdM hosts for an IdM system administrator logged in to a web console client.

### Prerequisites

- The RHEL web console service is installed on a RHEL 9 system.  
For details, see [Installing the web console](#).
- The IdM client is installed on the system where the RHEL web console service is running.  
For details, see [IdM client installation](#).

## 7.1. JOINING A RHEL 9 SYSTEM TO AN IDM DOMAIN USING THE WEB CONSOLE

You can use the web console to join a Red Hat Enterprise Linux 9 system to the Identity Management (IdM) domain.

### Prerequisites

- The IdM domain is running and reachable from the client you want to join.
- You have the IdM domain administrator credentials.
- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.



For details, see [Logging in to the web console](#).

2. In the **Configuration** field of the **Overview** tab click **Join Domain**.
3. In the **Join a Domain** dialog box, enter the host name of the IdM server in the **Domain Address** field.
4. In the **Domain administrator name** field, enter the user name of the IdM administration account.
5. In the **Domain administrator password**, add a password.
6. Click **Join**.

### Verification

1. If the RHEL 9 web console did not display an error, the system has been joined to the IdM domain and you can see the domain name in the **System** screen.
2. To verify that the user is a member of the domain, click the Terminal page and type the **id** command:

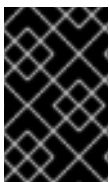
```
$ id
uid=548800004(example_user) gid=548800004(example_user)
groups=548800004(example_user) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
```

### Additional resources

- [Planning Identity Management](#)
- [Installing Identity Management](#)
- [Managing IdM users, groups, hosts, and access control rules](#)

## 7.2. LOGGING IN TO THE WEB CONSOLE USING KERBEROS AUTHENTICATION

As an Identity Management (IdM) user, you can use Single Sign-On (SSO) authentication to automatically access the RHEL web console in your browser.



### IMPORTANT

With SSO, you usually do not have any administrative privileges in the web console. This only works if you configure passwordless sudo. The web console does not interactively ask for a sudo password.

### Prerequisites

- The IdM domain is resolvable by DNS. For instance, the SRV records of the Kerberos server are resolvable:

```
$ host -t SRV _kerberos._udp.idm.example.com
_kerberos._udp.idm.example.com has SRV record 0 100 88 dc.idm.example.com
```

If the system where you are running your browser is a RHEL 9 system and has been [joined to the IdM domain](#), you are using the same DNS as the web console server and no DNS configuration is necessary.

- You have configured the web console server for SSO authentication.
- The host on which the web console service is running is an IdM client.
- You have configured the web console client for SSO authentication.

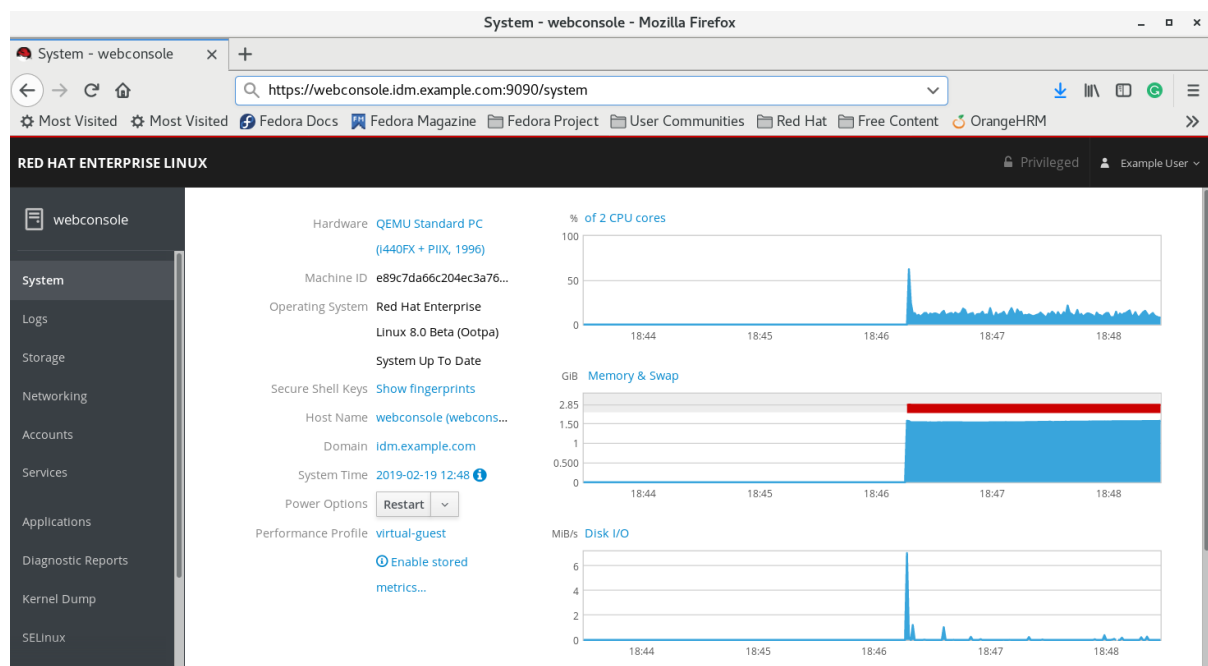
## Procedure

1. Obtain your Kerberos ticket-granting ticket:

```
$ kinit idmuser@IDM.EXAMPLE.COM
Password for idmuser@IDM.EXAMPLE.COM:
```

2. Enter the fully qualified name of the host on which the web console service is running into your browser:

```
https://<dns_name>:9090
```



At this point, you are successfully connected to the RHEL web console and you can start with configuration. For example, you can [join a RHEL 9 system to the IdM domain in the web console](#).

## 7.3. ENABLING THE RHEL WEB CONSOLE SINGLE SIGN-ON WITH GSSAPI ON IDM SERVERS

The RHEL web console can use the Generic Security Services Application Program Interface (GSSAPI) authentication. However, the IdM framework already owns an `HTTP/server.idm.example.com@IDM.EXAMPLE.COM` Kerberos service and its keytab. Therefore, to implement GSSAPI authentication on Identity Management (IdM) servers, create a symlink `/etc/cockpit/krb5.keytab` to `/var/lib/ipa/gssproxy/http.keytab` and then generate a certificate-key pair.

### Prerequisites

- You have **root** privileges.
- You are using RHEL 9.6 or later.

### Procedure

1. Create a symlink:

```
# ln -s /var/lib/ipa/gssproxy/http.keytab /etc/cockpit/krb5.keytab
```

2. Set a certificate file Bash variable:

```
# CERT_FILE=/etc/cockpit/ws-certs.d/50-certmonger.crt
```

3. Set a certificate key Bash variable:

```
# KEY_FILE=/etc/cockpit/ws-certs.d/50-certmonger.key
```

4. Generate a certificate-key pair:

```
# ipa-getcert request -f ${CERT_FILE} -k ${KEY_FILE} -D $(hostname --fqdn)
```

### Additional resources

- [man cockpit-tls\(8\)](#)

## 7.4. ENABLING SUDO ACCESS FOR IDM ADMINISTRATORS ON IDM HOSTS

You can configure Identity Management (IdM) system administrators to have **sudo** privileges on an IdM host.

### Prerequisites

- You are logged in as an IdM administrator to an IdM host.
- You have **root** privileges on the host.

### Procedure

- Enable **sudo** access on the host:

```
# ipa-adviser enable-admins-sudo | sh -ex
```

## CHAPTER 8. CONFIGURING SMART CARD AUTHENTICATION WITH THE WEB CONSOLE FOR CENTRALLY MANAGED USERS

You can configure smart card authentication in the RHEL web console for users who are centrally managed by:

- Identity Management
- Active Directory which is connected in the cross-forest trust with Identity Management

### Prerequisites

- The system for which you want to use the smart card authentication must be a member of an Active Directory or Identity Management domain.  
For details about joining the RHEL 9 system into a domain using the web console, see [Joining a RHEL system to an IdM domain using the web console](#).
- The certificate used for the smart card authentication must be associated with a particular user in Identity Management or Active Directory.  
For more details about associating a certificate with the user in Identity Management, see [Adding a certificate to a user entry in the IdM Web UI](#) or [Adding a certificate to a user entry in the IdM CLI](#).

### 8.1. SMART CARD AUTHENTICATION FOR CENTRALLY MANAGED USERS

A smart card is a physical device, which can provide personal authentication using certificates stored on the card. Personal authentication means that you can use smart cards in the same way as user passwords.

You can store user credentials on the smart card in the form of a private key and a certificate. Special software and hardware is used to access them. You insert the smart card into a reader or a USB socket and supply the PIN code for the smart card instead of providing your password.

Identity Management (IdM) supports smart card authentication with:

- User certificates issued by the IdM certificate authority.
- User certificates issued by the Active Directory Certificate Service (ADCS) certificate authority.



#### NOTE

If you want to start using smart card authentication, see the hardware requirements: [Smart Card support in RHEL8+](#).

### 8.2. INSTALLING TOOLS FOR MANAGING AND USING SMART CARDS

Before you can configure your smart card, you must install the corresponding tools that can generate certificates and start the **pscd** service.

#### Prerequisites

- You have **root** permissions.

### Procedure

1. Install the **opensc** and **gnutls-utils** packages:

```
# dnf -y install opensc gnutls-utils
```

2. Start the **pcscd** service.

```
# systemctl start pcscd
```

### Verification

- Verify that the **pcscd** service is up and running:

```
# systemctl status pcscd
```

## 8.3. PREPARING YOUR SMART CARD AND UPLOADING YOUR CERTIFICATES AND KEYS TO YOUR SMART CARD

Follow this procedure to configure your smart card with the **pkcs15-init** tool, which helps you to configure:

- Erasing your smart card
- Setting new PINs and optional PIN Unblocking Keys (PUKs)
- Creating a new slot on the smart card
- Storing the certificate, private key, and public key in the slot
- If required, locking the smart card settings as certain smart cards require this type of finalization

The **pkcs15-init** tool may not work with all smart cards. You must use the tools that work with the smart card you are using.

### Prerequisites

- The **opensc** package, which includes the **pkcs15-init** tool, is installed.  
For more details, see [Installing tools for managing and using smart cards](#).
- The card is inserted in the reader and connected to the computer.
- You have a private key, a public key, and a certificate to store on the smart card. In this procedure, **testuser.key**, **testuserpublic.key**, and **testuser.crt** are the names used for the private key, public key, and the certificate.
- You have your current smart card user PIN and Security Officer PIN (SO-PIN).

### Procedure

1. Erase your smart card and authenticate yourself with your PIN:

■

```
$ pkcs15-init --erase-card --use-default-transport-keys
```

Using reader with a card: *Reader name*

PIN [Security Officer PIN] required.

Please enter PIN [Security Officer PIN]:

The card has been erased.

2. Initialize your smart card, set your user PIN and PUK, and your Security Officer PIN and PUK:

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
  --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

The **pkcs15-init** tool creates a new slot on the smart card.

3. Set a label and the authentication ID for the slot:

```
$ pkcs15-init --store-pin --label testuser \
  --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

The label is set to a human-readable value, in this case, **testuser**. The **auth-id** must be two hexadecimal values, in this case it is set to **01**.

4. Store and label the private key in the new slot on the smart card:

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
  --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



#### NOTE

The value you specify for **--id** must be the same when storing your private key and storing your certificate in the next step. Specifying your own value for **--id** is recommended as otherwise a more complicated value is calculated by the tool.

5. Store and label the certificate in the new slot on the smart card:

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
  --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. Optional: Store and label the public key in the new slot on the smart card:

```
$ pkcs15-init --store-public-key testuserpublic.key \
  --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



#### NOTE

If the public key corresponds to a private key or certificate, specify the same ID as the ID of the private key or certificate.

- Optional: Certain smart cards require you to finalize the card by locking the settings:

```
$ pkcs15-init -F
```

At this stage, your smart card contains the certificate, private key, and public key in the newly created slot. You have also created your user PIN and PUK and the Security Officer PIN and PUK.

## 8.4. ENABLING SMART CARD AUTHENTICATION FOR THE WEB CONSOLE

To use smart card authentication in the web console, enable this authentication method in the **cockpit.conf** file.

Additionally, you can disable password authentication in the same file.

### Prerequisites

- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

- Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).
- Click **Terminal**.
- In the **/etc/cockpit/cockpit.conf**, set the **ClientCertAuthentication** to **yes**:

```
[WebService]
ClientCertAuthentication = yes
```

- Optional: Disable password based authentication in **cockpit.conf** with:

```
[Basic]
action = none
```

This configuration disables password authentication and you must always use the smart card.

- Restart the web console to ensure that the **cockpit.service** accepts the change:

```
# systemctl restart cockpit
```

## 8.5. LOGGING IN TO THE WEB CONSOLE WITH SMART CARDS

You can use smart cards to log in to the web console.

### Prerequisites

- A valid certificate stored in your smart card that is associated to a user account created in a Active Directory or Identity Management domain.

- PIN to unlock the smart card.
- The smart card has been put into the reader.
- You have installed the RHEL 9 web console.  
For instructions, see [Installing and enabling the web console](#).

### Procedure

1. Log in to the RHEL 9 web console.  
For details, see [Logging in to the web console](#).

The browser asks you to add the PIN protecting the certificate stored on the smart card.

2. In the **Password Required** dialog box, enter PIN and click **OK**.
3. In the **User Identification Request** dialog box, select the certificate stored in the smart card.
4. Select **Remember this decision**.  
The system does not open this window next time.



#### NOTE

This step does not apply to Google Chrome users.

5. Click **OK**.

You are now connected and the web console displays its content.

## 8.6. ENABLING PASSWORDLESS SUDO AUTHENTICATION FOR SMART CARD USERS

You can configure passwordless authentication to **sudo** and other services for smart card users in the web console.

As an alternative, if you use Red Hat Enterprise Linux Identity Management, you can declare the initial web console certificate authentication as trusted for authenticating to **sudo**, SSH, or other services. For that purpose, the web console automatically creates an S4U2Proxy Kerberos ticket in the user session.

### Prerequisites

- Identity Management installed.
- Active Directory connected in the cross-forest trust with Identity Management.
- Smart card set up to log in to the web console. See [Configuring smart card authentication with the web console for centrally managed users](#) for more information.

### Procedure

1. Set up constraint delegation rules to list which hosts the ticket can access.

**Example 8.1. Setting up constraint delegation rules**



The web console session runs host **host.example.com** and should be trusted to access its own host with **sudo**. Additionally, we are adding second trusted host - **remote.example.com**.

- Create the following delegation:
  - Run the following commands to add a list of target machines a particular rule can access:

```
# ipa servicedelegationtarget-add cockpit-target
# ipa servicedelegationtarget-add-member cockpit-target \
  --principals=host/host.example.com@EXAMPLE.COM \
  --principals=host/remote.example.com@EXAMPLE.COM
```

- To allow the web console sessions (HTTP/principal) to access that host list, use the following commands:

```
# ipa servicedelegationrule-add cockpit-delegation
# ipa servicedelegationrule-add-member cockpit-delegation \
  --principals=HTTP/host.example.com@EXAMPLE.COM
# ipa servicedelegationrule-add-target cockpit-delegation \
  --servicedelegationtargets=cockpit-target
```

## 2. Enable GSS authentication in the corresponding services:

- a. For sudo, enable the **pam\_sss\_gss** module in the **/etc/sss/sss.conf** file:
  - i. As root, add an entry for your domain to the **/etc/sss/sss.conf** configuration file.

```
[domain/example.com]
pam_gssapi_services = sudo, sudo-i
```

- ii. Enable the module in the **/etc/pam.d/sudo** file on the first line.

```
auth sufficient pam_sss_gss.so
```

- b. For SSH, update the **GSSAPIAuthentication** option in the **/etc/ssh/sshd\_config** file to **yes**.



### WARNING

The delegated S4U ticket is not forwarded to remote SSH hosts when connecting to them from the web console. Authenticating to sudo on a remote host with your ticket will not work.

## Verification

1. Log in to the web console using a smart card.
2. Click the **Limited access** button.

3. Authenticate using your smart card.

Alternatively:

- Try to connect to a different host with SSH.

## 8.7. LIMITING USER SESSIONS AND MEMORY TO PREVENT A DOS ATTACK

A certificate authentication is protected by separating and isolating instances of the **cockpit-ws** web server against attackers who wants to impersonate another user. However, this introduces a potential denial of service (DoS) attack: A remote attacker could create a large number of certificates and send a large number of HTTPS requests to **cockpit-ws** each using a different certificate.

To prevent such DoS attacks, the collective resources of these web server instances are limited. By default, limits for the number of connections and memory usage are set to 200 threads and 75 % (soft) or 90 % (hard) memory limit.

The example procedure demonstrates resource protection by limiting the number of connections and memory.

### Procedure

1. In the terminal, open the **system-cockpithttps.slice** configuration file:

```
# systemctl edit system-cockpithttps.slice
```

2. Limit the **TasksMax** to *100* and **CPUQuota** to *30%*:

```
[Slice]
# change existing value
TasksMax=100
# add new restriction
CPUQuota=30%
```

3. To apply the changes, restart the system:

```
# systemctl daemon-reload
# systemctl stop cockpit
```

Now, the new memory and user session lower the risk of DoS attacks on the **cockpit-ws** web server.

## 8.8. ADDITIONAL RESOURCES

- [Configuring Identity Management for smart card authentication](#) .
- [Configuring certificates issued by ADCS for smart card authentication in IdM](#) .
- [Configuring and importing local certificates to a smart card](#) .

## CHAPTER 9. SATELLITE HOST MANAGEMENT AND MONITORING IN THE WEB CONSOLE

After enabling RHEL web console integration on a Red Hat Satellite Server, you manage many hosts at scale in the web console.

Red Hat Satellite is a system management solution for deploying, configuring, and maintaining your systems across physical, virtual, and cloud environments. Satellite provides provisioning, remote management and monitoring of multiple Red Hat Enterprise Linux deployments with a centralized tool.

By default, RHEL web console integration is disabled in Red Hat Satellite. To access RHEL web console features for your hosts from within Red Hat Satellite, you must first enable RHEL web console integration on a Red Hat Satellite Server.

To enable the RHEL web console on your Satellite Server, enter the following command as **root**:

```
# satellite-installer --enable-foreman-plugin-remote-execution-cockpit --reset-foreman-plugin-remote-execution-cockpit-ensure
```

### Additional resources

- [Host management and monitoring by using the RHEL web console](#) in the Managing hosts in Red Hat Satellite guide