Red Hat Enterprise Linux 9.0 Beta

Managing systems using the RHEL 9 web console

A guide to using the web console for managing systems in RHEL 9

Last Updated: 2021-11-15

Red Hat Enterprise Linux 9.0 Beta Managing systems using the RHEL 9 web console

A guide to using the web console for managing systems in RHEL 9

Legal Notice

Abstract

This document describes how to manage physical and virtual Linux-based systems using the RHEL 9 web console. The instructions assume that the server used for management is running in RHEL 9.

Table of Contents

RHEL BETA RELEASE 3

MAKING OPEN SOURCE MORE INCLUSIVE 4

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 5

CHAPTER 1. GETTING STARTED USING THE RHEL WEB CONSOLE 6
1.1. WHAT IS THE RHEL WEB CONSOLE 6
1.2. INSTALLING AND ENABLING THE WEB CONSOLE 7
1.3. LOGGING IN TO THE WEB CONSOLE 7
1.4. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE 8
1.5. LOGGING IN TO THE WEB CONSOLE USING A ONE-TIME PASSWORD 9
1.6. RESTARTING THE SYSTEM USING THE WEB CONSOLE 10
1.7. SHUTTING DOWN THE SYSTEM USING THE WEB CONSOLE 10
1.8. CONFIGURING TIME SETTINGS USING THE WEB CONSOLE 10
1.9. DISABLING SMT TO PREVENT CPU SECURITY ISSUES USING THE WEB CONSOLE 11
1.10. ADDING A BANNER TO THE LOGIN PAGE 12
1.11. CONFIGURING AUTOMATIC IDLE LOCK IN THE WEB CONSOLE 14

CHAPTER 2. CONFIGURING THE HOST NAME IN THE WEB CONSOLE 16
2.1. HOST NAME 16
2.2. PRETTY HOST NAME IN THE WEB CONSOLE 16
2.3. SETTING THE HOST NAME USING THE WEB CONSOLE 16

CHAPTER 3. RED HAT WEB CONSOLE ADD-ONS 19
3.1. INSTALLING ADD-ONS 19
3.2. ADD-ONS FOR THE RHEL WEB CONSOLE 19

CHAPTER 4. CONFIGURING SMART CARD AUTHENTICATION WITH THE WEB CONSOLE FOR CENTRALLY MANAGED USERS 20
4.1. SMART CARD AUTHENTICATION FOR CENTRALLY MANAGED USERS 20
4.2. INSTALLING TOOLS FOR MANAGING AND USING SMART CARDS 20
4.3. STORING A CERTIFICATE ON A SMART CARD 21
4.4. ENABLING SMART CARD AUTHENTICATION FOR THE WEB CONSOLE 23
4.5. LOGGING IN TO THE WEB CONSOLE WITH SMART CARDS 23
4.6. ENABLING PASSWORD-LESS SUDO FOR SMART CARD USERS 24
4.7. LIMITING USER SESSIONS AND MEMORY TO PREVENT A DOS ATTACK 25

RHEL BETA RELEASE

Red Hat provides Red Hat Enterprise Linux Beta access to all subscribed Red Hat accounts. The purpose of Beta access is to:

- Provide an opportunity to customers to test major features and capabilities prior to the general availability release and provide feedback or report issues.
- Provide Beta product documentation as a preview. Beta product documentation is under development and is subject to substantial change.

Note that Red Hat does not support the usage of RHEL Beta releases in production use cases. For more information, see What does Beta mean in Red Hat Enterprise Linux and can I upgrade a RHEL Beta installation to a General Availability (GA) release?.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

CHAPTER 1. GETTING STARTED USING THE RHEL WEB CONSOLE

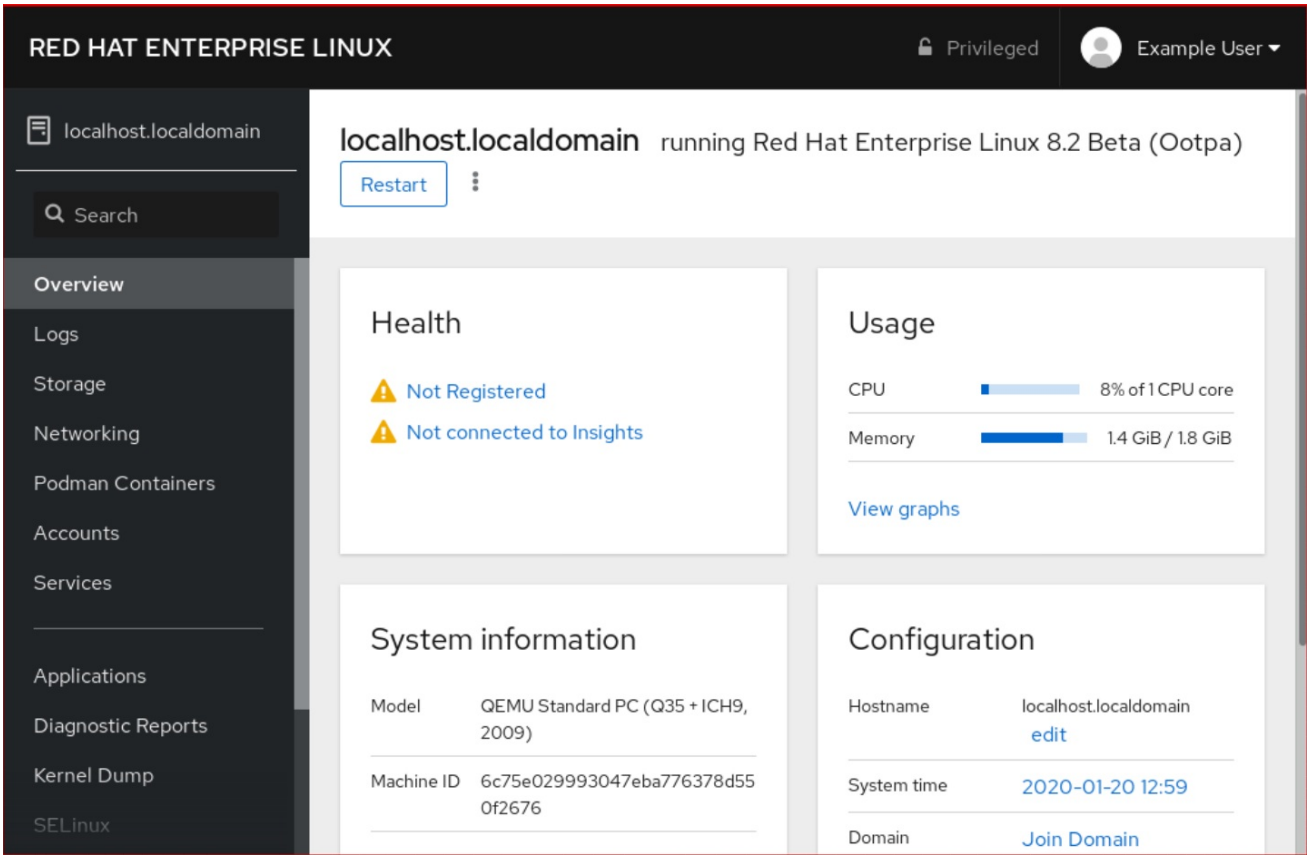
Install the web console in Red Hat Enterprise Linux 9 and learn how to add remote hosts and monitor them in the RHEL 9 web console.

Prerequisites

- Installed Red Hat Enterprise Linux 9.
- Enabled networking.
- Registered system with appropriate subscription attached.

1.1. WHAT IS THE RHEL WEB CONSOLE

The RHEL web console is a Red Hat Enterprise Linux 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 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 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 9 web console, first enable the **cockpit.socket** service.

Red Hat Enterprise Linux 8 includes the RHEL 9 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:

```
# yum 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 steps

1. To verify the previous installation and configuration,

1.3. LOGGING IN TO THE WEB CONSOLE

Use the steps in this procedure for the first login to the RHEL web console using a system user name and password.

Prerequisites

- Use one of the following browsers for opening 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 PAM stack located at **/etc/pam.d/cockpit**. Authentication with PAM allows you to log in with the user name and password of any local account on the system.

Procedure

1. Open the web console in your web browser:
   - Locally: **https://localhost:9090**
   - Remotely with the server's hostname: **https://example.com:9090**
   - Remotely with the server's IP address: **https://192.0.2.2:9090**
     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. In the login screen, enter your system user name and password.
3. Click **Log In**.

After successful authentication, the RHEL web console interface opens.

1.4. CONNECTING TO THE WEB CONSOLE FROM A REMOTE MACHINE

It is possible to connect to your web console interface from any client operating system and also from mobile phones or tablets.

Prerequisites

- Device with a supported internet browser, such as:
  - Mozilla Firefox 52 and later
  - Google Chrome 57 and later
  - Microsoft Edge 16 and later
- RHEL 9 server you want to access with an installed and accessible 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: **server.hostname.example.com:port_number**
   b. With the server's IP address: **server.IP_address:port_number**
3. After the login interface opens, log in with your RHEL machine credentials.

1.5. 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

- The RHEL web console has been installed.
- 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.

1.6. RESTARTING THE SYSTEM USING THE WEB CONSOLE

You can use the web console to restart a RHEL system that the web console is attached to.

Prerequisites

- The web console is installed and accessible.

Procedure

1. Log into the RHEL 8 web console.
2. Click **Overview**.
3. Click the **Restart** restart button.
4. If any users are logged into the system, write a reason for the restart in the **Restart** dialog box.
5. Optional: In the **Delay** drop down list, select a time interval.
6. Click **Restart**.

1.7. SHUTTING DOWN THE SYSTEM USING THE WEB CONSOLE

You can use the web console to shut down a RHEL system that the web console is attached to.

Prerequisites

- The web console is installed and accessible.

Procedure

1. Log into the RHEL 8 web console.
2. Click **Overview**.
3. In the **Restart** drop down list, select **Shut Down**.
4. If any users are logged in to the system, write a reason for the shutdown in the **Shut Down** dialog box.
5. Optional: In the **Delay** drop down list, select a time interval.
6. Click **Shut Down**.

1.8. CONFIGURING TIME SETTINGS USING THE WEB CONSOLE

You can set a time zone and synchronize the system time with a Network Time Protocol (NTP) server.

Prerequisites

- The web console is installed and accessible.

Procedure

1. Log in to the RHEL 8 web console.
2. Click the current system time in **Overview**.
3. In the **Change System Time** dialog box, change the time zone if necessary.
4. In the **Set Time** drop down menu, select one of the following:

   Manually

   Use this option if you need to set the time manually, without an NTP server.

   Automatically using NTP server

   This is a default option, which synchronizes time automatically with the preset NTP servers.

   Automatically using specific NTP servers

   Use this option only if you need to synchronize the system with a specific NTP server. Specify the DNS name or the IP address of the server.

5. Click **Change**.

- Check the system time displayed in the **System** tab.

1.9. DISABLING SMT TO PREVENT CPU SECURITY ISSUES USING THE WEB CONSOLE

Disable Simultaneous Multi Threading (SMT) in case of attacks that misuse CPU SMT. Disabling SMT can mitigate security vulnerabilities, such as L1TF or MDS.

IMPORTANT

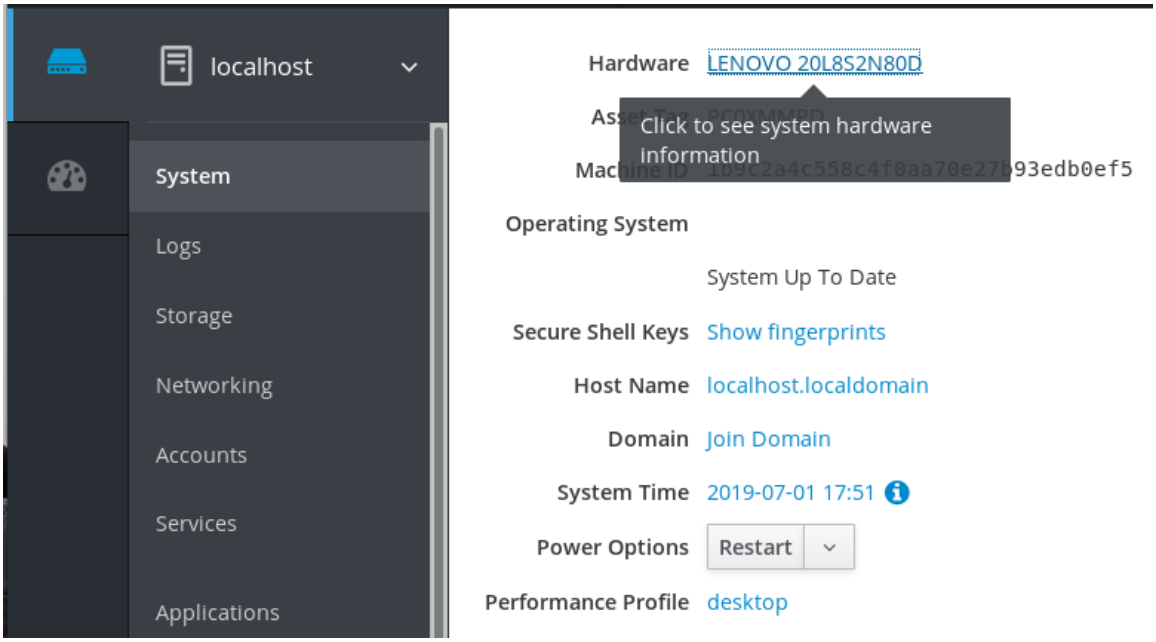
Disabling SMT might lower the system performance.

Prerequisites

- The web console must be installed and accessible.

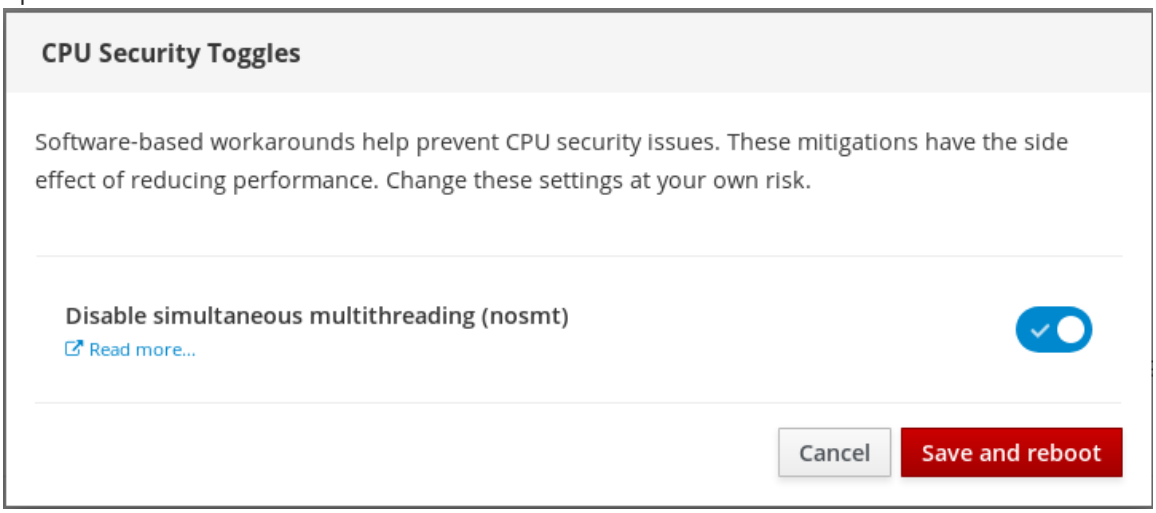
Procedure

1. Log in to the RHEL 8 web console.
2. Click **System**.
3. In the **Hardware** item, click the hardware information.



4. In the **CPU Security** item, click **Mitigations**.
   If this link is not present, it means that your system does not support SMT, and therefore is not vulnerable.

5. In the **CPU Security Toggles**, switch on the **Disable simultaneous multithreading (nosmt)** option.



6. Click on the **Save and reboot** button.

After the system restart, the CPU no longer uses SMT.

Additional resources

- L1TF - L1 Terminal Fault Attack - CVE-2018-3620 & CVE-2018-3646
- MDS - Microarchitectural Data Sampling - CVE-2018-12130, CVE-2018-12126, CVE-2018-12127, and CVE-2019-11091

1.10. ADDING A BANNER TO THE LOGIN PAGE

Companies or agencies sometimes need to show a warning that usage of the computer is for lawful purposes, the user is subject to surveillance, and anyone trespassing will be prosecuted. The warning must be visible before login. Similarly to SSH, the web console can optionally show the content of a

banner file on the login screen. To enable banners in your web console sessions, you need to modify the **/etc/cockpit/cockpit.conf** file. Note that the file is not required and you may need to create it manually.

Prerequisites

- The web console is installed and accessible.
- You must have sudo privileges.

Procedure

1. Create the **/etc/issue.cockpit** file in a text editor of your preference if you do not have it yet. Add the content you want to display as the banner to the file.
   Do not include any macros in the file as there is no re-formatting done between the file content and the displayed content. Use intended line breaks. It is possible to use ASCII art.
2. Save the file.
3. Open or create the **cockpit.conf** file in the **/etc/cockpit/** directory in a text editor of your preference.

```
$ sudo vi cockpit.conf
```

4. Add the following text to the file:

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

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

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

Verification steps

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

Example 1.1. Adding an example banner to the login page

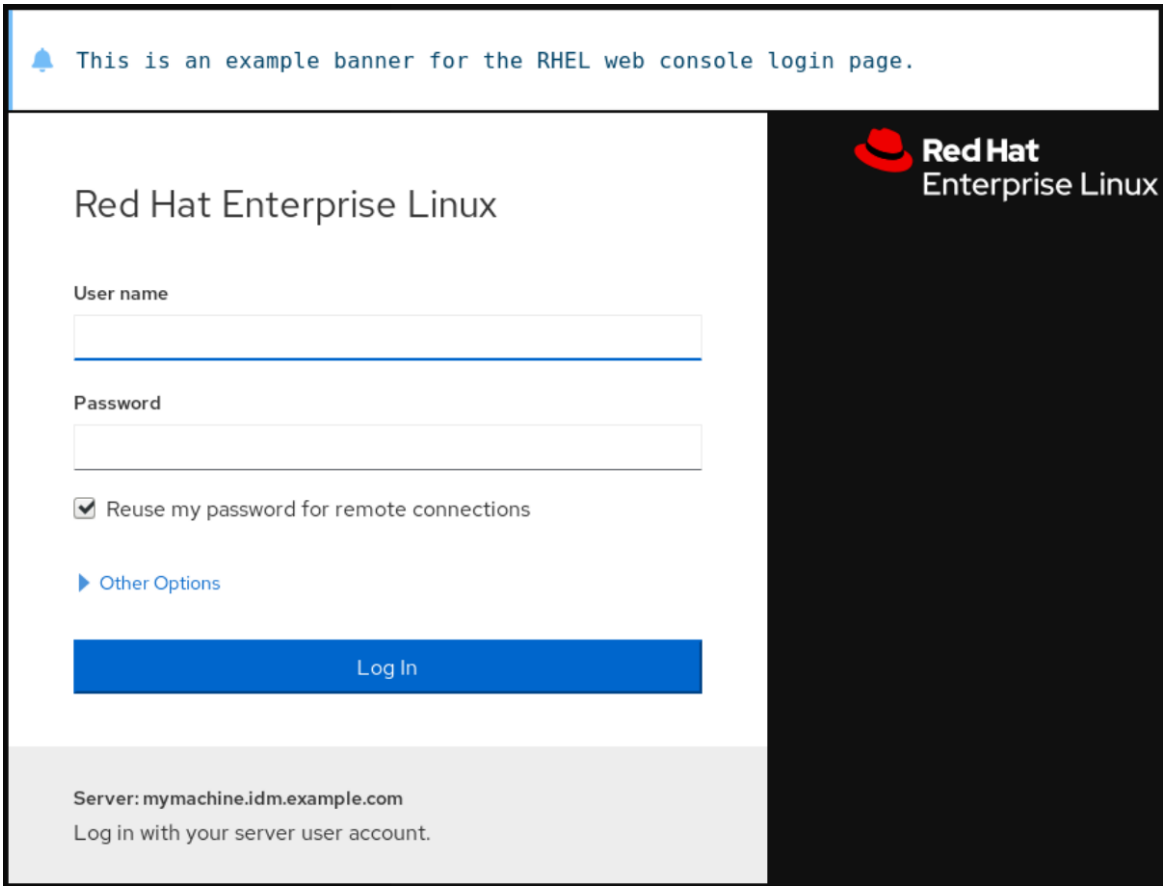
1. Create an **/etc/issue.cockpit** file with a desired text using a text editor:

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

2. Open or create the **/etc/cockpit/cockpit.conf** file and add the following text:

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

3. Restart the web console.
4. Open the web console login screen again.



1.11. CONFIGURING AUTOMATIC IDLE LOCK IN THE WEB CONSOLE

By default, there is no idle timeout set in the web console interface. If you wish to enable an idle timeout on your system, you can do so by modifying the **/etc/cockpit/cockpit.conf** configuration file. Note that the file is not required and you may need to create it manually.

Prerequisites

- The web console must be installed and accessible.
- You must have sudo privileges.

Procedure

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

```
$ sudo vi 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 steps

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

CHAPTER 2. CONFIGURING THE HOST NAME IN THE WEB CONSOLE

Learn how to use the Red Hat Enterprise Linux web console to configure different forms of the host name on the system that the web console is attached to.

2.1. HOST NAME

The host name identifies the system. By default, the host name is set to **localhost**, but you can change it.

A host name consists of two parts:

Host name

It is a unique name which identifies a system.

Domain

Add the domain as a suffix behind the host name when using a system in a network and when using names instead of just IP addresses.

A host name with an attached domain name is called a fully qualified domain name (FQDN). For example: **mymachine.example.com**.

Host names are stored in the **/etc/hostname** file.

2.2. PRETTY HOST NAME IN THE WEB CONSOLE

You can configure a pretty host name in the RHEL web console. The pretty host name is a host name with capital letters, spaces, and so on.

The pretty host name displays in the web console, but it does not have to correspond with the host name.

> **Example 2.1. Host name formats in the web console**
>
> **Pretty host name**
>
> **My Machine**
>
> **Host name**
>
> **mymachine**
>
> **Real host name - fully qualified domain name (FQDN)**
>
> **mymachine.idm.company.com**

2.3. SETTING THE HOST NAME USING THE WEB CONSOLE

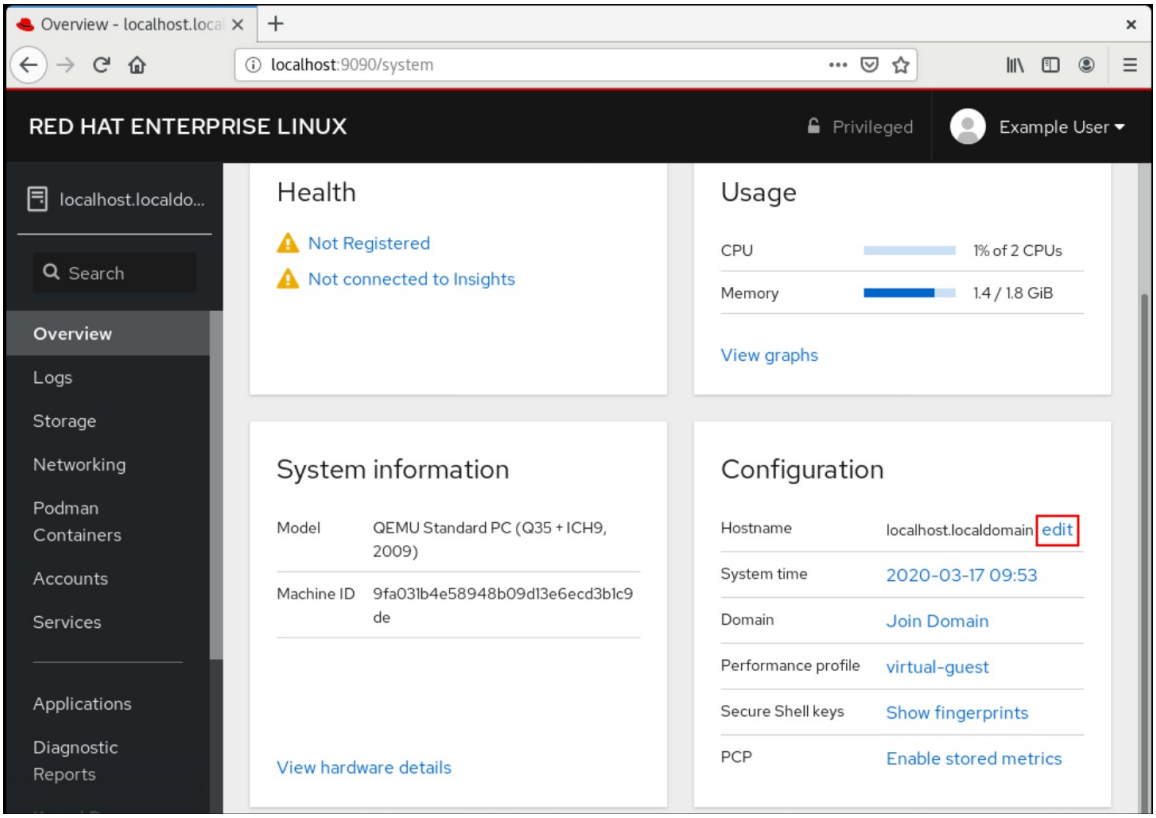
This procedure sets the real host name or the pretty host name in the web console.

Prerequisites

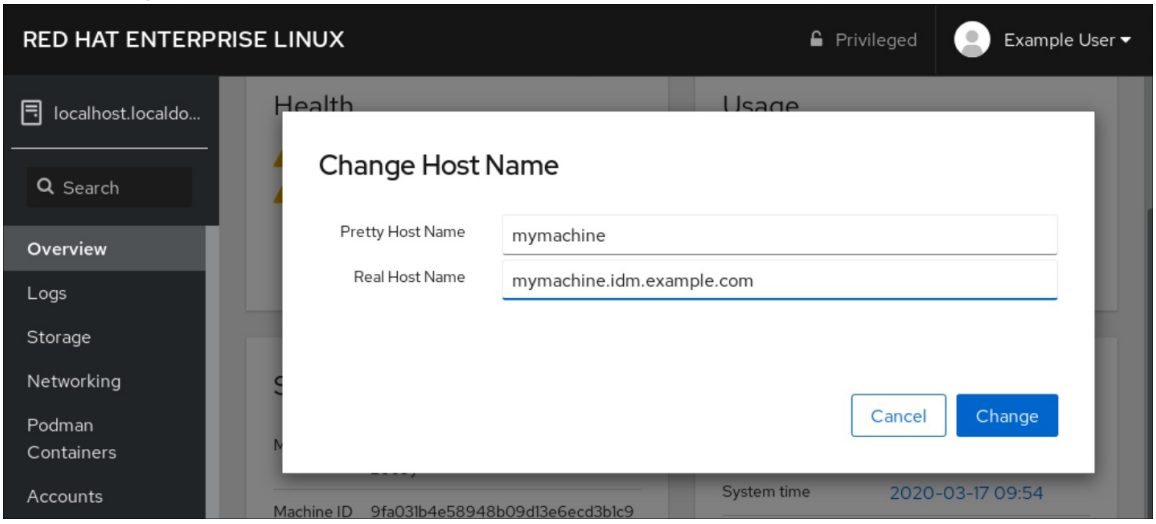
- The web console is installed and accessible.

Procedure

1. Log into the web console.
2. Click **Overview**.
3. Click **edit** next to the current host name.



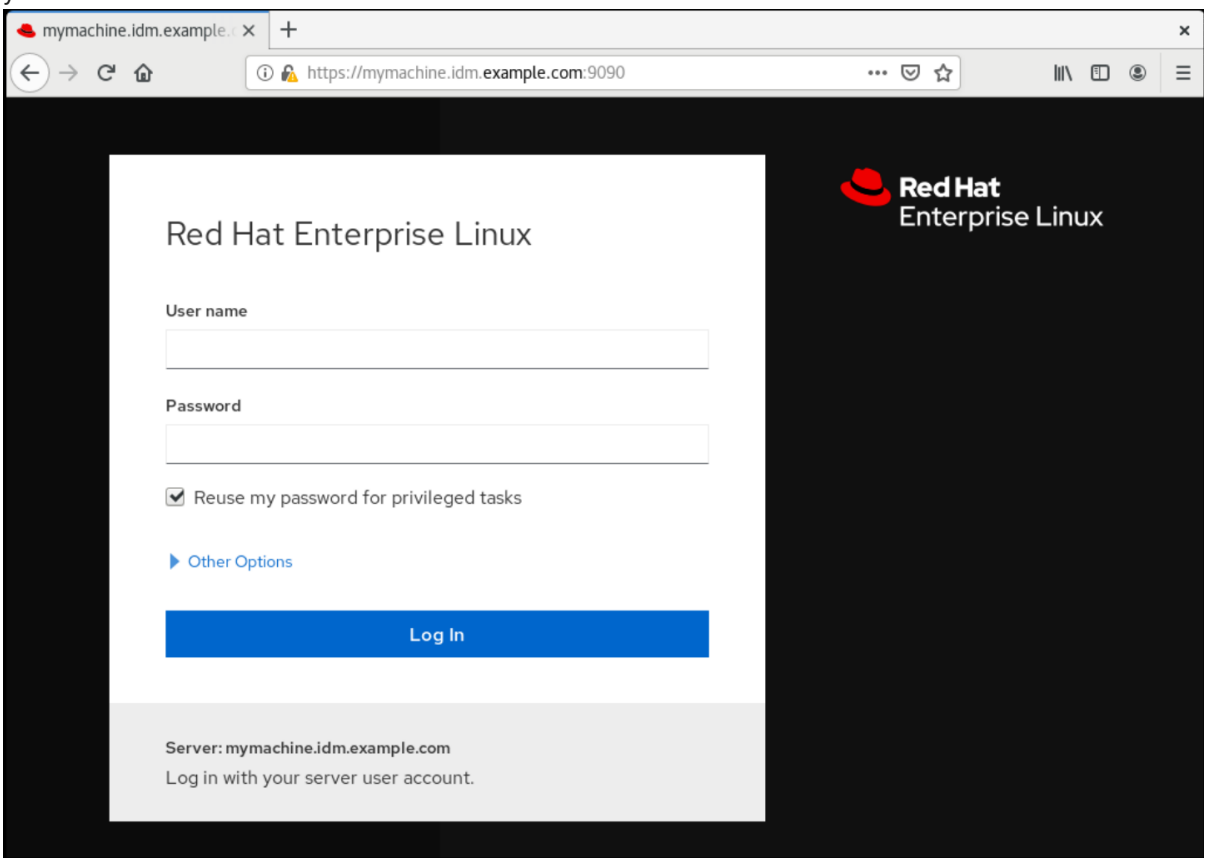
4. In the **Change Host Name** dialog box, enter the host name in the **Pretty Host Name** field.
5. The **Real Host Name** field attaches a domain name to the pretty name.
   You can change the real host name manually if it does not correspond with the pretty host name.
6. Click **Change**.



Verification steps

1. Log out from the web console.

2. Reopen the web console by entering an address with the new host name in the address bar of your browser.



CHAPTER 3. RED HAT WEB CONSOLE ADD-ONS

Install add-ons in the RHEL web console and learn what add-on applications are available for you.

3.1. INSTALLING ADD-ONS

The **cockpit** package is a part of Red Hat Enterprise Linux by default. To be able to use add-on applications you must install them separately.

Prerequisites

- Installed and enabled the **cockpit** package.

Procedure

- Install an add-on.

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

3.2. ADD-ONS FOR THE RHEL WEB CONSOLE

The following table lists available add-on applications for the RHEL web console.

Feature name	Package name	Usage
Composer	cockpit-composer	Building custom OS images
Dashboard	cockpit-dashboard	Managing multiple servers in one UI
Machines	cockpit-machines	Managing libvirt virtual machines
PackageKit	cockpit-packagekit	Software updates and application installation (usually installed by default)
PCP	cockpit-pcp	Persistent and more fine-grained performance data (installed on demand from the UI)
podman	cockpit-podman	Managing podman containers
Session Recording	cockpit-session-recording	Recording and managing user sessions

CHAPTER 4. CONFIGURING SMART CARD AUTHENTICATION WITH THE WEB CONSOLE FOR CENTRALLY MANAGED USERS

Configure smart card authentication in the RHEL 8 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.
- 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.

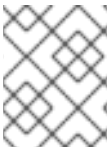
4.1. SMART CARD AUTHENTICATION FOR CENTRALLY MANAGED USERS

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 .

4.2. INSTALLING TOOLS FOR MANAGING AND USING SMART CARDS

To configure your smart card, you need tools which can generate certificates and store them on a smart card.

You must:

- Install the **gnutls-utils** package which helps you to manage certificates.

- Install the **opensc** package which provides a set of libraries and utilities to work with smart cards.
- Start the **pcscd** service which communicates with the smart card reader.

Procedure

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

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

2. Start the **pcscd** service.

```
# systemctl start pcscd
```

Verify that the **pcscd** service is up and running.

4.3. STORING A CERTIFICATE ON A SMART CARD

This section describes smart card configuration 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
- Locking the smart card settings (some smart cards require this type of finalization)

Prerequisites

- The **opensc** package, which includes the **pkcs15-init** tool is installed.
  For details, see Installing tools for managing and using smart cards .
- The card is inserted in the reader and connected to the computer.
- You have the private key, public key, and 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.
- 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 **pcks15-init** tool creates a new slot on the smart card.

3. Set the 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 certificate. If you do not specify a value for **--id**, a more complicated value is calculated by the tool and it is therefore easier to define your own value.

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 and/or certificate, you should specify the same ID as that private key and/or certificate.

7. (Optional) Some smart cards require you to finalize the card by locking the settings:

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

At this stage, your smart card includes 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.

4.4. ENABLING SMART CARD AUTHENTICATION FOR THE WEB CONSOLE

To be able to use smart card authentication in the web console, enable smart card authentication in the **cockpit.conf** file.

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

Prerequisites

- The RHEL 8 web console has been installed.

Procedure

1. Log in to the RHEL web console with administrator privileges.
2. Click **Terminal**.
3. In the **/etc/cockpit/cockpit.conf**, set the **ClientCertAuthentication** to **yes**:

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

4. Optionally, disable password based authentication in **cockpit.conf** with:

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

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

5. Restart the web console to make sure that the **cockpit.service** accepts the change:

```
# systemctl restart cockpit
```

4.5. 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.

Procedure

1. Open your web browser and add the web console's address in the address bar.
   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.
5. Click **OK**.

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

4.6. ENABLING PASSWORD-LESS SUDO FOR SMART CARD USERS

Once you logged into web console with a certificate, you may need to switch to administrative mode (root privileges through **sudo**). If your user account has a password, it can be used to authenticate to **sudo**.

As an alternative, if you are using Red Hat 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
- 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 4.1. 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 web console sessions (HTTP/principal) to access that host list, run 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/sssd/sssd.conf** file:

      i. As root, add an entry for your domain to the **/etc/sssd/sssd.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 literal:[GSSAPIAuthentication] option in the **/etc/ssh/sshd_config** file to literal:[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.

OR

1. Try to connect to a different host with SSH.

4.7. LIMITING USER SESSIONS AND MEMORY TO PREVENT A DOS ATTACK

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 this DoS, the collective resources of these web server instances are limited. By default, limits to the number of connections and to memory usage are set to 200 threads and a 75% (soft) / 90% (hard) memory limit.

The following procedure describes 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 limits protect the **cockpit-ws** web server from DoS attacks.