



## Red Hat Enterprise Linux 9

# RHEL 9 Web コンソールを使用したシステムの管理

グラフィカルな Web ベースのインターフェイスによるサーバー管理



# Red Hat Enterprise Linux 9 RHEL 9 Web コンソールを使用したシステムの管理

---

グラフィカルな Web ベースのインターフェイスによるサーバー管理

## 法律上の通知

Copyright © 2024 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.

## 概要

RHEL Web コンソールは、アップストリームの Cockpit プロジェクトに基づいた Web ベースのグラフィカルインターフェイスです。これを使用すると、systemd サービスの検査と制御、ストレージの管理、ネットワークの設定、ネットワークの問題の分析、ログの検査などのシステム管理タスクを実行できます。

## 目次

RED HAT ドキュメントへのフィードバック (英語のみ)	4
<b>第1章 RHEL WEB コンソールの使用</b>	<b>5</b>
1.1. RHEL WEB コンソールの概要	5
1.2. WEB コンソールのインストールおよび有効化	6
1.3. WEB コンソールへのログイン	6
1.4. WEB コンソールでの基本認証の無効化	7
1.5. リモートマシンから WEB コンソールへの接続	8
1.6. ROOT ユーザーとしてリモートマシンからの WEB コンソールへの接続	9
1.7. ワンタイムパスワードを使用した WEB コンソールへのログイン	9
1.8. ログインページへのバナーの追加	10
1.9. WEB コンソールでの自動アイドルロックの設定	11
1.10. WEB コンソールのリッスンポートの変更	12
<b>第2章 RHEL システムロールを使用して WEB コンソールをインストールおよび設定する</b>	<b>15</b>
2.1. COCKPIT RHEL システムロールを使用した WEB コンソールのインストール	15
<b>第3章 WEB コンソールアドオンのインストールとカスタムページの作成</b>	<b>17</b>
3.1. RHEL WEB コンソールのアドオン	17
3.2. WEB コンソールでの新しいページの作成	17
3.3. WEB コンソールでのマニフェスト設定のオーバーライド	18
<b>第4章 WEB コンソールでソフトウェア更新の管理</b>	<b>19</b>
4.1. WEB コンソールでの手動ソフトウェア更新の管理	19
4.2. WEB コンソールで自動ソフトウェア更新の管理	19
4.3. WEB コンソールでソフトウェア更新適用後のオンデマンド再起動の管理	20
4.4. WEB コンソールでのカーネルライブパッチを使用したパッチ適用	21
<b>第5章 WEB コンソールでサブスクリプションの管理</b>	<b>23</b>
5.1. WEB コンソールでサブスクリプションの管理	23
5.2. WEB コンソールで認証情報を使用してサブスクリプションを登録	23
5.3. WEB コンソールでアクティベーションキーを使用してサブスクリプションを登録	25
<b>第6章 WEB コンソールでリモートシステムの管理</b>	<b>27</b>
6.1. WEB コンソールのリモートシステムマネージャー	27
6.2. WEB コンソールへのリモートシステムの追加	28
6.3. 新しいホストの SSH ログインの有効化	29
6.4. スマートカードで認証されたユーザーが、再度認証を要求されることなくリモートホストに SSH 接続できるようにするための WEB コンソールの設定	32
6.5. ANSIBLE を使用して WEB コンソールを設定し、スマートカードで認証されたユーザーが再認証を求められることなくリモートホストに SSH 接続できるようにする	34
<b>第7章 IDM ドメインで RHEL 9 WEB コンソールにシングルサインオンを設定</b>	<b>38</b>
7.1. WEB コンソールを使用した RHEL 9 システムの IDM ドメインへの参加	38
7.2. KERBEROS 認証を使用した WEB コンソールへのログイン	39
<b>第8章 集中管理ユーザー向けに WEB コンソールを使用したスマートカード認証の設定</b>	<b>41</b>
8.1. 集中管理ユーザーのスマートカード認証	41
8.2. スマートカードを管理および使用するツールのインストール	41
8.3. スマートカードを準備し、証明書と鍵をスマートカードにアップロードする	42
8.4. WEB コンソールのスマートカード認証の有効化	44
8.5. スマートカードを使用して WEB コンソールへのログイン	44
8.6. スマートカードユーザーに対するパスワードなしの SUDO 認証の有効化	45
8.7. DOS 攻撃を防ぐためのユーザーセッションおよびメモリーの制限	47

8.8. 関連情報	48
第9章 WEB コンソールでの SATELLITE ホストの管理と監視 .....	49



## RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見や感想をお寄せください。また、改善点があればお知らせください。

### Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

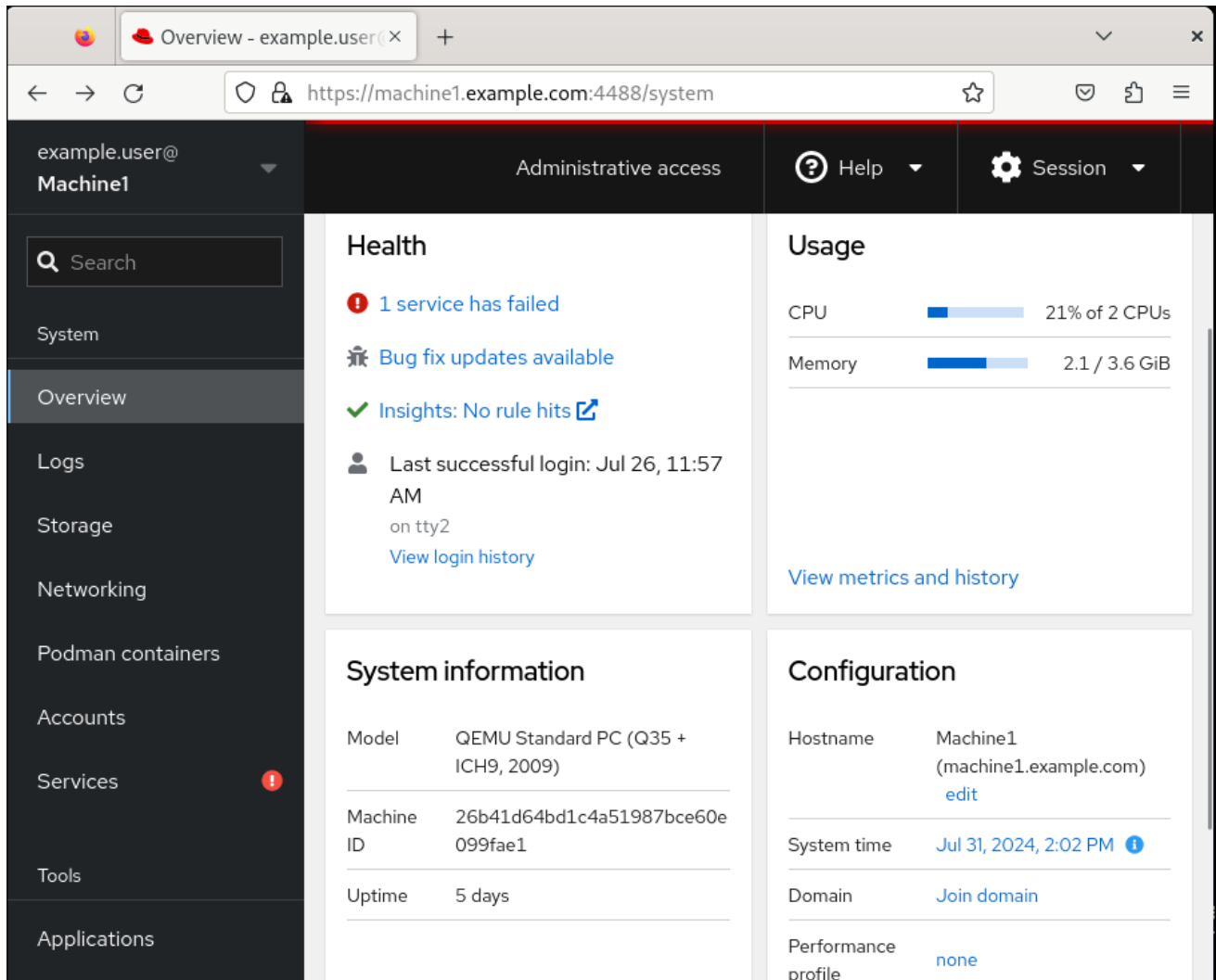


## 第1章 RHEL WEB コンソールの使用

Red Hat Enterprise Linux 9 Web コンソールのインストール方法、便利なグラフィカルインターフェイスから [リモートホストを追加および管理](#) する方法、Web コンソールによって管理されるシステムを監視する方法を説明します。

### 1.1. RHEL WEB コンソールの概要

RHEL Web コンソールは、ローカルシステム、およびネットワーク環境にある Linux サーバーを管理および監視するために設計された Web ベースのインターフェイスです。



RHEL Web コンソールでは、以下を含むさまざまな管理タスクの実行が可能です。

- サービスの管理
- ユーザーアカウントの管理
- システムサービスの管理および監視
- ネットワークインターフェイスおよびファイアウォールの設定
- システムログの確認
- 仮想マシンの管理

- 診断レポートの作成
- カーネルダンプ設定の設定
- SELinux の設定
- ソフトウェアの更新
- システムサブスクリプションの管理

RHEL Web コンソールは、ターミナルで使用するのと同じシステム API を使用します。ターミナルで実行した操作は、即座に RHEL Web コンソールに反映されます。

ネットワーク環境のシステムのログや、パフォーマンスをグラフで監視できます。さらに、Web コンソールで設定を直接変更したり、ターミナルから設定を変更できます。

## 1.2. WEB コンソールのインストールおよび有効化

RHEL Web コンソールにアクセスするには、最初に **cockpit.socket** サービスを有効にします。

Red Hat Enterprise Linux 9 では、多くのインストール方法で、Web コンソールがデフォルトでインストールされます。ご使用のシステムがこれに該当しない場合は、**cockpit** パッケージをインストールしてから **.socket** サービスを有効にしてください。

### 手順

1. Web コンソールがインストールバリエーションにデフォルトでインストールされていない場合は、**cockpit** パッケージを手動でインストールします。

```
# dnf install cockpit
```

2. 必要に応じて、Web サーバーを実行する **cockpit.socket** サービスを有効にして起動します。

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

3. Web コンソールがインストールバリエーションにデフォルトでインストールされておらず、カスタムのファイアウォールプロファイルを使用している場合は、**cockpit** サービスを **firewalld** に追加して、ファイアウォールの 9090 番ポートを開きます。

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

### 検証

- 以前のインストールと設定を確認するには、[Web コンソールを開きます](#)。

## 1.3. WEB コンソールへのログイン

**cockpit.socket** サービスが実行中で、対応するファイアウォールポートが開いている場合、ブラウザで Web コンソールに初めてログインできます。

### 前提条件

- 次のブラウザーのいずれかを使用して Web コンソールを開いている。
  - Mozilla Firefox 52 以降
  - Google Chrome 57 以降
  - Microsoft Edge 16 以降
- システムユーザーアカウントの認証情報  
RHEL Web コンソールは、`/etc/pam.d/cockpit`にある特定のプラグ可能な認証モジュール (PAM) スタックを使用します。デフォルト設定では、システム上の任意のローカルアカウントのユーザー名とパスワードを使用してログインできます。
- ファイアウォールでポート 9090 が開いている。

## 手順

1. Web ブラウザーに次のアドレスを入力して Web コンソールにアクセスします。

`https://localhost:9090`



### 注記

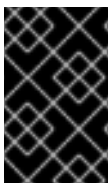
これにより、ローカルマシン上で Web コンソールログインが可能になります。リモートシステムの Web コンソールにログインする場合、[「リモートマシンから Web コンソールへの接続」](#) を参照してください。

自己署名証明書を使用する場合は、ブラウザーに警告が表示されます。証明書を確認し、セキュリティ例外を許可して、ログインを続行します。

コンソールは `/etc/cockpit/ws-certs.d` ディレクトリーから証明書をロードし、アルファベット順で最後となる `.cert` 拡張子のファイルを使用します。セキュリティの例外を承認しなくてもすむように、認証局 (CA) が署名した証明書をインストールします。

2. ログイン画面で、システムユーザー名とパスワードを入力します。
3. **Log In** をクリックします。

認証に成功すると、RHEL Web コンソールインターフェイスが開きます。



### 重要

制限付きアクセスと管理アクセスを切り替えるには、Web コンソールページのトップパネルで **Administrative access** または **Limited access** をクリックします。管理者アクセスを取得するには、ユーザーパスワードを入力する必要があります。

## 1.4. WEB コンソールでの基本認証の無効化

`cockpit.conf` ファイルを変更することで、認証方式の動作を変更できます。**none** アクションを使用して認証方式を無効にし、GSSAPI とフォームによる認証のみを許可します。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。

手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

- **sudo** を使用して管理コマンドを入力するための **root** 権限または権限がある。

## 手順

1. 任意のテキストエディターで、**/etc/cockpit/** ディレクトリーの **cockpit.conf** ファイルを開くか、作成します。次に例を示します。

```
# vi cockpit.conf
```

2. 次のテキストを追加します。

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

3. ファイルを保存します。
4. Web コンソールを再起動して、変更を有効にします。

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

## 1.5. リモートマシンから WEB コンソールへの接続

Web コンソールインターフェイスには、クライアントオペレーティングシステムだけでなく、携帯電話やタブレットからも接続できます。

### 前提条件

- 対応しているインターネットブラウザを備えたデバイス。以下に例を示します。
  - Mozilla Firefox 52 以降
  - Google Chrome 57 以降
  - Microsoft Edge 16 以降
- インストール済みのアクセス可能な Web コンソールを使用してアクセスする RHEL 9。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

## 手順

1. Web ブラウザーを開きます。
2. リモートサーバーのアドレスを次のいずれかの形式で入力します。
  - a. サーバーのホスト名を使用する場合:

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

以下に例を示します。

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

- b. サーバーの IP アドレスを使用する場合:

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

以下に例を示します。

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

3. ログインインターフェイスが開いたら、RHEL システムの認証情報を使用してログインします。

## 1.6. ROOT ユーザーとしてリモートマシンからの WEB コンソールへの接続

RHEL 9.2 以降の新規インストールでは、セキュリティ上の理由から、RHEL Web コンソールはデフォルトで root アカountのログインを許可しません。`/etc/cockpit/disallowed-users` ファイルで root ログインを許可できます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. `/etc/cockpit/` ディレクトリーにある **disallowed-users** ファイルを任意のテキストエディターで開きます。次に例を示します。

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

2. このファイルを編集して、**root** ユーザーの行を削除します。

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

3. 変更を保存し、エディターを終了します。

### 検証

- Web コンソールに **root** ユーザーとしてログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。

## 1.7. ワンタイムパスワードを使用した WEB コンソールへのログイン

ワンタイムパスワード (OTP) 設定が有効になっている Identity Management (IdM) ドメインにシステムが含まれている場合には、OTP を使用して RHEL Web コンソールにログインできます。



### 重要

ワンタイムパスワードを使用してログインできるのは、OTP 設定が有効な Identity Management (IdM) ドメインに、お使いのシステムが含まれる場合のみです。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。
- Identity Management サーバーで OTP 設定を有効しておく。
- OTP トークンを生成する設定済みのハードウェアまたはソフトウェアのデバイス

## 手順

1. ブラウザーで RHEL Web コンソールを開きます。

- ローカルの場合 - **`https://localhost:PORT_NUMBER`**
- リモートでサーバーのホスト名を使用する場合 - **`https://example.com:PORT_NUMBER`**
- リモートでサーバーの IP アドレスを使用する場合 -  
**`https://EXAMPLE.SERVER.IP.ADDR:PORT_NUMBER`**

自己署名証明書を使用する場合は、ブラウザーに警告が表示されます。証明書を確認し、セキュリティ例外を許可してから、ログインを続行します。

コンソールは `/etc/cockpit/ws-certs.d` ディレクトリーから証明書をロードし、アルファベット順で最後となる `.cert` 拡張子のファイルを使用します。セキュリティの例外を承認しなくてもすむように、認証局 (CA) が署名した証明書をインストールします。

2. ログイン画面が表示されます。ログイン画面で、システムユーザーの名前とパスワードを入力します。
3. デバイスでワンタイムパスワードを生成します。
4. パスワードを確認してから、Web コンソールインターフェイスに表示される新規フィールドにワンタイムパスワードを入力します。
5. **Log in** をクリックします。
6. ログインに成功すると、Web コンソールインターフェイスの **Overview** ページに移動します。

## 1.8. ログインページへのバナーの追加

ログイン画面にバナーファイルの内容を表示するように Web コンソールを設定できます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。
- **sudo** を使用して管理コマンドを入力するための **root** 権限または権限がある。

## 手順

1. 任意のテキストエディターで `/etc/issue.cockpit` ファイルを開きます。

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

2. バナーとして表示するコンテンツをファイルに追加します。次に例を示します。

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

ファイルにマクロを含めることはできませんが、改行と ASCII アートは使用できます。

3. ファイルを保存します。
4. 任意のテキストエディターで、**/etc/cockpit/** ディレクトリーの **cockpit.conf** ファイルを開きます。次に例を示します。

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

5. 以下のテキストをファイルに追加します。

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

6. ファイルを保存します。
7. Web コンソールを再起動して、変更を有効にします。

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

## 検証

- Web コンソールのログイン画面を再度開き、バナーが表示されていることを確認します。

## 1.9. WEB コンソールでの自動アイドルロックの設定

Web コンソールインターフェイスを使用して、自動アイドルロックを有効にし、システムのアイドルタイムアウトを設定できます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。
- **sudo** を使用して管理コマンドを入力するための **root** 権限または権限がある。

### 手順

1. 任意のテキストエディターで、**/etc/cockpit/** ディレクトリーの **cockpit.conf** ファイルを開きます。次に例を示します。

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

2. 以下のテキストをファイルに追加します。

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

<X> は、任意の期間の数値 (分単位) に置き換えます。

3. ファイルを保存します。
4. Web コンソールを再起動して、変更を有効にします。

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

### 検証

- 設定の期間後にセッションがログアウトされているかどうかを確認します。

## 1.10. WEB コンソールのリッスンポートの変更

デフォルトでは、RHEL Web コンソールは TCP ポート 9090 を介して通信します。このポート番号は、デフォルトのソケット設定をオーバーライドすることで変更できます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。
- **sudo** を使用して管理コマンドを入力するための **root** 権限または権限がある。
- **firewalld** サービスが実行している。

### 手順

1. 使用されていないポート (例: <4488/tcp>) を選択し、**cockpit** サービスがそのポートにバインドできるように SELinux に指示します。



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

ポートは1つのサービスのみで一度に使用できるため、すでに使用しているポートを使用しようとすると、**ValueError:Port already defined** エラーが発生します。

2. ファイアウォールで新しいポートを開き、以前のポートを閉じます。

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

3. **cockpit.socket** サービスのオーバーライドファイルを作成します。

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

4. 次に表示されるエディター画面で、**/etc/systemd/system/cockpit.socket.d/** ディレクトリーにある空の **override.conf** ファイルが開きます。次の行を追加して、Web コンソールのデフォルトポートを、9090 から、先ほど選択した番号に変更します。

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

最初の **ListenStream=** ディレクティブの値が空になっているのは意図的であることに注意してください。単一のソケットユニットで複数の **ListenStream** ディレクティブを宣言します。このドロップインファイルに空の値を指定すると、リストがリセットされ、元のユニットのデフォルトポート 9090 が無効になります。



### 重要

上記のコードスニペットは、**# Anything between here** と **# Lines below this** で始まる行の間に挿入してください。それ以外の場合、システムによって変更が破棄されます。

5. **Ctrl + O** と **Enter** を押して変更を保存します。 **Ctrl + X** を押してエディターを終了します。
6. 変更した設定を再読み込みします。

```
# systemctl daemon-reload
```

7. 設定が機能していることを確認します。

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

8. **cockpit.socket** を再起動します。

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

### 検証

- Web ブラウザーを開き、更新したポートで Web コンソールにアクセスします。次に例を示します。

-

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

## 関連情報

- システム上の **firewall-cmd(1)**、**semanage(8)**、**systemd.unit(5)**、および **systemd.socket(5)**  
man ページ

## 第2章 RHEL システムロールを使用して WEB コンソールをインストールおよび設定する

**cockpit** RHEL システムロールを使用すると、複数の RHEL システムに Web コンソールを自動的にデプロイして有効にできます。

### 2.1. COCKPIT RHEL システムロールを使用した WEB コンソールのインストール

**cockpit** システムロールを使用すると、複数のシステムで RHEL Web コンソールのインストールと有効化を自動化できます。

この例では、**cockpit** システムロールを使用して次のことを行います。

- RHEL Web コンソールをインストールする
- カスタムポート番号 (9050/tcp) を使用するように Web コンソールを設定します。デフォルトでは、Web コンソールはポート 9090 を使用します。
- 新しいポートを開くためにシステムを設定できるように、**firewalld** および **selinux** システムロールを許可します。
- 自己署名証明書を使用する代わりに、**ipa** の信頼された認証局からの証明書を使用するように Web コンソールを設定する



#### 注記

ファイアウォールを管理したり証明書を作成したりするために、Playbook で **firewall** または **certificate** システムロールを呼び出す必要はありません。**cockpit** システムロールが、必要に応じてそれらを自動的に呼び出します。

#### 前提条件

- **コントロールノードと管理対象ノードの準備が完了している。**
- 管理対象ノードで Playbook を実行できるユーザーとしてコントロールノードにログインしている。
- 管理対象ノードへの接続に使用するアカウントに、そのノードに対する **sudo** 権限がある。

#### 手順

1. 次の内容を含む Playbook ファイル (例: **~/playbook.yml**) を作成します。

```
---
- name: Manage the RHEL web console
  hosts: managed-node-01.example.com
  tasks:
    - name: Install RHEL web console
      ansible.builtin.include_role:
        name: rhel-system-roles.cockpit
      vars:
        cockpit_packages: default
        cockpit_port: 9050
```

```
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
```

サンプル Playbook で指定されている設定は次のとおりです。

#### **cockpit\_manage\_selinux: true**

**selinux** システムロールを使用して、**websm\_port\_t** SELinux タイプで正しいポート権限を設定するように SELinux を設定できるようにします。

#### **cockpit\_manage\_firewall: true**

**cockpit** システムロールが **firewalld** システムロールを使用してポートを追加できるようにします。

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

デフォルトでは、RHEL Web コンソールは自己署名証明書を使用します。または、**cockpit\_certificates** 変数を Playbook に追加し、IdM 認証局 (CA) から証明書を要求するか、管理対象ノードで使用可能な既存の証明書と秘密鍵を使用するようにロールを設定することもできます。

Playbook で使用されるすべての変数の詳細は、コントロールノードの **/usr/share/ansible/roles/rhel-system-roles.cockpit/README.md** ファイルを参照してください。

2. Playbook の構文を検証します。

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

このコマンドは構文を検証するだけであり、有効だが不適切な設定から保護するものではないことに注意してください。

3. Playbook を実行します。

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

## 関連情報

- **/usr/share/ansible/roles/rhel-system-roles.cockpit/README.md** ファイル
- **/usr/share/doc/rhel-system-roles/cockpit** ディレクトリー
- [RHEL システムロールを使用した証明書の要求](#)

## 第3章 WEB コンソールアドオンのインストールとカスタムページの作成

Red Hat Enterprise Linux システムの使用方法に応じて、**使用可能** なアプリケーションを Web コンソールに追加したり、ユースケースに基づいてカスタムページを作成したりできます。

### 3.1. RHEL WEB コンソールのアドオン

**cockpit** パッケージはデフォルトで Red Hat Enterprise Linux の一部ですが、次のコマンドを使用してオンデマンドでアドオンアプリケーションをインストールできます。

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

上記コマンドの **<add-on>** は、RHEL Web コンソールで使用可能なアドオンアプリケーションのリストのパッケージ名に置き換えます。

機能名	パッケージ名	用途
Composer	<b>cockpit-composer</b>	カスタム OS イメージの構築
マシン	<b>cockpit-machines</b>	<b>libvirt</b> 仮想マシンの管理
PackageKit	<b>cockpit-packagekit</b>	ソフトウェア更新およびアプリケーションインストール (通常はデフォルトでインストールされている)
PCP	<b>cockpit-pcp</b>	永続的かつ、より詳細なパフォーマンスデータ (UI からオンデマンドでインストール)
Podman	<b>cockpit-podman</b>	<a href="#">コンテナの管理</a> と <a href="#">コンテナイメージの管理</a>
セッションの録画	<b>cockpit-session-recording</b>	ユーザーセッションの記録および管理
ストレージ	<b>cockpit-storaged</b>	<b>udisk</b> によるストレージの管理

### 3.2. WEB コンソールでの新しいページの作成

カスタマイズした関数を Red Hat Enterprise Linux Web コンソールに追加する場合は、必要な関数を実行するページの HTML および JavaScript ファイルを含むパッケージディレクトリを追加する必要があります。

カスタムページの追加の詳細は、[Cockpit Project Web サイトの Creating Plugins for the Cockpit User Interface](#) を参照してください。

#### 関連情報

- [Cockpit Project Developer Guide](#) の [Cockpit Packages](#) セクション

### 3.3. WEB コンソールでのマニフェスト設定のオーバーライド

システムの特定のユーザーおよび全ユーザーの Web コンソールのメニューを変更できます。**cockpit** プロジェクトでは、パッケージ名はディレクトリー名です。パッケージには、**manifest.json** ファイルと他のファイルが含まれています。デフォルト設定は、**manifest.json** ファイルに存在します。指定したユーザーの特定の場所に **<package-name>.override.json** ファイルを作成することで、デフォルトの **cockpit** メニュー設定をオーバーライドできます。

#### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

#### 手順

1. 任意のテキストエディターで **<systemd>.override.json** ファイルのマニフェスト設定をオーバーライドします。次に例を示します。

- a. すべてのユーザーの設定を編集するには、次のように入力します。

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

- b. 単一ユーザーの設定を編集するには、次のように入力します。

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

2. 次の詳細を含む必要なファイルを編集します。

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

- **null** 値を指定すると、**services** タブが非表示になります。
- **-1** 値を指定すると、**logs** タブが一番目に移動します。

3. **cockpit** サービスを再起動します。

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

#### 関連情報

- システム上の **cockpit(1)** man ページ
- [Manifest overrides](#)

## 第4章 WEB コンソールでソフトウェア更新の管理

RHEL 9 Web コンソールでソフトウェア更新を管理する方法と、それらを自動化する方法を説明します。

Web コンソールのソフトウェア更新モジュールは、**dnf** ユーティリティーに基づいています。**dnf** を使用したソフトウェアの更新の詳細は、[パッケージの更新](#) セクションを参照してください。

### 4.1. WEB コンソールでの手動ソフトウェア更新の管理

Web コンソールを使用してソフトウェアを手動で更新できます。

#### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

#### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Software Updates** をクリックします。  
利用可能な更新のリストは、24 時間が経過すると自動的に更新されます。再読み込みさせるには、**Check for Updates** ボタンをクリックします。
3. 更新を適用します。更新の実行中に更新ログを見ることができます。
  - a. 利用可能な更新をすべてインストールするには、**Install all updates** ボタンをクリックします。
  - b. 利用可能なセキュリティ更新がある場合は、**Install Security Updates** ボタンをクリックして個別にインストールできます。
  - c. **kpatch** 更新が利用可能な場合は、**Install kpatch updates** ボタンをクリックして個別にインストールできます。
4. オプション: システムを自動的に再起動するために、**Reboot after completion** スイッチをオンにすることができます。  
この手順を実行する場合は、この手順の残りの手順をスキップできます。
5. システムが更新を適用すると、システムを再起動するように勧められます。新しいカーネルまたはシステムサービスが更新に含まれていて、それらを個別に再起動する必要がない場合は、システムを再起動します。
6. **無視** をクリックして再起動をキャンセルするか、**今すぐ再起動** をクリックしてシステムの再起動を続行します。  
システムの再起動後、Web コンソールにログインし、**Software Updates** ページに移動して更新が成功したことを確認します。

### 4.2. WEB コンソールで自動ソフトウェア更新の管理

Web コンソールでは、すべての更新またはセキュリティ更新の適用を選択し、自動更新の周期とタイミングを管理することもできます。

## 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

## 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Software Updates** をクリックします。
3. **Settings** テーブルで、**Edit** ボタンをクリックします。
4. 自動更新の種類を一つ選びます。**Security updates only**、または **All updates** を選択できます。
5. 自動更新の日付を変更するには、ドロップダウンメニューの **毎日** をクリックして、特定の日付を選択します。
6. 自動更新の時刻を変更するには、**6:00** のフィールドをクリックして、特定の時刻を選択するか、入力します。
7. ソフトウェアの自動更新を無効にする場合は、**更新なし** を選択してください。

## 4.3. WEB コンソールでソフトウェア更新適用後のオンデマンド再起動の管理

インテリジェント再起動機能は、ソフトウェア更新後にシステム全体を再起動する必要があるのか、それとも特定のサービスだけを再起動すればよいのかをユーザーに通知する機能です。

## 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

## 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Software Updates** をクリックします。
3. システムの更新を適用します。
4. 更新が成功したら、**Reboot system...**、**Restart services...**、または **Ignore** をクリックします。
5. 無視することにした場合には、次のいずれかの方法で再起動またはリブートメニューに戻ることができます。
  - a. リブート:
    - i. **Software Updates** ページの **Status** フィールドにある **Reboot system** ボタンをクリックします。
    - ii. オプション: ログインしているユーザーにメッセージを書き込みます。



- iii. **Delay** ドロップダウンメニューから、**delay** を選択します。
- iv. **Reboot** をクリックします。
- b. サービスの再起動:
  - i. Software Updates ページの Status フィールドの **Restart services...** ボタンをクリックします。  
再起動が必要なすべてのサービスのリストが表示されます。
  - ii. **サービスの再起動** をクリックします。  
選択した内容に応じて、システムを再起動するか、サービスを再起動します。

## 4.4. WEB コンソールでのカーネルライブパッチを使用したパッチ適用

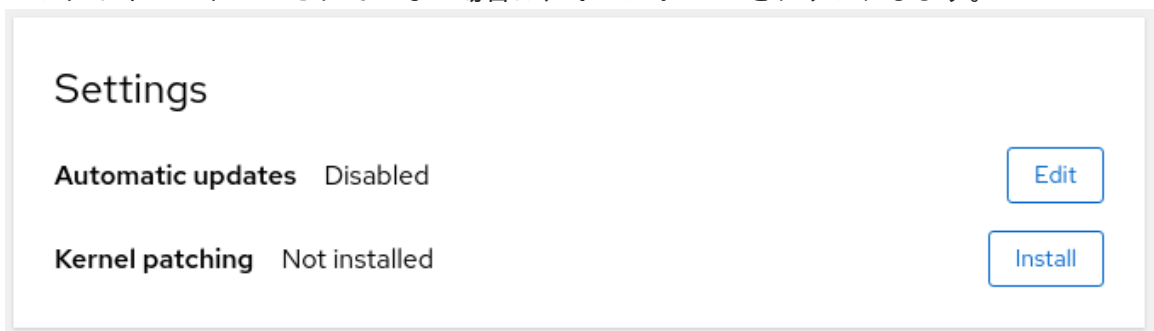
RHEL Web コンソールで、強制的に再起動せずにカーネルセキュリティパッチを適用する **kpatch** フレームワークを設定できます。

### 前提条件

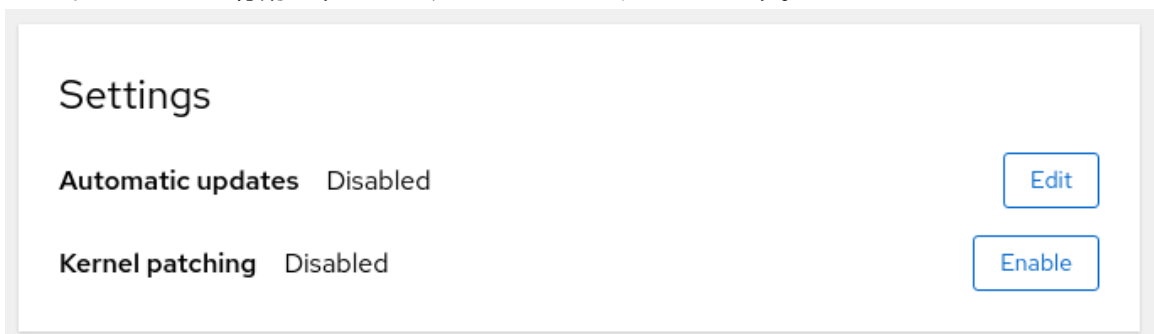
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Software Updates** をクリックします。
3. カーネルパッチの設定状況を確認します。
  - a. パッチがインストールされていない場合は、**インストール** をクリックします。

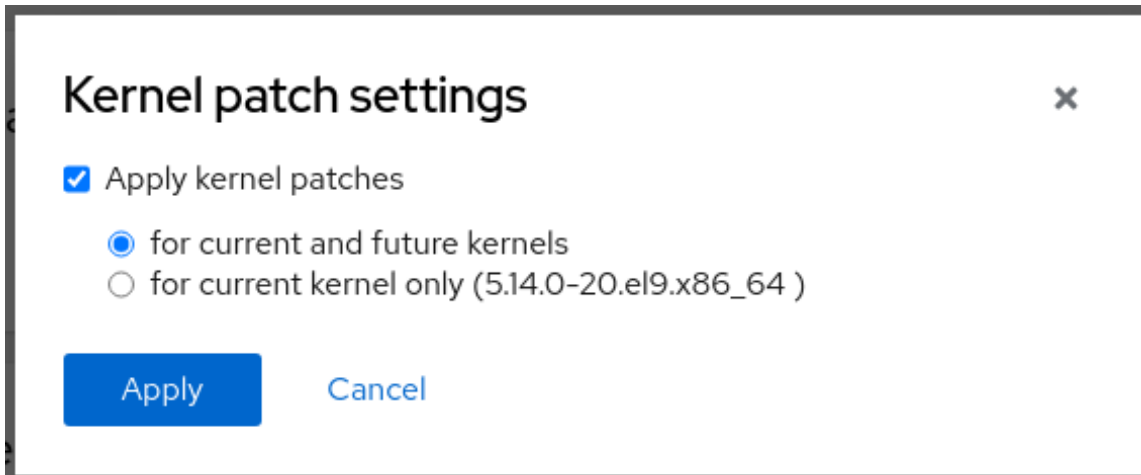


- b. カーネルパッチを有効にするには、**Enable** をクリックします。



- c. カーネルパッチを適用する場合はチェックを入れます。

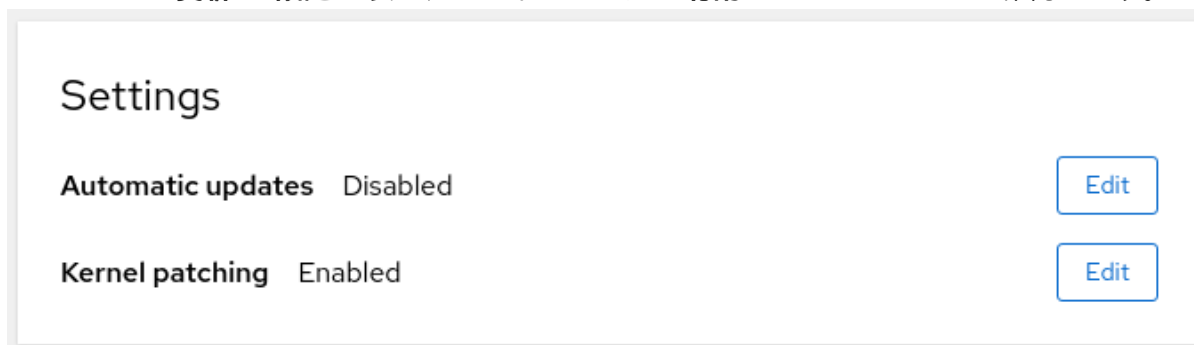
- d. 現在および今後のカーネルにパッチを適用するか、現在のカーネルにのみ適用するかを選択します。今後のカーネルにパッチを適用することを選択した場合、システムは今後リリースされるカーネルのパッチも適用します。

A dialog box titled "Kernel patch settings" with a close button (X) in the top right corner. It contains a checked checkbox labeled "Apply kernel patches". Below this, there are two radio button options: "for current and future kernels" (which is selected) and "for current kernel only (5.14.0-20.el9.x86\_64)". At the bottom, there are two buttons: "Apply" (in blue) and "Cancel" (in light blue).

- e. **Apply** をクリックします。

## 検証

- ソフトウェア更新 の 設定 の表で、カーネルパッチが **有効** になっていることを確認します。

A screenshot of the "Settings" page. It features a table with two rows. The first row is "Automatic updates" with the status "Disabled" and an "Edit" button. The second row is "Kernel patching" with the status "Enabled" and an "Edit" button.

## 関連情報

- [カーネルライブパッチでパッチの適用](#)

## 第5章 WEB コンソールでサブスクリプションの管理

Red Hat Enterprise Linux 9 Web コンソールで Red Hat 製品のサブスクリプションを管理できます。

### 前提条件

- [Red Hat カスタマーポータル](#) またはサブスクリプションのアクティベーションキー。

### 5.1. WEB コンソールでサブスクリプションの管理

RHEL 9 Web コンソールは、ローカルシステムにインストールされている Red Hat Subscription Manager を使用するインターフェイスを提供します。

Subscription Manager は、Red Hat カスタマーポータルに接続し、次のサブスクリプションを確認します。

- アクティブなサブスクリプション
- 期限が切れたサブスクリプション
- 更新されたサブスクリプション

Red Hat カスタマーポータルでサブスクリプションを更新したり、別のサブスクリプションを取得したりする場合、Subscription Manager のデータを手動で更新する必要はありません。

Subscription Manager は、データを Red Hat カスタマーポータルと自動的に同期します。

### 5.2. WEB コンソールで認証情報を使用してサブスクリプションを登録

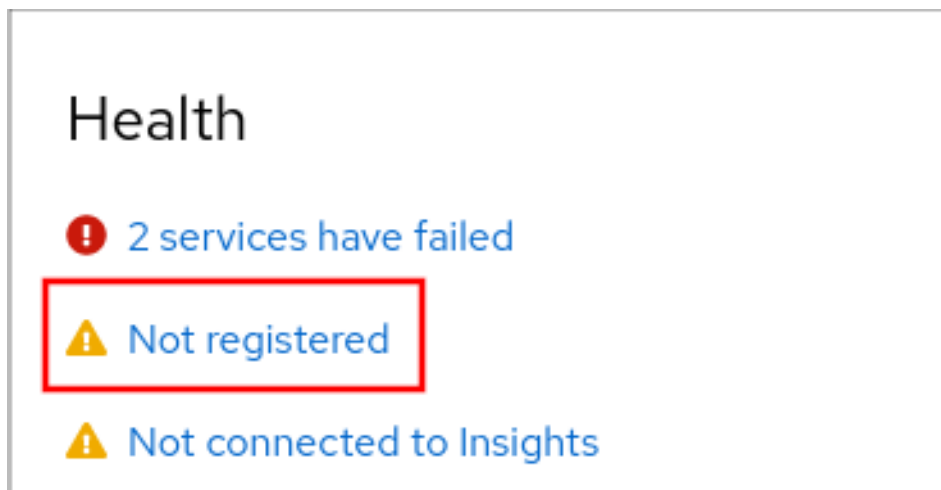
RHEL Web コンソールでアカウント認証情報を使用して、新しくインストールした Red Hat Enterprise Linux を登録できます。

### 前提条件

- Red Hat カスタマーポータルに有効なユーザーアカウントがある。  
[Red Hat アカウントの作成](#) ページを参照してください。
- RHEL システムに使用するアクティブなサブスクリプションがある。
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **概要** ページの **ヘルス** ファイル内の **未登録** の警告をクリックするか、メインメニューの **サブスクリプション** をクリックして、サブスクリプション情報のあるページに移動します。



3. Overview フィールドで、**Register** をクリックします。
4. Register system ダイアログボックスで、アカウント認証情報を使用して登録する **Account** を選択します。

A screenshot of the 'Register System' dialog box. It has a title bar 'Register System'. Below the title bar, there are several sections. The 'URL' section has a dropdown menu set to 'Default'. Below it is a checkbox 'Use proxy server' which is unchecked. The 'Method' section has two radio buttons: 'Account' (which is selected and highlighted by a red box) and 'Activation key'. Below the 'Method' section are three text input fields: 'Username', 'Password', and 'Organization'. The 'Subscriptions' section has a checked checkbox 'Attach automatically'. The 'Insights' section has a checked checkbox 'Connect this system to Red Hat Insights' with a link icon. At the bottom, there are two buttons: 'Register' (in blue) and 'Cancel'.

5. ユーザー名を入力します。
6. パスワードを入力します。
7. オプション: 組織名または ID を入力します。  
アカウントが Red Hat カスタマーポータルで複数の組織に所属している場合には、組織名または組織 ID を追加する必要があります。組織 ID を取得するには、Red Hat の連絡先にお問い合わせください。
  - システムを Red Hat Insights に接続しない場合は、**Insights** チェックボックスをオフにします。
8. **Register** をクリックします。

### 5.3. WEB コンソールでアクティベーションキーを使用してサブスクリプションを登録

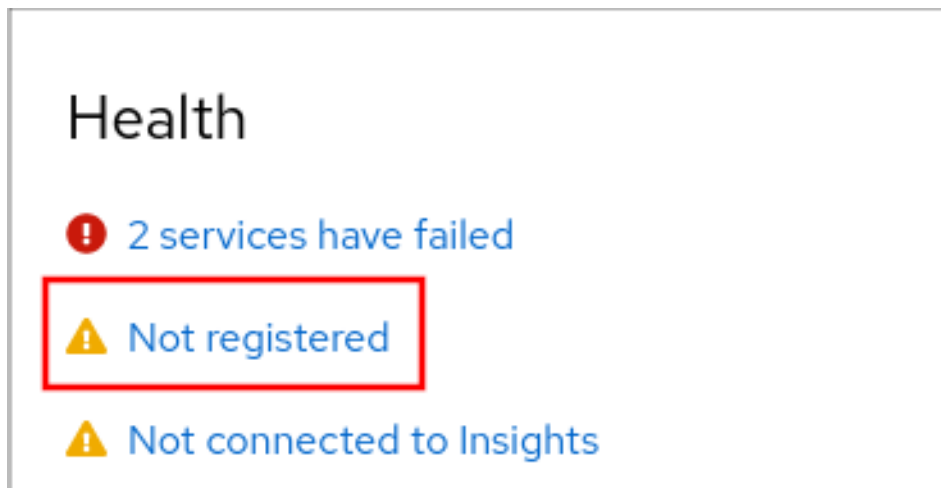
RHEL Web コンソールでアクティベーションキーを使用して、新しくインストールした Red Hat Enterprise Linux を登録できます。

#### 前提条件

- Red Hat 製品サブスクリプションのアクティベーションキー。
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

#### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Overview** ページの **Health** フィールドで、**Not registered** の警告をクリックするか、メインメニューの **Subscriptions** をクリックして、サブスクリプション情報を含むページに移動します。



をクリックします。

3. **Overview** フィールドの **Register** をクリックします。
4. **Register system** ダイアログボックスで、**Activation key** を選択して、アクティベーションキーを使用して登録します。

## 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. キーを入力します。

6. 組織名または ID を入力します。  
組織 ID の取得は、Red Hat にお問い合わせください。

- Red Hat Insights にシステムを接続しない場合は、**Insights** チェックボックスのチェックを外してください。

7. **Register** をクリックします。

## 第6章 WEB コンソールでリモートシステムの管理

リモートシステムに接続し、RHEL 9 Web コンソールで管理できます。

以下を説明します。

- 接続したシステムで最適なトポロジー
- リモートシステムを追加および削除する方法
- リモートシステム認証に SSH 鍵を使用する時、理由、および方法
- スマートカードで認証されたユーザーがリモートホストに **SSH** 接続してサービスにアクセスできるように Web コンソールクライアントを設定する方法。

### 前提条件

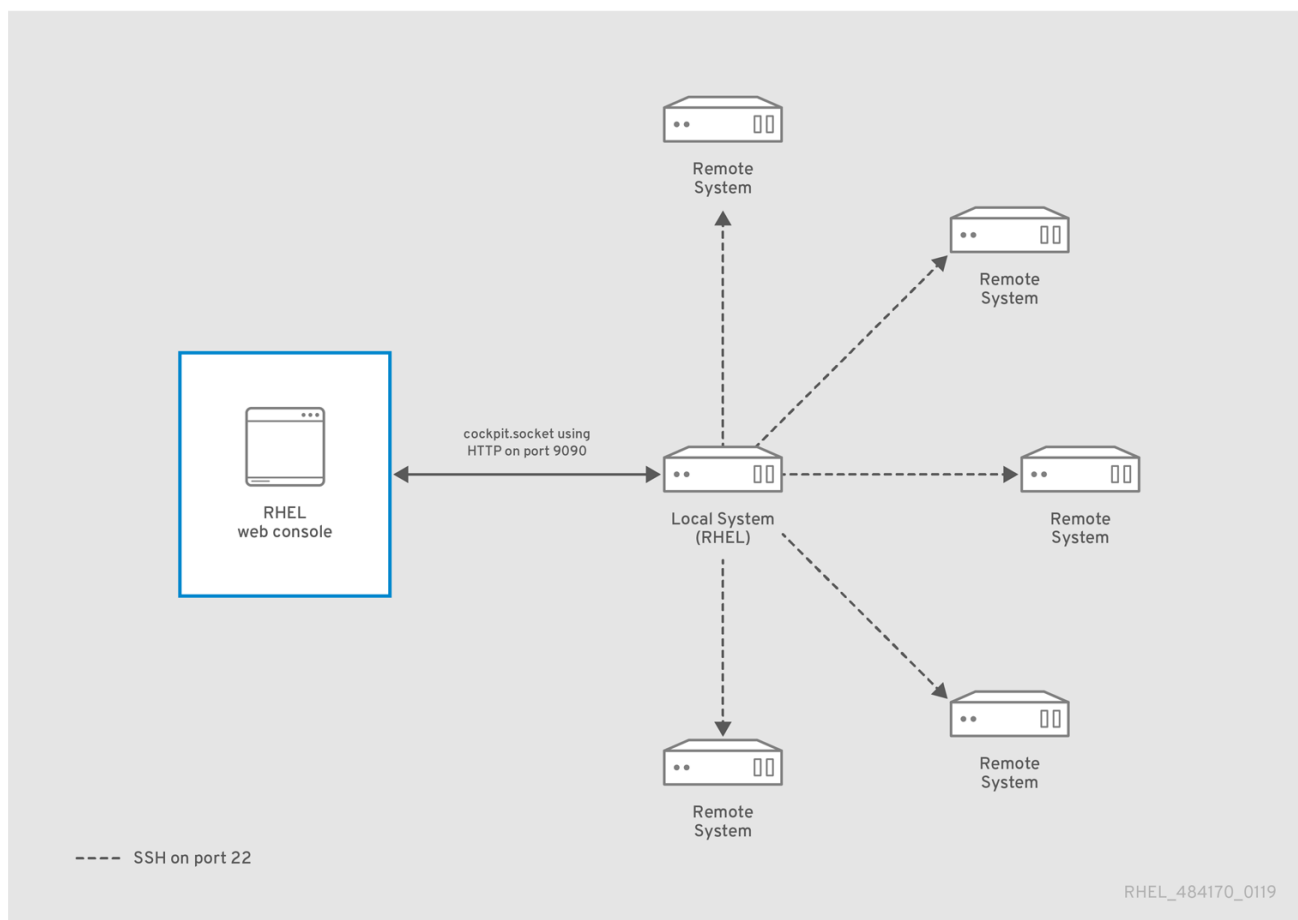
- SSH サービスがリモートシステムで実行されている。

### 6.1. WEB コンソールのリモートシステムマネージャー

セキュリティ上の理由から、RHEL 9 Web コンソールによって管理されるリモートシステムの次のネットワーク設定を使用します。

- Web コンソールを使用して、システム 1 台を要塞ホストとして設定します。要塞ホストは、開いている HTTPS ポートを使用するシステムです。
- その他のすべてのシステムは SSH を介して通信します。

要塞ホストで Web インターフェイスを使用して、デフォルト設定でポート 22 を使用して、SSH プロトコルを介して他のすべてのシステムに到達できます。



## 6.2. WEB コンソールへのリモートシステムの追加

RHEL Web コンソールで、対応する認証情報を使用してリモートシステムを追加した後、そのリモートシステムを管理できます。

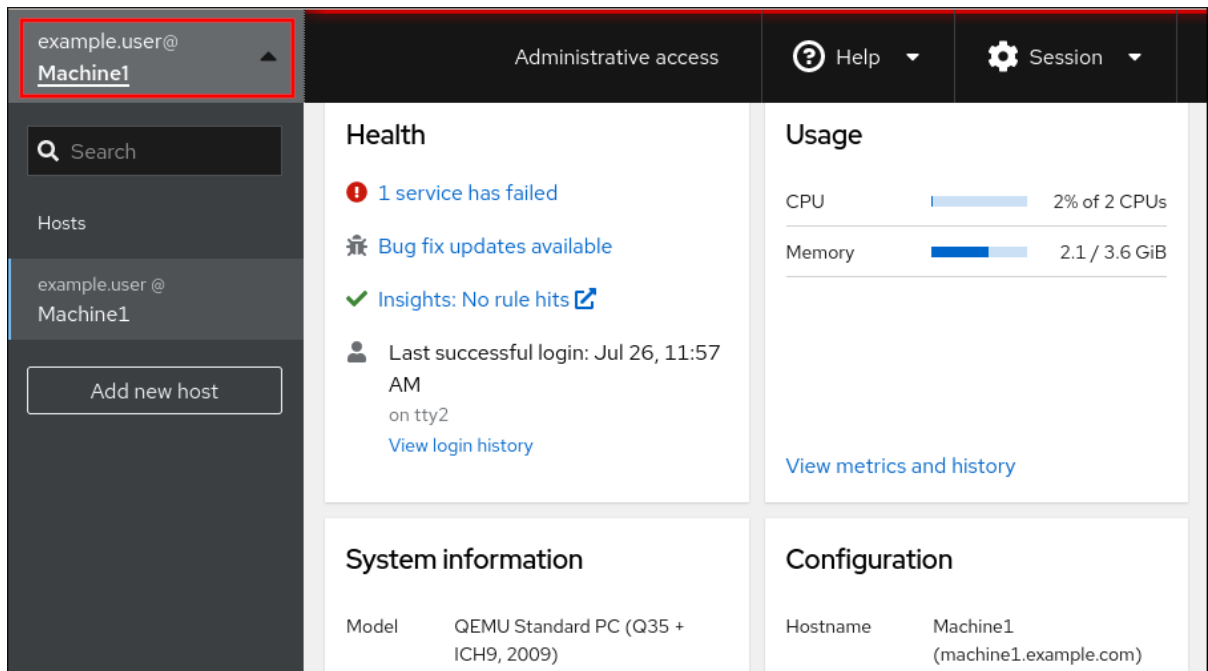
### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. RHEL 9 Web コンソールで、**Overview** ページの左上隅にある `<username>@<hostname>` をクリックします。





3. ドロップダウンメニューから、**Add new host** をクリックします。
4. **新規ホストの追加** ダイアログボックスで、追加するホストを指定します。
5. オプション: 接続するアカウントのユーザー名を追加します。  
リモートシステムのユーザーアカウントを使用できます。ただし、管理者権限のないユーザーアカウントの認証情報を使用すると、管理タスクを実行できません。

ローカルシステムと同じ認証情報を使用すると、ログインするたびに Web コンソールによってリモートシステムが自動的に認証されます。複数のシステムで同じ認証情報を使用すると、セキュリティが弱まることに注意してください。

6. オプション: システムの色を変更するには、**Color** フィールドをクリックします。
7. **Add** をクリックします。



### 重要

Web コンソールはリモートシステムへのログインに使用されるパスワードを保存しません。そのため、システムを再起動するたびに再度ログインする必要があります。次回ログインするときは、切断されたリモートシステムのメイン画面にある **Log in** をクリックして、ログインダイアログを開いてください。

### 検証

- 新しいホストが、`<username>@<hostname>` ドロップダウンメニューに表示されます。

## 6.3. 新しいホストの SSH ログインの有効化

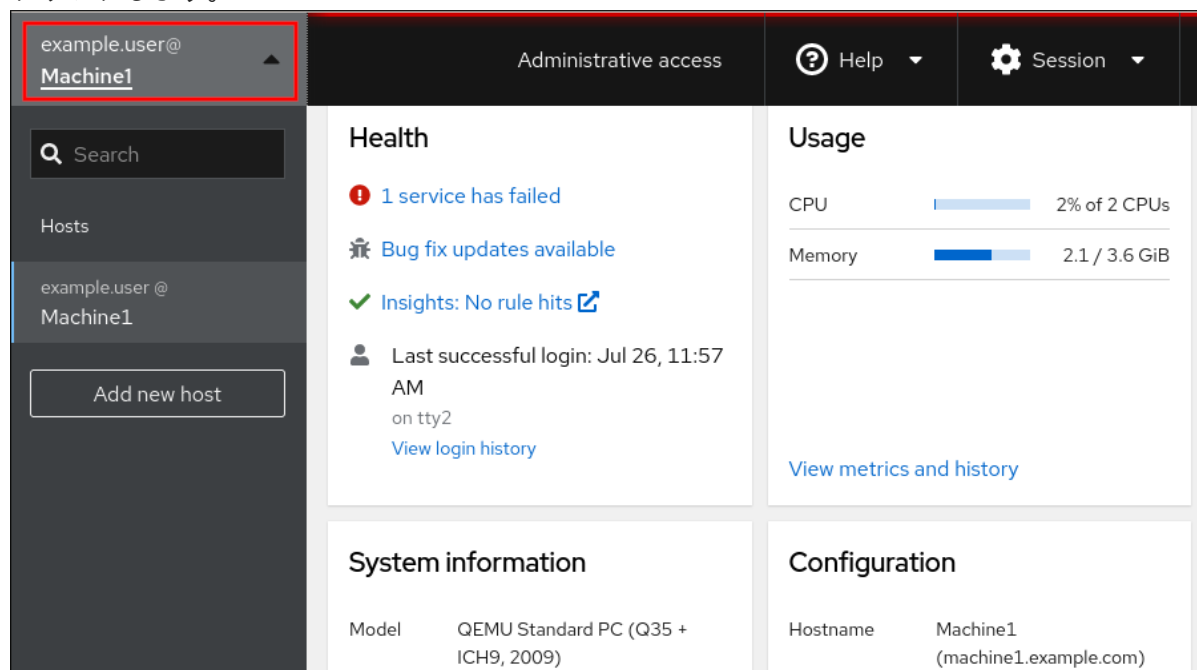
Web コンソールに新しいホストを追加すると、SSH 鍵を使用してホストにログインできるようになります。システム上にすでに SSH 鍵がある場合、Web コンソールは既存の鍵を使用します。そうでない場合、Web コンソールは鍵を作成できます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

## 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. RHEL 9 Web コンソールで、**Overview** ページの左上隅にある `<username>@<hostname>` をクリックします。



3. ドロップダウンメニューから、**Add new host** をクリックします。
4. **新規ホストの追加** ダイアログボックスで、追加するホストを指定します。
5. 接続するアカウントのユーザー名を追加します。  
リモートシステムのユーザーアカウントを使用できます。ただし、管理者権限のないユーザーアカウントを使用すると、管理タスクを実行できません。
6. オプション: システムの色を変更するには、**Color** フィールドをクリックします。
7. **Add** をクリックします。  
新しいダイアログウィンドウが表示され、パスワードの入力が求められます。
8. ユーザーアカウントのパスワードを入力します。
9. すでに SSH 鍵がある場合は、**Authorize SSH key** チェックボックスをオンにします。

## 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. SSH 鍵がない場合は、**Create new SSH key and authorize it**にチェックを入れてください。Web コンソールが鍵を作成します。

## 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

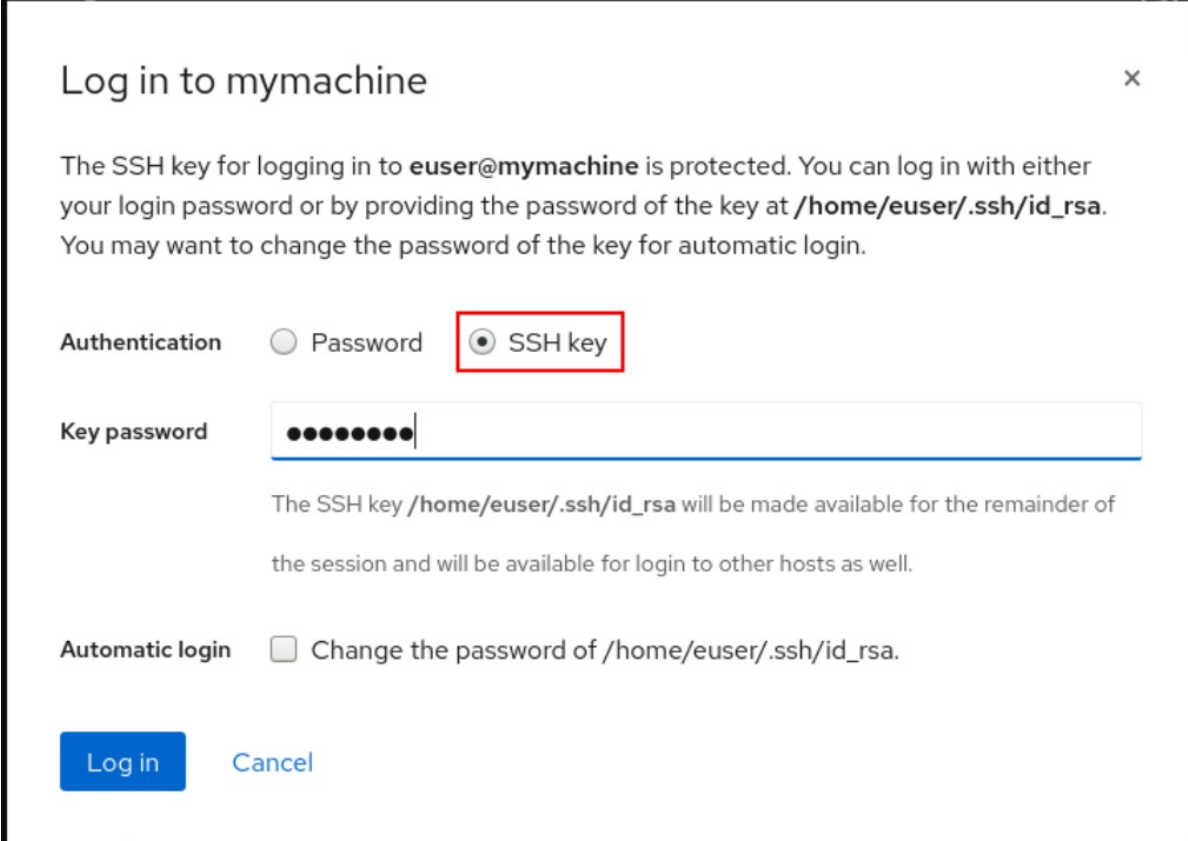
In order to allow log in to **mymachine** as **euser** without password in the future, use the login password of **euser** on **localhost** as the key password, or leave the key password blank.

- SSH 鍵のパスワードを追加します。
- パスワードを確認します。

11. **Log in** をクリックします。

## 検証

1. ログアウトします。
2. ログインし直してください。
3. **Not connected to host** 画面の **Log in** をクリックします。
4. 認証オプションとして、**SSH 鍵** を選択します。



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. 鍵のパスワードを入力します。
6. **Log in** をクリックします。

## 関連情報

- [2 台のシステム間で OpenSSH を使用した安全な通信の使用](#)

## 6.4. スマートカードで認証されたユーザーが、再度認証を要求されることなくリモートホストに SSH 接続できるようにするための WEB コンソールの設定

RHEL の Web コンソールでユーザーアカウントにログインした後、Identity Management (IdM) システム管理者として、**SSH** プロトコルを使用してリモートマシンに接続する必要がある場合があります。[制約付き委任](#) 機能を使用すると、再度認証を求められることなく **SSH** を使用することができます。

制約付き委任を使用するように Web コンソールを設定するには、次の手順に従います。以下の例では、Web コンソールセッションは `myhost.idm.example.com` ホストで実行され、認証されたユーザー

の代わりに **SSH** を使用して **remote.idm.example.com** ホストにアクセスするように設定されています。

## 前提条件

- IdM **admin** Ticket-Granting Ticket (TGT) を取得している
- **remote.idm.example.com** への **root** アクセス権がある
- Web コンソールサービスが IdM に存在する
- **remote.idm.example.com** ホストが IdM に存在する
- Web コンソールは、ユーザーセッションに **S4U2Proxy** Kerberos チケットを作成している。これを確認するために、IdM ユーザーで Web コンソールにログインし、**Terminal** ページを開き、以下を入力します。

```
$ 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
```

## 手順

1. 委任ルールでアクセス可能な対象ホストのリストを作成します。

- a. サービス委任ターゲットを作成します。

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

- b. 委任対象に対象ホストを追加します。

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

2. サービス委任ルールを作成し、**HTTP** サービスの Kerberos プリンシパルを追加することで、**cockpit** セッションが対象ホストのリストにアクセスできるようにします。

- a. サービス委任ルールを作成します。

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

- b. Web コンソールクライアントを委任ルールに追加します。

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

- c. 委任対象を委任ルールに追加します。

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

3. remote.idm.example.com ホストで Kerberos 認証を有効にします。
  - a. **root** として remote.idm.example.com に **SSH** 接続します。
  - b. /etc/ssh/sshd\_config ファイルを開いて編集します。
  - c. **GSSAPIAuthentication no** 行のコメントを外し、**GSSAPIAuthentication yes** に置き換えて、**GSSAPIAuthentication** を有効にします。
4. 上記の変更がすぐに有効になるように、remote.idm.example.com の **SSH** サービスを再起動します。

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

## 関連情報

- [スマートカードを使用して Web コンソールへのログイン](#)
- [アイデンティティ管理における制約付き委任](#)

## 6.5. ANSIBLE を使用して WEB コンソールを設定し、スマートカードで認証されたユーザーが再認証を求められることなくリモートホストに SSH 接続できるようにする

RHEL の Web コンソールでユーザーアカウントにログインした後、Identity Management (IdM) システム管理者として、**SSH** プロトコルを使用してリモートマシンに接続する必要がある場合があります。[制約付き委任](#) 機能を使用すると、再度認証を求められることなく **SSH** を使用することができま

**servicedelegationrule** および **servicedelegationtarget ansible-freeipa** モジュールを使用して、制約付き委任を使用するように Web コンソールを設定するには、この手順に従います。以下の例では、Web コンソールセッションは myhost.idm.example.com ホストで実行され、認証されたユーザーの代わりに **SSH** を使用して remote.idm.example.com ホストにアクセスするように設定されています。

## 前提条件

- IdM **admin** パスワードがある
- remote.idm.example.com への **root** アクセスがある
- Web コンソールサービスが IdM に存在する
- remote.idm.example.com ホストが IdM に存在する
- Web コンソールは、ユーザーセッションに **S4U2Proxy** Kerberos チケットを作成している。これを確認するために、IdM ユーザーで Web コンソールにログインし、**Terminal** ページを開き、以下を入力します。

```
$ 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
```

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して **Ansible インベントリーファイル** を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible Vault に **ipaadmin\_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

## 手順

1. `~/MyPlaybooks/` ディレクトリーに移動します。

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

2. 以下の内容で **web-console-smart-card-ssh.yml** Playbook を作成します。
  - a. 委任対象の存在を確認するタスクを作成します。

```
---
- 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. 対象ホストを委任ターゲットに追加するタスクを追加します。

```
- 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. 委任ルールが存在を確認するタスクを追加します。

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

- d. Web コンソールクライアントサービスの Kerberos プリンシパルが制約付き委任ルールのものであることを確認するタスクを追加します。

```
- name: Ensure the Kerberos principal of the web console client service is added to the
  servicedelegationrule web-console-delegation-rule
  ipaservicedelegationrule:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: web-console-delegation-rule
    principal: HTTP/myhost.idm.example.com
    action: member
```

- e. 制約付き委任ルールが web-console-delegation-target 委任対象と関連付けられることを確認するタスクを追加します。

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

3. ファイルを保存します。
4. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

5. **remote.idm.example.com** で Kerberos 認証を有効にします。
  - a. **root** として **remote.idm.example.com** に **SSH** 接続します。
  - b. **/etc/ssh/sshd\_config** ファイルを開いて編集します。
  - c. **GSSAPIAuthentication no** 行のコメントを外し、**GSSAPIAuthentication yes** に置き換えて、**GSSAPIAuthentication** を有効にします。

## 関連情報

- [スマートカードを使用して Web コンソールへのログイン](#)
- [アイデンティティ管理における制約付き委任](#)
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-servicedelegationrule.md** および **README-servicedelegationtarget.md**



- **/usr/share/doc/ansible-freeipa/playbooks/servicedelegationtarget** および  
**/usr/share/doc/ansible-freeipa/playbooks/servicedelegationrule** ディレクトリーのサンプル  
Playbook

## 第7章 IDM ドメインで RHEL 9 WEB コンソールにシングルサインオンを設定

RHEL 9 Web コンソールで Identity Management (IdM) によって提供されるシングルサインオン (SSO) 認証を使用すると、次の利点を活用できます。

- IdM ドメインの管理者は、RHEL 9 Web コンソールを使用して、ローカルマシンを管理できます。
- IdM ドメインに Kerberos チケットがあると、Web コンソールにアクセスする際にログイン認証情報を指定する必要がなくなりました。
- IdM ドメインが認識しているすべてのホストは、RHEL 9 Web コンソールのローカルインスタンスから SSH 経由でアクセスできます。
- 証明書設定は必須ではありません。コンソールの Web サーバーでは、IdM 認証局が発行した証明書に自動的に切り替わり、ブラウザに許可されます。

RHEL Web コンソールにログインするための SSO を設定するには、次のことが必要です。

1. RHEL 9 Web コンソールを使用して IdM ドメインにマシンを追加します。
2. 認証に Kerberos を使用する場合は、マシンで Kerberos チケットを取得する必要があります。
3. IdM サーバーの管理者が、任意のホストで任意のコマンドを実行できます。

### 前提条件

- RHEL Web コンソールが RHEL 9 システムにインストールされている。  
詳細は、[Web コンソールのインストール](#) を参照してください。
- RHEL Web コンソールを使用して IdM クライアントがシステムにインストールされている。  
詳細は [IdM クライアントのインストール](#) を参照してください。

## 7.1. WEB コンソールを使用した RHEL 9 システムの IDM ドメインへの参加

Web コンソールを使用して、Red Hat Enterprise Linux 9 システムを Identity Management (IdM) ドメインに参加させることができます。

### 前提条件

- IdM ドメインが実行中で参加するクライアントから到達可能
- IdM ドメインの管理者認証情報がある。
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Overview** タブの **Configuration** フィールドで、**Join Domain** をクリックします。

3. **ドメイン参加** ダイアログボックスの **ドメインアドレス** フィールドに、IdM サーバーのホスト名を入力します。
4. **ドメイン管理者名** フィールドで、IdM 管理アカウントのユーザー名を入力します。
5. **Domain administrator password** にパスワードを追加します。
6. **Join** をクリックします。

## 検証

1. システムが IdM ドメインに参加していると、RHEL 9 Web コンソールにエラーが表示されず、**システム** 画面でドメイン名を確認できます。
2. ユーザーがドメインのメンバーであることを確認するには、Terminal ページをクリックし、**id** コマンドを実行します。

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

## 関連情報

- [Identity Management の計画](#)
- [Identity Management のインストール](#)
- [IdM ユーザー、グループ、ホスト、およびアクセス制御ルールの管理](#)

## 7.2. KERBEROS 認証を使用した WEB コンソールへのログイン

Kerberos 認証を使用するように RHEL 9 システムを設定します。



### 重要

SSO を使用した場合は、通常、Web コンソールに管理者権限がありません。これは、パスワードがない sudo を設定した場合に限り機能します。Web コンソールは、対話的に sudo パスワードを要求しません。

## 前提条件

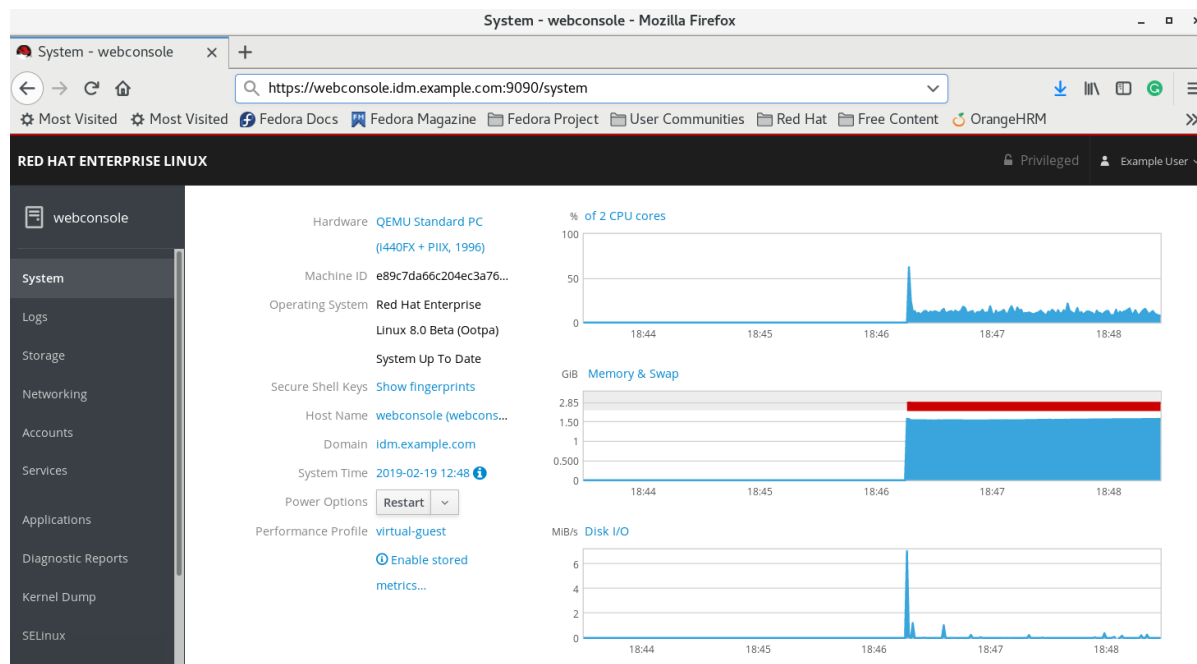
- 稼働中で、会社の環境で到達可能な IdM ドメイン  
詳細は[Web コンソールで IdM ドメインに RHEL 9 システムを参加させる](#) を参照してください。
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。
- SSSD クライアントによって管理される Kerberos チケットをシステムが使用していない場合は、**kinit** ユーティリティを使用して手動でチケットを要求する。

## 手順

- Web ブラウザーに次の URL を入力して、RHEL Web コンソールにログインします。

–

https://<dns\_name>:9090



この時点で、RHEL Web コンソールへの接続に成功しており、設定を開始できます。

## 第8章 集中管理ユーザー向けに WEB コンソールを使用したスマートカード認証の設定

RHEL Web コンソールで、以下の方法で集中管理されているユーザーに対してスマートカード認証を設定できます。

- Identity Management
- Identity Management を使用してフォレスト間の信頼に接続する Active Directory

### 前提条件

- スマートカード認証を使用するシステムは、Active Directory または Identity Management ドメインのメンバーである必要があります。  
Web コンソールを使用して RHEL 9 システムをドメインに参加させる方法の詳細は、[Web コンソールを使用して RHEL システムを IdM ドメインに参加させる](#) を参照してください。
- スマートカード認証に使用される証明書は、Identity Management または Active Directory の特定のユーザーに関連付けられている必要があります。  
Identity Management のユーザーと証明書の関連付けの詳細は、[Adding a certificate to a user entry in the IdM Web UI](#) または [Adding a certificate to a user entry in the IdM CLI](#) を参照してください。

### 8.1. 集中管理ユーザーのスマートカード認証

スマートカードは、カードに保存されている証明書を使用して個人認証を提供できる物理デバイスです。個人認証とは、ユーザーパスワードと同じ方法でスマートカードを使用できることを意味します。

秘密鍵と証明書の形式で、スマートカードにユーザーの認証情報を保存できます。特別なソフトウェアおよびハードウェアを使用して、そのソフトウェアにアクセスします。スマートカードをリーダーまたは USB ソケットに挿入して、パスワードを入力する代わりに、スマートカードの PIN コードを入力します。

Identity Management (IdM) では、以下によるスマートカード認証に対応しています。

- IdM 認証局が発行するユーザー証明書。
- Active Directory Certificate Service (ADCS) 認証局が発行するユーザー証明書。



#### 注記

スマートカード認証の使用を開始する場合は、ハードウェアの要件を参照してください。[RHEL 8 以降でのスマートカードのサポート](#)

### 8.2. スマートカードを管理および使用するツールのインストール

#### 前提条件

- **gnutls-utils** パッケージがインストールされている。
- **opensc** パッケージがインストールされている。
- **pcscd** サービスを実行している。

スマートカードを設定する前に、対応するツール (証明書を作成して **pcscd** サービスを起動できるもの) をインストールする必要があります。

## 手順

1. **opensc** パッケージおよび **gnutls-utils** パッケージをインストールします。

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

2. **pcscd** サービスを開始します。

```
# systemctl start pcscd
```

## 検証

- **pcscd** サービスが稼働していることを確認します。

```
# systemctl status pcscd
```

## 8.3. スマートカードを準備し、証明書と鍵をスマートカードにアップロードする

**pkcs15-init** ツールを使用してスマートカードを設定するには、この手順に従います。このツールは、以下を設定するのに役立ちます。

- スマートカードの消去
- 新しい PIN および任意の PIN ブロック解除キー (PUK) の設定
- スマートカードでの新規スロットの作成
- スロットへの証明書、秘密鍵、および公開鍵の保存
- 必要に応じて、特定のスマートカードではこのタイプのファイナライズが必要なため、スマートカードの設定をロックします。



### 注記

**pkcs15-init** ツールは、すべてのスマートカードで機能するとは限りません。使用しているスマートカードで動作するツールを使用する必要があります。

## 前提条件

- **pkcs15-init** ツールを含む **opensc** パッケージがインストールされている。  
詳細は [スマートカードを管理および使用するツールのインストール](#) を参照してください。
- カードがリーダーに挿入され、コンピューターに接続されている。
- スマートカードに保存する秘密鍵、公開鍵、および証明書がある。この手順の **testuser.key**、**testuserpublic.key**、および **testuser.crt** は、秘密鍵、公開鍵、および証明書に使用される名前です。
- 現在のスマートカードユーザー PIN およびセキュリティオフィス PIN (SO-PIN)

## 手順

1. スマートカードを消去して PIN で自身を認証します。

```
$ 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]:
```

カードが削除されました。

2. スマートカードを初期化し、ユーザーの PIN と PUK を設定します。また、セキュリティー担当者の PIN と PUK を設定します。

```
$ 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
```

**pkcs15-init** ツールは、スマートカードに新しいスロットを作成します。

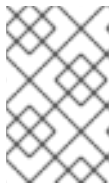
3. スロットのラベルと認証 ID を設定します。

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

ラベルは人間が判読できる値に設定されます (この場合は **testuser**)。 **auth-id** は 16 進数の値である必要があります。この場合、**01** に設定されます。

4. スマートカードの新しいスロットに秘密鍵を保存し、ラベルを付けます。

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



## 注記

--id に指定する値は、秘密鍵を保存するときと、次の手順で証明書を保存するときと同じである必要があります。--id に独自の値を指定することを推奨します。そうしないと、より複雑な値がツールによって計算されます。

5. スマートカードの新しいスロットに証明書を保存し、ラベル付けします。

```
$ 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. オプション: スマートカードの新しいスロットに公開鍵を保存し、ラベルを付けます。

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



### 注記

公開鍵が秘密鍵または証明書に対応する場合は、秘密鍵または証明書の ID と同じ ID を指定します。

7. オプション: スマートカードの中には、設定をロックしてカードをファイナライズする必要があるものもあります。

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

この段階では、スマートカードには、新たに作成されたスロットに証明書、秘密鍵、および公開鍵が含まれます。ユーザーの PIN と PUK、およびセキュリティー担当者の PIN と PUK も作成しました。

## 8.4. WEB コンソールのスマートカード認証の有効化

Web コンソールでスマートカード認証を使用するには、**cockpit.conf** ファイルでスマートカード認証方式を有効にします。

また、同じファイルでパスワード認証を無効にすることもできます。

### 前提条件

- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

### 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。
2. **Terminal** をクリックします。
3. **/etc/cockpit/cockpit.conf** で **ClientCertAuthentication** を **yes** に設定します。

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

4. オプション: 次のようにして、**cockpit.conf** でパスワードベースの認証を無効にします。

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

この設定ではパスワード認証が無効になり、常にスマートカードを使用する必要があります。

5. Web コンソールを再起動して、**cockpit.service** が変更を受け入れることを確認します。

```
# systemctl restart cockpit
```

## 8.5. スマートカードを使用して WEB コンソールへのログイン

スマートカードを使用して、Web コンソールにログインできます。



## 前提条件

- 有効な証明書が、Active Directory または Identity Management ドメインで作成されたユーザーアカウントに関連付けられているスマートカードに保存されている。
- スマートカードのロックを解除するピン。
- スマートカードがリーダーに追加されている。
- RHEL 9 Web コンソールがインストールされている。  
手順は、[Web コンソールのインストールおよび有効化](#) を参照してください。

## 手順

1. RHEL 9 Web コンソールにログインします。  
詳細は、[Web コンソールへのログイン](#) を参照してください。  
  
ブラウザーは、スマートカードに保存されている証明書を PIN で保護するよう要求します。
2. **Password Required** ダイアログボックスで PIN を入力し、**OK** をクリックします。
3. **User Identification Request** ダイアログボックスで、スマートカードに保存されている証明書を  
選択します。
4. **Remember this decision** を選択します。  
次回、このウィンドウが開きません。



### 注記

この手順は、Google Chrome ユーザーには適用されません。

5. **OK** をクリックします。

これで接続され、Web コンソールがそのコンテンツを表示します。

## 8.6. スマートカードユーザーに対するパスワードなしの SUDO 認証の有効化

Web コンソールで、スマートカードユーザー向けに **sudo** およびその他のサービスへのパスワードなし認証を設定できます。

別の方法として、Red Hat Identity Management を使用している場合は、最初の Web コンソール証明書認証を **sudo**、SSH、またはその他のサービスに対する認証で信頼できるものとして宣言することができます。そのために、Web コンソールはユーザーセッションに S4U2Proxy Kerberos チケットを自動的に作成します。

## 前提条件

- Identity Management がインストールされている。
- Identity Management を使用してフォレスト間の信頼に接続された Active Directory。
- Web コンソールへのログイン用に設定されたスマートカード。詳細は、[Configuring smart card authentication with the web console for centrally managed users](#) を参照してください。

## 手順

1. チケットがアクセスできるホストをリストアップする制約委譲ルールを設定します。

### 例8.1 制約委譲ルールの設定

Web コンソールセッションはホスト **host.example.com** で実行されており、**sudo** を使用して自分のホストにアクセスできるように信頼されている必要があります。さらに、2 つ目の信頼できるホストとして **remote.example.com** を追加します。

- 以下の委譲を作成します。
  - 以下のコマンドを実行して、特定のルールがアクセスできるターゲットマシンのリストを追加します。

```
# 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
```

- Web コンソールセッション (HTTP/プリンシパル) がそのホストリストにアクセスできるようにするには、次のコマンドを使用します。

```
# 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. 対応するサービスで GSS 認証を有効にします。

- a. **sudo** の場合は、**/etc/sss/sss.conf** ファイルで **pam\_sss\_gss** モジュールを有効にします。

- i. **root** で、**/etc/sss/sss.conf** 設定ファイルにドメイン用のエントリーを追加します。

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

- ii. **/etc/pam.d/sudo** ファイルの 1 行目でモジュールを有効にします。

```
auth sufficient pam_sss_gss.so
```

- b. SSH の場合、**/etc/ssh/sshd\_config** ファイルの **GSSAPIAuthentication** オプションを **yes** に更新します。



### 警告

委譲された S4U チケットが、Web コンソールからリモートの SSH ホストに接続するときに転送されません。チケットを使用してリモートホストの `sudo` を認証してうまくいきません。

### 検証

1. スマートカードを使用して Web コンソールにログインします。
2. **Limited access** ボタンをクリックします。
3. スマートカードを使用して認証を行います。

または、次のようになります。

- 別のホストに SSH で接続を試みます。

## 8.7. DOS 攻撃を防ぐためのユーザーセッションおよびメモリーの制限

証明書認証は、別のユーザーの権限を借用しようとする攻撃者に対して Web サーバー **cockpit-ws** のインスタンスを分離して孤立させることで保護されます。ただし、これによりサービス拒否 (DoS) 攻撃が発生する可能性があります。リモートの攻撃者は、多数の証明書を作成し、別の証明書を使用して、多数の HTTPS 要求を **cockpit-ws** に送信できます。

このような DoS 攻撃を防ぐには、これらの Web サーバーインスタンスの共有リソースを制限します。デフォルトでは、接続数とメモリー使用量の制限として、200 スレッドと 75 % (ソフト) または 90 % (ハード) のメモリー制限が設定されています。

この例の手順では、接続数とメモリーを制限することでリソースを保護する方法を示します。

### 手順

1. 端末で **system-cockpithttps.slice** 設定ファイルを開きます。

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

2. **TasksMax** を 100 に、**CPUQuota** を 30% に制限します。

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

3. 変更を適用するには、システムを再起動します。

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

新しいメモリーとユーザーセッションにより、**cockpit-ws** Web サーバーに対する DoS 攻撃のリスクが低減されます。

## 8.8. 関連情報

- [Configuring Identity Management for smart card authentication](#) .
- [Configuring certificates issued by ADCS for smart card authentication in IdM](#) .
- ローカル証明書のスマートカードへの設定およびインポート

## 第9章 WEB コンソールでの SATELLITE ホストの管理と監視

Red Hat Satellite Server で RHEL Web コンソール統合を有効にすると、Web コンソールで多数のホストを大規模に管理できます。

Red Hat Satellite は、物理環境、仮想環境、およびクラウド環境全体のシステムをデプロイ、設定、保守するためのシステム管理ソリューションです。Satellite では、一元化されたツールを使用して複数の Red Hat Enterprise Linux デプロイメントのプロビジョニング、リモート管理、監視が可能です。

デフォルトでは、Red Hat Satellite では RHEL Web コンソールの統合が無効になっています。Red Hat Satellite 内からホストの RHEL Web コンソール機能にアクセスするには、まず Red Hat Satellite Server で RHEL Web コンソールの統合を有効にする必要があります。

Satellite Server で RHEL Web コンソールを有効にするには、**root** として次のコマンドを入力します。

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

### 関連情報

- Red Hat Satellite の「ホストの管理」ガイドの [RHEL Web コンソールを使用したホストの管理と監視](#)