



# OpenShift Container Platform 4.3

## Web コンソール

OpenShift Container Platform Web コンソールのスタートガイド



# OpenShift Container Platform 4.3 Web コンソール

---

OpenShift Container Platform Web コンソールのスタートガイド

## 法律上の通知

Copyright © 2020 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

本書では、OpenShift Container Platform Web コンソールにアクセスし、カスタマイズする方法を説明します。

---

## 目次

<b>第1章 WEB コンソールへのアクセス</b> .....	<b>3</b>
1.1. 前提条件	3
1.2. WEB コンソールの理解および WEB コンソールへのアクセス	3
<b>第2章 OPENSIFT CONTAINER PLATFORM ダッシュボードを使用したクラスター情報の取得</b> .....	<b>4</b>
2.1. OPENSIFT CONTAINER PLATFORM ダッシュボードページについて	4
<b>第3章 OPENSIFT CONTAINER PLATFORM の WEB コンソールの設定</b> .....	<b>6</b>
3.1. 前提条件	6
3.2. WEB コンソールの設定	6
<b>第4章 OPENSIFT CONTAINER PLATFORM の WEB コンソールのカスタマイズ</b> .....	<b>7</b>
4.1. カスタムロゴおよび製品名の追加	7
4.2. WEB コンソールでのカスタムリンクの作成	8
4.3. ログインページのカスタマイズ	9
4.4. 外部ログリンクのテンプレートの定義	10
4.5. カスタム通知バナーの作成	10
4.6. CLI ダウンロードのカスタマイズ	11
4.7. YAML サンプルの KUBERNETES リソースへの追加	11
<b>第5章 WEB コンソールの DEVELOPER パースペクティブ</b> .....	<b>13</b>
5.1. 前提条件	13
5.2. DEVELOPER パースペクティブへのアクセス	13
<b>第6章 OPENSIFT CONTAINER PLATFORM の WEB コンソールの無効化</b> .....	<b>15</b>
6.1. 前提条件	15
6.2. WEB コンソールの無効化	15



## 第1章 WEB コンソールへのアクセス

OpenShift Container Platform Web コンソールは、Web ブラウザーからアクセスできるユーザーインターフェースです。開発者は Web コンソールを使用してプロジェクトのコンテンツを視覚的に把握し、参照し、管理することができます。

### 1.1. 前提条件

- Web コンソールを使用するために JavaScript が有効にされている必要があります。WebSocket をサポートする Web ブラウザーを使用することが最も推奨されます。
- 「[OpenShift Container Platform 4.x のテスト済みインテグレーション](#)」のページを確認してから、クラスターのサポートされるインフラストラクチャーを作成します。

### 1.2. WEB コンソールの理解および WEB コンソールへのアクセス

Web コンソールはマスター上で Pod として実行されます。Web コンソールを実行するために必要な静的アセットは Pod によって提供されます。OpenShift Container Platform が正常にインストールされた後に、Web コンソールの URL およびインストールされたクラスターのログイン認証情報を、インストールプログラムの CLI 出力で確認します。以下は例になります。

```
INFO Install complete!
INFO Run 'export KUBECONFIG=<your working directory>/auth/kubeconfig' to manage the cluster
with 'oc', the OpenShift CLI.
INFO The cluster is ready when 'oc login -u kubeadmin -p <provided>' succeeds (wait a few minutes).
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.demo1.openshift4-beta-abcorp.com
INFO Login to the console with user: kubeadmin, password: <provided>
```

これらの詳細を使用してログインし、Web コンソールにアクセスします。

## 第2章 OPENSIFT CONTAINER PLATFORM ダッシュボードを使用したクラスター情報の取得

OpenShift Container Platform Web コンソールから **Home** → **Dashboards** → **Overview** に移動し、クラスターの概要情報をキャプチャーする OpenShift Container Platform ダッシュボードにアクセスします。

OpenShift Container Platform ダッシュボードは、個別のダッシュボードカードでキャプチャーされるさまざまなクラスター情報を提供します。

### 2.1. OPENSIFT CONTAINER PLATFORM ダッシュボードページについて

OpenShift Container Platform ダッシュボードは以下のカードで構成されます。

- **Details** は、クラスターの詳細情報の概要を表示します。  
ステータスには、**ok**、**error**、**warning**、**in progress**、および **unknown** が含まれます。リソースでは、カスタムのステータス名を追加できます。
  - クラスター
  - プロバイダー
  - バージョン
- **Cluster Inventory** は、リソースの数および関連付けられたステータスの詳細を表示します。これは、問題の解決に介入が必要な場合に役立ちます。以下についての情報が含まれます。
  - ノード数
  - Pod 数
  - 永続ストレージボリューム要求 (Persistent Storage Volume Claim)
  - クラスター内のベアメタルホスト。これらはステータス別に一覧表示されます (**metal3** 環境でのみ利用可能です)。
- **Cluster Capacity** チャートは、管理者が追加リソースがクラスターで必要になるタイミングを把握するのに役立ちます。このチャートには、現在の消費量を表示する内側の円が含まれ、外側の円には、以下の情報を含む、リソースに対して設定されたしきい値が表示されます。
  - CPU 時間
  - メモリー割り当て
  - 消費されるストレージ
  - 消費されるネットワークリソース
- **Cluster Utilization** は指定された期間における各種リソースの容量を表示します。これは、管理者がリソースの高い消費量の規模および頻度を理解するのに役立ちます。
- **Events** は、Pod の作成または別のホストへの仮想マシンの移行などのクラスター内の最近のアクティビティに関連したメッセージを一覧表示します。



- **Top Consumers** は、管理者がクラスターリソースが消費される状況を把握するのに役立ちます。リソースをクリックし、指定されたクラスターリソース (CPU、メモリー、またはストレージ) の最大量を消費する Pod およびノードを一覧表示する詳細ページに切り替えます。

## 第3章 OPENSIFT CONTAINER PLATFORM の WEB コンソールの設定

OpenShift Container Platform の Web コンソールを変更してログアウトリダイレクト URL を設定したり、コンソールを無効にしたりすることができます。

### 3.1. 前提条件

- OpenShift Container Platform クラスターをデプロイします。

### 3.2. WEB コンソールの設定

`console.config.openshift.io` リソースを編集して Web コンソールを設定できます。

- `console.config.openshift.io` リソースを編集します。

```
$ oc edit console.config.openshift.io cluster
```

以下の例は、コンソールのリソース定義のサンプルを示しています。

```
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  authentication:
    logoutRedirect: "" ❶
status:
  consoleURL: "" ❷
```

- ❶ ユーザーが Web コンソールからログアウトする際に読み込むページの URL を指定します。値を指定しない場合、ユーザーは Web コンソールのログインページに戻ります。**logoutRedirect** URL を指定することにより、ユーザーはアイデンティティプロバイダー経由でシングルログアウト (SLO) を実行し、シングルサインオンセッションを破棄することができます。
- ❷ Web コンソール URL。このパラメーターの値を変更することはできません。変更しても、クラスターはこれをデフォルト値にリセットします。

## 第4章 OPENSIFT CONTAINER PLATFORM の WEB コンソールのカスタマイズ

OpenShift Container Platform の Web コンソールをカスタマイズして、カスタムロゴ、製品名、リンク、通知、およびコマンドラインのダウンロードを設定できます。これは、Web コンソールを企業や政府の特定要件を満たすように調整する必要がある場合にとくに役立ちます。

### 4.1. カスタムロゴおよび製品名の追加

カスタムロゴまたはカスタム製品名を追加することで、カスタムブランディングを作成できます。これらの設定は相互に独立しているため、両方またはいずれかを設定できます。

#### 前提条件

- 管理者の権限があること。
- 使用するロゴのファイルを作成します。ロゴは、GIF、JPG、PNG、または SVG を含む共通のイメージ形式のファイルであり、**60px** の **max-height** に制限されます。

#### 手順

1. ロゴファイルを **openshift-config** namespace の ConfigMap にインポートします。

```
$ oc create configmap console-custom-logo --from-file /path/to/console-custom-logo.png -n openshift-config
```

2. Web コンソールの Operator 設定を編集して、**customLogoFile** および **customProductName** を組み込みます。

```
$ oc edit console.operator.openshift.io cluster
```

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  customization:
    customLogoFile:
      key: console-custom-logo.png
      name: console-custom-logo
    customProductName: My Console
```

Operator 設定が更新されると、カスタムロゴ ConfigMap をコンソール namespace に同期し、これをコンソール Pod にマウントし、再デプロイします。

3. 正常に実行されたかどうかを確認します。問題がある場合は、コンソールクラスター Operator は **Degraded** を報告し、コンソール Operator 設定も **CustomLogoDegraded** を **KeyOrFilenameInvalid** または **NoImageProvided** などの理由と共に報告します。**clusteroperator** を確認するには、以下を実行します。

```
$ oc get clusteroperator console -o yaml
```

コンソール Operator 設定を確認するには、以下を実行します。

```
$ oc get console.operator.openshift.io -o yaml
```

## 4.2. WEB コンソールでのカスタムリンクの作成

### 前提条件

- 管理者の権限があること。

### 手順

1. **Administration** → **Custom Resource Definitions**から、 **ConsoleLink** をクリックします。
2. **YAML** をクリックし、ファイルを編集します。

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: example
spec:
  href: 'https://www.example.com'
  location: HelpMenu ❶
  text: Link 1
```

- ❶ 有効な場所の設定は、**HelpMenu**、**UserMenu**、**ApplicationMenu**、および **NamespaceDashboard** です。

カスタムリンクがすべての namespace に表示されるようにするには、以下の例に従います。

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: namespaced-dashboard-link-for-all-namespaces
spec:
  href: 'https://www.example.com'
  location: NamespaceDashboard
  text: This appears in all namespaces
```

カスタムリンクが一部の namespace のみに表示されるようにするには、以下の例に従います。

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: namespaced-dashboard-for-some-namespaces
spec:
  href: 'https://www.example.com'
  location: NamespaceDashboard
  # This text will appear in a box called "Launcher" under "namespace" or "project" in the web console
  text: Custom Link Text
  namespaceDashboard:
    namespaces:
      # for these specific namespaces
```

- my-namespace
- your-namespace
- other-namespace

カスタムリンクがアプリケーションメニューに表示されるようにするには、以下の例に従います。

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: application-menu-link-1
spec:
  href: 'https://www.example.com'
  location: ApplicationMenu
  text: Link 1
  applicationMenu:
    section: My New Section
    # image that is 24x24 in size
    imageURL: https://via.placeholder.com/24
```

3. **Save** ボタンをクリックして変更を適用します。

### 4.3. ログインページのカスタマイズ

サービス利用規約情報をカスタムログインページを使用して作成します。カスタムログインページは、GitHub や Google などのサードパーティログインプロバイダーを使用している場合にも、ユーザーが信頼し、予想できるブランドのページを提示して、その後にユーザーを認証プロバイダーにリダイレクトする際に役立ちます。また、認証プロセス中にカスタムエラーページをレンダリングすることもできます。

#### 前提条件

- 管理者の権限があること。

#### 手順

1. 以下のコマンドを実行して、変更可能なテンプレートを作成します。

```
$ oc adm create-login-template > login.html
$ oc adm create-provider-selection-template > providers.html
$ oc adm create-error-template > errors.html
```

2. シークレットを作成します。

```
$ oc create secret generic login-template --from-file=login.html -n openshift-config
$ oc create secret generic providers-template --from-file=providers.html -n openshift-config
$ oc create secret generic error-template --from-file=errors.html -n openshift-config
```

3. 以下を実行します。

```
$ oc edit oauths cluster
```

4. 仕様を更新します。

```
spec:
  templates:
    error:
      name: error-template
    login:
      name: login-template
  providerSelection:
    name: providers-template
```

**oc explain oauths.spec.templates** を実行して、オプションを把握します。

## 4.4. 外部ログリンクのテンプレートの定義

ログの参照に役立つサービスに接続しているものの、特定の 방법으로 URL を生成する必要がある場合は、リンクのテンプレートを定義できます。

### 前提条件

- 管理者の権限があること。

### 手順

1. **Administration** → **Custom Resource Definitions** から、 **ConsoleExternalLogLink** をクリックします。
2. **YAML** をクリックし、ファイルを編集します。

```
apiVersion: console.openshift.io/v1
kind: ConsoleExternalLogLink
metadata:
  name: example
spec:
  hrefTemplate: >-
    https://example.com/logs?
    resourceName=${resourceName}&containerName=${containerName}&resourceNamespace=${
    resourceNamespace}&podLabels=${podLabels}
  text: Example Logs
```

## 4.5. カスタム通知バナーの作成

### 前提条件

- 管理者の権限があること。

### 手順

1. **Administration** → **Custom Resource Definitions** から、 **ConsoleNotification** をクリックします。
2. **YAML** をクリックし、ファイルを編集します。

```
apiVersion: console.openshift.io/v1
kind: ConsoleNotification
```

```

metadata:
  name: example
spec:
  backgroundColor: '#0088ce'
  color: '#fff'
  link:
    href: 'https://www.example.com'
    text: Optional link text
  location: BannerTop 1
  text: This is an example notification message with an optional link.

```

- 1** 有効な場所の設定は、**BannerTop**、**BannerBottom**、および **BannerTopBottom** です。

3. **Save** ボタンをクリックして変更を適用します。

## 4.6. CLI ダウンロードのカスタマイズ

ファイルパッケージを直接ポイントしたり、パッケージを提供する外部ページをポイントできるカスタムのリンクテキストおよび URL を使用して、CLI をダウンロードするリンクを設定できます。

### 前提条件

- 管理者の権限があること。

### 手順

1. **Administration** → **Custom Resource Definitions** に移動します。
2. カスタムリソース定義 (CRD) の一覧から **ConsoleCLIDownload** を選択します。
3. **YAML** タブをクリックし、編集を行います。

```

apiVersion: console.openshift.io/v1
kind: ConsoleCLIDownload
metadata:
  name: example-cli-download-links-for-foo
spec:
  description: |
    This is an example of download links for foo
  displayName: example-foo
  links:
    - href: 'https://www.example.com/public/foo.tar'
      text: foo for linux
    - href: 'https://www.example.com/public/foo.mac.zip'
      text: foo for mac
    - href: 'https://www.example.com/public/foo.win.zip'
      text: foo for windows

```

4. **Save** ボタンをクリックします。

## 4.7. YAML サンプルの KUBERNETES リソースへの追加

YAML サンプルはいつでも Kubernetes リソースに動的に追加できます。

## 前提条件

- クラスター管理者の権限があること。

## 手順

1. **Administration** → **Custom Resource Definitions**から、**ConsoleYAMLSample** をクリックします。
2. **YAML** をクリックし、ファイルを編集します。

```
apiVersion: console.openshift.io/v1
kind: ConsoleYAMLSample
metadata:
  name: example
spec:
  targetResource:
    apiVersion: batch/v1
    kind: Job
  title: Example Job
  description: An example Job YAML sample
  yaml: |
    apiVersion: batch/v1
    kind: Job
    metadata:
      name: countdown
    spec:
      template:
        metadata:
          name: countdown
        spec:
          containers:
            - name: counter
              image: centos:7
              command:
                - "bin/bash"
                - "-c"
                - "for i in 9 8 7 6 5 4 3 2 1 ; do echo $i ; done"
          restartPolicy: Never
```

**spec.snippet** を使用して、YAML サンプルが完全な YAML リソース定義ではなく、ユーザーのカーソルで既存の YAML ドキュメントに挿入できる断片を示します。

3. **Save** をクリックします。



## 第5章 WEB コンソールの DEVELOPER パースペクティブ

OpenShift Container Platform Web コンソールは、**Administrator** パースペクティブと **Developer** パースペクティブという 2 つのパースペクティブを提供します。

**Developer** パースペクティブは、以下を実行する機能を含む、開発者のユースケースに固有のワークフローを提供します。

- 既存のコードベース、イメージ、および Dockerfile をインポートして、OpenShift Container Platform でアプリケーションを作成し、デプロイします。
- アプリケーション、コンポーネント、およびプロジェクト内のこれらに関連付けられたサービスと視覚的に対話し、それらのデプロイとビルドの状態をモニターします。
- アプリケーション内のコンポーネントを分類し、コンポーネントの接続をアプリケーション内で、また複数のアプリケーションにまたがって実行します。
- Serverless 機能 (テクノロジープレビュー) を統合します。
- Eclipse Che を使用してアプリケーションコードを編集するためのワークスペースを作成します。

### 5.1. 前提条件

**Developer** パースペクティブにアクセスするために、Web コンソールにログインしていること。

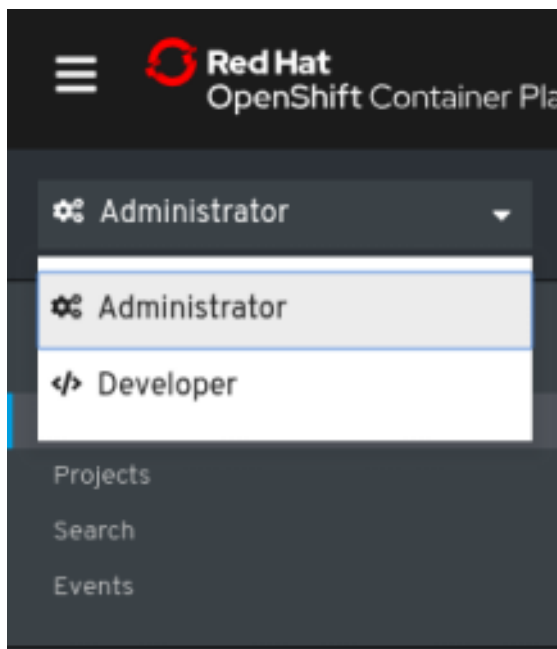
### 5.2. DEVELOPER パースペクティブへのアクセス

OpenShift Container Platform Web コンソールの **Developer** パースペクティブは、開発者のユースケースに固有のワークフローを提供します。

以下のように、Web コンソールから **Developer** パースペクティブにアクセスできます。

#### 手順

1. ログイン認証情報を使用して OpenShift Container Platform Web コンソールにログインします。OpenShift Container Platform Web コンソールのデフォルトビューは **Administrator** パースペクティブです。
2. パースペクティブスイッチャーを使用して、**Developer** パースペクティブに切り替えます。**Topology** ビューがアプリケーションを作成するオプションと共に表示されます。



### 追加リソース

- [Developer](#) パースペクティブを使用して OpenShift Container Platform でアプリケーションを作成し、デプロイする
- [Topology](#) ビューを使用してプロジェクトにアプリケーションを表示し、デプロイメントのステータスを確認し、それらと対話する

## 第6章 OPENSIFT CONTAINER PLATFORM の WEB コンソールの無効化

OpenShift Container Platform の Web コンソールを無効にすることができます。

### 6.1. 前提条件

- OpenShift Container Platform クラスターをデプロイします。

### 6.2. WEB コンソールの無効化

**console.config.openshift.io** リソースを編集して Web コンソールを無効にすることができます。

- **console.operator.openshift.io** リソースを編集します。

```
$ oc edit console.operator.openshift.io cluster
```

以下の例は、変更できるリソースのパラメーターを表示しています。

```
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  managementState: Removed ❶
```

- ❶ **managementState** パラメーター値を **Removed** に設定し、Web コンソールを無効にします。このパラメーターの他の有効な値には以下が含まれます。**Managed** ではクラスターの制御下でコンソールを有効にし、**Unmanaged** は Web コンソール管理を制御するのがユーザーであることを意味します。