



# OpenShift Container Platform 4.4

## Web 控制台

在 OpenShift Container Platform 中使用 web 控制台



# OpenShift Container Platform 4.4 Web 控制台

---

在 OpenShift Container Platform 中使用 web 控制台

## 法律通告

Copyright © 2021 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.

## 摘要

本文档提供了有关使用和定制 OpenShift Container Platform web 控制台的信息。

---

# 目录

<b>第 1 章 访问WEB控制台</b> .....	<b>3</b>
1.1. 先决条件	3
1.2. 了解和访问WEB控制台	3
<b>第 2 章 使用 OPENSIFT CONTAINER PLATFORM DASHBOARD 获取集群信息</b> .....	<b>4</b>
2.1. 关于 OPENSIFT CONTAINER PLATFORM 仪表板页	4
<b>第 3 章 在OPENSIFT CONTAINER PLATFORM中配置WEB控制台</b> .....	<b>5</b>
3.1. 先决条件	5
3.2. 配置WEB控制台	5
<b>第 4 章 在 OPENSIFT CONTAINER PLATFORM 中定制 WEB 控制台</b> .....	<b>6</b>
4.1. 添加自定义徽标和产品名称	6
4.2. 在 WEB 控制台中创建自定义链接	7
4.3. 自定义登录页面	8
4.4. 为外部日志链接定义模板	9
4.5. 创建自定义通知标语	9
4.6. 自定义 CLI 下载	10
4.7. 在 KUBERNETES 资源中添加 YAML 示例	11
<b>第 5 章 关于 WEB 控制台中的开发者视角</b> .....	<b>12</b>
5.1. 先决条件	12
5.2. 访问 DEVELOPER 视角	12
<b>第 6 章 在OPENSIFT CONTAINER PLATFORM中禁用WEB控制台</b> .....	<b>14</b>
6.1. 先决条件	14
6.2. 禁用WEB控制台	14



# 第1章 访问WEB控制台

OpenShift Container Platform Web控制台是可从Web浏览器访问的用户界面。开发人员可以使用Web控制台来直观地浏览并管理项目的内容。

## 1.1. 先决条件

- 必须启用JavaScript才能使用Web控制台。为获得最佳体验，请使用支持WebSockets的Web浏览器。
- 在为集群创建支持基础结构之前，请参阅[OpenShift Container Platform 4.x Tested Integrations](#)页。

## 1.2. 了解和访问WEB控制台

Web控制台作为一个pod 在主服务器（master）上运行。这个 pod 提供了运行Web控制台所需的静态环境。成功安装 OpenShift Container Platform 后，在安装程序的 CLI 输出中可以找到已安装集群的 Web 控制台的 URL 及登录凭据。例如：

```
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>
```

使用这些信息登录并访问Web控制台。

## 第 2 章 使用 OPENSIFT CONTAINER PLATFORM DASHBOARD 获取集群信息

OpenShift Container Platform 仪表板 (dashboard) 包括了集群的高级别信息。在 OpenShift Container Platform web 控制台中通过 **Home** → **Dashboards** → **Overview** 访问它。

OpenShift Container Platform 仪表板提供各种集群信息，被分别显示在独立的仪表板卡中。

### 2.1. 关于 OPENSIFT CONTAINER PLATFORM 仪表板页

OpenShift Container Platform 仪表板由以下各卡组成：

- **Details** 提供有关信息型集群详情的简单概述。  
状态包括 **ok**、**error**、**warning**、**in progress** 和 **unknown**。资源可添加自定义状态名称。
  - 集群 ID
  - 提供者
  - 版本
- **Cluster Inventory** 详细列出资源数目和相关状态。这在通过干预解决问题时非常有用，其中包含以下相关信息：
  - 节点数
  - pod 数量
  - 持久性存储卷声明
  - 集群中的裸机主机，根据其状态列出（只在 **metal3** 环境中可用）。
- **Cluster Capacity** 图表可帮助管理员了解在什么时候集群需要额外的资源。此表包含一个内环和一个外环。内环显示当前的消耗，外环显示为资源配置的阈值，其中包括以下信息：
  - CPU 时间
  - 内存分配
  - 所消耗的存储
  - 所消耗的网络资源
- **Cluster Utilization** 显示在指定时间段内各种资源的能力，以帮助管理员了解高资源消耗的范围和频率。
- **Events** 列出了与集群中最近活动相关的消息，如创建 pod 或虚拟机迁移到另一台主机。
- **Top Consumers** 可帮助管理员了解集群资源是如何被消耗的。点一个资源可以进入一个包括详细信息的页面，它列出了对指定集群资源（CPU、内存或者存储）消耗最多的 Pod 和节点。



## 第 3 章 在 OPENSIFT CONTAINER PLATFORM 中配置 WEB 控制台

您可以修改 OpenShift Container Platform Web 控制台以设置注销重定向 URL，或禁用控制台。

### 3.1. 先决条件

- 部署一个 OpenShift Container Platform 集群。

### 3.2. 配置 WEB 控制台

您可以通过编辑 `console.config.openshift.io` 资源来配置 Web 控制台设置。

- 编辑 `console.config.openshift.io` 资源：

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

以下是控制台的资源定义示例：

```
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  authentication:
    logoutRedirect: "" 1
status:
  consoleURL: "" 2
```

- 1** 指定用户注销后，Web 控制台要加载页面的 URL。如果未指定，则用户将会返回到 Web 控制台的登录页面。通过指定 `logoutRedirect` URL，用户可以使用身份供应商的单点注销（SLO）功能销毁其单点登录会话。
- 2** Web 控制台 URL。您无法修改此参数的值。如果修改了这个值，集群会将其重置为默认值。

## 第 4 章 在 OPENSHIFT CONTAINER PLATFORM 中定制 WEB 控制台

您可以对 OpenShift Container Platform web 控制台进行定制，如设置自定义徽标、产品名、链接、通知标语和命令行下载。这在您需要定制 Web 控制台以满足具体公司或政府要求时特别有用。

### 4.1. 添加自定义徽标和产品名称

您可以通过添加自定义徽标或自定义产品名称来创建自定义品牌。因为这些设置相互独立，因此可以两者都设置或只设置其中的一个。

#### 先决条件

- 您必须具有管理员特权。
- 创建一个要使用的徽标文件。徽标可以是通用图像格式的文件，包括 GIF、JPG、PNG 或 SVG，并有 **max-height** 为 **60px**的限制。

#### 流程

1. 在 **openshift-config** 命名空间中将您的徽标文件导入到配置映射中：

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

2. 编辑 web 控制台的 Operator 配置使其包含 **customLogoFile** 和 **customProductName**：

```
$ 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
```

更新 Operator 配置后，它将会把自定义的 logo 配置映射同步到控制台命名空间中，并将其挂载到 console pod 并重新部署。

3. 检查操作是否成功。如果有任何问题，控制台集群 Operator 将报告 **Degraded** 状态，控制台 Operator 配置也会报告 **CustomLogoDegraded** 状态，但状态类似于 **KeyOrFilenameInvalid** 或 **NoImageProvided**。

运行以下命令检查 **clusteroperator**：

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

运行以下命令检查 console Operator 配置：

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

## 4.2. 在 WEB 控制台中创建自定义链接

### 先决条件

- 您必须具有管理员特权。

### 流程

1. 在 **Administration** → **Custom Resource Definitions** 中点 **ConsoleLink**。
2. 选择 **Instances** 标签
3. 点击 **Create Console Link** 并编辑文件：

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

- 1** 有效的位置设置为 **HelpMenu**、**UserMenu**、**ApplicationMenu** 和 **NamespaceDashboard**。

要使自定义链接出现在所有命名空间中，请按照以下示例操作：

```
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
```

要使自定义链接只出现在某些命名空间中，请按照以下示例操作：

```
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
```

要使自定义链接出现在应用程序菜单中，请按照以下示例操作：

```
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
```

4. 点 **Save** 按钮应用所做的改变。

### 4.3. 自定义登录页面

使用自定义登录页面创建服务条款信息。如果您使用第三方登录提供程序（如 GitHub 或 Google），在将用户信任并期望它重定向到认证提供程序之前，自定义登录页面也会很有用。您还可以在验证过程中显示自定义的错误页。

#### 先决条件

- 您必须具有管理员特权。

#### 流程

1. 运行以下命令来创建您可以修改的模板：

```
$ oc adm create-login-template > login.html
```

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

```
$ oc adm create-error-template > errors.html
```

2. 创建 secret:

```
$ 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. 运行：

■

```
$ oc edit oauths cluster
```

#### 4. 更新规格：

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

运行 **oc explain oauths.spec.templates** 以了解选项。

## 4.4. 为外部日志链接定义模板

如果您连接到可帮助您浏览日志的服务，但需要以特定的方式生成 URL，则可以为链接定义一个模板。

### 先决条件

- 您必须具有管理员特权。

### 流程

1. 在 **Administration** → **Custom Resource Definitions** 中点 **ConsoleExternalLogLink**。
2. 选择 **Instances** 标签
3. 点击 **Create Console External Log Link** 并编辑文件：

```
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. 创建自定义通知标语

### 先决条件

- 您必须具有管理员特权。

### 流程

1. 在 **Administration** → **Custom Resource Definitions** 中点 **ConsoleNotification**。
2. 选择 **Instances** 标签

3. 点击 **Create Console Notification** 并编辑文件：

```

apiVersion: console.openshift.io/v1
kind: ConsoleNotification
metadata:
  name: example
spec:
  text: This is an example notification message with an optional link.
  location: BannerTop 1
  link:
    href: 'https://www.example.com'
    text: Optional link text
    color: '#fff'
    backgroundColor: '#0088ce'

```

- 1** 有效的位置设置为 **BannerTop**、**BannerBottom** 和 **BannerTopBottom**。

4. 点 **Create** 按钮应用所做的改变。

## 4.6. 自定义 CLI 下载

您可以使用自定义链接文本和 URL 来配置用于下载 CLI 的链接。它们可以直接指向软件包的文件或提供软件包的外部页面。

### 先决条件

- 您必须具有管理员特权。

### 流程

1. 进入 **Administration** → **Custom Resource Definitions**。
2. 从 Custom Resource Definitions (CRDs) 列表中选 **ConsoleCLIDownload**。
3. 点 **YAML** 标签页，然后进行编辑：

```

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. 点 **Save** 按钮。

## 4.7. 在 KUBERNETES 资源中添加 YAML 示例

您可以随时动态地将 YAML 示例添加到任何 Kubernetes 资源中。

### 先决条件

- 您必须具有集群管理员特权。

### 流程

1. 在 **Administration** → **Custom Resource Definitions** 中点 **ConsoleYAMLSample**。
2. 点 **YAML** 并编辑该文件：

```

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
  
```

使用 **spec.snippet** 表示 YAML 样本不是完整的 YAML 资源定义，而是可在用户光标处的现有 YAML 文档中插入的片段。

3. 点 **Save**。

## 第 5 章 关于 WEB 控制台中的开发者视角

OpenShift Container Platform web 控制台提供两种视角: **Administrator** 视角和 **Developer** 视角。

**Developer** 视角提供开发人员用例特有的工作流, 比如 :

- 通过导入现有代码基、镜像和 dockerfile 在 OpenShift Container Platform 中创建和部署应用程序。
- 在一个项目中, 以可视的形式和与其关联的应用程序、组件和服务进行交互, 并监控它们的部署和构建状态。
- 在应用程序中对组件进行分组, 并在应用程序内部及跨应用程序间连接组件。
- 集成无服务器功能 (技术预览)。
- 使用 Eclipse Che 创建开发平台来编辑应用程序代码。

### 5.1. 先决条件

要访问 **Developer** 视角, 需要已登陆到 web 控制台。

### 5.2. 访问 DEVELOPER 视角

OpenShift Container Platform web 控制台中的 **Developer** 视角提供了针对于开发人员用例的工作流。

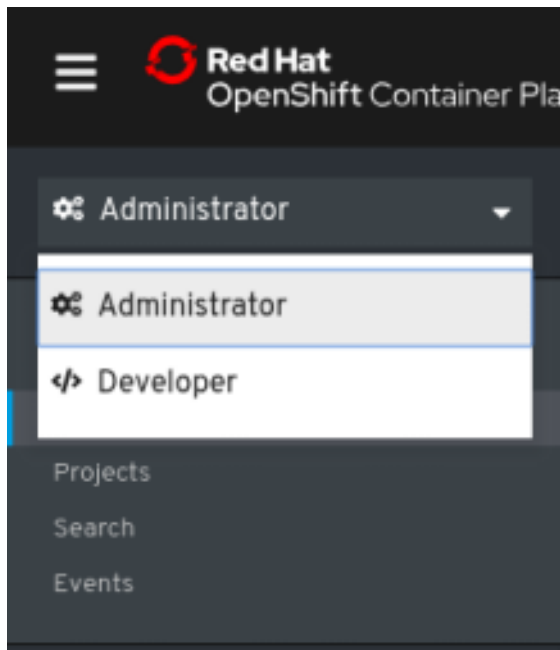
您可以使用以下方法来访问 **Developer** 视角 :

#### 流程

1. 使用您的登录凭证登录到 OpenShift Container Platform web 控制台。OpenShift Container Platform Web 控制台的默认视图是 **Administrator** 视角。
2. 使用视角切换功能把它切换到 **Developer** 视角。带有创建应用程序选项的 **Topology** 视图会被显示。



图 5.1. Developer Perspective (开发者视角)



### 其他资源

- 使用 [Developer](#) 视角在 OpenShift Container Platform 中创建并部署应用程序
- 使用 [Topology](#) 视图查看项目中的应用程序，验证应用程序的部署状态，并与应用程序进行交互。

## 第 6 章 在 OPENSHIFT CONTAINER PLATFORM 中禁用 WEB 控制台

您可以禁用 OpenShift Container Platform Web 控制台。

### 6.1. 先决条件

- 部署一个 OpenShift Container Platform 集群。

### 6.2. 禁用 WEB 控制台

您可以通过编辑 `console.operator.openshift.io` 资源来禁用 Web 控制台。

- 编辑 `console.operator.openshift.io` 资源：

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

以下示例显示了资源中可以修改的参数：

```
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  managementState: Removed 1
```

- 1** 将 `managementState` 参数值设置为 **Removed** 以禁用 Web 控制台。此参数的其他有效值是 **Managed**（启用由集群控制的控制台），**Unmanaged**（启用由用户控制管理的 Web 控制台）。