



# OpenShift Container Platform 4.6

## Web 控制台

在 OpenShift Container Platform 中使用 web 控制台



## OpenShift Container Platform 4.6 Web 控制台

---

在 OpenShift Container Platform 中使用 web 控制台

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Web\_console.ent file |.

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 WEB 控制台概述</b> .....	<b>3</b>
1.1. 关于 WEB 控制台中的 ADMINISTRATOR 视角	3
1.1.1. 访问管理员视角	3
1.2. 关于 WEB 控制台中的开发者视角	4
1.2.1. 访问 Developer 视角	4
<b>第 2 章 访问WEB控制台</b> .....	<b>6</b>
2.1. 先决条件	6
2.2. 了解和访问WEB控制台	6
<b>第 3 章 使用 OPENSIFT CONTAINER PLATFORM DASHBOARD 获取集群信息</b> .....	<b>7</b>
3.1. 关于 OPENSIFT CONTAINER PLATFORM 仪表板页	7
<b>第 4 章 在OPENSIFT CONTAINER PLATFORM中配置WEB控制台</b> .....	<b>8</b>
4.1. 先决条件	8
4.2. 配置WEB控制台	8
<b>第 5 章 在 OPENSIFT CONTAINER PLATFORM 中定制 WEB 控制台</b> .....	<b>9</b>
5.1. 添加自定义徽标和产品名称	9
5.2. 在 WEB 控制台中创建自定义链接	10
5.3. 自定义 WEB 控制台 URL	11
5.4. 自定义登录页面	12
5.5. 为外部日志链接定义模板	13
5.6. 创建自定义通知标语	14
5.7. 自定义 CLI 下载	14
5.8. 在 KUBERNETES 资源中添加 YAML 示例	15
<b>第 6 章 关于 WEB 控制台中的 WEB 终端</b> .....	<b>17</b>
6.1. 安装 WEB 终端	17
6.2. 使用 WEB 终端	18
6.3. 卸载 WEB 终端	18
6.3.1. 删除 Web 终端组件和自定义资源	18
6.3.2. 使用 Web 控制台卸载 Operator	20
<b>第 7 章 在OPENSIFT CONTAINER PLATFORM中禁用WEB控制台</b> .....	<b>21</b>
7.1. 先决条件	21
7.2. 禁用WEB控制台	21



# 第 1 章 WEB 控制台概述

Red Hat OpenShift Container Platform Web 控制台提供了一个图形用户界面，用于视觉化您的项目数据，并执行管理、管理和故障排除任务。Web 控制台在 openshift-console 项目的 control plane 节点上运行。它由一个 **console-operator** pod 管理。支持 **管理员**和 **开发人员**透视图。

**管理员**和 **开发者视角**都允许您为 OpenShift Container Platform 创建快速启动指南。快速开始是用户任务的指导教程，可用于熟悉应用程序、Operator 或其他产品。

## 1.1. 关于 WEB 控制台中的 ADMINISTRATOR 视角

**Administrator** 视角可让您查看集群清单、容量、常规和特定使用信息以及重要事件的流，它们可帮助您简化计划和故障排除任务。项目管理员和开发人员可以查看 **Administrator** 视角。

集群管理员还可在 OpenShift Container Platform 4.7 及之后的版本中为 web 终端 Operator 打开内嵌的命令行终端实例。



### 注意

显示的默认 Web 控制台视角取决于用户的角色。如果用户被视为管理员，则默认显示 **Administrator** 视角。

**Administrator** 视角提供特定于管理员用例的工作流，例如：

- 管理工作负载、存储、网络 and 集群设置。
- 使用 Operator Hub 安装和管理 Operator。
- 添加允许用户通过角色和角色绑定登录和管理用户访问权限的身份提供程序。
- 查看和管理各种高级设置，如集群更新、部分集群更新、集群 Operator、自定义资源定义 (CRD)、角色绑定和资源配额。
- 访问和管理监控功能，如指标、警报和监控仪表盘。
- 查看并管理有关集群的日志记录、指标和高状态信息。
- 在 OpenShift Container Platform 中，以视觉方式与与 **Administrator** 视角关联的应用程序、组件和服务交互。

### 其他资源

如需有关 [web 终端 Operator](#) 的更多信息，请参阅 [web 控制台中的 web 终端](#)。

### 1.1.1. 访问管理员视角

OpenShift Container Platform Web 控制台中的 **Administrator** 视角提供了特定于管理员用例的工作流。您可以通过 Web 控制台访问 **Administrator** 视角，如下所示：

#### 流程

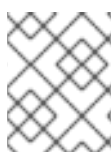
- 使用登录凭证登录到 OpenShift Container Platform web 控制台，以访问 **Administrator** 视角。

## 1.2. 关于 WEB 控制台中的开发者视角

**Developer** 视角提供了几个用来部署应用程序、服务和数据库的内置方法。在 **Developer** 视角中，您可以：

- 查看组件上滚动和重新创建推出部署的实时视觉化。
- 查看应用状态、资源利用率、项目事件流和配额消耗。
- 将您的项目与他人共享。
- 通过在项目上运行 Prometheus Query Language(PromQL)查询并查看图表中呈现的指标来排除应用程序的问题。指标提供有关集群状态以及您要监控的任何用户定义的工作负载的信息。

集群管理员也可以在 OpenShift Container Platform 4.7 及之后的版本中的 Web 控制台中打开内嵌的命令终端实例。



### 注意

显示的默认 Web 控制台视角取决于用户的角色。如果用户是开发人员，则 **Developer** 视角会被默认显示。

**Developer** 视角提供开发人员用例特有的工作流，比如：

- 通过导入现有代码库、镜像和容器文件，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。

### 1.2.1. 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：

#### 先决条件

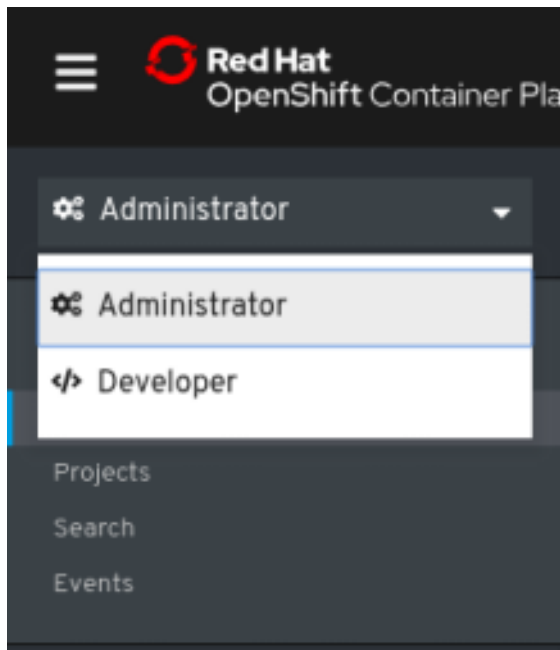
要访问 **Developer** 视角，需要已登陆到 web 控制台。OpenShift Container Platform Web 控制台的默认视图是 **Developer** 视角。

#### 流程

1. 使用视角切换功能把它切换到 **Developer** 视角。此时会显示包含集群中所有项目的列表的 **Topology** 视图。



图 1.1. Developer Perspective (开发者视角)



2. 从列表中选择现有项目，或使用 **Project** 下拉列表创建新项目。

如果您项目中没有工作负载或应用程序，则 **Topology** 视图会显示可用来创建应用程序的选项。如果已有工作负载，则 **Topology** 视图会以图形的形式显示工作负载节点。

#### 其他资源

- [查看集群信息](#)
- [配置Web控制台](#)
- [自定义 Web 控制台](#)
- [在 web 控制台中启动内嵌的命令行终端实例](#)
- [禁用Web控制台](#)

## 第 2 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。

### 2.1. 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。

### 2.2. 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。当使用 **openshift-install create cluster** 成功安装 OpenShift Container Platform 后，在安装程序的 CLI 输出中找到已安装集群的 Web 控制台的 URL 及登录凭证。例如：

#### 输出示例

```
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。

对于不是您安装的、已存在的集群，可以使用 **oc whoami --show-console** 查看 web 控制台 URL。

## 第 3 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。

### 3.1. 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 4 章 在 OPENSHIFT 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。

### 4.1. 先决条件

- 部署一个 OpenShift Container Platform 集群。

### 4.2. 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。要将其更新为自定义值，请参阅 [自定义 Web 控制台 URL](#)。

## 第 5 章 在 OPENSIFT 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。

### 5.1. 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
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: console-custom-logo
  namespace: openshift-config
data:
  console-custom-logo.png: <base64-encoded_logo> ... 1
```

- 1 提供有效的 base64 编码徽标。

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

```
$ oc edit consoles.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。

- 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 consoles.operator.openshift.io -o yaml
```

## 5.2. 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

- 在 **Administration** → **Custom Resource Definitions** 中点 **ConsoleLink**。
- 选择 **Instances** 标签
- 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
```

- 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. 单击 **Save** 以应用您的更改。

### 5.3. 自定义 WEB 控制台 URL

您可以将 Web 控制台 URL (**consoleURL**) 更新为自定义值。

#### 流程

1. 在 **consoles.operator.openshift.io** 自定义资源中修改安装过程中默认创建的集群实例：

```
$ oc patch consoles.operator.openshift.io cluster --patch '{"spec":{"route":{"hostname":"console.example.com"}}}' --type=merge
```

```

apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  route:
    hostname: 'console.example.com'

```

2. 如果指定自定义证书，则必须在 **openshift-config** 命名空间中创建具有密钥和证书的 secret。例如：

```
$ oc create secret tls console-tls --key=key.pem --cert=cert.pem -n openshift-config
```

```
apiVersion: v1
kind: Secret
metadata:
  name: console-tls
  namespace: openshift-config
type: kubernetes.io/tls
data:
  tls.crt: >-
    <base64-encoded_cert> ... ❶
  tls.key: >-
    <base64-encoded_key> ... ❷
```

❶ 提供有效的 TLS 证书。

❷ 提供有效的 TLS 密钥。

然后，编辑 Web 控制台的 Operator 配置：

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

在配置资源中添加以下小节：

```
spec:
  route:
    hostname: console.example.com
  secret:
    name: console-tls
```

## 5.4. 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。



### 注意

自定义错误模板仅限于使用重定向的身份提供程序（IDP），如请求标头和基于 OIDC 的操作。它对使用直接密码验证的 IDP（如 LDAP 和 HTTPasswd）不起作用。

### 先决条件

- 您必须具有管理员特权。

### 流程

- 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。

## 5.5. 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
```

## 5.6. 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. 点击 **Create** 以应用您的更改。

## 5.7. 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。

## 5.8. 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 6 章 关于 WEB 控制台中的 WEB 终端

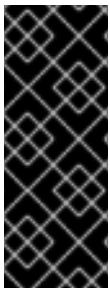
您可以在 OpenShift web 控制台中启动内嵌的命令行终端实例。您必须首先安装 Web Terminal Operator 来使用 Web 终端。



### 注意

集群管理员可以访问 OpenShift Container Platform 4.7 及之后的版本中的 Web 终端。

此终端实例预安装了与集群交互的通用 CLI 工具，如 **oc**、**kubectl**、**odo**、**kn**、**tkn**、**helm**、**kubens**、**subctl** 和 **kubectx**。它还包含正在处理的项目的上下文，并自动记录您使用凭证的项目。



### 重要

Web 终端只是一个技术预览功能。技术预览功能不被红帽产品服务等级协议 (SLA) 支持，且可能在功能方面有缺陷。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的详情，请参阅 <https://access.redhat.com/support/offerings/techpreview/>。

## 6.1. 安装 WEB 终端

您可以使用 OpenShift Container Platform OperatorHub 中列出的 Web Terminal Operator 来安装 Web 终端。安装 Web Terminal Operator 时，会自动安装命令行配置（如 **DevWorkspace** CRD）所需的自定义资源定义（CRD）。打开 web 终端时，web 控制台会创建所需的资源。

### 先决条件

- 使用具有 **cluster-admin** 权限的账户访问 OpenShift Container Platform 集群。


### 流程

1. 在 Web 控制台的 **Administrator** 视角中，导航到 **Operators → OperatorHub**。
2. 使用 **Filter by keyword** 复选框在目录中搜索 **Web Terminal Operator**，然后点击 **Web Terminal** 标题。
3. 参阅 **Web Terminal** 页面中有关 Operator 的简单描述，然后点击 **Install**。
4. 在 **Install Operator** 页面中，保留所有字段的默认值。
  - **Update Channel** 菜单中的 **alpha** 选项启用 Web Terminal Operator 最新版本的安装。
  - **Installation Mode** 菜单中的 **All namespaces on the cluster** 选项可让 Operator 监视并可供集群中的所有命名空间使用。
  - **Installed Namespace** 菜单中的 **openshift-operators** 选项会在默认的 **openshift-operators** 命名空间中安装 Operator。
  - **Approval Strategy** 菜单中的 **Automatic** 选项确保以后对 Operator 的升级由 Operator Lifecycle Manager 自动处理。

5. 点击 **Install**。
6. 在 **Installed Operators** 页面中，点 **View operator** 来验证 **Installed Operators** 页中列出的 Operator。
7. 安装 Operator 后，刷新页面以查看控制台右上角的命令行终端图标。

## 6.2. 使用 WEB 终端

安装 Web Terminal Operator 后，您可以使用 Web 终端，如下所示：

1. 要启动 web 终端，请点击控制台右上角的命令行终端图标(  )。在 **Command line terminal** 窗格中会显示 web 终端实例。此实例使用您的凭证自动登录。
2. 从 Project 下拉列表中选择创建 **DevWorkspace** CR 的 项目。默认情况下会选择当前项目。



### 注意

- 只有在不存在 **DevWorkspace** CR 时才会创建 DevWorkspace CR。
- **openshift-terminal** 项目是集群管理员使用的默认项目。它们没有选择其他项目的选项。

3. 点 **Start** 使用所选项目初始化 Web 终端。

初始化 web 终端后，您可以在 web 终端中使用预安装的 CLI 工具，如 **oc**、**kubectl**、**odo**、**kn**、**tkn**、**helm**、**kubens**、**subctl** 和 **kubectx**。

## 6.3. 卸载 WEB 终端

卸载 web 终端需要两步：

1. 删除安装 Operator 时添加的组件和自定义资源（CR）。
2. 卸载 Web Terminal Operator。

卸载 Web Terminal Operator 不会移除安装 Operator 时创建的任何自定义资源定义（CRD）或受管资源。为了安全起见，必须手动卸载这些组件。删除这些组件还允许您通过确保在卸载 Operator 时不会闲置终端来保存集群资源。

### 先决条件

- 使用具有 **cluster-admin** 权限的账户访问 OpenShift Container Platform 集群。

### 6.3.1. 删除 Web 终端组件和自定义资源

使用 CLI 删除安装 Web Terminal Operator 期间创建的 CR。

### 流程

1. 运行以下命令以确保所有 **DevWorkspace** CR 及其相关的 Kubernetes 对象（如部署）被删除。

```
$ oc delete devworkspaces.workspace.devfile.io --all-namespaces --all --wait
```

```
$ oc delete workspaceroutings.controller.devfile.io --all-namespaces --all --wait
```

```
$ oc delete components.controller.devfile.io --all-namespaces --all --wait
```



### 警告

如果此步骤未完成，则终结器很难轻松地完全卸载 Operator。

## 2. 运行以下命令以删除 CRD:

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io  
workspaceroutings.controller.devfile.io
```

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io components.controller.devfile.io
```

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io  
devworkspaces.workspace.devfile.io
```

## 3. 删除 **DevWorkspace-Webhook-Server** 部署 :

```
$ oc delete deployment/devworkspace-webhook-server -n openshift-operators
```



### 注意

运行此步骤及以下步骤时，无法使用 **oc exec** 命令在容器中运行命令。删除 Webhook 后，您将可以再次使用 **oc exec** 命令。

## 4. 运行以下命令以删除所有闲置服务、secret 和配置映射 :

```
$ oc delete all --selector app.kubernetes.io/part-of=devworkspace-  
operator,app.kubernetes.io/name=devworkspace-webhook-server
```

```
$ oc delete serviceaccounts devworkspace-webhook-server -n openshift-operators
```

```
$ oc delete configmap devworkspace-controller -n openshift-operators
```

```
$ oc delete clusterrole devworkspace-webhook-server
```

```
$ oc delete clusterrolebinding devworkspace-webhook-server
```


## 5. 运行以下命令以删除变异或验证 Webhook 配置 :

```
$ oc delete mutatingwebhookconfigurations controller.devfile.io
```

```
$ oc delete validatingwebhookconfigurations controller.devfile.io
```

### 6.3.2. 使用 Web 控制台卸载 Operator

#### 流程

1. 在 web 控制台的 **Administrator** 视角中，导航到 **Operators → Installed Operators**。
2. 滚动过滤器列表或在 **Filter by name** 框中输入关键字以查找 **Web Terminal Operator**。
3. 点击 Web Terminal Operator 的 Options 菜单 ，然后选择 **Uninstall Operator**。
4. 在 **Uninstall Operator** 确认对话框中，点 **Uninstall** 从集群中删除 Operator、Operator 部署和 pod。Operator 会停止运行，并且不再接收更新。



## 第 7 章 在 OPENSIFT CONTAINER PLATFORM 中禁用 WEB 控制台

您可以禁用 OpenShift Container Platform Web 控制台。

### 7.1. 先决条件

- 部署一个 OpenShift Container Platform 集群。

### 7.2. 禁用 WEB 控制台

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

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

```
$ oc edit consoles.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）。