OpenShift Container Platform 4.7

Web 控制台

在 OpenShift Container Platform 中使用 web 控制台

Last Updated: 2021-05-06

OpenShift Container Platform 4.7 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 ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® 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 ® 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

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：

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。

第 2 章 使用 OPENSHIFT 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 OPENSHIFT 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。

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。要将它更新为自定义值，请参阅**自定义 Web 控制台 URL**。

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 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。

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 consoles.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 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
```

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
```

然后，在配置资源中添加以下小节：

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

4.4. 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。

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。

4.5. 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
```

4.6. 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 **Create** 按钮应用所做的改变。

4.7. 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。

4.8. 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 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。

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

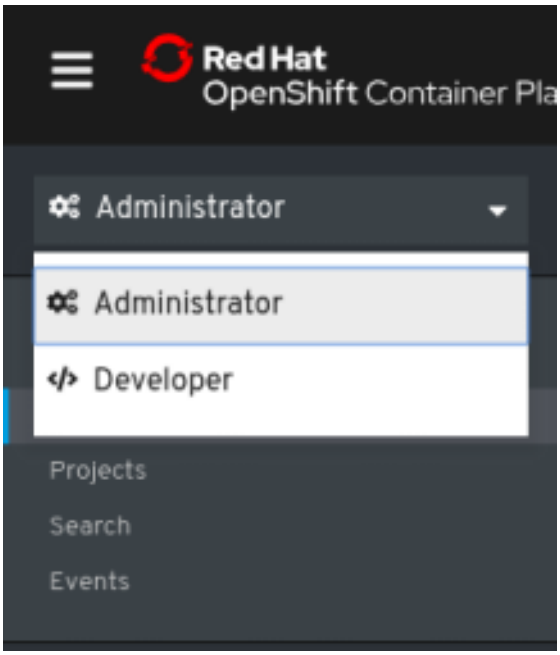
OpenShift Container Platform web 控制台中的 **Developer** 视角提供了针对于开发人员用例的工作流。

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。此时会显示包含集群中所有项目的列表的 **Topology** 视图。

图 5.1. Developer Perspective (开发者视角)



3. 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

- 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。

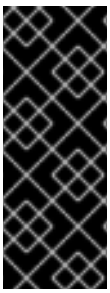
第 6 章 关于 WEB 控制台中的 WEB 终端

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。

此终端实例已预先安装与集群交互的通用 CLI 工具，如 **oc**、**kubectl**、**odo**、**kn**、**tkn**、**helm** 、 kubens 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。

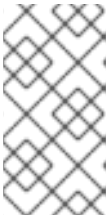
7. 安装 Operator 后，刷新页面以查看控制台右上角的命令行终端图标。

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** 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:

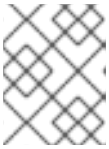
```
$ oc delete customresourcedefinitions.apiextensions.k8s.io
workspaceroutings.controller.devfile.io
```

```
$ 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OPENSHIFT 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）。

第 8 章 在 WEB 控制台中创建快速启动指南

如果您要为 OpenShift Container Platform Web 控制台创建快速启动指南，请按照以下步骤保留所有快速启动的用户体验。

8.1. 了解快速开始

快速开始是用户任务的指导教程。在 Web 控制台中，您可以在 **Help** 菜单下快速启动访问。它们在使用应用程序、Operator 或其他产品时特别有用。

快速开始主要由任务和步骤组成。每个任务都有多个步骤，每个快速开始都有多个任务。例如：

- 任务 1
  - 第 1 步
  - 第 2 步
  - 第 3 步
- 任务 2
  - 第 1 步
  - 第 2 步
  - 第 3 步
- 任务 3
  - 第 1 步
  - 第 2 步
  - 第 3 步

8.2. 快速启动用户工作流

当您与现有快速启动指南交互时，这是预期的工作流体验：

1. 在 **Administrator** 或 **Developer** 视角中，点击 **Help** 图标并选择 **Quick Starts**。
2. 点快速启动卡。
3. 在出现的面板中点 **Start**。
4. 完成屏幕的说明，然后点 **Next**。
5. 在出现 **Check your work** 模块时，回答问题以确认您成功完成了该任务。
   a. 如果您选择 **Yes**，点 **Next** 来继续到下一个任务。
   b. 如果您选择 **No**，重复任务说明并再次检查您的工作。
6. 重复以上第 1 到 6 步，以便快速完成剩余的任务。
7. 完成最后的任务后，点 **Close** 关闭快速启动。

8.3. 快速启动组件

快速开始由以下部分组成：

- **Card**：提供快速启动基本信息的 catalog 标题，其中包括标题、描述、时间提交和完成状态
- **Introduction**：概述快速开始的目标和任务
- **Task headings**：快速启动中每个任务的超链接标题
- **Check your work module**：一个模块用户使用一个模块来确认他们成功完成了任务，然后到快速启动的下一个任务为止
- **Hints**：帮助用户识别产品特定部分的动画
- **Buttons**
  - **Next and back buttons**：用来浏览快速开始任务里的步骤和模块的按钮
  - **Final screen buttons**：关闭快速启动，返回到快速启动中的先前任务，并查看所有快速启动

快速启动的主要内容包括以下部分：

- **Card copy**
- **简介**
- **Task steps**
- **Modals and in-app messaging**
- **Check your work module**

8.4. 快速开始

OpenShift Container Platform 引入了由 **ConsoleQuickStart** 对象定义的快速启动自定义资源。operator 和管理员可以使用此资源来快速使用集群。

先决条件

- 您必须具有集群管理员特权。

流程

1. 要创建新快速启动，请运行：

```
$ oc get -o yaml consolequickstart spring-with-s2i > my-quick-start.yaml
```

2. 运行：

```
$ oc create -f my-quick-start.yaml
```

3. 根据本文档中介绍的指南更新 YAML 文件。
4. 保存您的编辑。

8.4.1. 查看快速启动 API 文档

流程

- 要查看快速启动 API 文档，请运行：

```
$ oc explain consolequickstarts
```

运行 **oc explain -h** 以了解有关 **oc explain** 使用的更多信息。

8.4.2. 将快速启动 CR 中的元素映射到快速启动 CR

本节帮助您视觉地映射快速启动自定义资源（CR）的部分，使其出现在 web 控制台中快速启动的自定义资源（CR）中。

8.4.2.1. conclusion 元素

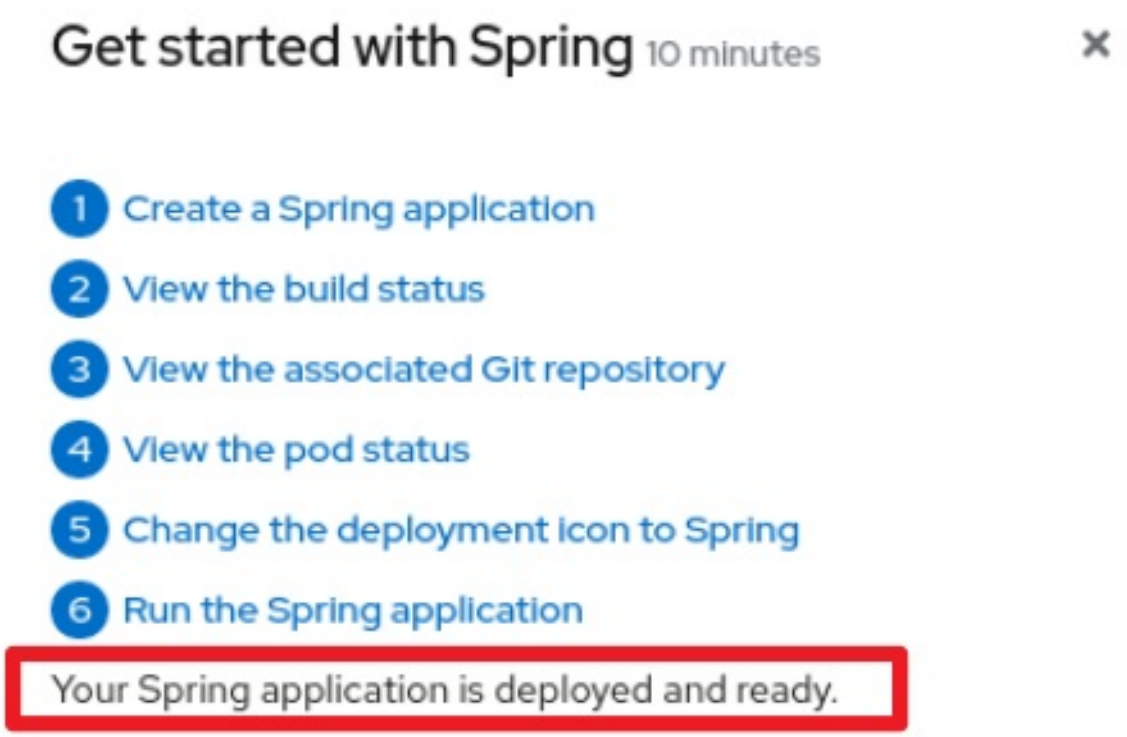
查看 YAML 文件中的 conclusion 元素

```
...
summary:
  failed: Try the steps again.
  success: Your Spring application is running.
title: Run the Spring application
conclusion: >-
Your Spring application is deployed and ready. 1
```

1 conclusion 文本

在 web 控制台中查看 conclusion 元素

最后会出现在快速开始的最后部分。



8.4.2.2. description 元素

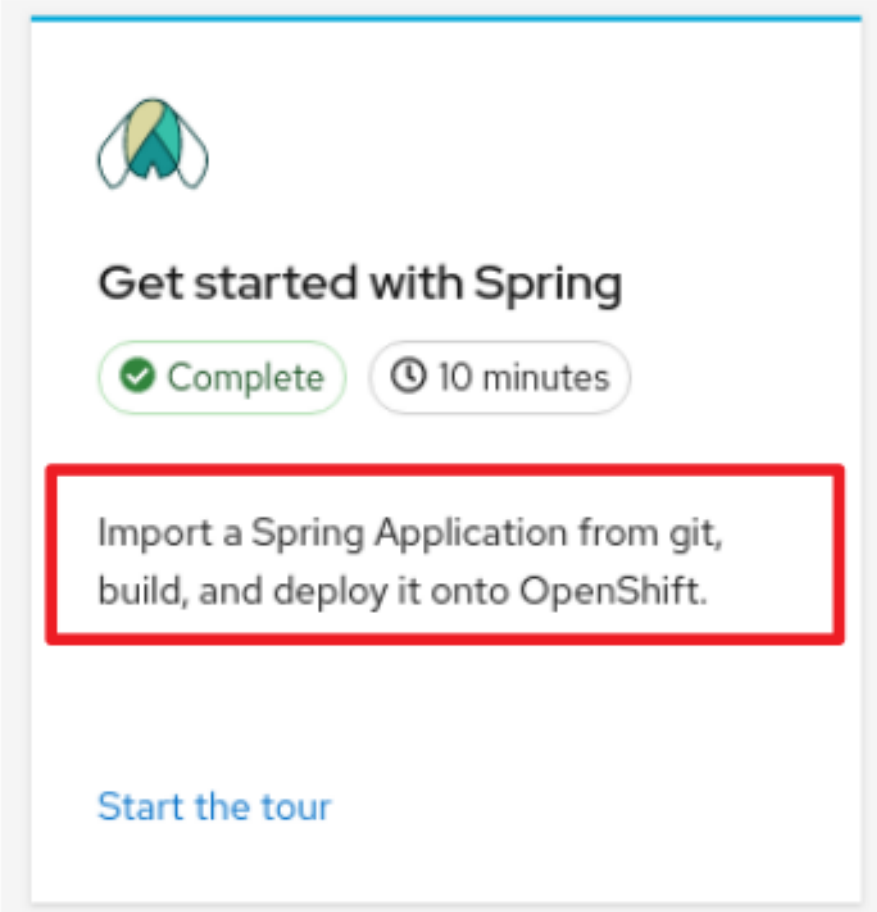
查看 YAML 文件中的 description 元素

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.' (1)
...
```

(1) description 文本

在 web 控制台中查看 description 元素

这个描述会出现在**快速开始**页的介绍中。



8.4.2.3. displayName 元素

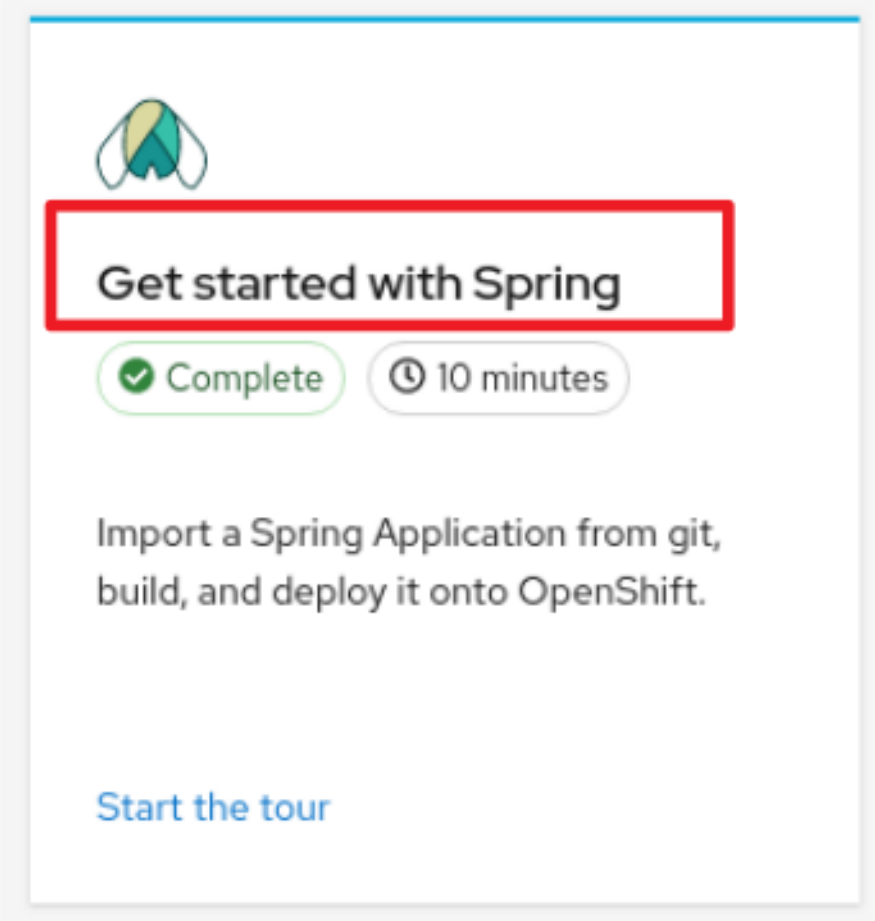
查看 YAML 文件中的 displayName 元素

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.'
  displayName: Get started with Spring (1)
  durationMinutes: 10
```

(1) **displayName** 文本。

在 web 控制台中查看 displayName 元素

显示名称会出现在**快速启动**页的介绍中。



8.4.2.4. durationMinutes 元素

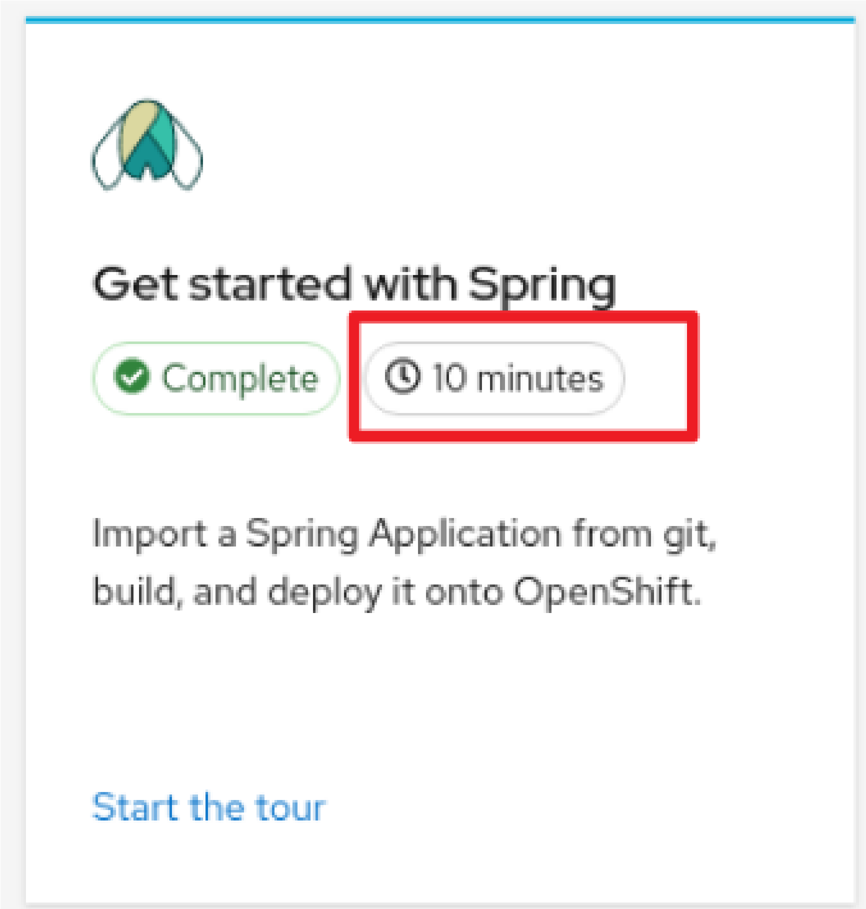
在 YAML 文件中查看 durationMinutes 元素

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.'
  displayName: Get started with Spring
  durationMinutes: 10 (1)
```

(1) **durationMinutes** 值，以分钟为单位。这个值定义了快速启动完成所需的时间。

在 web 控制台中查看 durationMinutes 元素

durationMinutes 元素会出现在**快速开始**页的介绍中。



8.4.2.5. icon 元素

查看 YAML 文件中的 icon 元素

```
...
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.'
  displayName: Get started with Spring
  durationMinutes: 10
  icon: >- (1)
```

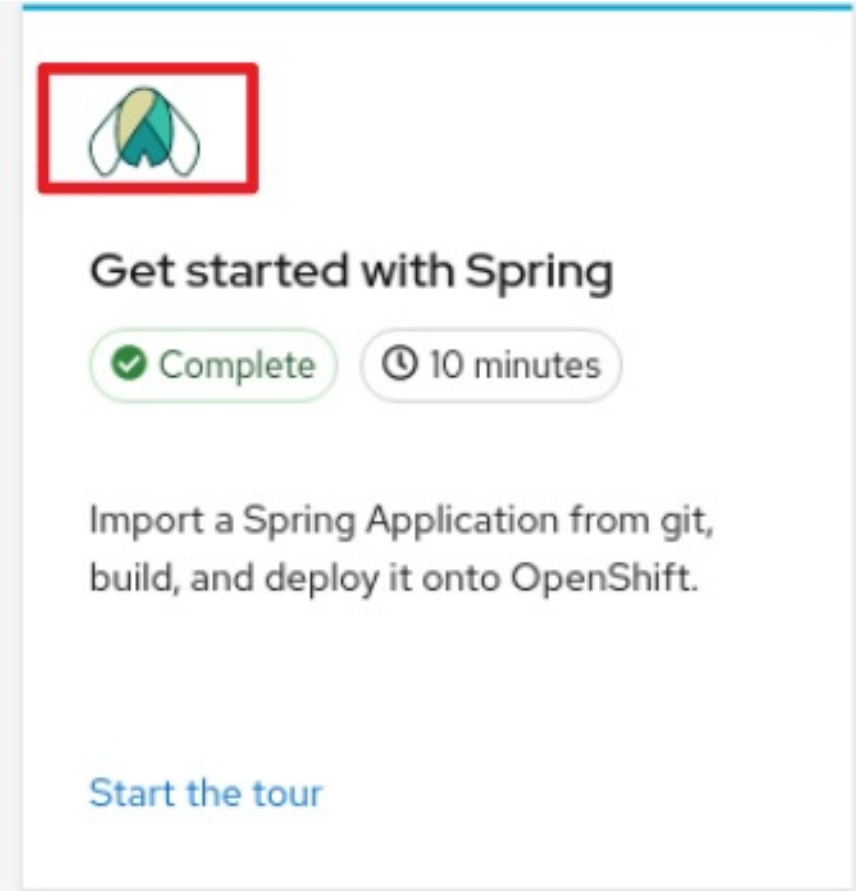

```

JMYXllcl8xIiBkYXRhLW5hbWU9IkxheWVyIDEiIHZpZXdCb3g9IjAgMCAxMDI0IDEwMjQiPjxkZWZzPjxzc
HlsZT4uY2xzLTF7ZmlsbDojMTUzZDNjO30uY2xzLTJ7ZmlsbDojZDhkYTlkO30uY2xzLTN7ZmlsbDojNT
hjMGE4O30uY2xzLTR7ZmlsbDojZmZmO30uY2xzLTV7ZmlsbDojM2Q5MTkxO308L3N0eWxlPjwvZGV
mcz48dGl0bGU+c25vd2Ryb3BfaWNvbl9yZ2JfZGVmYXVsdDwvdGl0bGU+PHBhdGggY2xhc3M9ImNsc
y0xIiBkPSJNMTAxMi42OSw1OTNjLTExLjEyLTM4LjA3LTMxLTczLTU5LjIxLTEwMy44LTkuNS0xMS4zL
TIzLjIxLTI4LjI5LTM5LjA2LTQ3Ljk0QzgzMy41MywzNDEsNzQ1LjM3LDIzNC4xOCw2NzQsMTY4Ljk0Yy
01LTUuMjYtMTAuMjYtMTAuMzEtMTUuNjUtMTUuMDdhMjQ2LjQ5LDI0Ni40OSwwLDAsMC0zNi41NS
0yNi44LDE4Mi41LDE4Mi41LDAsMCwwLTIwLjMtMTEuNzcsMjAxLjUzLDIwMS41MywwLDAsMC00My4
xOS0xNUExNTUuMjQsMTU1LjI0LDAsMCwwLDUyOCw5NS4yYy02Ljc2LS42OC0xMS43NC0uODEtM
TQuMzktLjgxaDBsLTEuNjIsMC0xLjYyLDBhMTc3LjMsMTc3LjMsMCwwLDAtMzEuNzcsMy4zNSwyMDg
uMjMsMjA4LjIzLDAsMCwwLTU2LjEyLDE3LjU2LDE4MSwxODEsMCwwLDAtMjAuMjcsMTEuNzUsMjQ
3LjQzLDI0Ny40MywwLDAsMC0zNi41NywyNi44MUMzNjAuMjUsMTU4LjYyLDM1NSwxNjMuNjgsMzUw
LDE2OWMtNzEuMzUsNjUuMjUtMTU5LjUsMTcyLTI0MC4zOSwyNzIuMjhDOTMuNzMsNDYwLjg4LDg
wLDQ3Ny44Nyw3MC41Miw0ODkuMTcsNDIuMzUsNTIwLDIyLjQzLDU1NC45LDExLjMxLDU5MywwLjNz
IsNjI5LjIyLTEuNzMsNjY3LjY5LDQsNzA3LjMxLDE1LDc4Mi40OSw1NS43OCw4NTkuMTIsMTE4LjkzLD
kyMy4wOWEyMiwyMiwwLDAsMCwxNS41OSw2LjUyaDEuODNsMS44Ny0uMzJjODEuMDYtMTMuOT
```

EsMTEwLTc5LjU3LDE0My40OC0xNTUuNiwzLjkxLTguODgsNy45NS0xOC4wNSwxMi4yLTI3LjQzcTU
uNDIsOC41NCwxMS4zOSwxNi4yM2MzMS44NSw0MC45MSw3NS4xMiw2NC42NywxMzIuMzIsNzIuNj
NsMTguOCwyLjYyLDQuOTUtMTguMzNjMTMuMjYtNDkuMDcsMzUuMy05MC44NSw1MC42NC0xMT
YuMTksMTUuMzQsMjUuMzQsMzcuMzgsNjcuMTIsNTAuNjQsMTE2LjE5bDUsMTguMzMsMTguOC0yL
jYyYzU3LjItOCwxMDAuNDctMzEuNzIsMTMyLjMyLTcyLjYzcTYtNy42OCwxMS4zOS0xNi4yM2M0LjI1L
DkuMzgsOC4yOSwxOC41NSwxMi4yLDI3LjQzLDMzLjQ5LDc2LDYyLjQyLDE0MS42OSwxNDMuNDgs
MTU1LjZsMS44MS4zMWgxLjg5YTIyLDIyLDAsMCwwLDE1LjU5LTYuNTJjNjMuNjMtNjQsMTAzLjk1LT
E0MC42LDExNC44OS0yMTUuNzhhMTAyNS43Myw2NjcuNjksMTAyMy4yOCw2MjkuMjIsMTAxMi42O
Sw1OTNaIi8+PHBhdGggY2xhc3M9ImNscy0yIiBkPSJNMzY0LjE1LDE4NS4yM2MxNy44OS0xNi40LDM
0LjctMzAuMTUsNDkuNzctNDAuMTFhMjEyLDIxMiwwLDAsMSw2NS45My0yNS43M0ExOTgsMTk4LDA
sMCwxLDUxMiwxMTYuMjdhMTk2LjExLDE5Ni4xMSwwLDAsMSwzMiwzLjFjNC41LjkxLDkuMzYsMi4wN
wxNC41MywzLjUyLDYwLjQxLDIwLjQ4LDg0LjkyLDkxLjA1LTQ3LjQ0LDI0OC40NDi0yOC43NSwzNC4x
Mi0xNDAuNywxOTQuODQtMTg0LjY2LDI2OC40MmE2MzAuODYsNjMwLjg2LDAsMCwwLTMzLjIyLD
U4LjMyQzI3Niw2NTUuMzQsMjY1LjQsNTk4LDI2NS40LDU0OS4xMiwyNjUuNCw0MzUuNjEsMzExLjY
5LDI0MC43NCwzNjQuMTUsMTg1LjIzWiIvPjxwYXRoIGNsYXNzPSJjbHMtMyIgZD0iTTUyNy41NCwzO
DQuODNjMjguNDMtMTg0Ljc0LTk5LjkxLTI0NC4xOS01MTkuMTYtMjEzLjEsMTE4LjU3NSw0MC43MzU
0LjI2NC44LDI2Mi4yNy40NDIwLjI0MTUuMTkwNi4xMCwxMDQuNjQuNDcuMzQ4NTEyLjg0NyYsNTEzN
MiM3LDgxNi44OCwyMTYuOTMsODkzLjkzLDEzNC41OCw5MDguMDdhIi8+PHBhdGggY2xhc3M9ImN
scy01IiBkPSJNNTgzLjQzLDgxMy43OUM1NjAuMTgsNzI3LjcyLDUxMiw2NjQuMTUsNTEyLDY2NC4xN
XMtNDguMTcsNjMuNTctNzEuNDMsMTQ5LjY0Yy00OC40NS02Ljc0LTEwMC45MS0yNy41Mi0xMzUu
NjYtOTEuMTEuMTEuMjQ1LjkxNjMuMTY1LjM1My0xNTEuNzAuNDU5LTEzNS4xOS4wNzUuMy0yODYuMi4wMy4wMi
0ODM4Ny4yOGNwNDEuNTMuOTIuNjg0LjM5NTc2MS4yNjg1NDAuNTMuNTIuNDQuOTIuMzYuNTcuMzAu
NjYtOTEuMThhNjQ1LjY4LDY0NS42OCwwLDAsMSwzOS4yLTkxLjIzLTkxLjMMTY3LjUzLTE1Ni4xMi0xMzk
DIsMTY3LjEyLDI0NS42NmwuMTkuMzEuMjEuMzUuMzAuMzY1.40.NjE2Mi4xNy4xMzg5LjE5MkQ0LDg4
3MS41NEM2ODQuMzQsNzg2LjI3LDYzMS44NCw4MDcuMDUsNTgzLjQzLDgxMy43OVoiLz48cGF0a
CBjbGFzcz0iY2xzLTQiIGQ9Ik04ODkuNzUsOTA4YS4zOS4zOSwwLDAsMS0uMjcuMTFoLS4wNkM4M
DcuMDcsODkzLjkzLjc4My4yNzMsODQzLjc3LjM3LjM5LjQ4OSwMDAwLDcyM2EzMDcuNDEsNzQ3LjQ3LDcwLDlwL
jQ1LTU1LjU0YzExLTQ1LjExLTYxLjItMTA5OTEyLjE3NzY5LjlwLTIwLjQ1LTExLjI4LTkxLjE1LTM2Ljc1LTI4LTE3M
y0zNjguNTI6LEk5LjI2LTAyMzMuzNMwMjcuNDE0MC00MkAGTIwwLjYxLTE1OTgwLTI2MTksNTNDX
wyMDIuNzUsMjQyLjE5QzEwNDQuNzksNjM2LjcuNDYsMzA4NjIuMjc3Ni40LDgxMDIuMzA0OS41NS43OTkwZDhaIi8+
PC9zdmc+Cg==
...

1. 定义为 base64 值的图标。

在 web 控制台中查看 icon 元素

这个描述会出现在**快速开始**页的介绍中。



8.4.2.6. introduction 元素

查看 YAML 文件中的 introduction 元素

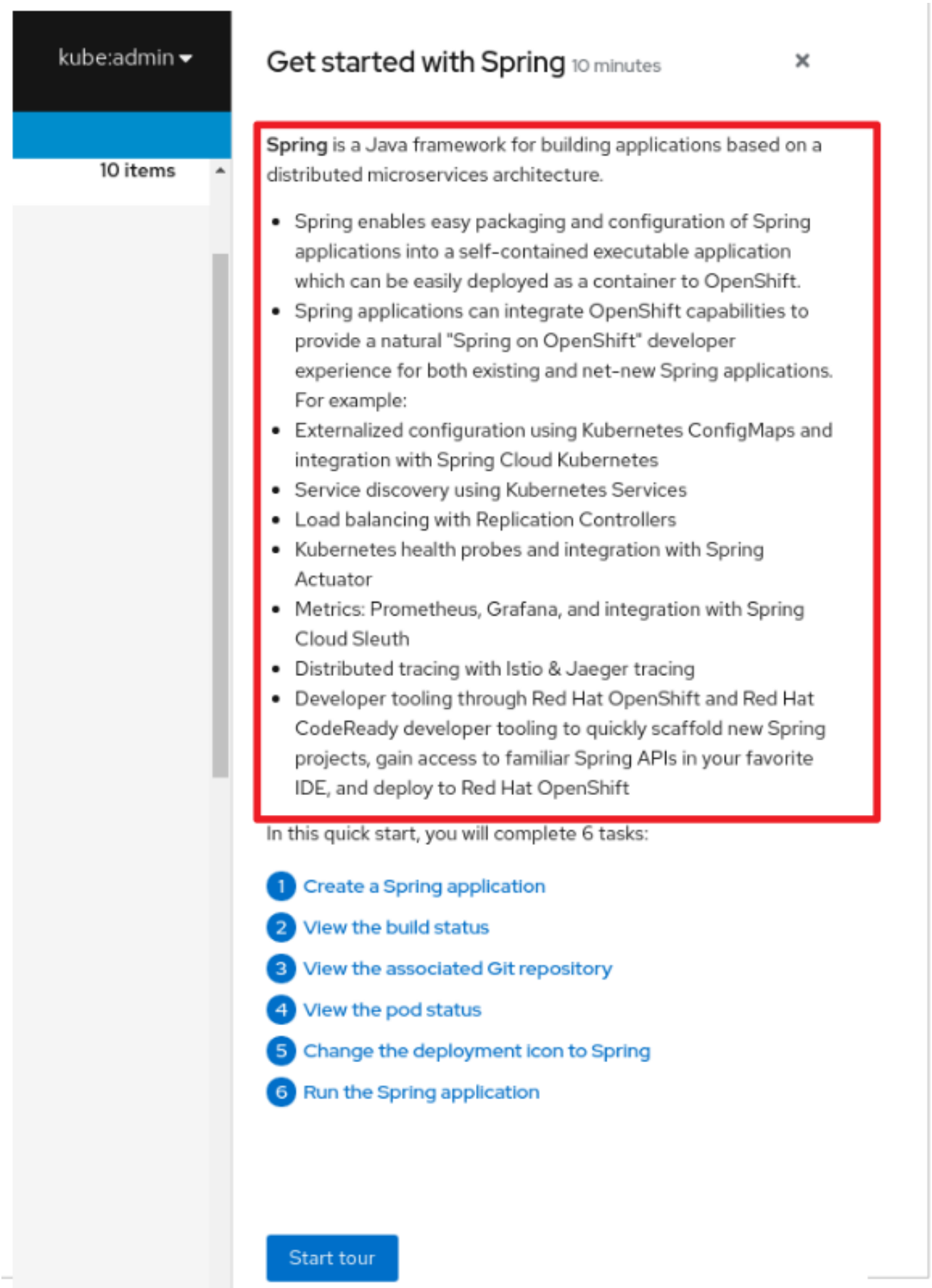
```
...
  introduction: >- 1
    **Spring** is a Java framework for building applications based on a distributed microservices architecture.

    - Spring enables easy packaging and configuration of Spring applications into a self-contained executable application which can be easily deployed as a container to OpenShift.

    - Spring applications can integrate OpenShift capabilities to provide a natural "Spring on OpenShift" developer experience for both existing and net-new Spring applications. For example:

    - Externalized configuration using Kubernetes ConfigMaps and integration with Spring Cloud Kubernetes

    - Service discovery using Kubernetes Services

    - Load balancing with Replication Controllers

    - Kubernetes health probes and integration with Spring Actuator

    - Metrics: Prometheus, Grafana, and integration with Spring Cloud Sleuth

    - Distributed tracing with Istio & Jaeger tracing

    - Developer tooling through Red Hat OpenShift and Red Hat CodeReady developer tooling to
```

```
quickly scaffold new Spring projects, gain access to familiar Spring APIs in your favorite IDE, and
deploy to Red Hat OpenShift
...
```

1 简介介绍了快速启动并列出其中的任务。

在 web 控制台中查看 introduction 元素

点一个快速启动卡后，一个侧边面板滑盘将快速启动并列出它里面的任务。



8.4.3. 为快速启动添加自定义图标

为所有快速启动提供了默认图标。您可以提供自己的自定义图标。

流程

流程

1. 查找您要用作自定义图标的 **.svg** 文件。
2. 使用在线工具将文本转换为 base64。
3. 在 YAML 文件中，添加 **icon: >-**，然后在下一行中包含 **data:image/svg+xml;base64**，后面接的是 base64 转换的输出。例如：

```
icon: >-


ciIHJvbGU9ImltZyIgdmlld.
```

8.4.4. 限制对快速开始的访问

并不是所有的快速开始都应该对所有人可用。YAML 文件的 **accessReviewResources** 部分提供限制对快速启动的访问的能力。

只有有权创建 **HelmChartRepository** 资源的用户，才能访问快速开始，使用以下配置：

```
accessReviewResources:
  - group: helm.openshift.io
    resource: helmchartrepositories
    verb: create
```

只有用户具有列出 Operator 组和软件包清单（因此能够安装 Operator）时允许用户访问快速开始，使用以下配置：

```
accessReviewResources:
  - group: operators.coreos.com
    resource: operatorgroups
    verb: list
  - group: packages.operators.coreos.com
    resource: packagemanifests
    verb: list
```

8.4.5. 连接到其他快速开始

流程

- 在 YAML 文件的 **nextQuickStart** 部分，提供您要链接的快速开始的 **name** 而不是 **displayName**。例如：

```
nextQuickStart:
  - add-healthchecks
```

8.4.6. 支持的标签快速启动

使用这些标签在标记中写入快速开始内容。标记将转换为 HTML。

Tag	描述
'b',	粗体文本。
'img',	嵌入图像。
'i',	斜体文本。
'strike',	带有横线贯穿的文本。
's',	较小的文本
'del',	较小的文本。
'em',	加重文本。
'strong',	重要文本。
'a',	anchor 标签。
'p',	段落文本。
'h1',	1 级标题。
'h2',	2 级标题。
'h3',	3 级标题。
'h4',	4 级标题。
'ul',	一个没有顺序的列表。
'ol',	一个有顺序的列表。
'li',	一个列表项。
'code',	代码文本。
'pre',	预格式化的文本块。
'button',	文本中的一个按钮。

8.5. 快速开始内容指南

8.5.1. Card copy

您可以在快速开始卡上自定义标题及描述，但您无法自定义状态。

- 将您的描述长度限制为一到两句。
- 从操作动词开始，并告知用户其目的。正确的示例：

> Create a serverless application.

8.5.2. 简介

点一个快速启动卡后，一个侧边面板滑盘将快速启动并列出它里面的任务。

- 使您的简介内容更加简洁、明确、易于理解。
- 说明快速开始的目的。用户应在开始之前了解快速开始的目的。
- 为用户提供一个操作，而不是快速开始。
  - **正确的示例**：

> In this quick start, you will deploy a sample application to {product-title}.

  - **不正确的示例**：

> This quick start shows you how to deploy a sample application to {product-title}.

- 根据特性的复杂程度，简介应保持在最多 4 到 5 句。介绍不要太长。
- 列出简介内容后快速开始的任务，并以操作动词启动每个任务。不要指定任务数量，因为每次添加或删除任务时都需要更新副本。
  - **正确的示例**：

> Tasks to complete: Create a serverless application; Connect an event source; Force a new revision

  - **不正确的示例**：

> You will complete these 3 tasks: Creating a serverless application; Connecting an event source; Forcing a new revision

8.5.3. Task steps

用户点 **Start** 后会出现一系列步骤，它们必须执行这些操作来完成快速开始。

在编写任务步骤时遵循这些常规指南：

- 对于按钮和标签使用 "Click"。在选择框、单选按钮和下拉菜单中使用"选择"。
- 使用 "Click" 而不是 "Click on"
  - **正确的示例**：

> Click OK.

- **不正确的示例：**

> Click on the OK button.

- 告诉用户如何在**管理**员和**开发者**视角间如何切换。即使您认为某个用户可能已经处于正确的视角，最好仍为用户提供相应的操作说明，使其确定位于正确的位置。
示例：

> Enter the Developer perspective: In the main navigation, click the dropdown menu and select Developer.
> Enter the Administrator perspective: In the main navigation, click the dropdown menu and select Admin.

- 使用 "Location, action" 结构。告诉用户要做什么前先告诉用户到什么地方。
  - **正确的示例：**

> In the node.js deployment, hover over the icon.

  - **不正确的示例：**

> Hover over the icon in the node.js deployment.

- 保持您的产品术语大写一致。
- 如果您必须指定一个菜单类型或使用列表作为下拉菜单，使用 "dropdown"（一个单词，没有短横线）。
- 明确区分用户动作和产品功能的附加信息。
  - **User action：**

> Change the time range of the dashboard by clicking the dropdown menu and selecting time range.

  - **Additional information：**

> To look at data in a specific time frame, you can change the time range of the dashboard.

- 避免方向性的语言，如 "In the top-right corner, click the icon"。因为当 UI 布局改变时，方向语言就有可能变为不正确。另外，桌面用户对于具有不同屏幕大小的用户来说，方向可能是不正确的。反之，使用它的名称来标识项。
  - **正确的示例：**

> In the navigation menu, click Settings.

  - **不正确的示例：**

> In the left-hand menu, click Settings.

- 不要只使用颜色来标识项，如 "Click the gray circle"。颜色标识符对受限制的用户，尤其是无法识别颜色的用户可能不可用。相反，使用它的名称或复制来标识项目，如 button copy。

- **正确的示例：**

> The success message indicates a connection.

- **不正确的示例：**

> The message with a green icon indicates a connection.

- 一致性地使用第二人称（you）：
  - **正确的示例：**

> Set up your environment.

  - **不正确的示例：**

> Let's set up our environment.

8.5.4. Check your work module

- 用户完成一个步骤后会出现一个 **Check your work** 模块。这个模块提示用户回答"是"或对步骤结果没有问题，这使得他们有机会复核他们的工作。对于这个模块，您只需要写一个是或不需要问题。
  - 如果用户的回答是 **Yes**，会出现一个标记。
  - 如果用户的回答是 **No**， 会出现一个出错信息，其中包含相关文档的链接。然后，用户可以选择返回并再次进行尝试。

8.5.5. 格式化 UI 元素

使用以下指南格式化 UI 元素：

- 按钮、下拉菜单、标签、字段和其他 UI 控制的副本复制：在 UI 中写入副本并加粗体。
- 所有其他 UI 元素-包括页面、窗口和面板名称：在 UI 中写入该文件并加粗体。
- 代码或用户输入的文本：使用 monospaced 字体。
- 提示：如果包含到导航或 masthead 元素的提示，则使用您链接的文本。
- CLI 命令：使用 monospaced 字体。
- 在运行文本时，在命令中使用粗体 monospaced 字体。
- 如果参数或选项是一个变量值，使用 monospaced 字体。
- 参数使用粗体的 monospaced 字体，选项使用 monospaced 字体。

8.6. 其他资源

- 关于声音和音调的要求，请参考 PatternFly's brand voice and tone guidelines。
- 关于其他 UX 内容指南，请参考 PatternFly 的 UX 编写风格指南。