OpenShift Container Platform 4.19

教程

OpenShift Container Platform 入门

Last Updated: 2025-12-10

OpenShift Container Platform 4.19 教程

OpenShift Container Platform 入门

Legal Notice

Abstract

本文档提供了帮助您快速开始使用 OpenShift Container Platform 的信息。这包括 Kubernetes 和 OpenShift Container Platform 中常见术语的定义。它还包含 OpenShift Container Platform Web 控制台的步骤，以及使用命令行界面创建和构建应用程序。

Table of Contents

第 1 章 教程概述 **3**
1.1. 其他学习资源 3

第 2 章 教程：使用 WEB 控制台部署应用程序 **4**
2.1. 先决条件 4
2.2. 创建一个项目 4
2.3. 授予查看权限 5
2.4. 部署前端应用程序 6
2.5. 部署后端应用程序 9
2.6. 部署数据库应用程序 10
2.7. 在网页浏览器中查看应用程序 13

第 3 章 教程：使用 CLI 部署应用程序 **15**
3.1. 先决条件 15
3.2. 创建一个项目 15
3.3. 授予查看权限 16
3.4. 部署前端应用程序 17
3.5. 部署后端应用程序 20
3.6. 部署数据库应用程序 22
3.7. 在网页浏览器中查看应用程序 24

第 4 章 其他实践资源 **26**
4.1. 红帽开发人员学习路径 26
4.2. RED HAT 培训课程 26
4.3. RED HAT CHEAT SHEETS 27

第 1 章 教程概述

您可以使用 OpenShift CLI (**oc**)或 Web 控制台在 OpenShift Container Platform 上部署应用程序的端到端示例。

- 教程：使用 CLI 部署应用程序
- 教程：使用 Web 控制台部署应用程序

1.1. 其他学习资源

要发现 OpenShift Container Platform 的其他教程和实践学习资源，请参阅 其他实践学习。

第 2 章 教程：使用 WEB 控制台部署应用程序

本教程介绍了在 OpenShift Container Platform 上部署服务来支持名为 **national-parks-app** 的应用程序，该应用程序显示了全球的国家公园地图。您将使用 OpenShift Container Platform Web 控制台来完成本教程的操作。

要完成本教程，您将执行以下步骤：

1. 为应用创建一个项目。
   此步骤的目录是将您的应用程序与其他集群用户的工作负载隔离。
2. 授予查看权限。
   此步骤授予与 OpenShift API 进行交互的 **view** 权限，以帮助发现项目内运行的服务和其他资源。
3. 部署前端应用。
   此步骤部署 **parksmap** 前端应用，将其公开给外部，并将其扩展至两个实例。
4. 部署后端应用。
   此步骤部署 **nationalparks** 后端应用程序，并在外部公开。
5. 部署数据库应用。
   此步骤部署 **mongodb-nationalparks** MongoDB 数据库，将数据加载到数据库中，并设置访问数据库所需的凭证。

在完成这些步骤后，您可以在 Web 浏览器中查看国家公园应用程序。

2.1. 先决条件

在开始本教程前，请确保您满足以下先决条件：

- 您可以访问一个用于测试目的的 OpenShift Container Platform 集群。
  如果您的机构没有用于测试的集群，您可以请求访问 Developer Sandbox 来试用 OpenShift Container Platform。
- 您有适当的权限，如 **cluster-admin** 集群角色，来创建一个项目并在其中创建应用程序。
  如果您没有所需的权限，请联络您的集群管理员。您需要 **self-provisioner** 角色来创建一个项目，以及项目的 **admin** 角色，以修改该项目中的资源。

  如果使用 Developer Sandbox，则会为您创建一个具有所需权限的项目。
- 您已登陆到 OpenShift Container Platform Web 控制台。

2.2. 创建一个项目

通过*项目（project）*，社区用户可以以一个隔离的方式来组织和管理其内容。项目是 OpenShift Container Platform 对 Kubernetes 命名空间的扩展。项目具有额外的功能，使用户自助配置。每个项目都有自己的一组对象、策略、约束和服务帐户的集合。

集群管理员可以允许开发人员创建自己的项目。在大多数情况下，您可以自动获得对自己的项目的访问权限。管理员可以根据需要授予其他项目的访问权限。

此流程创建一个名为 **user-get-started** 的新项目。在本教程的其余部分中，您将使用此项目。

重要

如果您使用 Developer Sandbox 完成本教程，请跳过这个过程。已为您创建了一个项目。

先决条件

- 已登陆到 OpenShift Container Platform Web 控制台。

流程

1. 浏览至 **Home → Project**。
2. 点击 **Create Project**。
3. 在 **Name** 字段中输入 **user-getting-started**。
4. 点 **Create**。

其他资源

- 使用 Web 控制台查看项目

2.3. 授予查看权限

OpenShift Container Platform 会在每个项目中自动创建多个服务帐户。**default** 服务帐户负责运行 pod。OpenShift Container Platform 使用并将此服务帐户注入到启动的每个 pod 中。

默认情况下，**default** 服务帐户具有与 OpenShift API 交互的有限权限。

根据应用的要求，您必须将 **view** 角色分配给 **default** 服务帐户，以允许它与 OpenShift API 通信，以了解项目中的 pod、服务和资源。

先决条件

- 有 **cluster-admin** 或项目级别的 **admin** 权限。

流程

1. 进入到 **User Management → RoleBindings**。
2. 点 **Create binding**。
3. 在 **Name** 字段中，输入 **sa-user-account**。
4. 在 **Namespace** 字段中，搜索并选择 **user-getting-started**。

重要

如果您使用其他项目，请选择项目的名称。

5. 在 **Role name** 字段中，搜索并选择 **view**。
6. 在 **Subject** 下，选择 **ServiceAccount**。

7. 在 Subject namespace 字段中，搜索并选择 **user-getting-started**。

重要

如果您使用其他项目，请选择项目的名称。

8. 在 Subject name 字段中，输入 **default**。
9. 点 Create。

其他资源

- RBAC 概述

2.4. 部署前端应用程序

在 OpenShift Container Platform 中部署应用程序的最简单方法是运行一个提供的容器镜像。

以下流程会部署 **parksmap**，它是 **national-parks-app** 应用的前端组件。Web 应用显示全球国家公园位置的交互式地图。

流程

1. 在右上角的 Quick create ( )菜单中点 Container images。
2. 选择 Image name from external registry 并输入 **quay.io/openshiftroadshow/parksmap:latest**。
3. 滚动到 General 部分。
4. 在 Application name 字段中，输入 **national-parks-app**。
5. 在 Name 字段中，确保值是 **parksmap**。
6. 滚动到 Deploy 部分。
7. 在 Resource type 字段中，确保选择了 Deployment。
8. 在 Advanced options 部分中，确保选择了 Create a route。
   默认情况下，在 OpenShift Container Platform 上运行的服务无法从外部访问。您必须选择这个选项来创建一个路由，以便外部客户端可以访问您的服务。
9. 点 Labels 超链接。
   应用程序代码需要设置某些标签。
10. 在文本区中添加以下标签，并在每个键/值对后按 Enter 键：
    - **app=national-parks-app**
    - **component=parksmap**
    - **role=frontend**
11. 点 Create。

您会被重定向到 Topology 页面，您可以在其中查看 **national-parks-app** 应用程序中的 **parksmap** 部署。

其他资源

- 查看应用程序拓扑

2.4.1. 查看 pod 详情

OpenShift Container Platform 使用 Kubernetes 的 *pod* 概念，它是共同部署在同一主机上的一个或多个容器，也是可被定义、部署和管理的最小计算单元。对容器而言，Pod 大致相当于一个机器实例（物理或虚拟）。

通过 **Overview** 面板，您可以访问 **parksmap** 部署的许多功能。**Details** 和 **Resources** 选项卡允许您扩展应用程序 pod，并检查构建、服务和路由的状态。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

1. 进入到 **Workloads → Topology**。
2. 点 **national-parks-app** 应用中的 **parksmap** 部署。

   图 2.1. Parksmap 部署

   这会打开一个概述面板，其中包含以下标签页：

   - **Details** ：查看部署的详细信息，编辑设置，扩展您的部署。
   - **Resources** ：查看与部署关联的 pod、服务和路由的详情。
   - **Observe** ：查看部署的指标和事件。
3. 要查看 pod 的日志，请选择 **Resources** 选项卡，再点 **parksmap** pod 旁边的 **View logs**。

其他资源

- 与应用程序和组件交互
- 扩展应用程序 Pod 以及检查构建和路由

- 用于 Topology 视图的标签和注解

2.4.2. 扩展应用程序

在 Kubernetes 中，**Deployment** 对象定义了应用的部署方式。在大多数情况下，当部署应用程序时，OpenShift Container Platform 会为您创建 **Pod**、**Service**、**ReplicaSet** 和 **Deployment** 资源。

当您部署 **parksmap** 镜像时，会创建一个部署资源。在本例中，只部署了一个 pod。您可能需要扩展应用程序，以便满足用户需求或确保应用程序始终运行（即使出现一个 pod 停机的情况）。

以下流程将 **parksmap** 部署扩展到使用两个实例。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

1. 进入到 **Workloads → Topology**，然后点 **parksmap** 部署。
2. 选择 **Details** 选项卡。
3. 使用向上箭头将容器集扩展到两个实例。

图 2.2. 扩展应用程序

提示

您可以使用下箭头将部署缩减为一个 pod 实例。

其他资源

- 扩展集群的建议实践

2.5. 部署后端应用程序

以下流程部署 **nationalparks**，这是 **national-parks-app** 应用的后端组件。Python 应用程序针对 MongoDB 数据库执行 2D geo-spatial 查询，以定位和返回世界上的所有国家公园的信息。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

1. 在右上角的 **Quick create** ()菜单，点 **Import from Git**。
2. 在 **Git Repo URL** 字段中，输入 **https://github.com/openshift-roadshow/nationalparks-py.git**。
   构建器镜像会被自动检测到，但导入策略默认为 Dockerfile，而不是 Python。
3. 更改导入策略：
   a. 点 **Edit Import Strategy**。
   b. 选择 **Builder Image**。
   c. 选择 **Python**。
4. 滚动到 **General** 部分。
5. 在 **Application** 字段中，确保值为 **national-parks-app**。
6. 在 **Name** 字段中输入 **nationalparks**。
7. 滚动到 **Deploy** 部分。
8. 在 **Resource type** 字段中，确保选择了 **Deployment**。
9. 在 **Advanced options** 部分中，确保选择了 **Create a route**。
   默认情况下，在 OpenShift Container Platform 上运行的服务无法从外部访问。您必须选择这个选项来创建一个路由，以便外部客户端可以访问您的服务。
10. 点 **Labels** 超链接。
    应用程序代码需要设置某些标签。
11. 在文本区中添加以下标签，并在每个键/值对后按 Enter 键：
    - **app=national-parks-app**
    - **component=nationalparks**
    - **role=backend**
    - **type=parksmap-backend**
12. 点 **Create**。

您将被重定向到 **Topology** 页，您可以在其中看到 **national-parks-app** 应用中的 **nationalparks** 部署。

验证

1. 进入到 **Workloads → Topology**。
2. 点 **national-parks-app** 应用中的 **nationalparks** 部署。
3. 单击 **Resources** 选项卡。
   等待构建成功完成。

其他资源

- 在应用程序中添加服务
- 从 Git 导入代码库来创建应用程序

2.6. 部署数据库应用程序

以下流程部署 **mongodb-nationalparks**，它是一个 MongoDB 数据库，它将保存国家公园位置信息。

先决条件

- 您已部署了 **parksmap** 前端应用程序。
- 您已部署了 **nationalparks** 后端应用程序。

流程

1. 在右上角的 **Quick create** ( )菜单中点 **Container images**。
2. 选择 **Image name from external registry** 并输入 **registry.redhat.io/rhmap47/mongodb**。
3. 在 **Runtime icon** 字段中，搜索并选择 **mongodb**。
4. 滚动到 **General** 部分。
5. 在 **Application name** 字段中，输入 **national-parks-app**。
6. 在 **Name** 字段中输入 **mongodb-nationalparks**。
7. 滚动到 **Deploy** 部分。
8. 在 **Resource type** 字段中，确保选择了 **Deployment**。
9. 点 **Show advanced Deployment option**。
10. 在 **Environment variables (runtime only)** 下，添加以下名称和值：

    表 2.1. 环境变量名称和值

Name	value
MONGODB_USER	**mongodb**
MONGODB_PASSWORD	**mongodb**

Name	value
MONGODB_DATABASE	**mongodb**
MONGODB_ADMIN_PASSWORD	**mongodb**

提示

点 Add value 来添加额外的环境变量。

11. 在 Advanced options 部分中，清除 Create a route。
数据库应用不需要从外部访问，因此不需要路由。

12. 点 Create。

您将被重定向到 Topology 页，您可以在其中查看 **national-parks-app** 应用程序中的 **mongodb-nationalparks** 部署。

2.6.1. 通过创建 secret 提供对数据库的访问

nationalparks 应用程序需要信息，如数据库名称、用户名和密码来访问 MongoDB 数据库。但是，由于此信息是敏感的，因此不应将这些信息直接存储在 pod 中。

您可以使用 *secret* 存储敏感信息，并与工作负载共享该 secret。

Secret 对象提供了一种机制来保存敏感信息，如密码、OpenShift Container Platform 客户端配置文件和私有源存储库凭证等。secret 将敏感内容与 Pod 分离。您可以使用卷插件或作为环境变量传递 secret 来将 secret 挂载到容器中。然后，系统可以使用 secret 为 pod 提供敏感信息。

以下流程创建 **nationalparks-mongodb-parameters** secret，并将其挂载到 **nationalparks** 工作负载。

先决条件

- 您已部署了 **nationalparks** 后端应用程序。
- 您已部署了 **mongodb-nationalparks** 数据库应用程序。

流程

1. 导航到 Workloads → Secrets。
2. 点 Create → Key/value secret。
3. 在 Secret name 字段中，输入 **nationalparks-mongodb-parameters**。
4. 输入 Key 和 Value 的以下值：

表 2.2. Secret 键和值

键	value
DATABASE_SERVICE_NAME	**mongodb-nationalparks**
MONGODB_USER	**mongodb**
MONGODB_PASSWORD	**mongodb**
MONGODB_DATABASE	**mongodb**
MONGODB_ADMIN_PASSWORD	**mongodb**

提示

点 Add key/value 来添加每个额外的键/值对。

5. 点 Create。
6. 点 Add Secret to workload。
7. 从 Add this secret to workload列表中，选择 **nationalparks**。
8. 点击 Save。

这个配置的更改会触发一个新的 **nationalparks** 部署推出部署，并正确注入环境变量。

其他资源

- 了解 secret

2.6.2. 将数据加载到数据库中

部署 **mongodb-nationalparks** 数据库后，您可以将国家公园位置信息加载到数据库中。

先决条件

- 您已部署了 **nationalparks** 后端应用程序。
- 您已部署了 **mongodb-nationalparks** 数据库应用程序。

流程

1. 进入到 Workloads → Topology。
2. 点 **nationalparks** 部署，再选择 Resources 选项卡。
3. 从您的路由复制 Location URL。
4. 将 URL 复制并粘贴到您的网页浏览器中，并在 URL 的末尾添加以下内容：

```
/ws/data/load
```

例如：

```
https://nationalparks-user-getting-started.apps.cluster.example.com/ws/data/load
```

输出示例

```
Items inserted in database: 2893
```

2.7. 在网页浏览器中查看应用程序

在部署了必要的应用程序并将数据加载到数据库后，您现在可以通过浏览器查看国家公园应用程序。

您可以通过打开前端应用程序的 URL 来访问应用程序。

先决条件

- 您已部署了 **parksmap** 前端应用程序。
- 您已部署了 **nationalparks** 后端应用程序。
- 您已部署了 **mongodb-nationalparks** 数据库应用程序。
- 您已将数据加载到 **mongodb-nationalparks** 数据库中。

流程

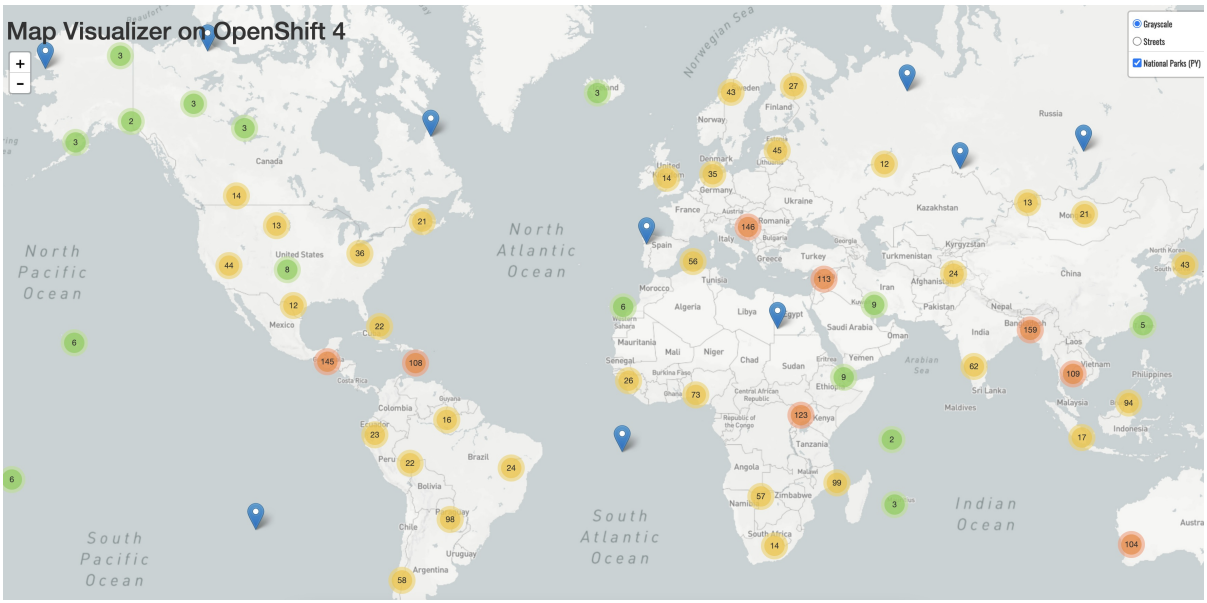
1. 进入到 **Workloads → Topology**。
2. 点 **parksmap** 部署中的 **Open URL** 链接。

   图 2.3. 跨世界的国家公园

3. 验证您的 Web 浏览器是否显示了全球国家公园地图。

图 2.4. 跨世界的国家公园



如果您允许应用程序访问您的位置信息，则地图会以您的位置为中心。

第 3 章 教程：使用 CLI 部署应用程序

本教程介绍了在 OpenShift Container Platform 上部署服务来支持名为 **national-parks-app** 的应用程序，该应用程序显示了全球的国家公园地图。您将使用 OpenShift CLI (**oc**)完成本教程。

要完成本教程，您将执行以下步骤：

1. 为应用创建一个项目。
   此步骤的目录是将您的应用程序与其他集群用户的工作负载隔离。
2. 授予查看权限。
   此步骤授予与 OpenShift API 进行交互的 **view** 权限，以帮助发现项目内运行的服务和其他资源。
3. 部署前端应用。
   此步骤部署 **parksmap** 前端应用，将其公开给外部，并将其扩展至两个实例。
4. 部署后端应用。
   此步骤部署 **nationalparks** 后端应用程序，并在外部公开。
5. 部署数据库应用。
   此步骤部署 **mongodb-nationalparks** MongoDB 数据库，将数据加载到数据库中，并设置访问数据库所需的凭证。

完成这些步骤后，您可以在 Web 浏览器中查看国家公园应用程序。

3.1. 先决条件

在开始本教程前，请确保您满足以下先决条件：

- 已安装 OpenShift CLI (**oc**)。
- 您可以访问一个用于测试目的的 OpenShift Container Platform 集群。
  如果您的机构没有用于测试的集群，您可以请求访问 Developer Sandbox 来试用 OpenShift Container Platform。
- 您有适当的权限，如 **cluster-admin** 集群角色，来创建一个项目并在其中创建应用程序。
  如果您没有所需的权限，请联络您的集群管理员。您需要 **self-provisioner** 角色来创建一个项目，以及项目的 **admin** 角色，以修改该项目中的资源。

  如果使用 Developer Sandbox，则会为您创建一个具有所需权限的项目。
- 已使用 OpenShift CLI (**oc**)登录到集群。

3.2. 创建一个项目

通过*项目（project）*，社区用户可以以一个隔离的方式来组织和管理其内容。项目是 OpenShift Container Platform 对 Kubernetes 命名空间的扩展。项目具有额外的功能，使用户自助配置。每个项目都有自己的一组对象、策略、约束和服务帐户的集合。

集群管理员可以允许开发人员创建自己的项目。在大多数情况下，您可以自动获得对自己的项目的访问权限。管理员可以根据需要授予其他项目的访问权限。

此流程创建一个名为 **user-get-started** 的新项目。在本教程的其余部分中，您将使用此项目。

重要

如果您使用 Developer Sandbox 完成本教程，请跳过这个过程。已为您创建了一个项目。

先决条件

- 已登陆到 OpenShift CLI (**oc**)。

流程

- 运行以下命令来创建项目：

```
$ oc new-project user-getting-started
```

输出示例

```
Now using project "user-getting-started" on server "https://openshift.example.com:6443".
...
```

其他资源

- oc new-project

3.3. 授予查看权限

OpenShift Container Platform 会在每个项目中自动创建多个服务帐户。**default** 服务帐户负责运行 pod。OpenShift Container Platform 使用并将此服务帐户注入到启动的每个 pod 中。

默认情况下，**default** 服务帐户具有与 OpenShift API 交互的有限权限。

根据应用的要求，您必须将 **view** 角色分配给 **default** 服务帐户，以允许它与 OpenShift API 通信，以了解项目中的 pod、服务和资源。

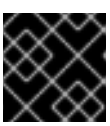
先决条件

- 有访问 OpenShift Container Platform 集群的权限。
- 已安装 OpenShift CLI(**oc**)。
- 有 **cluster-admin** 或项目级别的 **admin** 权限。

流程

- 运行以下命令，将 **view** 角色添加到 **user-get-started** 项目中的 **default** 服务帐户：

```
$ oc adm policy add-role-to-user view -z default -n user-getting-started
```



重要

如果您使用其他项目，请将 **user-getting-started** 替换为项目的名称。

其他资源

- RBAC 概述
- oc adm policy add-role-to-user

3.4. 部署前端应用程序

在 OpenShift Container Platform 中部署应用程序的最简单方法是运行一个提供的容器镜像。

以下流程会部署 **parksmap**，它是 **national-parks-app** 应用的前端组件。Web 应用显示全球国家公园位置的交互式地图。

先决条件

- 有访问 OpenShift Container Platform 集群的权限。
- 已安装 OpenShift CLI(**oc**)。

流程

- 运行以下命令部署 **parksmap** 应用程序：

```
$ oc new-app quay.io/openshiftroadshow/parksmap:latest --name=parksmap -l
'app=national-parks-app,component=parksmap,role=frontend,app.kubernetes.io/part-
of=national-parks-app'
```

输出示例

```
--> Found container image 0c2f55f (4 years old) from quay.io for
"quay.io/openshiftroadshow/parksmap:latest"

    * An image stream tag will be created as "parksmap:latest" that will track this image

--> Creating resources with label app=national-parks-app,app.kubernetes.io/part-of=national-
parks-app,component=parksmap,role=frontend ...
    imagestream.image.openshift.io "parksmap" created
    deployment.apps "parksmap" created
    service "parksmap" created
--> Success
    Application is not exposed. You can expose services to the outside world by executing one
or more of the commands below:
     'oc expose service/parksmap'
    Run 'oc status' to view your app.
```

其他资源

- oc new-app

3.4.1. 公开前端服务

默认情况下，在 OpenShift Container Platform 上运行的服务无法从外部访问。

要公开您的服务，以便外部客户端可以访问该服务，您可以创建一个 *路由*。**Route** 对象是类似于 Kubernetes **Ingress** 对象的 OpenShift Container Platform 网络资源。默认的 OpenShift Container Platform 路由器(HAProxy)使用传入请求的 HTTP 标头来确定连接的位置。

另外，您可以为路由定义安全性，如 TLS。

先决条件

- 您已部署了 **parksmap** 前端应用程序。
- 有 **cluster-admin** 或项目级别的 **admin** 权限。

流程

- 运行以下命令，创建一个路由来公开 **parksmap** 前端应用程序：

```
$ oc create route edge parksmap --service=parksmap
```

验证

- 运行以下命令验证应用程序路由是否已成功创建：

```
$ oc get route parksmap
```

输出示例

```
NAME        HOST/PORT                                          PATH   SERVICES   PORT
TERMINATION   WILDCARD
parksmap    parksmap-user-getting-started.apps.cluster.example.com             parksmap
8080-tcp    edge          None
```

其他资源

- oc create route edge
- oc get

3.4.2. 查看 pod 详情

OpenShift Container Platform 使用 Kubernetes 的 *pod* 概念，它是共同部署在同一主机上的一个或多个容器，也是可被定义、部署和管理的最小计算单元。对容器而言，Pod 大致相当于一个机器实例（物理或虚拟）。

您可以查看集群中的 pod，并确定这些 pod 和整个集群的健康状态。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

- 运行以下命令，列出当前项目中的所有 pod：

```
$ oc get pods
```

输出示例

```
NAME                  READY   STATUS    RESTARTS   AGE
parksmap-5f9579955-6sng8   1/1     Running   0          77s
```

- 运行以下命令显示 pod 的详情：

```
$ oc describe pod parksmap-5f9579955-6sng8
```

输出示例

```
Name:             parksmap-5f9579955-6sng8
Namespace:        user-getting-started
Priority:         0
Service Account:  default
Node:             ci-ln-fr1rt92-72292-4fzf9-worker-a-g9g7c/10.0.128.4
Start Time:       Wed, 26 Mar 2025 14:03:19 -0400
Labels:           app=national-parks-app
                  app.kubernetes.io/part-of=national-parks-app
                  component=parksmap
                  deployment=parksmap
                  pod-template-hash=848bd4954b
                  role=frontend
...
```

- 运行以下命令，查看 pod 的日志：

```
$ oc logs parksmap-5f9579955-6sng8
```

输出示例

```
...
2025-03-26 18:03:24.774  INFO 1 --- [           main] o.s.m.s.b.SimpleBrokerMessageHandler
: Started.
2025-03-26 18:03:24.798  INFO 1 --- [           main]
s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
2025-03-26 18:03:24.801  INFO 1 --- [           main] c.o.evg.roadshow.ParksMapApplication
: Started ParksMapApplication in 4.053 seconds (JVM running for 4.46)
```

其他资源

- oc describe
- oc get
- 查看 pod
- 查看 pod 日志

3.4.3. 扩展部署

在 Kubernetes 中，**Deployment** 对象定义了应用的部署方式。在大多数情况下，当部署应用程序时，OpenShift Container Platform 会为您创建 **Pod**、**Service**、**ReplicaSet** 和 **Deployment** 资源。

当您部署 **parksmap** 镜像时，会创建一个部署资源。在本例中，只部署了一个 pod。您可能需要扩展应用程序，以便满足用户需求或确保应用程序始终运行（即使出现一个 pod 停机的情况）。

以下流程将 **parksmap** 部署扩展到使用两个实例。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

- 运行以下命令，将部署从一个 pod 实例扩展到两个 pod 实例：

```
$ oc scale --replicas=2 deployment/parksmap
```

输出示例

```
deployment.apps/parksmap scaled
```

验证

- 运行以下命令验证您的部署是否已正确扩展：

```
$ oc get pods
```

输出示例

```
NAME                  READY   STATUS    RESTARTS   AGE
parksmap-5f9579955-6sng8   1/1     Running   0          7m39s
parksmap-5f9579955-8tgft   1/1     Running   0          24s
```

验证是否列出了两个 **parksmap** pod。

提示

要将部署缩减为一个 pod 实例，请将 **1** 传递给 **--replicas** 选项：

```
$ oc scale --replicas=1 deployment/parksmap
```

其他资源

- oc scale

3.5. 部署后端应用程序

以下流程部署 **nationalparks**，这是 **national-parks-app** 应用的后端组件。Python 应用程序针对 MongoDB 数据库执行 2D geo-spatial 查询，以定位和返回世界上的所有国家公园的信息。

先决条件

- 您已部署了 **parksmap** 前端应用程序。

流程

- 运行以下命令来创建 **nationalparks** 后端应用程序：

```
$ oc new-app python~https://github.com/openshift-roadshow/nationalparks-py.git --name
nationalparks -l 'app=national-parks-
app,component=nationalparks,role=backend,app.kubernetes.io/part-of=national-parks-
app,app.kubernetes.io/name=python' --allow-missing-images=true
```

输出示例

```
--> Found image 9531750 (2 weeks old) in image stream "openshift/python" under tag "3.11-
ubi8" for "python"

    Python 3.11
    -----------
...

--> Creating resources with label app=national-parks-
app,app.kubernetes.io/name=python,app.kubernetes.io/part-of=national-parks-
app,component=nationalparks,role=backend ...
    imagestream.image.openshift.io "nationalparks" created
    buildconfig.build.openshift.io "nationalparks" created
    deployment.apps "nationalparks" created
    service "nationalparks" created
--> Success
    Build scheduled, use 'oc logs -f buildconfig/nationalparks' to track its progress.
    Application is not exposed. You can expose services to the outside world by executing one
or more of the commands below:
     'oc expose service/nationalparks'
    Run 'oc status' to view your app.
```

3.5.1. 公开后端服务

与为外部客户端公开前端服务的方式类似，现在您必须通过创建路由来公开后端服务。

先决条件

- 您已部署了 **nationalparks** 后端应用程序。
- 有 **cluster-admin** 或项目级别的 **admin** 权限。

流程

1. 运行以下命令，创建一个路由来公开 **nationalparks** 后端应用程序：

```
$ oc create route edge nationalparks --service=nationalparks
```

2. 运行以下命令标记 **nationalparks** 路由：

```
$ oc label route nationalparks type=parksmap-backend
```

应用程序代码预期 **nationalparks** 路由使用 **type=parksmap-backend** 进行标记。

其他资源

- oc label

3.6. 部署数据库应用程序

以下流程部署 **mongodb-nationalparks**，它是一个 MongoDB 数据库，它将保存国家公园位置信息。

先决条件

- 您已部署了 **parksmap** 前端应用程序。
- 您已部署了 **nationalparks** 后端应用程序。

流程

- 运行以下命令部署 **mongodb-nationalparks** 数据库应用程序：

```
$ oc new-app registry.redhat.io/rhmap47/mongodb --name mongodb-nationalparks -e
MONGODB_USER=mongodb -e MONGODB_PASSWORD=mongodb -e
MONGODB_DATABASE=mongodb -e MONGODB_ADMIN_PASSWORD=mongodb -l
'app.kubernetes.io/part-of=national-parks-app,app.kubernetes.io/name=mongodb'
```

输出示例

```
--> Found container image 7a61087 (12 days old) from quay.io for
"quay.io/mongodb/mongodb-enterprise-server"

    * An image stream tag will be created as "mongodb-nationalparks:latest" that will track this
image

--> Creating resources with label app.kubernetes.io/name=mongodb,app.kubernetes.io/part-
of=national-parks-app ...
    imagestream.image.openshift.io "mongodb-nationalparks" created
    deployment.apps "mongodb-nationalparks" created
    service "mongodb-nationalparks" created
--> Success
    Application is not exposed. You can expose services to the outside world by executing one
or more of the commands below:
     'oc expose service/mongodb-nationalparks'
    Run 'oc status' to view your app.
```

3.6.1. 通过创建 secret 提供对数据库的访问

nationalparks 应用程序需要信息，如数据库名称、用户名和密码来访问 MongoDB 数据库。但是，由于此信息是敏感的，因此不应将这些信息直接存储在 pod 中。

您可以使用 *secret* 存储敏感信息，并与工作负载共享该 secret。

Secret 对象提供了一种机制来保存敏感信息，如密码、OpenShift Container Platform 客户端配置文件和私有源存储库凭证等。secret 将敏感内容与 Pod 分离。您可以使用卷插件或作为环境变量传递 secret 来将 secret 挂载到容器中。然后，系统可以使用 secret 为 pod 提供敏感信息。

以下流程创建 **nationalparks-mongodb-parameters** secret，并将其挂载到 **nationalparks** 工作负载。

先决条件

- 您已部署了 **nationalparks** 后端应用程序。
- 您已部署了 **mongodb-nationalparks** 数据库应用程序。

流程

1. 运行以下命令，使用所需的数据库访问信息创建 secret：

```
$ oc create secret generic nationalparks-mongodb-parameters --from-literal=DATABASE_SERVICE_NAME=mongodb-nationalparks --from-literal=MONGODB_USER=mongodb --from-literal=MONGODB_PASSWORD=mongodb --from-literal=MONGODB_DATABASE=mongodb --from-literal=MONGODB_ADMIN_PASSWORD=mongodb
```

2. 运行以下命令，将环境从 secret 导入到 **nationalparks** 工作负载：

```
$ oc set env --from=secret/nationalparks-mongodb-parameters deploy/nationalparks
```

3. 等待 **nationalparks** 部署推出带有此环境信息的新修订版本。运行以下命令，检查 **nationalparks** 部署的状态：

```
$ oc rollout status deployment nationalparks
```

输出示例

```
deployment "nationalparks" successfully rolled out
```

其他资源

- 了解 secret
- oc create secret generic
- oc set env
- oc rollout status

3.6.2. 将数据加载到数据库中

部署 **mongodb-nationalparks** 数据库后，您可以将国家公园位置信息加载到数据库中。

先决条件

- 您已部署了 **nationalparks** 后端应用程序。

- 您已部署了 **mongodb-nationalparks** 数据库应用程序。

流程

- 运行以下命令载入国家公园数据：

```
$ oc exec $(oc get pods -l component=nationalparks | tail -n 1 | awk '{print $1;}') -- curl -s http://localhost:8080/ws/data/load
```

输出示例

```
"Items inserted in database: 2893"
```

验证

- 运行以下命令验证地图数据是否已正确载入：

```
$ oc exec $(oc get pods -l component=nationalparks | tail -n 1 | awk '{print $1;}') -- curl -s http://localhost:8080/ws/data/all
```

输出示例（修剪）

```
...
, {"id": "Great Zimbabwe", "latitude": "-20.2674635", "longitude": "30.9337986", "name": "Great Zimbabwe"}]
```

其他资源

- oc exec

3.7. 在网页浏览器中查看应用程序

在部署了必要的应用程序并将数据加载到数据库后，您现在可以通过浏览器查看国家公园应用程序。

您可以通过检索前端应用的路由信息来获取应用的 URL。

先决条件

- 您已部署了 **parksmap** 前端应用程序。
- 您已部署了 **nationalparks** 后端应用程序。
- 您已部署了 **mongodb-nationalparks** 数据库应用程序。
- 您已将数据加载到 **mongodb-nationalparks** 数据库中。

流程

1. 运行以下命令，获取您的路由信息以检索您的地图应用程序 URL：

```
$ oc get route parksmap
```

输出示例

```
NAME      HOST/PORT                                        PATH   SERVICES   PORT
TERMINATION   WILDCARD
parksmap   parksmap-user-getting-started.apps.cluster.example.com          parksmap
8080-tcp   edge          None
```

2. 从上面的输出中，复制 **HOST/PORT** 列中的值。

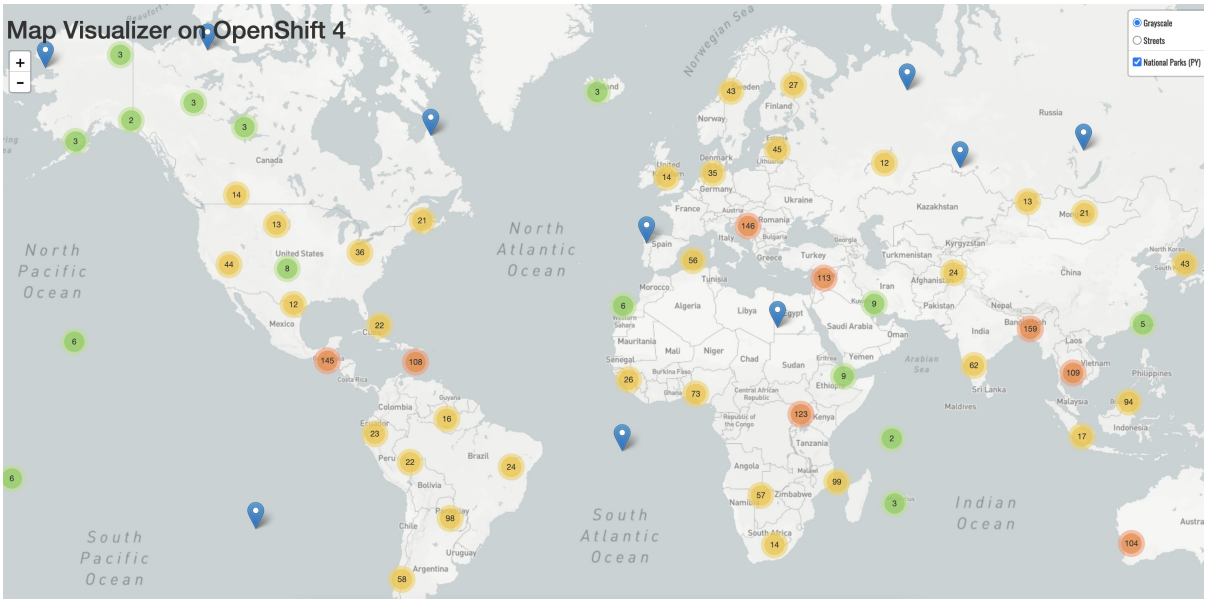
3. 在复制的值前面添加 **https://** 以获取应用程序 URL。这是必要的，因为路由是一个安全路由。

应用程序 URL 示例

```
https://parksmap-user-getting-started.apps.cluster.example.com
```

4. 将此应用程序 URL 粘贴到您的网页浏览器中。您的浏览器应当显示全球的国家公园地图。

图 3.1. 跨世界的国家公园



如果您允许应用程序访问您的位置信息，则地图会以您的位置为中心。

第 4 章 其他实践资源

红帽为管理员和开发人员提供了许多其他学习资源，以便获得 OpenShift Container Platform 的实践经验。

4.1. 红帽开发人员学习路径

Red Hat Developer 程序为开发人员提供了几个学习路径，以便开始使用 OpenShift Container Platform。

下表列出了 OpenShift Container Platform 的几个推荐学习路径：

表 4.1. 红帽开发人员学习路径

学习路径	描述
OpenShift 基础	本学习路径涵盖了基本的 Red Hat OpenShift 概念以及如何通过各种方法创建和部署应用程序。
使用 OpenShift	本学习路径涵盖了管理集群访问、数据库操作和资源管理。
在 OpenShift 中开发应用程序	本学习路径涵盖了从源代码和镜像部署应用程序，并使用 Node.js 进行开发。
如何在 OpenShift 中部署全堆栈 JavaScript 应用程序	本学习路径涵盖了如何在 OpenShift Container Platform 集群中部署全堆栈 JavaScript 应用程序。
使用 PVC 在 Red Hat OpenShift 中存储持久性数据	本学习路径涵盖了如何在 OpenShift Container Platform 中为持久性存储创建和使用持久性卷声明 (PVC)。

有关 OpenShift Container Platform 可用的红帽开发人员学习路径的完整列表，请参阅 OpenShift 和 Kubernetes 学习。

4.2. RED HAT 培训课程

Red Hat 培训提供各种在线课程，包括免费和付费课程，以帮助您学习红帽 OpenShift 及相关技术。

下表列出了针对开发人员和管理员的 OpenShift Container Platform 的一些推荐培训课程：

表 4.2. 面向开发人员的 Red Hat 培训课程

教程	描述
DO101: OpenShift 应用程序简介	此课程帮助开发人员在 OpenShift Container Platform 中部署、扩展应用程序并进行故障排除。
DO188: Red Hat OpenShift 开发一：使用 Podman 的容器简介	此课程帮助开发人员使用 Podman 和 OpenShift Container Platform 构建、运行和管理容器。

教程	描述
DO288：Red Hat OpenShift Developer II：构建和部署云原生应用程序	此课程可帮助开发人员在 OpenShift Container Platform 集群中设计、构建和部署容器化软件应用程序。

表 4.3. 针对管理员的 Red Hat 培训课程

教程	描述
DO180: Red Hat OpenShift Administration I: Operating a Production Cluster	此课程帮助集群管理员了解如何管理 OpenShift Container Platform 集群，并与开发人员合作支持应用程序工作负载。
DO280：Red Hat OpenShift 管理 II：配置生产集群	此课程帮助集群管理员了解如何配置安全功能、管理 Operator 并执行集群更新。
DO322: Red Hat OpenShift Installation Lab	此课程帮助集群管理员在各种环境中学习如何安装 OpenShift Container Platform 集群。

有关可用课程的完整列表，请参阅 Red Hat 培训和认证。您还可以参与评估，获得针对从何处开始学习的建议。

4.3. RED HAT CHEAT SHEETS

红帽提供了几个 cheat sheet，它提供常见 OpenShift CLI (**oc**)命令的快速参考来使用 OpenShift Container Platform。

下表列出了 OpenShift Container Platform 几个推荐的 cheat sheets：

表 4.4. Red Hat cheat sheets

cheat sheet	描述
Red Hat OpenShift Cheat Sheet	此 cheat sheet 提供了许多 OpenShift CLI (**oc**)命令来管理应用程序的生命周期。
OpenShift 命令行基本知识表	此 cheat sheet 提供了几个基本的 OpenShift CLI (**oc**)命令，如创建应用程序、调试和编辑部署。

有关可用 cheat sheet 的完整列表，请参阅 Red Hat Developer cheat sheets。