

# 04-Lab1 部署和部署策略实验

## 情景

在这个实验室中，你将提供不同部署策略的演示。你描绘了一个想要部署应用程序、扩展它并重新配置它的开发者。你还要演示部署类型之间的差异，因为它们会影响构成部署的Pod。

你要在Red Hat® OpenShift®容器平台界面上采取**行动**，并**解释**活动、产品功能、资源和概念。

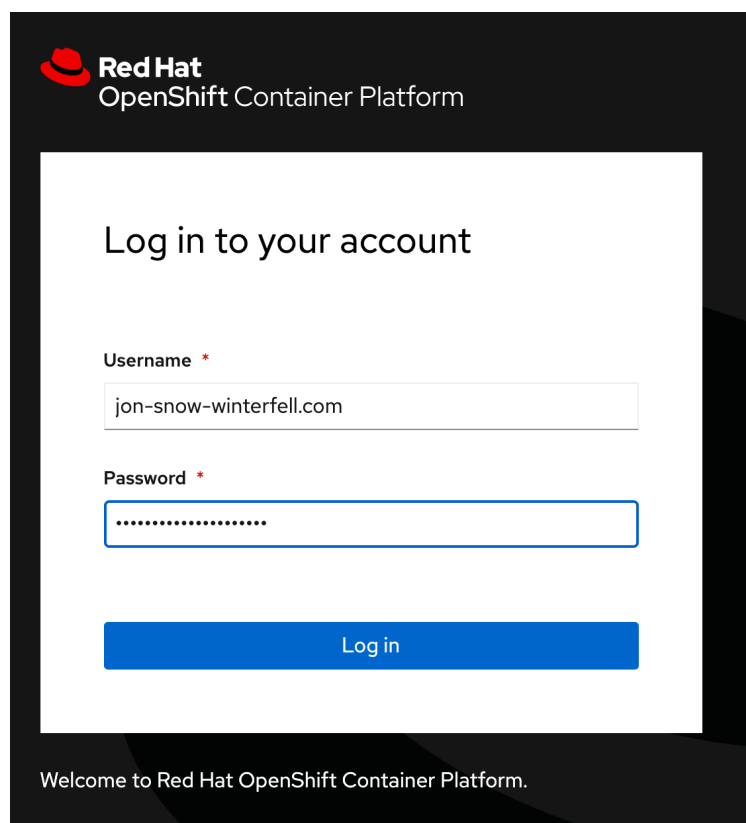
## 目标

- 部署一个应用程序
- 扩展应用程序
- 用一个新的环境变量重新部署应用程序
- 查看部署情况
- 改变部署类型
- 重新部署应用程序

## 1. Demonstrate Authentication to OpenShift

**提示：** ip this section if you already executed the steps in previous demonstrations.

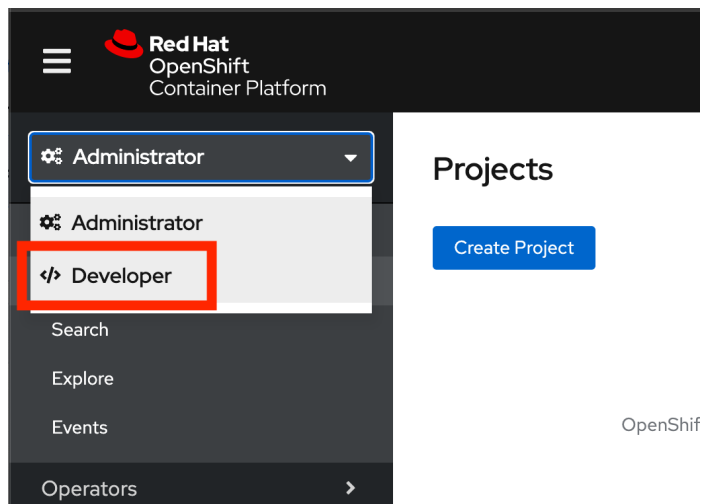
1. **行动：** 使用你的登录凭证登录到OpenShift容器平台网络控制台。



- **解释：** 你会被带到**项目**页面。OpenShift容器平台网络控制台的默认视图是管理员视角。

## 2. 演示开发者视图

1. **解释：** OpenShift容器平台网络控制台提供了两个视图：管理员视图和开发者视图。开发者视图提供了针对开发者用例的工作流程。
2. **行动：** 使用视图切换器来切换到开发者视图。显示带有创建应用程序选项的拓扑结构视图。



### 3. 演示新项目的创建

**解释：**项目允许社区的用户组织和管理他们的内容，与其他社区隔离。项目是OpenShift对Kubernetes命名空间的扩展，具有额外的功能，可以实现用户自我配置。在绝大多数情况下，它们是可以互换的。

1. **行动：**点击**项目**下拉菜单，看到所有可用项目的列表。选择**创建项目**。

◦ **解释：**什么是项目，以及不同的项目如何有不同的用户权限和配额附加到他们身上。

2. **行动：**填写**名称**、**显示名称**和**描述**字段，如下所示。

**注意：**确保用一个唯一的标识符替换 `GUID`，例如你的客户名称或你从OPENTLC收到的四字标识符。项目名称在OpenShift中必须是唯一的。

◦ **名称：** `GUID-deployments-demo`。

◦ **显示名称：** `Demonstrating Deployment Strategies`。

◦ **描述：** `This is the project to host the deployments strategy demonstration`。

### 4. 使用S2I演示应用程序的部署

#### 4.1. 演示 "从Git "部署

**解释：**OpenShift已经创建了你的项目，并将你的浏览器转发到**拓扑结构**页面。你的项目中没有工作负载可以显示在拓扑页面上，所以出现了部署工作负载的选项。

1. **行动：**选择**从Git**，查看**从git导入**页面。

2. **行动：**在**Git Repo URL**框中输入 `https://github.com/StefanoPicozzi/cotd2`。

3. 解释：OpenShift会检测Git repo的编程语言，并检测合适的构建器镜像。

## Import from git

### Git


**Git Repo URL \***  
  
Validated


> [Show Advanced Git Options](#)


### Builder


#### Builder Image \*


✓ **Builder image(s) detected.**  
Recommended builder images are represented by ★ icon.


  
Perl


  
PHP


  
Nginx


  
Modern Webapp


  
Httpd


  
.NET Core

  
Go

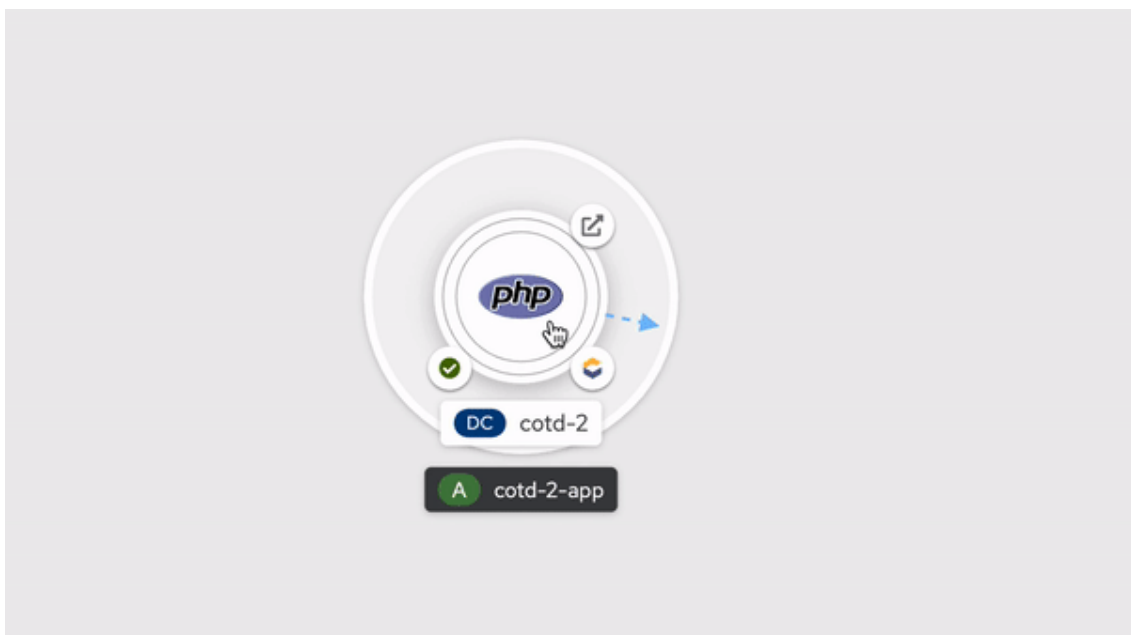
  
Ruby

  
Python

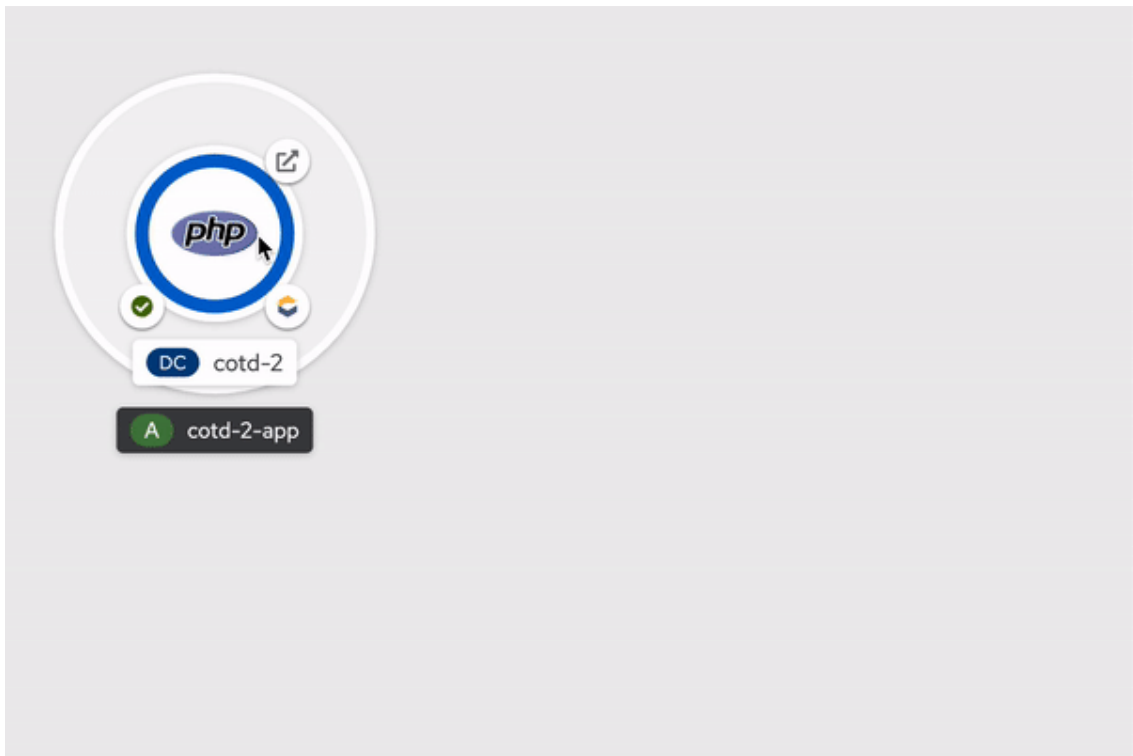
  
Java

  
Node.js

4. 行动：重点！ 点击 资源 → 部署配置.apps/Deployment目前不支持部署策略。
5. 行动：勾选创建通往应用程序的路由复选框。
6. 行动：点击创建，打开拓扑结构视图页面。
7. 解释：请耐心等待 PHP cotd2 应用程序将建立。



8. 行动：单击圆圈，然后出现Deployment Config Overview面板。



## 4.2. 演示设置环境变量

1. **行动：**单击**环境**标签。

◦ **解释：**以下**环境变量**部分内容。

- 环境变量是如何在OpenShift中使用的，以及如何轻松地设置环境变量，如用户名、数据库名称等来修改部署。

2. **行动：**输入一个名为 `SELECTOR` 的新环境变量，值为 `cities`。

Project: guid-deployments-demo ▼

Deployment Configs > Deployment Config Details

**DC cotd-2** Actions ▼

Details **YAML** Replication Controllers Pods **Environment** Events

Container: **C cotd-2** ▼

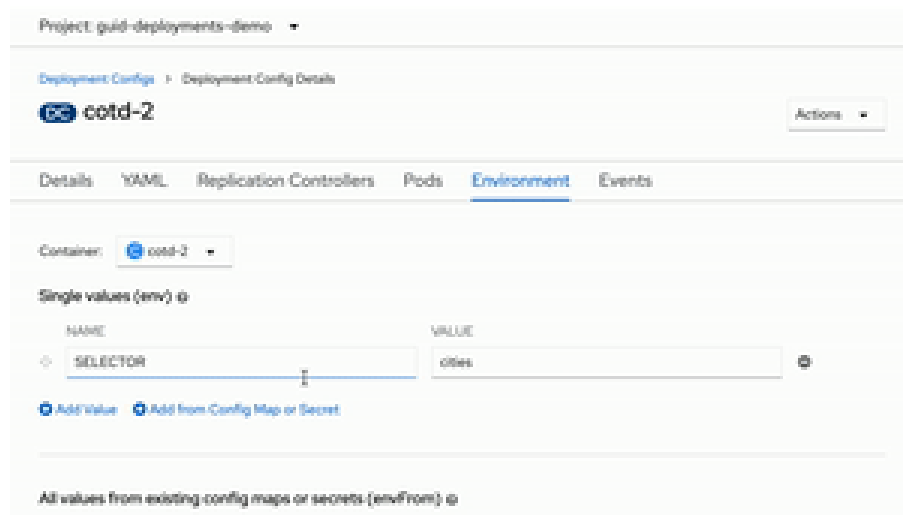
Single values (env) ⓘ

NAME	VALUE
<input type="text" value="SELECTOR"/>	<input type="text" value="cities"/>

[+ Add Value](#) [+ Add from Config Map or Secret](#)

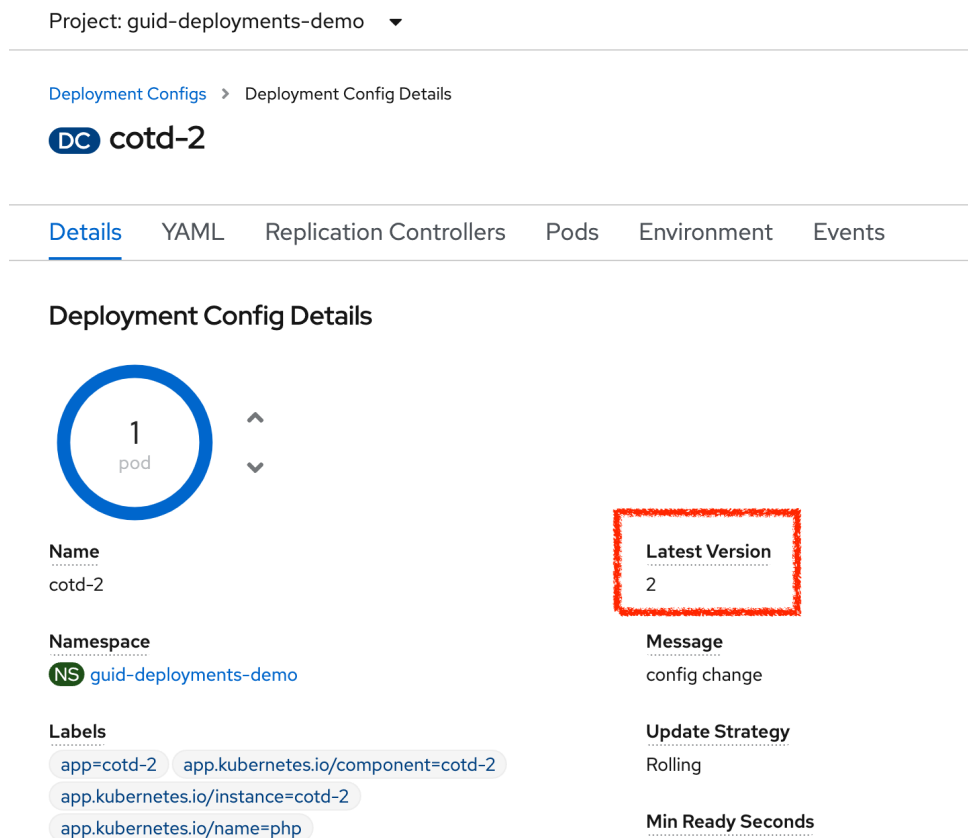
3. **行动：**点击**保存**。

4. **行动：**单击**细节**标签



5. 解释：更新的环境变量如何创建新版本的 cotd DeploymentConfig。

- DeploymentConfig部署了包含更新环境变量的替换Pod。
- **LATEST VERSION**等于`2`。
  - 解释：应用程序被部署为显示你在环境变量中定义的 cities。



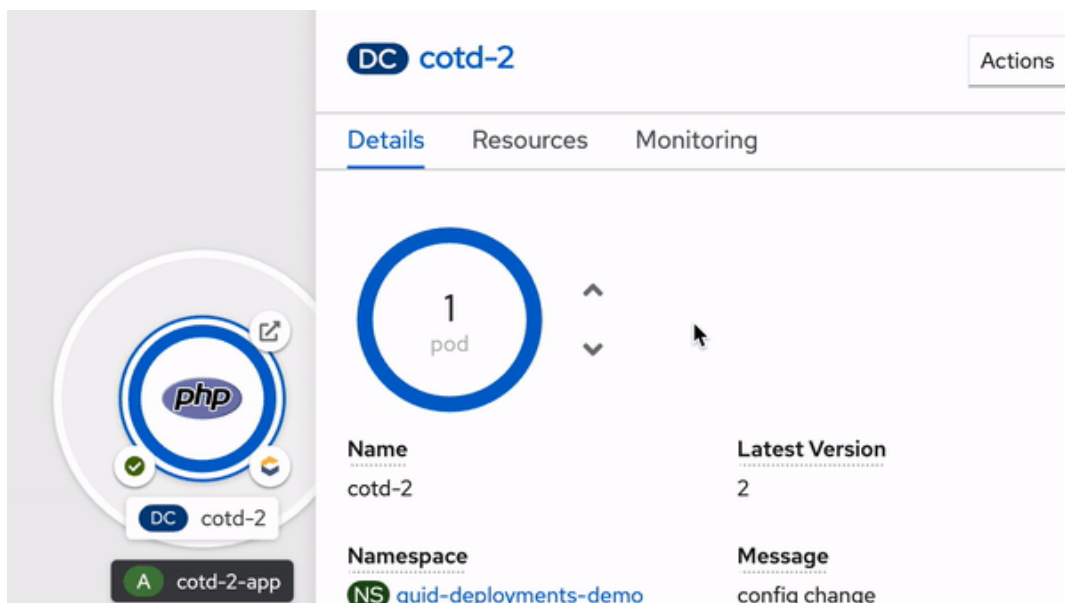
### 4.3. 演示环境变量对应用程序的影响

1. 行动：点击左边的**拓扑**菜单，在圆圈中点击，再次点击PHP标志，显示侧边面板。
2. 行动：点击**资源**标签并向下滚动到**路由**。
3. 解释：路由是为 GUID-deployments-demo 命名空间创建的，其值在**LOCATION**字段中。
4. 行动：点击**LOCATION**下的路由URL，一个新的标签打开，显示城市的镜像，与你之前设置的环境变量相匹配。

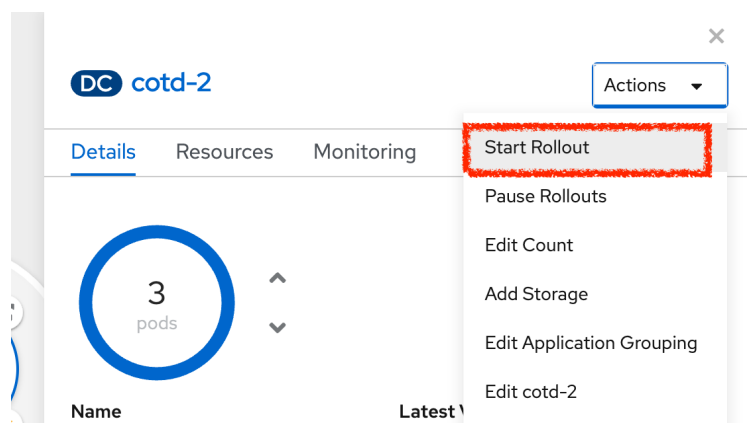
## 5. 演示应用程序的滚动部署

1. 行动：通过返回Web控制台，点击**Details**，然后点击向上的箭头，将你的应用程序扩展到三个 Pod。

- **解释：**你可以看到OpenShift随着 `available` 数量的增加和 `unavailable` 数量的减少而扩大了Pod的规模。



2. **行动：**在DC ctd-2概览面板中，单击**Actions** → **Start Rollout**。



- **解释：**这个页面显示你的部署配置。
  - **最新版本**的递增。
  - **信息**现在是 "手动更改"。
  - 指出部署被设置为 `Rolling`，并解释OpenShift会重复部署一个新的Pod副本并删除一个旧的部署Pod副本，直到新的部署达到所需的副本数量，而旧的部署为零。
- **解释：**现在正在使用 `Rolling` 部署策略进行部署的以下方面。
  - **部署配置概述**页的**COUNTS**部分。
  - 为新的部署创建了一个新的Pod，在进行健康检查测试后，旧的部署Pod被销毁。
  - OpenShift会继续增加新部署的规模，每次减少旧部署的Pod，更新页面的**可用**和**不可用**的Pod。
  - 使用这种部署策略有利于减少应用程序的停机时间，因为新的和旧的部署可以并存一段时间。

## 6. 演示通过配置变更触发部署

1. **行动：**单击 `DC ctd-2`，打开**部署配置详细信息**页面。
2. **行动：**选择**环境**标签，将 `SELECTOR` 环境变量的值改为 `cats`，并点击**保存**。

Project: guid-deployments-demo ▾

Deployment Configs > Deployment Config Details

**DC** ctd-2 Actions ▾

Details   YAML   Replication Controllers   Pods   Environment   Events

Container: C ctd-2 ▾

Single values (env) ⓘ

NAME	VALUE
<div><div>+</div>SELECTOR</div>	<div>cats</div> <div>⊖</div>

[+ Add Value](#) [+ Add from Config Map or Secret](#)

3. **行动**：点击**细节**标签，看到一个新的部署被触发了。
4. **行动**：一旦部署完成，通过刷新你先前打开的应用程序的网页，显示新的部署显示的是 `cats` 而不是 `cities`。

## 7. 演示改变部署策略

1. **行动**：单击**YAML**标签。
  - **解释**：你也可以在这个页面上添加存储，设置资源限制，添加一个自动调节器，并编辑健康检查。
2. **行动**：编辑部署配置YAML，删除 `spec.strategy` 下的所有行（从 `type: Rolling` 到 `triggers`），添加 `type: Recreate` 配置。











```
17   app: ctd-2
18   app.kubernetes.io/component: ctd-2
19   app.kubernetes.io/instance: ctd-2
20   app.kubernetes.io/name: php
21   app.kubernetes.io/part-of: ctd-2-app
22   app.openshift.io/runtime: php
23   app.openshift.io/runtime-version: '7.3'
24 spec:
25   strategy:
26     type: Rolling
27     rollingParams:
28       updatePeriodSeconds: 1
29       intervalSeconds: 1
30       timeoutSeconds: 600
31       maxUnavailable: 25%
32       maxSurge: 25%
33     resources: {}
34     activeDeadlineSeconds: 21600
35   triggers:
36   - type: ImageChange
37     imageChangeParams:
```

3. **行动**：点击**保存**。
  - **解释**：这种配置变化（部署规格变化）不会触发自动推出。只有对Pod规格的更改才会触发自动展开。
4. **行动**：转到**Pods**标签，通过点击**Actions** → **Start Rollout**触发新的部署。

Filter by name

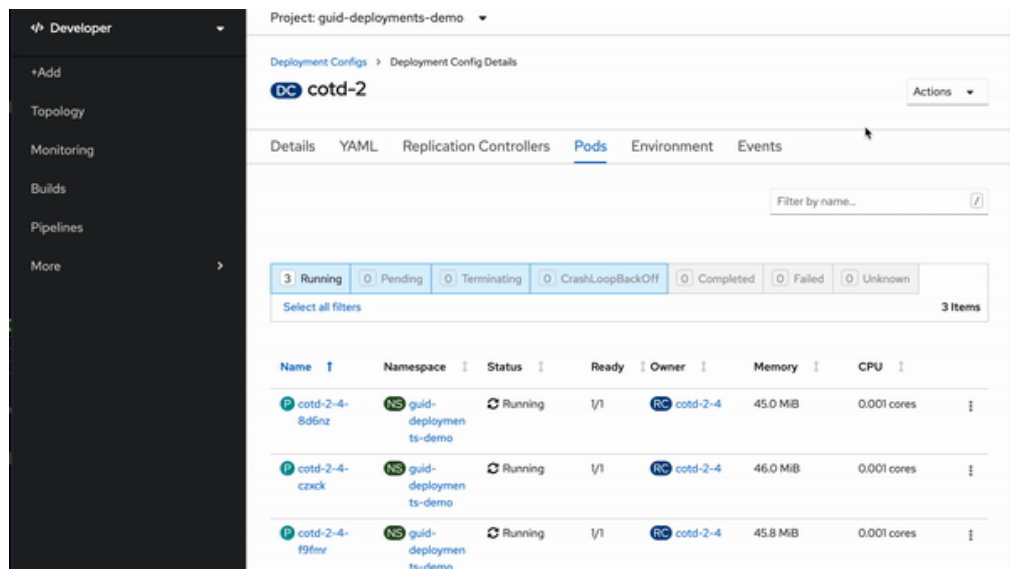
3 Running 0 Pending 0 Terminating 0 CrashLoopBackOff 0 Completed 0 Failed

Select all filters

Name ↑	Namespace ↑	Status ↑	Ready ↑	Owner ↑	Memory ↑	
 ctd-2-4-8d6nz	 guid-deploymen ts-demo	 Running	1/1	 ctd-2-4	45.0 MiB	
 ctd-2-4-czxc	 guid-deploymen ts-demo	 Running	1/1	 ctd-2-4	46.0 MiB	0.001 cores ⋮
 ctd-2-4-f9fmr	 guid-deploymen ts-demo	 Running	1/1	 ctd-2-4	45.8 MiB	0.001 cores ⋮

- Start Rollout
- Pause Rollouts
- Edit Count
- Add Storage
- Edit Application Grouping
- Edit ctd-2
- Edit Labels
- Edit Annotations
- Edit Deployment Config
- Delete Deployment Config

- 解释：在 Recreate 部署策略中，所有现有的副本被扩展到`0`，然后新的部署被扩展到所需的副本数量。
- 注意所有的Pod被删除，然后重新创建。
  - 提到这个策略对应用程序来说有很短的停机时间，所以它最常被用于两个版本的应用程序不能共存的情况，即使是很短的时间。



## 8. 清理环境

如果你不打算进行任何额外的演示，请回到OpenShift容器平台网络控制台的主页，删除你的项目。

## 9. 额外演示

你可以做以下可选的演示。

- 演示A/B测试。



- 演示为部署设置自动缩放。
- 演示改变健康和准备状态检查。