实验地址：<https://docs.aws.amazon.com/zh_cn/codepipeline/latest/userguide/tutorials.html>

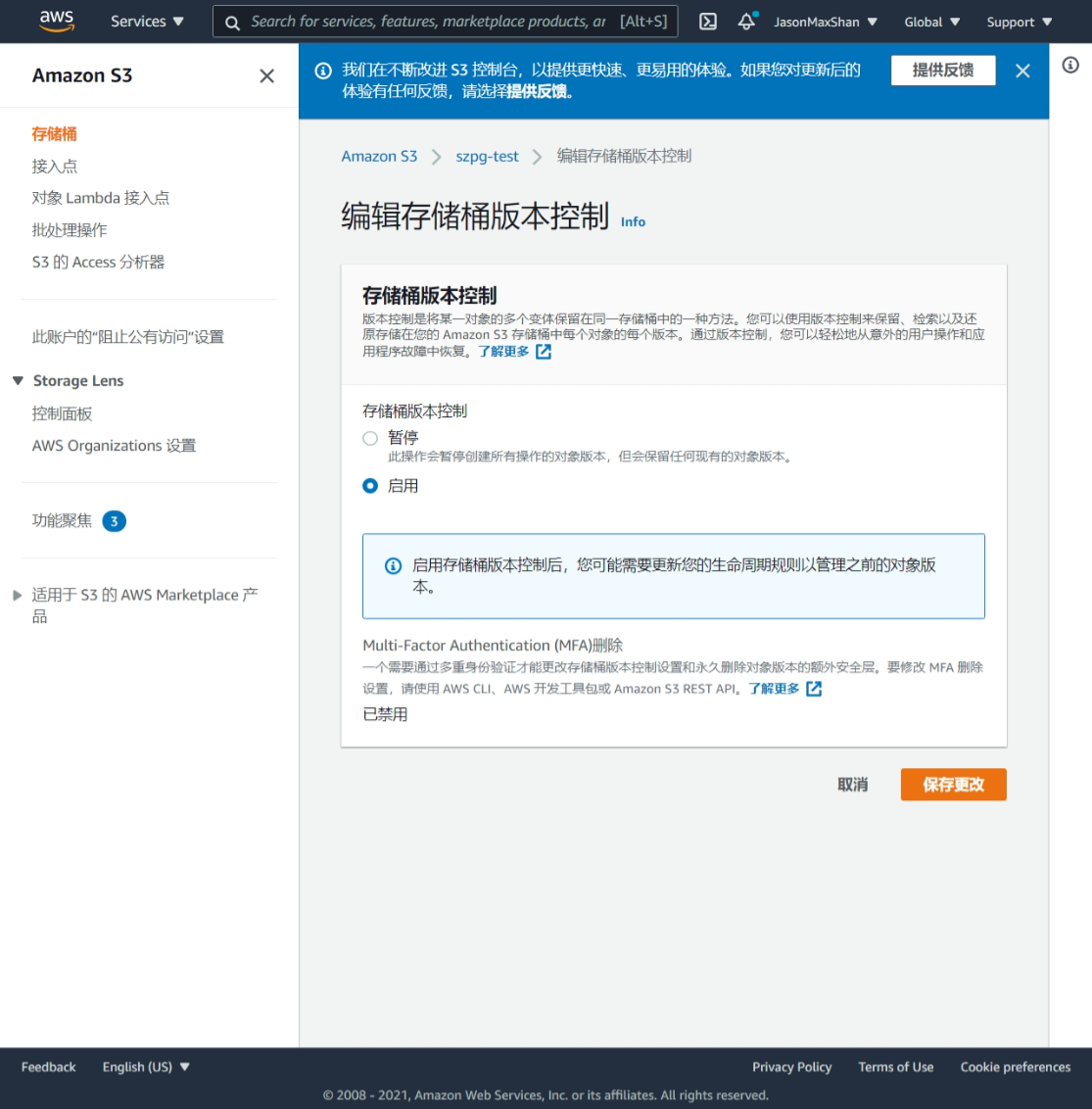
实验1：创建一个和S3有关的管道

步骤1：创建一个S3存储桶

进入S3界面，创建一个S3存储桶



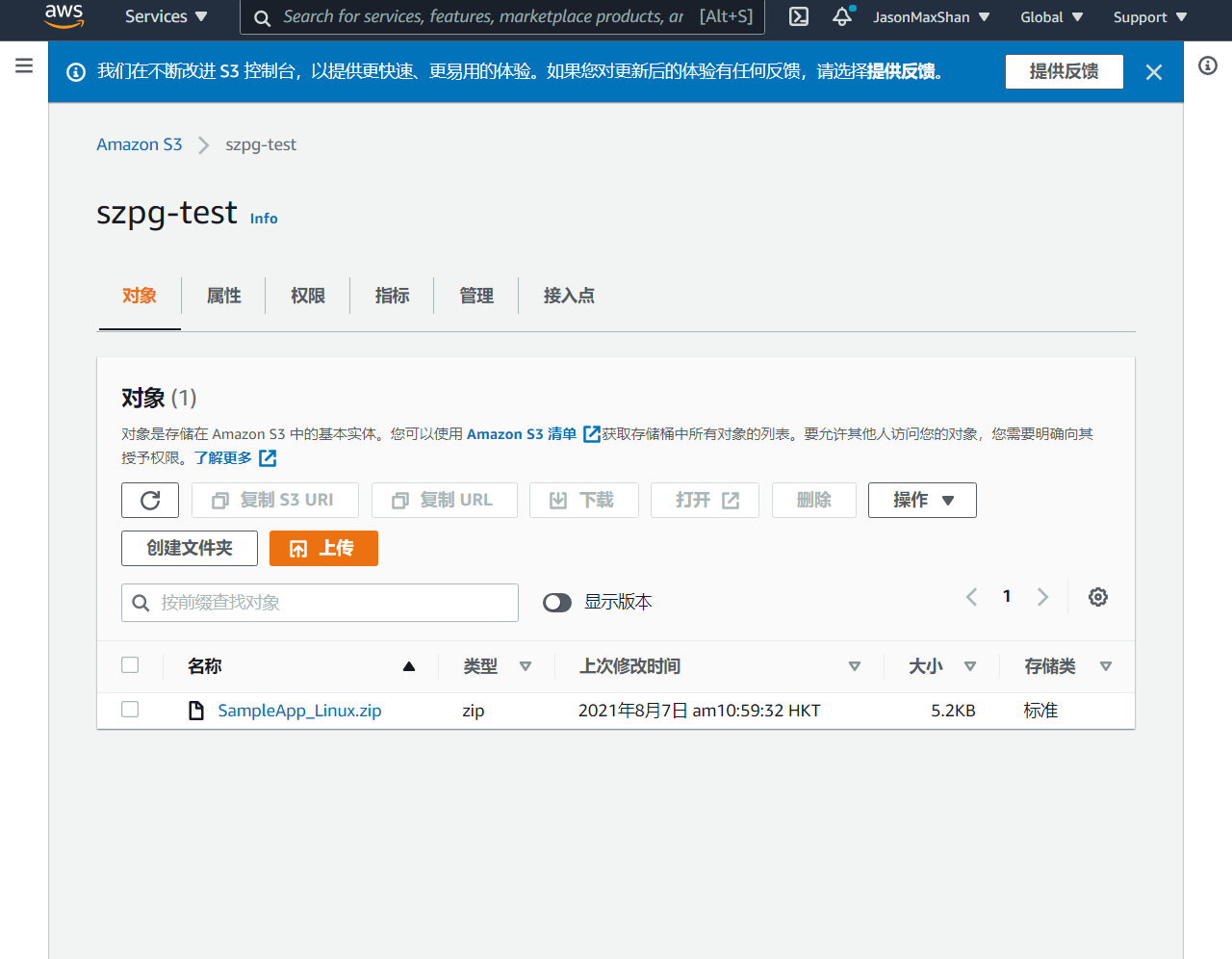
创建完成后，点击存储桶属性，启用存储桶版本控制



然后下载示例文件

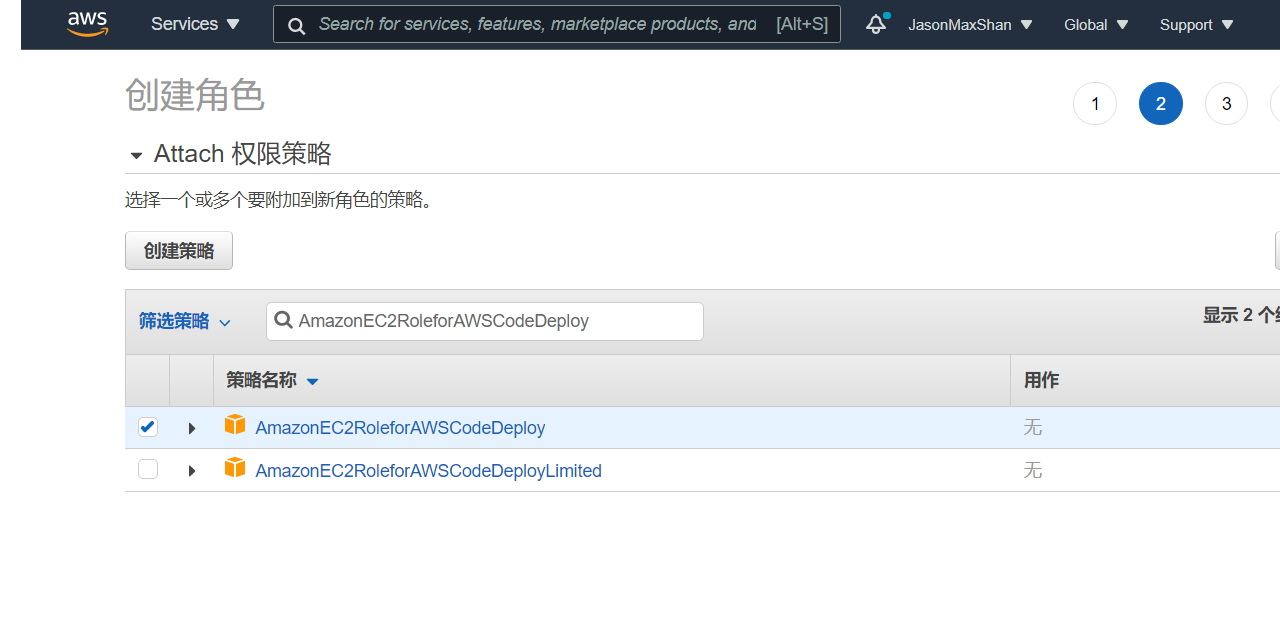
<https://docs.aws.amazon.com/zh_cn/codepipeline/latest/userguide/samples/SampleApp_Linux.zip>

并上传到存储桶中



步骤2：创建Linux实例并安装Codedeploy代理

首先打开IAM界面，创建一个role，使用案例选择EC2，策略搜索AmazonEC2RoleforAWSCodeDeploy，然后创建角色



启动一台Amazon Linux 2 AMI (HVM)的t2.micro实例

IAM角色选择刚刚的角色，用户数据中输入

#!/bin/bash

yum -y update

yum install -y ruby

yum install -y aws-cli

cd /home/ec2-user

wget https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install

chmod +x ./install

./install auto



点击创建

第3步：在CodeDeploy中创建应用程序

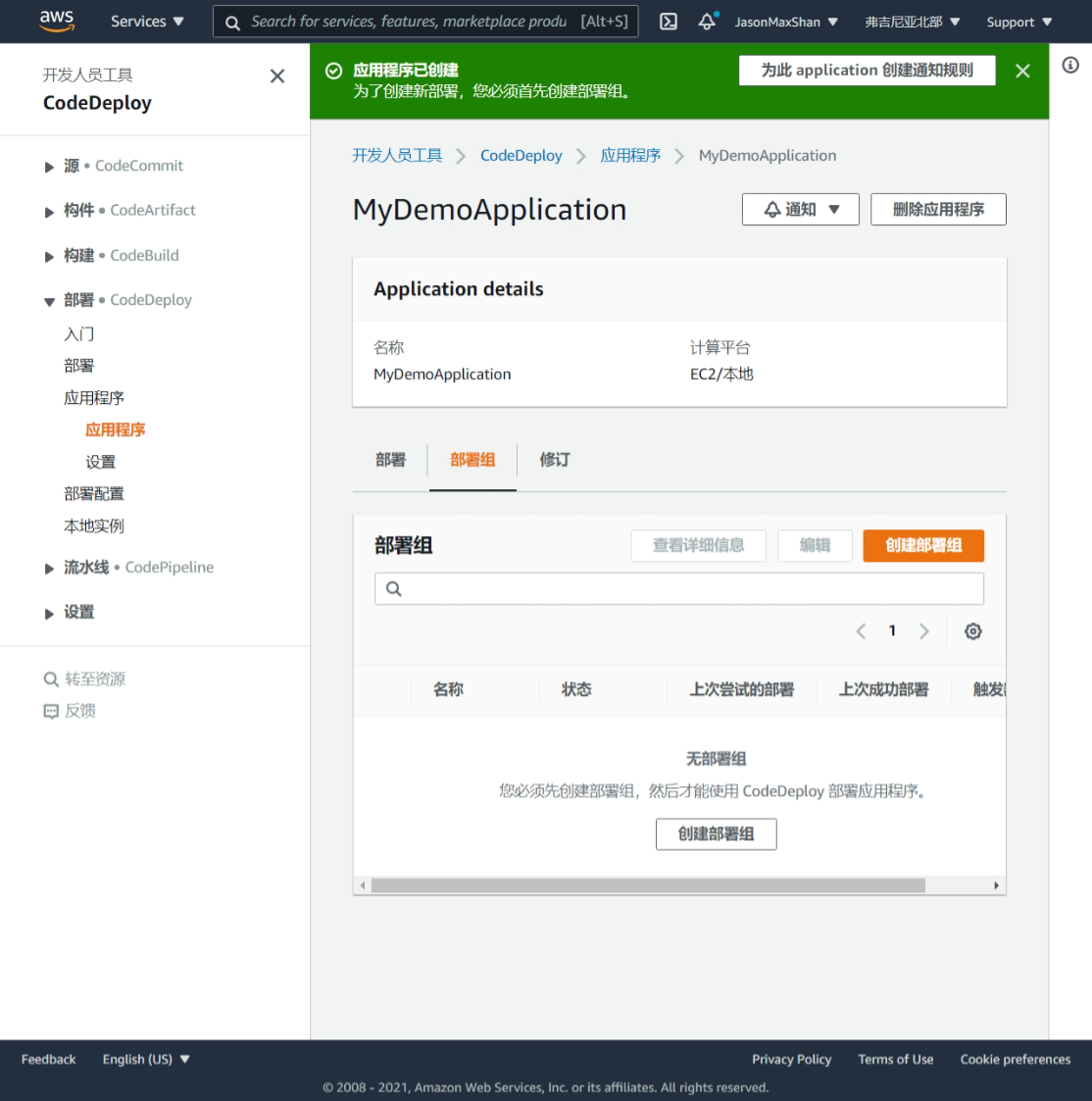
如果应用程序页面上，在 AWS CodeDeploy 菜单上选择应用程序。

选择 Create application。

In应用程序名称，输入MyDemoApplication。

在 Compute Platform (计算平台) 中，选择 EC2/On-premises (EC2/本地)。

选择 Create application。



在 CodeDeploy 中创建部署组

在显示应用程序的页面上，选择 Create deployment group (创建部署组)。

在 Deployment group name (部署组名称) 中，输入 MyDemoDeploymentGroup。

In服务角色中，选择信任 AWS CodeDeploy 的服务角色（至少具有创建 CodeDeploy 的服务角色。要获得服务角色 ARN，请参阅获得服务角色 ARN（控制台）。

在 Deployment type (部署类型) 下，选择 In-place (就地)。

在 Environment configuration (环境配置) 下，选择 Amazon EC2 Instances (Amazon EC2 实例)。选择名称中的密钥字段中，并在值字段中，输入MyCodePipelineDemo。

在 Deployment configuration (部署配置) 中，选择 CodeDeployDefault.OneAtaTime。

在 Load Balancer (负载均衡器) 中，取消选择 Enable load balancing (启用负载均衡)。您无需为此示例设置负载均衡器或选择目标组。

在 Advanced部分，将默认值。

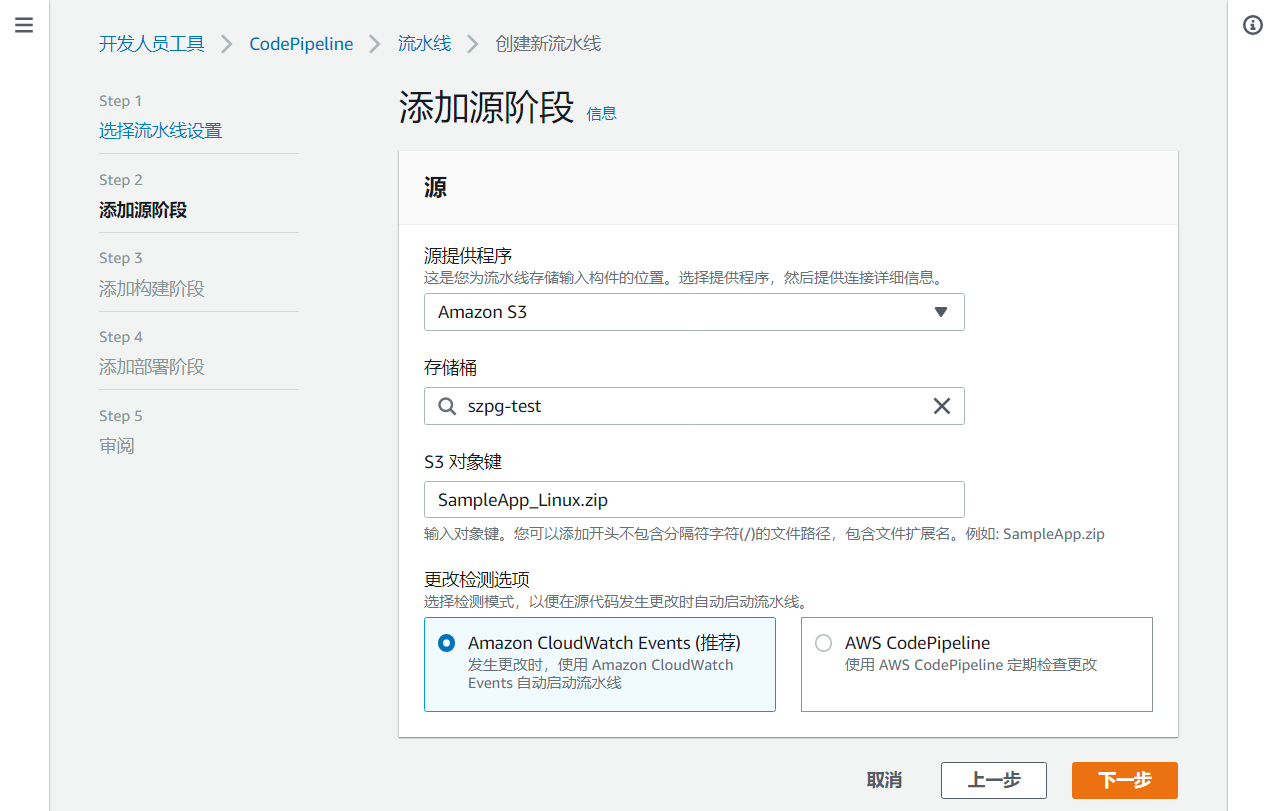
选择 Create deployment group (创建部署组)。



第 4 步：在 CodePipeline 中创建您的第一个管道

选择管道设置, 在Pipeline name (管道名称)，输入MyFirstPipeline。

添加源阶段, 在源提供商中，选择Amazon S3。在 Bucket (存储桶) 中，输入您在第 1 步：为您的应用程序创建 S3 存储桶中创建的 S3 存储桶的名称。在 S3 object key (S3 对象键) 中，输入带或不带文件路径的对象键。请记住加上文件的扩展名。



添加部署阶段, 在部署提供程序中，选择AWS CodeDeploy。Region (区域) 字段默认为与管道相同的 AWS 区域。在 Application name (应用程序名称) 中，输入 MyDemoApplication，或选择 Refresh (刷新) 按钮，然后从列表中选择应用程序名称。在 Deployment group (部署组) 中，输入 MyDemoDeploymentGroup 或者从列表中选择它，然后选择 Next



随后构建管道，等待CodeDeploy 将这些更改部署到 EC2 实例。

**验证您的管道是否已成功运行**

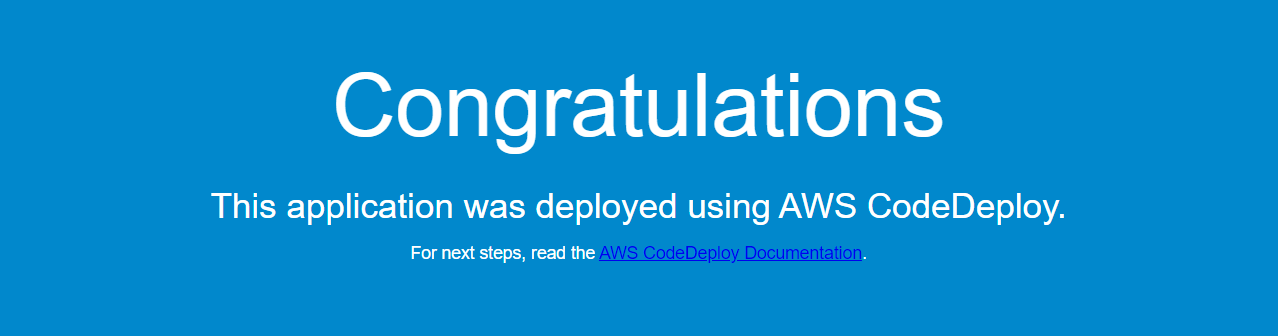
查看管道的初始进度。每个阶段的状态将从还没有任何执行变为正在进行，然后变为 成功或失败。管道将在几分钟内完成首次运行。

操作状态显示 Succeeded (成功) 之后，在 Deploy (部署) 阶段的状态区域中，选择 Details (详细信息)。此操作将打开 AWS CodeDeploy 控制台。

在部署组选项卡，在部署生命周期事件中，选择一个实例 ID。此操作将打开 EC2 控制台。

在 Description (描述) 选项卡上的 Public DNS (公有 DNS) 中，复制地址，然后将其粘贴到 Web 浏览器的地址栏中。查看上传到 S3 存储桶的示例应用程序的索引页面。

以下页面是您上传到 S3 存储桶的示例应用程序。



步骤5：向管道中添加另一个阶段

从打开 CodeDeploy 控制台https://console.aws.amazon.com/codedeploy。

选择 Applications (应用程序)，然后在应用程序列表中选择 MyDemoApplication。

选择 Deployment groups (部署组) 选项卡，然后选择 Create deployment group (创建部署组)。

在 Create deployment group (创建部署组) 页面的 Deployment group name (部署组名称) 中，输入第二个部署组的名称（例如 CodePipelineProductionFleet）。

In服务角色中，选择用于初始部署的相同 CodeDeploy 服务角色（而不是 CodePipeline 服务角色）。

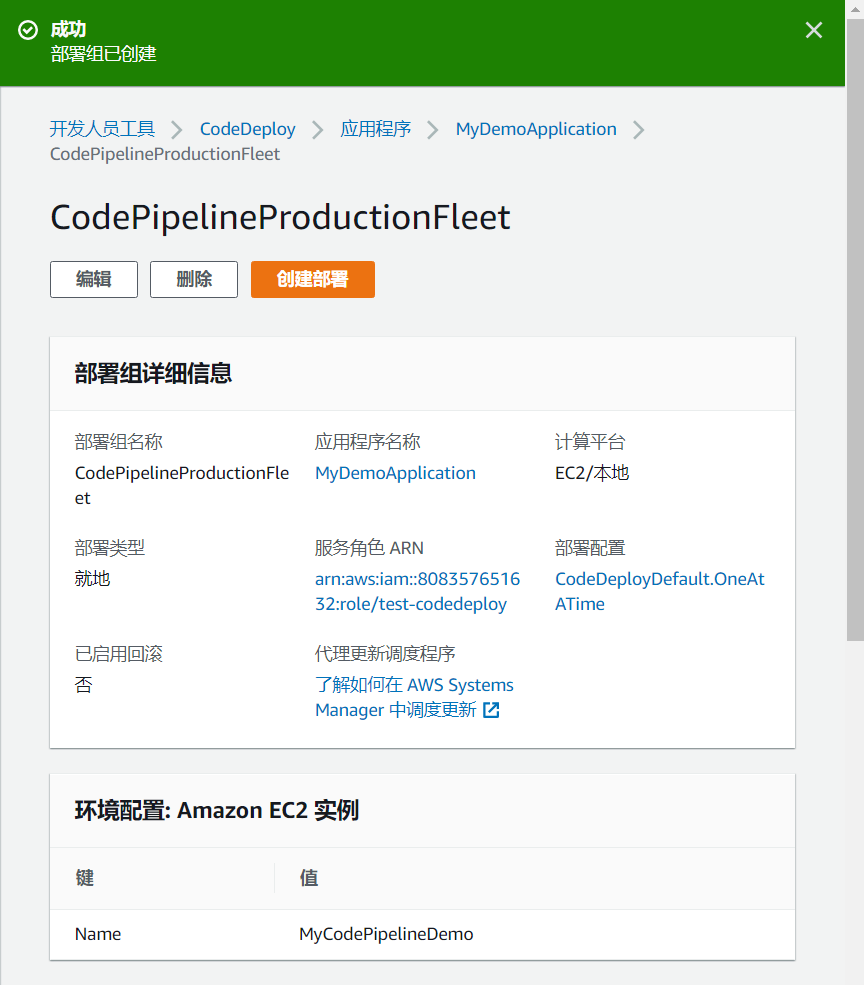
在 Deployment type (部署类型) 下，选择 In-place (就地)。

在 Environment configuration (环境配置) 下，选择 Amazon EC2 Instances (Amazon EC2 实例)。选择名称中的密钥框中，并在值框中，选择MyCodePipelineDemo从列表中。保留 Deployment settings (部署设置) 的默认配置。

在 Deployment configuration (部署配置) 中，选择 CodeDeployDefault.OneAtaTime。

在 Load Balancer (负载均衡器) 中，取消选择 Enable load balancing (启用负载均衡)。

选择 Create deployment group (创建部署组)。



有了另一个部署组后，就可以添加一个使用此部署组的阶段来部署到之前所使用的相同 EC2 实例。

选择您创建的管道的名称 “MyFirstPipeline”。

在管道详细信息页中，选择 Edit (编辑)。

在 Edit (编辑) 页面上，选择 + Add stage (+ 添加阶段) 以紧随 Deploy 阶段之后添加一个阶段。

在新阶段中，选择 + Add action group (+ 添加操作组)。

在 Edit action (编辑操作) 的 Action name (操作名称) 中，输入 Deploy-Second-Deployment。在 Action provider (操作提供方) 中的 Deploy (部署) 下，选择 AWS CodeDeploy。

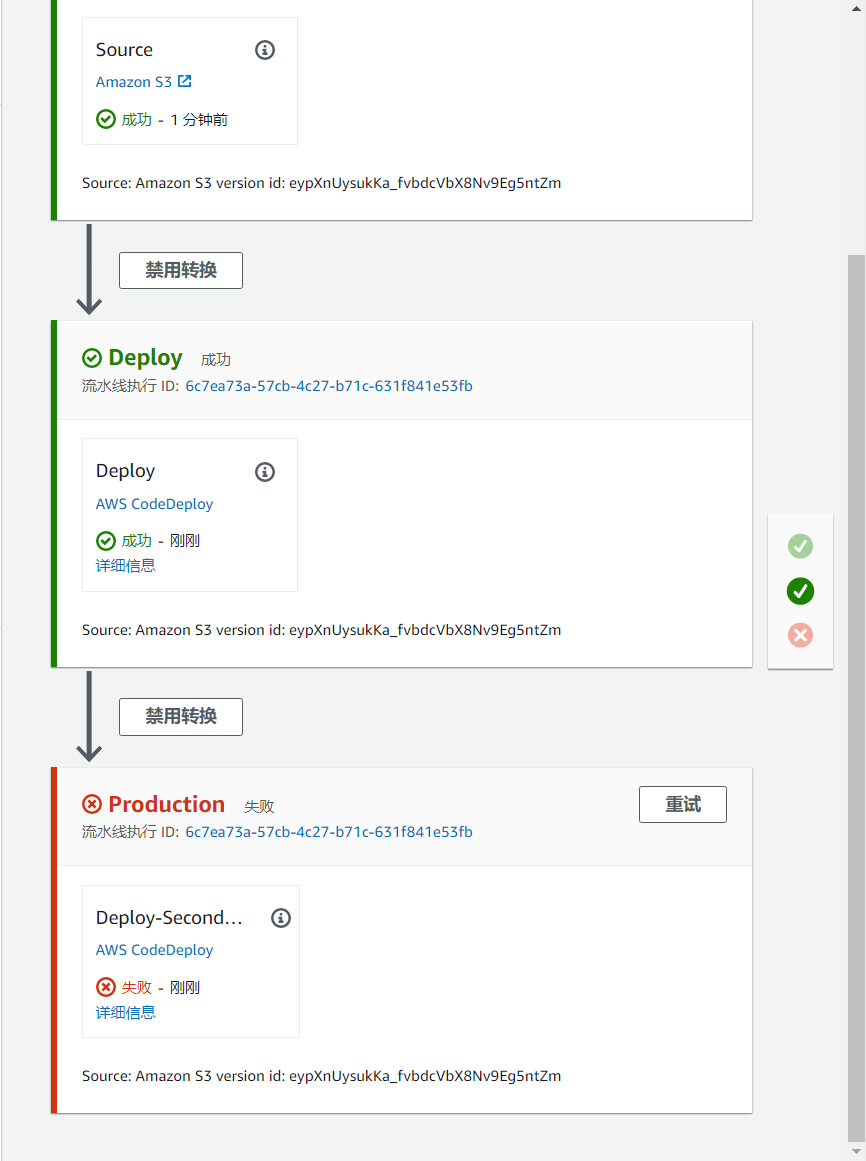
在 “CodeDeploy” 部分的应用程序名称中，选择MyDemoApplication，正如您在创建管道时所做的一样。在 Deployment group (部署组) 中，选择您刚刚创建的部署组 CodePipelineProductionFleet。在 Input artifacts (输入构件) 中，从源操作中选择输入构件。选择 Save。

在 Edit (编辑) 页面上，选择 Save (保存)。在 Save pipeline changes (保存管道更改) 中，选择 Save (保存)。

虽然新阶段已添加到您的管道中，但该阶段显示为 No executions yet 状态，因为没有发生触发管道再次运行的更改。您必须手动重新运行最新修订，以了解编辑后的管道的运行情况。在管道详细信息页中，选择版本更改，然后选择版本当系统提示时。这会通过管道运行在源操作中指定的每个源位置中提供的最新修订。

返回到 CodePipeline 控制台，然后在管道列表中选择MyFirstPipeline以打开视图页面。

管道显示三个阶段以及项目经历这三个阶段时的状态。管道经历所有阶段可能需要 5 分钟的时间。您将看到前两个阶段的部署取得成功，正如之前一样，但 Production 阶段显示 Deploy-Second-Deployment 操作失败。



在 Deploy-Second-Deployment 操作中，选择 Details。您将被重定向到 CodeDeploy 部署的页面。在这种情况下，操作失败是因为第一个实例组部署到所有 EC2 实例，导致第二个部署组没有实例可用。

