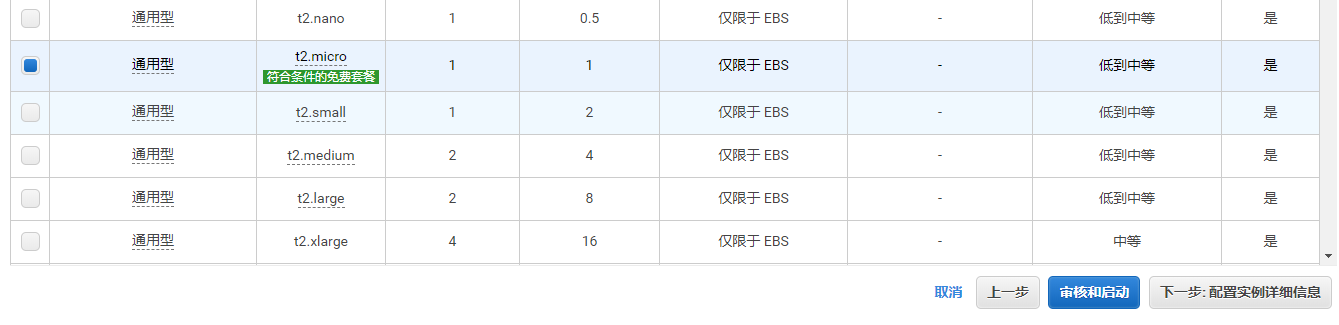
**创建实例**

<https://aws.amazon.com/cn/ec2/getting-started/>



免费用户最多使用30G，终止选项要打钩



安全组主要开通访问服务器的端口和允许访问的IP



报了个错



密钥：创建密钥，填写名称后下载密钥对（划重点）



注意新注册要审核4小时，否则

Your request for accessing resources in this region is being validated, and you will not be able to launch additional resources in this region until the validation is complete. We will notify you by email once your request has been validated. While normally resolved within minutes, please allow up to 4 hours for this process to complete. If the issue still persists, please let us know by writing to aws-verification@amazon.com for further assistance.

**二、链接实例**

链接：

SecureCRT 自带工具将私钥转成公钥

XXX.pem> XXX.pem.pub

两个放到同一目录

然后创建链接

用户名：默认用户名由启动实例时指定的 AMI 确定， Ubuntu，用户名称是 Ubuntu

**创建用户账户**

使用 **adduser** 命令创建用户账户并将其添加到系统（/etc/passwd 文件中会有一个条目）中。该命令还可以为账户创建一个组和一个主目录。在此示例中，用户账户名为 newuser。

[ec2-user ~]$ **sudo adduser *newuser***

[Ubuntu] 在将用户添加到 Ubuntu 系统时，请使用该命令包含 --disabled-password 参数，以避免向该账户添加密码。

[ubuntu ~]$ **sudo adduser *newuser*** --disabled-password

切换到新账户，以便将创建的目录和文件具有正确的所有权。

[ec2-user ~]$ **sudo su - *newuser***

[newuser ~]$

请注意，在此示例中，如果提示从 ec2-user 变为 newuser，则表示您已将 Shell 会话切换到新账户。

将 SSH 公有密钥添加到用户账户。首先在 SSH 密钥文件的用户主目录中创建一个目录，然后创建密钥文件，最后将公有密钥粘贴到密钥文件中。

在 .ssh 主目录中创建一个 newuser 目录，并将其权限更改为 700 (只有文件所有者能够读取、写入或打开该目录。)

[newuser ~]$ **mkdir .ssh**

[newuser ~]$ **chmod 700 .ssh**

**重要**

如果没有这些确切的文件权限，用户将无法登录。

在 authorized\_keys 目录中创建名为 .ssh 的文件并将其权限更改为 600 (只有文件所有者能够读取或写入此文件)。

[newuser ~]$ **touch .ssh/authorized\_keys**

[newuser ~]$ **chmod 600 .ssh/authorized\_keys**

**重要**

如果没有这些确切的文件权限，用户将无法登录。

使用您常用的文本编辑器 (如 **vim** 或 **nano**) 打开 authorized\_keys 文件。

[newuser ~]$ **nano .ssh/authorized\_keys**

将密钥对的公钥粘贴到该文件并保存更改。例如：

刚才生成的公钥 ：XXX.pem.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V

hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr

lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ

qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb

BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE

用户现在应该能够使用对应添加到 newuser 文件的公有密钥的私有密钥登录实例上的 authorized\_keys 账户。

**三、安装CodeDeploy Agent**

CodeDeploy是通过与部署在服务器上的Agent通信，实现代码部署的

由于Agent需要访问S3下载代码，所以EC2服务器需要配置角色（Role）以保证Agent对S3的读取权限。

**创建EC2时通过User Data安装**

在高级信息里添加启动脚本

*bucket-name* 是包含适用于您所在区域的 CodeDeploy 资源工具包文件的 Amazon S3 存储桶的名称。例如，对于 美国东部（俄亥俄）区域，将 *bucket-name* 替换为 aws-codedeploy-us-east-2

https://docs.aws.amazon.com/zh\_cn/codedeploy/latest/userguide/instances-ec2-create.html

**先创建后安装**

注意：存储桶名称和区域名称要替换，美国东部（俄亥俄）区域，为 aws-codedeploy-us-east-2 和 us-east-2

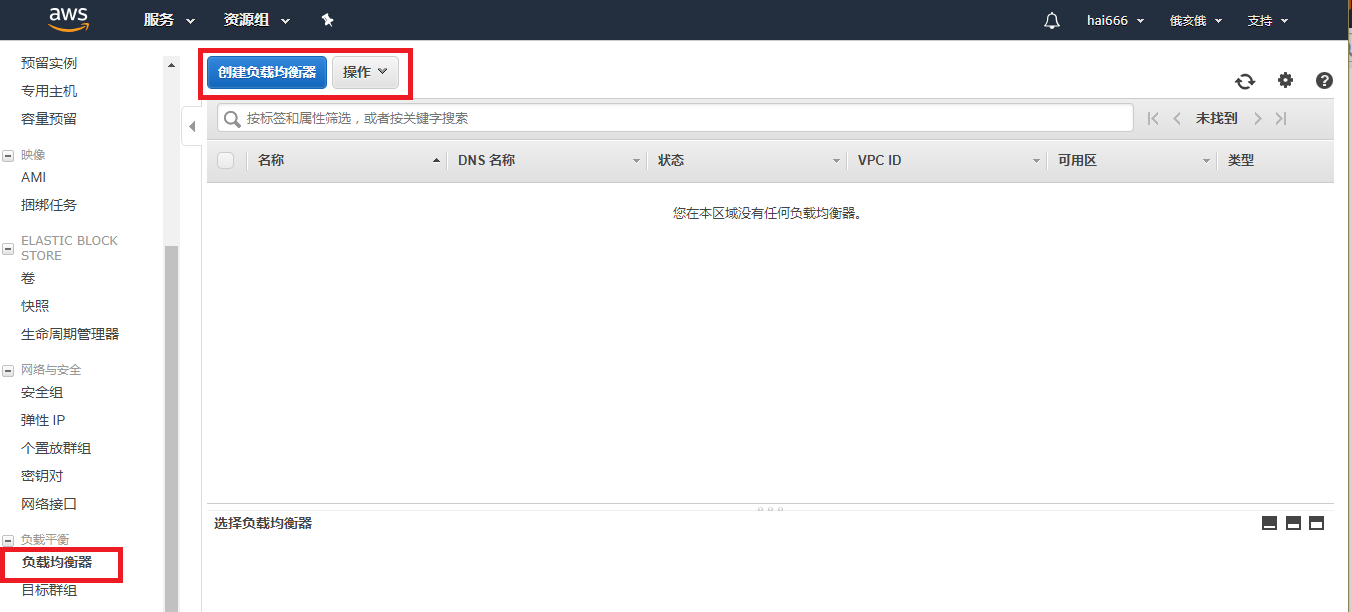
https://docs.aws.amazon.com/zh\_cn/codedeploy/latest/userguide/resource-kit.html#resource-kit-bucket-names

完成后查看：

sudo service codedeploy-agent status

如果显示：error: No AWS CodeDeploy agent running

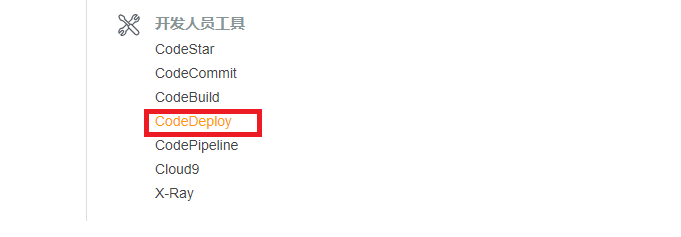
**四、创建负载均衡ALB**





**五、创建CodeDeploy环境**

用于集成Jenkins



输入名称和部署方式，就地部署和蓝绿部署

<https://docs.aws.amazon.com/zh_cn/codedeploy/latest/userguide/applications-create-in-place.html>

**简介**

CodeDeploy 是一项部署服务，可以向 Amazon EC2 实例、本地实例、无服务器 Lambda 函数或 Amazon ECS 服务自动执行应用程序部署

应用：

应用程序是部署的核心，由部署组（Deployment Group）和代码修订（Revisions）组成。一个应用可以包含多个部署组，一个部署组又可以包含多台EC2服务器。同时，一个服务器也可以属于多个部署组，因为一个服务器可能同时运行多个应用

**部署组：**

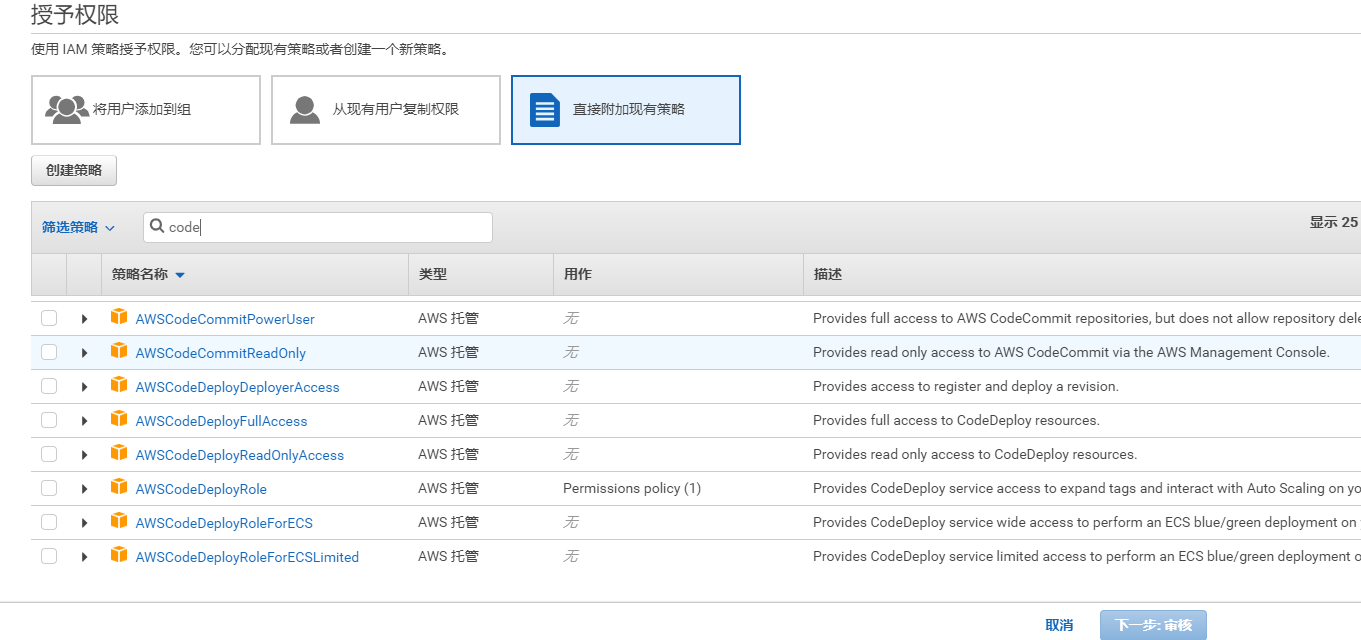
创建或修改部署组时，如果添加EC2服务器，可以通过标签（Tag）对已有的EC2服务器进行筛选。标签（Tag）可以在创建EC2时添加。

**代码修订：**

代码修订保存了当前应用涉及到得所有代码，代码的存放位置可以在S3或Github

创建IAM用户

IAM用户使用codedeploy，



创建角色

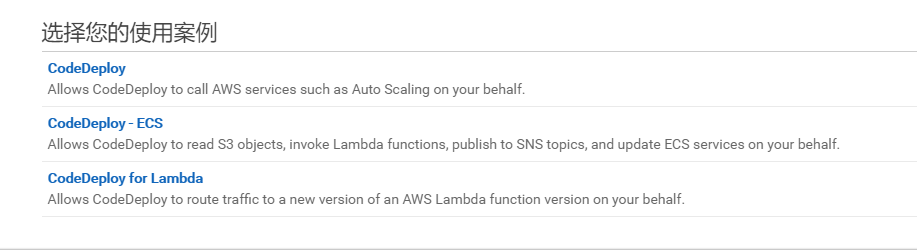
由于Agent需要访问S3下载代码，所以EC2服务器需要配置角色（Role）以保证Agent对S3的读取权限

**打开控制台：**<https://console.aws.amazon.com/iam/>。

服务 ==》IAM



选择创建角色

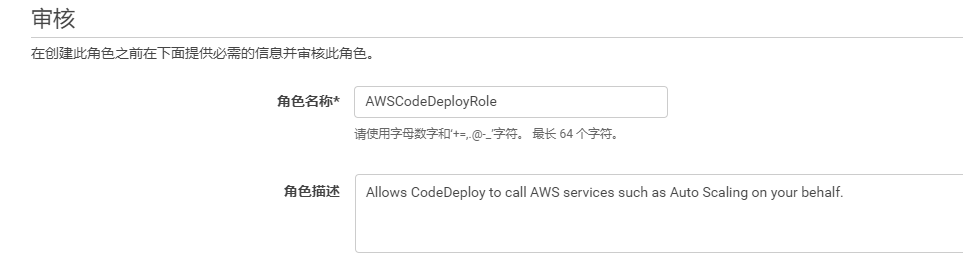


**对于 EC2/本地 部署，选择 CodeDeploy。**

**对于 Amazon ECS 部署，选择 CodeDeploy - ECS。**

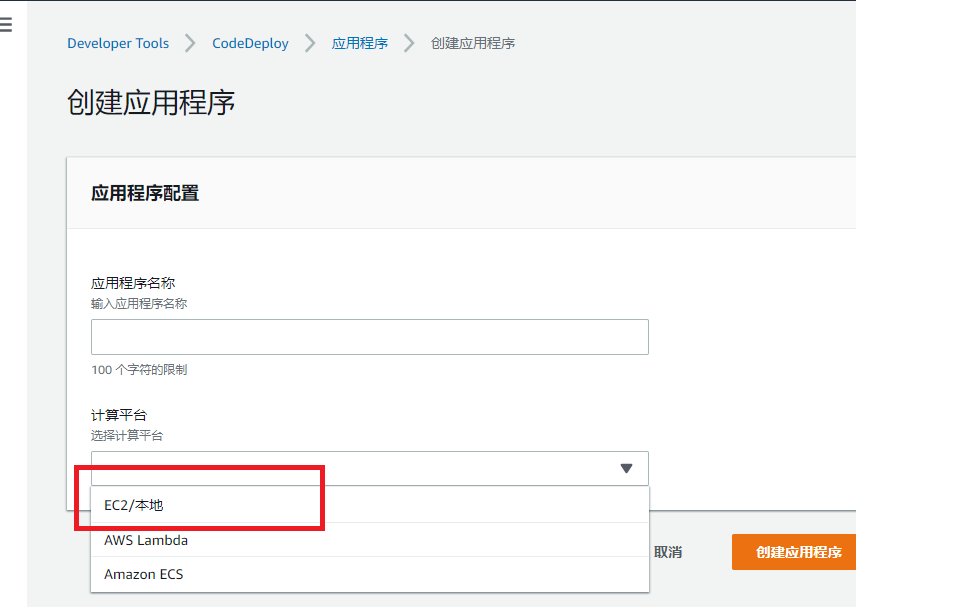
**对于 AWS Lambda 部署，选择 适用于 Lambda 的 CodeDeploy。**

**填入名称**

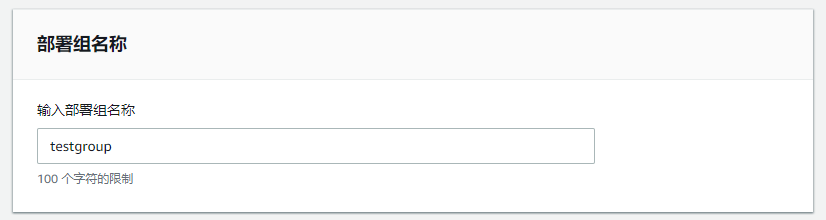


**AWSCodeDeployRole**

创建一个应用程序 test



**为应用程序创建部署组**

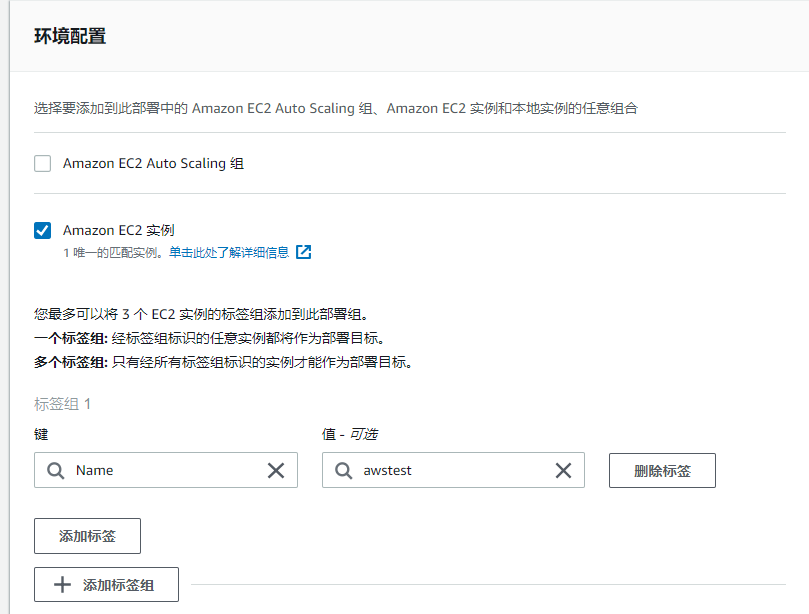


添加服务角色

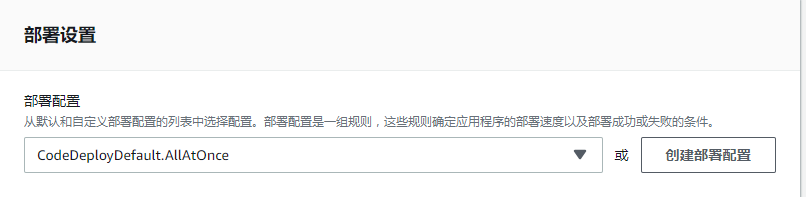


部署类型，选就地部署和蓝绿部署

环境配置，选EC2实例，关联到EC2实例

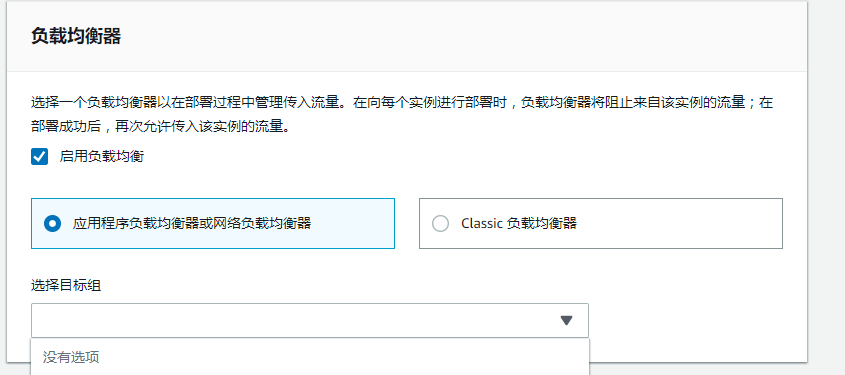


部署设置，比如单台依次部署



可以启用负载均衡，但要先创建目标组

由于只有一个EC2实例，先关闭，看看会怎么样



高级

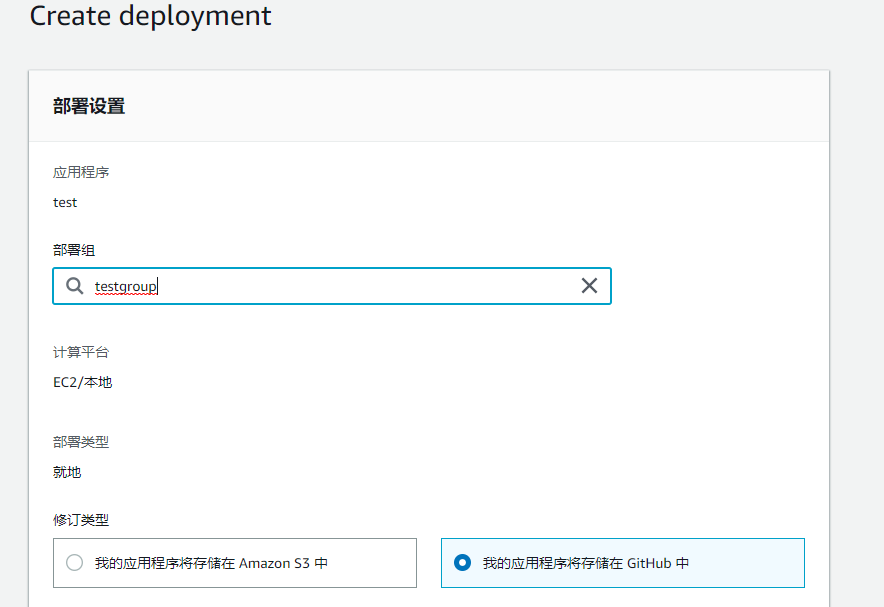
触发器：可以接受此部署组的事件通知

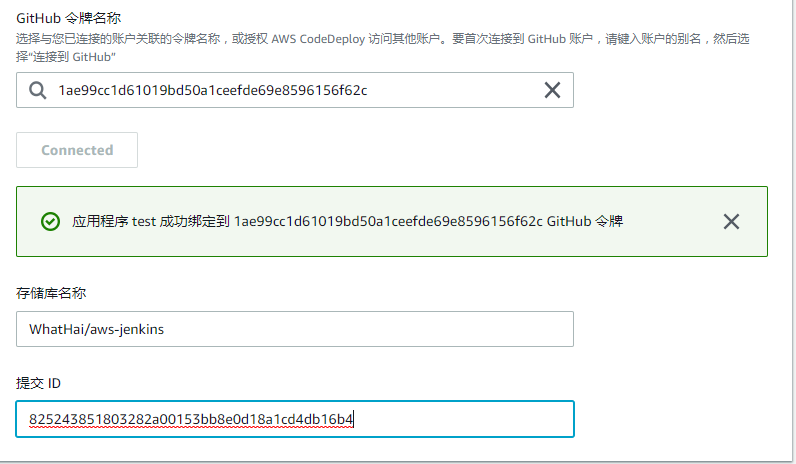
警报：当指标超过设定值，发出警报停止部署

回滚策略，可以部署失败回滚

**创建部署**

部署组选上一步创建的，源码选github



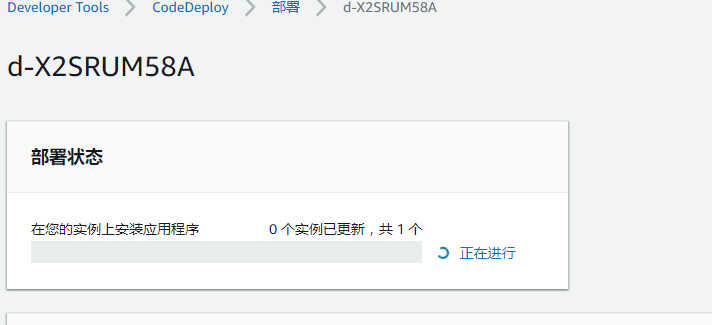


令牌是github生成的token

名称包含用户名和仓库明

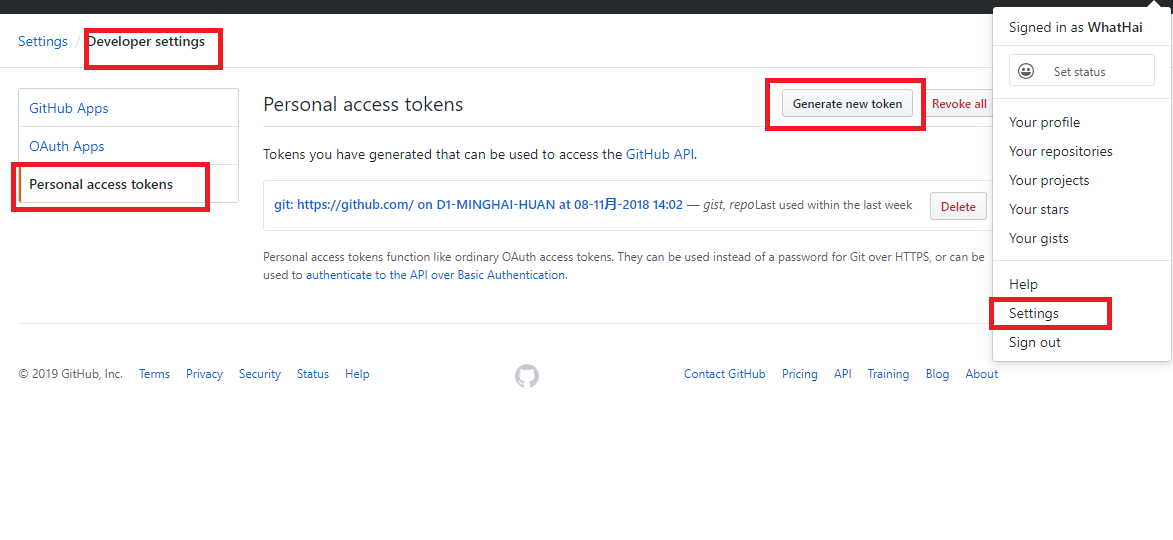
提交ID在提交列表中找到，通常为 40 个字符并包含字母和数字

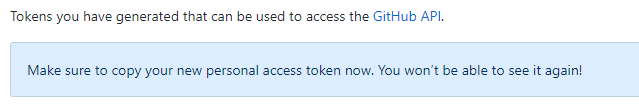
通常是master，然后选择commits，找到最新的commitid，粘贴到输入框



**六、源码托管自动触发，github**

生成GitHub token，这个token用于jenkins访问GitHub

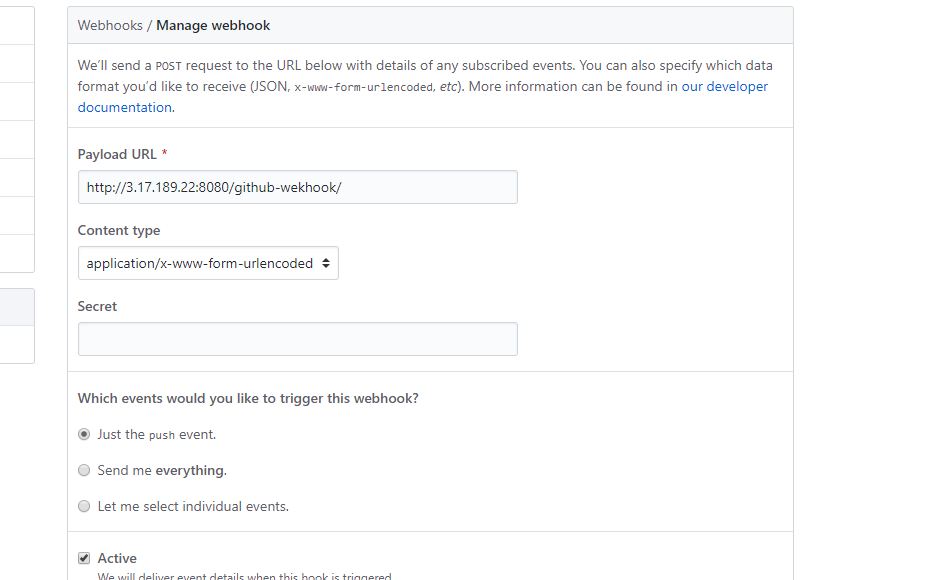




**1ae99cc1d61019bd50a1ceefde69e8596156f62c**

为需要做CI/CD的GitHub创建hook，实现代码更新自动通知Jenkins，Payload URL设置Jenkins Server的地址，默认Jenkins监听8080端口。记录下生成的token字符串

在Payload URL，输入http://EC2公网IP地址/github-wekhook/



**七、安装Jenkins，添加CodeDeploy插件**

移除openjdk，安装java1.8

**安装Jenkins**

在浏览器输入输入EC2的公网IP地址如54.223.215.xx:8080，输入上面得到的默认密码

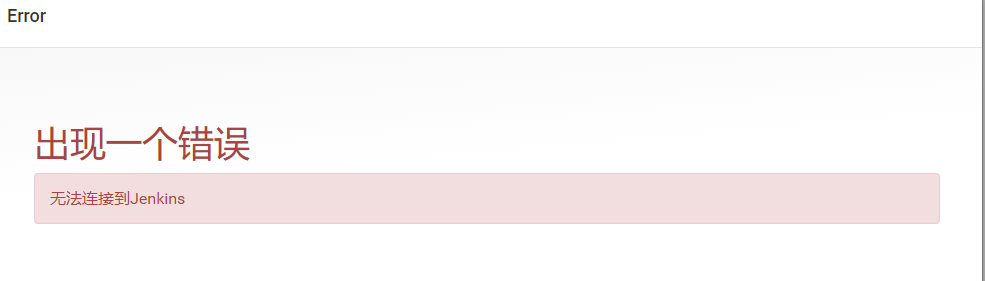
（访问老是超时，不知原因，猜是被墙了）

后来查文档发现，安全组配置不对，尴尬

[https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#troubleshoot-instance-ping](https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html)



输入密码进入后，点左边那个推荐的，不然会出现如下错误：



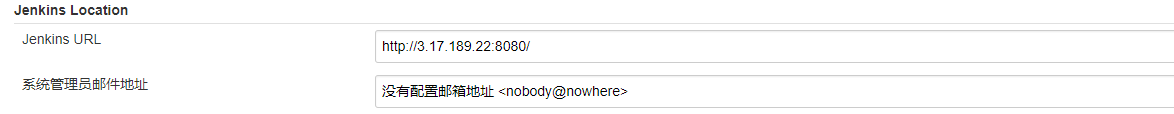
创建一个用户 hai666 p：hai666jenkins



**系统配置**

Manage ==》 Configure System

**Jenkins Location中配置**Jenkins URL： EC2公网ip,端口8080

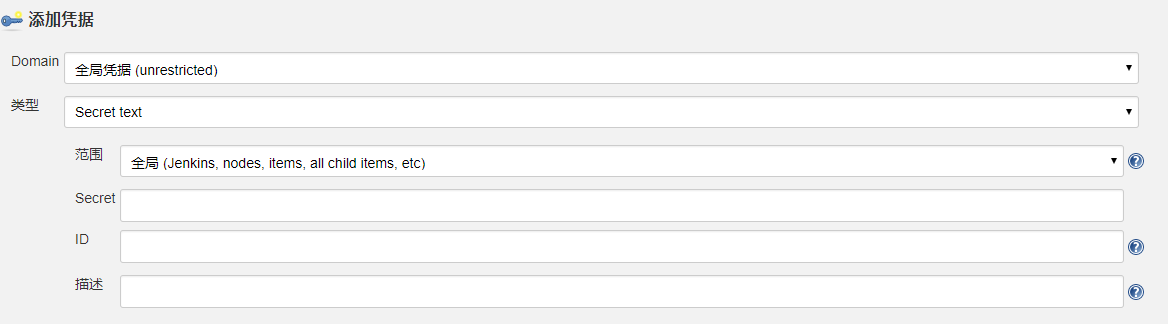


添加github服务器



配置凭据，在Secret 输入第六步获取的github token





Test Connection 测试配置是否成功。

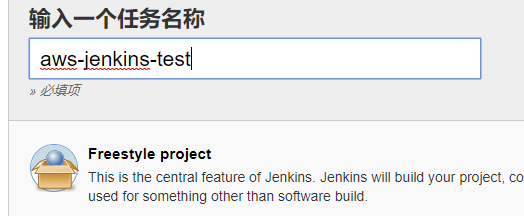
image32.gif

Credentials verified for user WhatHai, rate limit: 4998

**添加插件**

Manage Plugins 添加 AWS CodeDeploy Plugin for Jenkins 插件

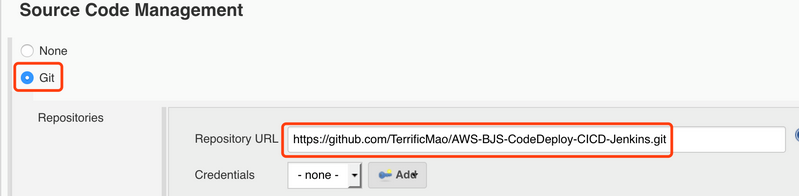
**创建项目**



配置github项目地址



源代码管理选git



其中报错：**Failed to connect to repository : Error performing command: git ls-remote -h https://github.com/WhatHai/aws-jenkins.git HEAD**

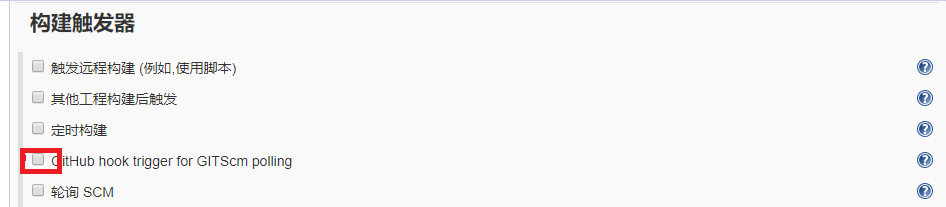
进入 系统管理 > Global Tool Configuration：

image32.gif

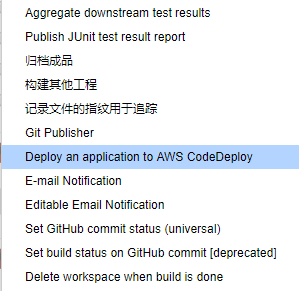
**There's no such executable git in PATH: /sbin, /usr/sbin, /bin, /usr/bin.**

解决：安装git，配置路径

构建触发器，选择Github hook trigger for GITScm polling



构建后操作

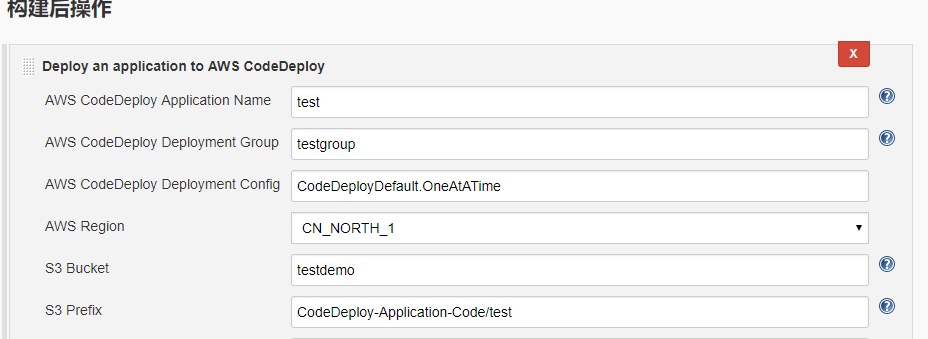


**配置CodeDeploy**

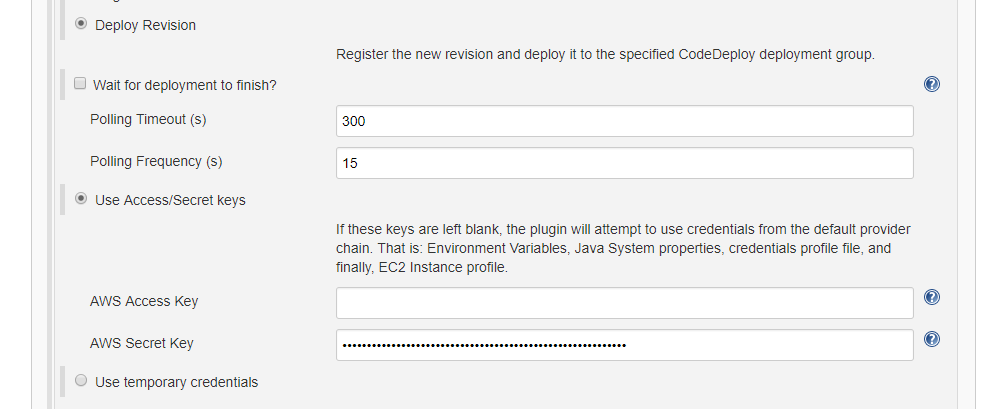
选择“Post-build Actions” 然后配置CodeDeploy，区域选择AWS中国（北京）区域cn-north-1

填入第五步创建的应用程序和部署组

S3为代码存储位置



认证方式可以输入AWS Access Key和Security Key，如果是生产环境建议使用临时的credentials ：Use temporary credentials



如何找到Access Key和Security Key？



进入选择Continue to Security Credentials ，然后创建密钥



访问密钥 ID:

AKIAIHIRTJWHHC7R5CRA

私有访问密钥:

N7HSlA6KLzOtWOR8W9i6p42dSZOxISzNQKZHXBVL

**八、总结**

