



链滴

通过 Github Action 发布项目到 Aliyun 阿里云 ECS

作者: [kisChang](#)

原文链接: <https://ld246.com/article/1602747692300>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



通过Github Action发布项目到Aliyun阿里云ECS，非常适合团队项目或是个人的小项目，操作简单便于维护。

本文将通过最详细的描述，从每个步骤的源头开始讲起，保证小白也能成功。

通过Github的Action发布到ECS有几个关键点：

1. 项目构建环境的初始化，以及项目的编译

这一步可以利用Action的构建模板，选择合适的基础环境，然后再定制化编译脚本完成初始化和编译程。

2. Action的编译环境，如何安全的连接到ECS完成发布

项目完成编译后，需要把文件上传到服务器，并完成服务的重启，此时需要通过SSH等服务。

本文以SSH为主进行介绍，SSH链接到ECS，建议通过密钥形式，安全性更高。

下面是一份我的个人项目中的一份脚本，供大家参考：

```
name: Build And Deploy To Aliyun ECS
on:
  push:
    branches: [ master ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2
        with:
          persist-credentials: false
```

```
  ref: master
  submodules: true
- name: Install and Build
  run: |
    yarn install
    yarn build
- name: Deploy to aliyun server
  uses: easingthemes/ssh-deploy@v2.0.7
  env:
    SSH_PRIVATE_KEY: ${{ secrets.ALIYUN_SERVER_ACCESS_TOKEN }}
    ARGS: "-avzr --delete"
    SOURCE: "dist/"
    REMOTE_HOST: ${{ secrets.ALIYUN_SERVER_HOST }}
    REMOTE_USER: "root"
    TARGET: "/root/web"
```

分部分介绍：

第一部分：

`name: Build And Deploy To Aliyun ECS`

这个是任务的名称，参考作用。

第二部分：

```
on:
push:
branches: [ master ]
```

这部分是触发器，当发生哪些行为时，触发Github Action的执行。

如上配置，则是当master分支，发生push时，触发执行。

你同样可以切换到其他的分支或其他的行为，比如新tag的push等等，具体请参考：

<https://docs.github.com/cn/free-pro-team@latest/actions/reference/events-that-trigger-workflows>

第三部分

```
jobs:
```

这里就开始正式的构建项目了，jobs就是工作流程，工作流程运行包括一项或多项作业。作业默认是行运行。

我们这里定义了一项作业，即build。

```
runs-on: ubuntu-latest
steps:
```

这里定义了我们这项作业的运行环境，机器可以是 GitHub 托管的运行器或自托管的运行器。

可选的GitHub运行器有windows、ubuntu和macos，你也可以构建自托管的运行器，满足您高度的定义运行环境需求。

我的项目是NodeJS编写的，默认的ubuntu运行器包含NodeJS环境，所以我这里直接采用了GitHub供的运行器。

steps则是作业的步骤，每个步骤可以是运行命令或是其他操作。

```
- name: Checkout
  uses: actions/checkout@v2
  with:
    persist-credentials: false
    ref: master
    submodules: true
```

这是我们作业的第一个步骤，这个步骤的任务是把我们的源码checkout下来，此处选择的分支是master。

```
- name: Install and Build
  run: |
    yarn install
    yarn build
```

这是作业的第二个步骤，NodeJS项目的构建环境初始化和编译。

```
- name: Deploy to aliyun server
  uses: easingthemes/ssh-deploy@v2.0.7
  env:
    SSH_PRIVATE_KEY: "${{ secrets.ALIYUN_SERVER_ACCESS_TOKEN }}"
    ARGS: "-avzr --delete"
    SOURCE: "dist/"
    REMOTE_HOST: "${{ secrets.ALIYUN_SERVER_HOST }}"
    REMOTE_USER: "root"
    TARGET: "/root/web"
```

这是作业的最后一步，项目编译完成后，将会在项目目录的dist下生成最终文件，我们只需要把该目录下的所有文件发布到ECS即可。

这里使用的是SSH链接到了ECS。

先详细介绍一下几个关键参数：

```
ARGS: "-avzr --delete"
SOURCE: "dist/"
TARGET: "/root/web"
```

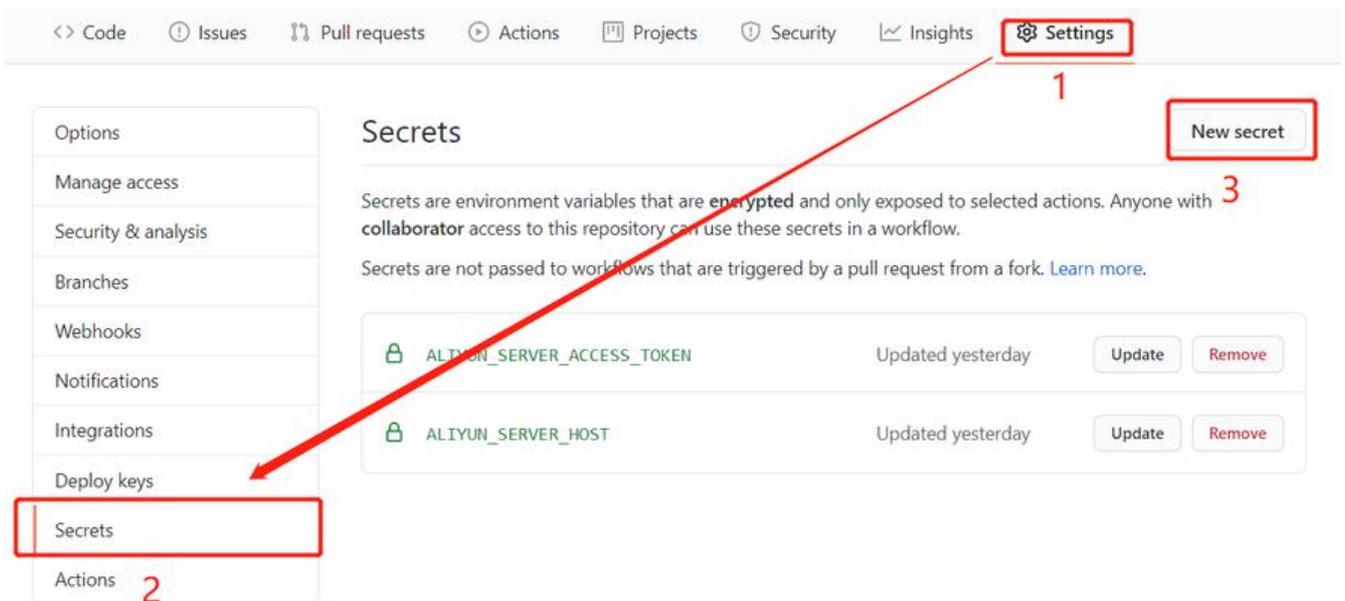
这三个参数，将本地的dist目录下的所有文件覆盖到了云端的/root/web目录

```
SSH_PRIVATE_KEY: "${{ secrets.ALIYUN_SERVER_ACCESS_TOKEN }}"
REMOTE_HOST: "${{ secrets.ALIYUN_SERVER_HOST }}"
REMOTE_USER: "root"
```

这三个参数是SSH的链接参数，user即链接服务的用户名（阿里云一般是root），host即服务器地址可以是IP也可以是域名）

SSH_PRIVATE_KEY链接私钥

其中HOST和KEY两个我们采用了环境变量的方式引用，对应secrets的环境变量，你可以在项目的如图所示位置进行配置：



我们配置了两个参数，HOST比较简单，关于私钥的配置，请参考如下步骤：

(其实是SSH免密码登录的配置，熟悉的同学可以掠过)

1. 我们需要创建一个私钥

我们可以利用阿里云控制台进行操作（同样，也可以用linux的命令行进行操作）

进入阿里云的ECS控制台，在左侧菜单找到<网络与安全> <密钥对>

点击创建密钥对，如下填写，名称和其他信息可以随意：

*密钥对名称：

长度为2~128个字符，不能以特殊字符及数字开头，只支持点号（.）、下划线（_）、连字符（-）和半角冒号（:）

*创建类型： 自动创建密钥对 导入已有密钥对

创建完成后请一定下载私钥，您只有一次下载私钥的机会

资源组：

标签：

创建成功后，就会自动下载私钥文件，我们保存到本地即可。

2. 利用私钥生成公钥

Windows可以使用WSL，Linux的进入终端

输入如下命令，即可生成公钥

```
ssh-keygen -y -f /path/to/private_key > /path/to/public_key.pub
```

3. 将公钥写入.ssh

将生成的公钥里面的内容，追加到ECS的~/.ssh/authorized_keys中即可。

4. 更新ssh配置，允许root登录

编辑/etc/ssh/sshd_config

修改如下配置：

```
PermitRootLogin yes
```

然后执行：

```
service sshd restart
```

应用配置

5. 更新GitHub配置

进入Secrets配置，点击<New Secret>，如下图所示填写

名称填写我们在脚本里写的变量名：ALIYUN_SERVER_ACCESS_TOKEN

值，填写私钥的内容。

Secrets / New secret

Name

ALIYUN_SERVER_ACCESS_TOKEN

Value

-----BEGIN RSA PRIVATE KEY-----

MIIl
waf
Aks
7b8
vXF

Add secret

创建完成后，记得删掉私钥，并清空回收站，不要让任何人获取到私钥内容。

至此，一切大功告成！