

基于Github Action的持续集成

什么是持续集成CI

持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽早地发现集成错误。

[持续集成](#)

Github Actions

Github Actions让我们可以在Github仓库中直接创建自定义的软件开发生命周期工作流程。

GitHub Actions enables you to create custom software development life cycle (SDLC) workflows directly in your GitHub repository.

[官方文档](#)

概念

- Workflow
 - 由一个或多个Job组成
 - 可以计划执行或通过[事件](#)激活
 - 可以在代码仓库中设置一个workflow，可配置自动化构建、测试、打包或部署项目
- Job
 - 由Step构成
 - 可以定制依赖关系
 - 可以并行运行
- Step
 - Job执行的一组任务
 - 在同一运行程序中执行
 - 文件系统共享信息
 - 运行命令或action
- Action
 - 工作流中最小的可移植构建块
 - 可以使用[GitHub社区共享的action库](#)，以及自定义action库。

workflow范例

下面配置可以在我们push代码时自动打包我们应用并部署到阿里云服务器上。

```
name: Build app and deploy to aliyun

on:
  push:
    branches:
      - ci-test

jobs:
  build:
    # runs-on 指定job任务运行所需要的虚拟机环境(必填字段)
    runs-on: ubuntu-latest
    steps:
      # 获取源码
      - name: Checkout
        # 使用action库 actions/checkout获取源码
        uses: actions/checkout@master
      # 安装Node10
      - name: use Node.js 13.2.0
        # 使用action库 actions/setup-node安装node
        uses: actions/setup-node@v1
        with:
          node-version: 13.2.0
      # 安装依赖
      - name: Yarn install
        run: npm install yarn && yarn install
      # 打包
      - name: Yarn build
        run: yarn run build:prod
      # 部署到阿里云
      - name: Deploy to Aliyun
        uses: easingthemes/ssh-deploy@v2.1.1
        env:
          # 私钥
          SSH_PRIVATE_KEY: ${ secrets.PRIVATE_KEY }
          # scp参数
          ARGS: "-avzr --delete"
          # 源目录
          SOURCE: "dist"
          # 服务器ip: 换成你的服务器IP
          REMOTE_HOST: "47.98.252.43"
          # 用户
          REMOTE_USER: "root"
```

```
# 目标地址
TARGET: "/root/kkb-honor-admin/client"
```

设置ssh秘钥

[linux](#)及[阿里云](#)相关操作

设置公钥

- 复制本地公钥, ~/.ssh/id_rsa_pub

```
cd .ssh/
cat id_rsa_pub
```

- 登录服务器

```
ssh root@47.98.252.43
```

- 增加公钥, .ssh/authorized_keys

```
cd .ssh/
vi authorized_keys
# 粘贴rsa_pub中内容
# 保存退出:wq
```

设置私钥

- 复制本地私钥, ~/.ssh/id_rsa

```
cd .ssh/
cat id_rsa
```

- github项目页 - settings - secrets - add a new secret

名称: `PRIVATE_KEY`, 值: 私钥内容

触发workflow

修改本地的几个关键配置:

如果不需要以上下文区分不同项目，可设置publicPath为'/'，vue.config.js

```
module.exports = {  
  // publicPath: process.env.NODE_ENV === 'production' ? '/xxx' : '/',  
  publicPath: '/', // 不需要子目录
```

路由模式，这里修改为 history，router/index.ts

```
const createRouter = () => new Router({  
  mode: 'history',
```

push我们代码即可，观察workflow执行情况：github - actions

Event ▾	Status ▾	Branch ▾	Actor ▾			
✓ PRIVATE_KEY				Build app and deploy to aliyun #10: Commit 1c8ffb1 pushed by 57code	ci-test	📅 1 hour ago 🕒 3m 20s ...

执行日志

Build app and deploy to aliyun / build		Search logs	< > ...
succeeded 1 hour ago in 3m 4s			
▶ ✓ Set up job			2s
▶ ✓ Checkout			1s
▶ ✓ use Node.js 13.2.0			3s
▶ ✓ Yarn install			1m 6s
▶ ✓ Yarn build			1m 35s
▶ ✓ Deploy to Aliyun			17s
▶ ✓ Post Checkout			0s
▶ ✓ Complete job			0s

验证部署结果

```
cd kkb-honor-admin/client  
ls
```

配置nginx

配置一个website

```
cd /etc/nginx/sites-enabled
ls
vi kkb-honor-admin
```

配置如下:

```
server {
    # 端口号: 此处没有配置子域名和publicPath, 故以端口号区分
    listen 8080;

    # 使用子域名, 需要在阿里云配置域名解析
    # server_name admin.josephxia.com;
    # 使用IP地址
    server_name 47.98.252.43

    location / {
        # history回退
        try_files $uri $uri/ /index.html;
        # 根目录
        root /root/kkb-honor-admin/client/dist;
    }
}
```

重新读取配置

```
nginx -s reload
```

测试~

接口服务器配置

开发时我们使用mock-server做测试, 生产中应该会有独立的api服务器, 只需在nginx配置代理即可。

添加接口代理,

```
location /mock-api/v1 {
    proxy_pass http://127.0.0.1:9528;
    proxy_redirect off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

客户端配置：修改 `VUE_APP_BASE_API`，`.env.production`

```
VUE_APP_BASE_API = 'http://47.98.252.43:8080/mock-api/v1'
```

上面的localhost修改为接口服务器ip或域名即可

由于我们目前没有服务端代码，暂且将mock中的代码打包上传至服务器代替

创建依赖描述文件，`mock/package.json`

```
{
  "name": "mock",
  "version": "1.0.0",
  "description": "",
  "main": "mock-server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "compression": "^1.7.4",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "faker": "^4.1.0",
    "morgan": "^1.9.1",
    "swagger-router-express": "^3.1.2",
    "tslib": "^2.0.0",
    "yamljs": "^0.3.0"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

添加ci配置，`.github/workflow/deploy.yml`

```
# 接口代码打包
- name: API build
  run: cd mock && tsc -P tsconfig.json && cp swagger.yml ./dist && cp
package.json ./dist
# 部署到阿里云
- name: Deploy to Aliyun
  uses: easingthemes/ssh-deploy@v2.1.1
  env:
    # 私钥
    SSH_PRIVATE_KEY: ${ secrets.PRIVATE_KEY }
```

```
# 源目录
SOURCE: "./mock/dist"
# 服务器ip
REMOTE_HOST: "47.98.252.43"
# 用户
REMOTE_USER: "root"
# 目标地址
TARGET: "/root/kkb-honor-admin/server"
```

启动api服务

```
cd kkb-honor-admin/server/dist
npm i
pm2 start mock-server.js
```

~ 测试

Kaikeba
开课吧