

Rancher下的CI/CD实践

张文哲@SodaLab

- 为什么需要CI/CD
- 容器化及Rancher
- Rancher2.X对CI/CD的支持
- 实践
 - gitlab和rancher对接
 - rancher流水线设置
 - 构建配置
 - 镜像制作配置
 - 部署配置

为什么需要CI/CD

- 让我们提交代码后，从容的点支烟，一支烟的时间，新代码已经部署完成

容器化及Rancher

- 小粒度的应用隔离
- 完整的应用管理
- 自动的灰度发布
- ○ ○ ○ ○ ○ ○

Rancher2.X对CI/CD的支持

新项目开启建议流程

- gitlab创建工程
- 本地下载工程
- IDE配置新建文件模版，增加描述性内容，如作者、时间等
- 工程目录中新建项目
- 提交代码
- 配置CI/CD
- 验证发布成果

图文教程

gitlab和rancher对接

- rancher->集群->项目->工作负载->流水线->设置代码库->认证同步代码库->gitlab
- gitlab->Avatar->setting->Applications
- 将rancher回调地址等填入gitlab，将生成的信息填回rancher
- rancher中点击授权，弹出gitlab认证窗口，点击Authorize

Application: rancher-demo

Application ID	7fa0e0abbaec5d053241	
Secret	dc7f3d1026ef38e42f6t	
Callback URL	https://console.rancher.lab.sodacar.com/verify-auth	

EditDestroy

Add new application

Name

rancher-demo

Redirect URI

https://console.rancher.lab.sodacar.com/verify-auth

Use one line per URI

Scopes

☐ api

Grants complete read/write access to the API, including all groups and projects.

☐ read_user

Grants read access to the user's profile information through the Open API endpoint.

2. 完成配置，并使用您的gitlab账号进行验证

应用ID *

7fa0e0abbaec5d05324f1771f8246f248426ac5fb3af6c3348b7a04c8e19f312

将新创建应用的Application Id和Secret复制粘贴到对应的输入框中。

密钥 *

.....

☒ 使用私有gitlab企业版部署

☐ 使用https连接

地址 *

sodagitlab.sodacar.com

授权

rancher流水线设置

- rancher->流水线->设置代码仓库
- 找到需要的项目，选中“启用”，点击项目列表底部“完成”
- 返回流水线项目列表，选择项目->设置
- 通过添加阶段按钮，依次添加“构建”、“镜像发布”、“部署”三个阶段

删除 

搜索

☐ 状态  名称 

最后运行 

☐ Untriggered soda-lab/test-ci

n/a



- 运行 
- 编辑配置 
- 查看/编辑YAML 
- 设置 
- API查看 
- 删除 

修改流水线配置

代码库

http://sodagitlab.sodacar.com/soda-lab/test-ci.git

用户



张文哲 (zhangwenzhe)

分支

dev

流水线配置



完成

取消

构建配置

- 编辑步骤
- 如果涉及到私有Maven仓库，需要自己制作镜像，将setting.xml打包进去
- 依次填入基础镜像和编译命令

编辑步骤

移除

步骤类型 *

运行脚本

基础镜像 *

docker-registry.sodacar.com:5000/lab/openjdk-maven:maven-3.6.0-jdk-8-alpine-springboot-2.1.3 (Custom)

脚本 *

1 mvn clean install -Dmaven.test.skip=true

显示高级选项

保存

取消

```
FROM maven:3.6.0-jdk-8-alpine
MAINTAINER zhangwenzhe

COPY src /opt/project/src
COPY pom.xml /opt/project
COPY maven-settings-dev.xml /root/.m2/settings.xml

WORKDIR /opt/project/

# 下载依赖包，最终一并打包
RUN mvn clean
RUN mvn compile
RUN mvn install
RUN rm -rf /opt/project/
RUN rm -rf /root/.m2/repository/com/sodacar/

# maven工作测试命令
#CMD ["mvn","compile"]

CMD ["mvn"]
```


镜像制作配置

- 配置镜像仓库（https要绑定权威证书）
- 配置镜像发布步骤



dev
Lab

工作负载

应用商店

资源

命名空间

成员



工作负载

负载均衡

服务发现

数据卷

重新部署

暂停编排

下载YAML

删



状态

名称

命名空间: car-hailing



Active

platform-api
80/http, 32287/tcp



Active

platform-auth
80/http, 31692/tcp

告警



证书



配置映射



日志



流水线



镜像库凭证

密文



导入YAML

部署服务

搜索

镜像

Pod伸缩



添加镜像库凭证

名称

添加描述

docker-registry.sodacar.com

作用域

☒ 此项目所有命名空间

☐ 单个命名空间

地址

☐ DockerHub

☐ Quay.io

☒ 自定义:

docker-registry.sodacar.com:5000

用户名

密码

保存

取消

编辑步骤

移除

步骤类型 *

构建并发布镜像

▼

Dockerfile路径 *

./Dockerfile

镜像名称 *

lab/test-ci:\${CICD_EXECUTION_SEQUENCE}

☒ 推送镜像到远端镜像仓库。

镜像库 *

docker-registry.sodacar.com:5000

▼

显示高级选项

保存

取消

部署配置

- 编写配置文件
 - 创建工作负载
 - 创建服务
- 配置发布步骤


```
1  kind: Service
2  apiVersion: v1
3  metadata:
4    name: test-ci-service
5    namespace: test
6  spec:
7    selector:
8      app: test-ci
9    type: NodePort
10   ports:
11     - protocol: TCP
12       port: 8080
13       targetPort: 8080
14   ---
15   apiVersion: apps/v1
16   kind: Deployment
17   metadata:
18     name: test-ci
19     namespace: test
20     labels:
21       app: test-ci
22   spec:
23     replicas: 2
24     selector:
25       matchLabels:
26         app: test-ci
27     template:
28       metadata:
29         labels:
30           app: test-ci
31       spec:
32         containers:
33           - env:
34             - name: MYSQL_HOST
35               value: mysql.sodacar.com
36             image: ${CICD_IMAGE}:${CICD_EXECUTION_SEQUENCE}
37             imagePullPolicy: Always
38             name: test-ci
39             ports:
```

编辑步骤

移除

步骤类型 *

部署YAML

YAML路径 *

./deployment.yaml

显示高级选项

保存

取消

注意事项

- Rancher环境，一定要安装官方要求的操作系统版本和Docker版本
- 添加节点，一定要分别填写局域网和公网IP
- 镜像仓库务必绑定权威证书，自签名证书会有很多麻烦
- 负载均衡要用自定义域名做绑定，xip.io没有备案

科技改变生活 技术促进生产

2019.3.5