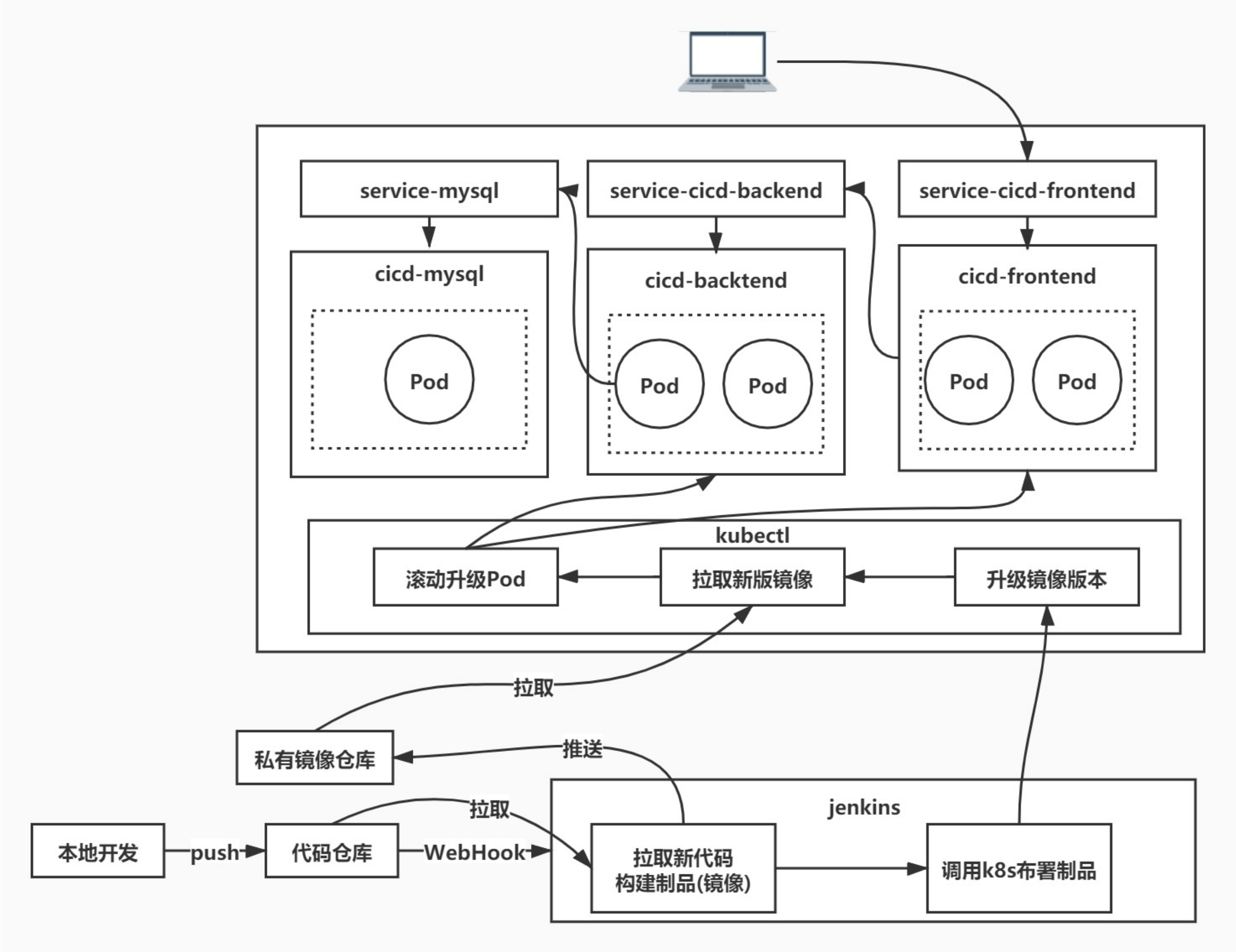
link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=87 sentences=120, words=747

## 1.项目仓库 #

- 前端技术栈为 React + craco
- 后端技术栈为 MySQL + eggjs
- cicd-frontend (https://gitee.com/zhufengpeixun/cicd-frontend)
- cicd-backend (https://gitee.com/zhufengpeixun/cicd-backend)



## 2.添加一个节点 #

- 增加一个node2的节点

## 2 布署MSYQL #

### 2.1 设置污点 #

- Node2节点机器只用于部署MySQL服务
- 可以给节点加污点，只用来布署MySQL服务
- node1增加webserver的污点
- node2增加mysql的污点

```
kubectl taint nodes node1 webserver:NoSchedule
kubectl taint nodes node2 mysql=true:NoSchedule
```

### 2.2 创建数据目录 #

- 在本地创建MYSQL数据文件夹然后挂载进 MySQL容器
- 以方便MySQL 数据可以持久化
- 在node2上创建mysql数据文件夹
- 此文件夹要为 &#x7A7A;，不然启动MYSQL会失败

```
mkdir /var/lib/mysql
```

- 将root密码存入 secret 内保存

```
kubectl create secret generic mysql-auth --from-literal=username=root  --from-literal=password=root
```

vi deployment-cicd-mysql.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: cicd-mysql
spec:
  replicas: 1
  selector:
    matchLabels:
      app: cicd-mysql
  template:
    metadata:
      labels:
        app: cicd-mysql
    spec:
      tolerations:
      - key: "mysql"
        operator: "Equal"
        value: "true"
        effect: "NoSchedule"
      containers:
      - name: cicd-mysql
        image: mysql:5.7
        imagePullPolicy: IfNotPresent
        args:
        - "--ignore-db-dir=lost+found"
        ports:
        - containerPort: 3306
        volumeMounts:
        - name: mysql-data
          mountPath: "/var/lib/mysql"
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-auth
              key: password
      volumes:
      - name: mysql-data
        hostPath:
          path: /var/lib/mysql
          type: Directory
```

```
[root@master project]# kubectl apply -f deployment-cicd-mysql.yaml
deployment.apps/cicd-mysql created

//查看容器内的日志 方便查看报错
kubectl get pods
kubectl describe pods cicd-mysql-bcb77c759-bdrd8
kubectl logs cicd-mysql-6cbd4f95-g64hh
```

vi service-cicd-mysql.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: service-cicd-mysql
spec:
  selector:
    app: cicd-mysql
  ports:
  - protocol: TCP
    port: 3306
    targetPort: 3306
  type: NodePort
```

- 让配置文件生效

```
kubectl apply -f service-cicd-mysql.yaml
```

- 连接数据库初始化数据
- -h 为任意节点的公网或内网IP

```
mysql -h172.31.178.169 -P32636 -uroot -proot
mysql -h118.190.156.138 -P32636 -uroot -proot
```

```
create database cicd;
use cicd;
CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT 'ID',
  `name` varchar(255) NOT NULL COMMENT '姓名',
  `age` int(11) NOT NULL COMMENT '年龄',
  `sex` varchar(255) NOT NULL COMMENT '性别：1男 2女',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
```

## 3 布署后端 #

### 3.1 新建jenkins项目 #

- cicd-backend
- 设置git源码地址
- 配置git私钥
- 配置DOCKER_LOGIN_USERNAME和DOCKER_LOGIN_PASSWORD

### 3.2 添加构建布署 #

```

time=$(date "+%Y%m%d%H%M%S")
npm install --registry=https://registry.npm.taobao.org
docker build -t 115.28.139.92:8082/cicd-backend:$time .
docker login -u $DOCKER_LOGIN_USERNAME -p $DOCKER_LOGIN_PASSWORD 115.28.139.92:8082
docker push 115.28.139.92:8082/cicd-backend:$time
```

### 3.3 配置信息 #

### 3.3.1 数据库地址 #

vi mysql.config.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-config
data:
  host: "service-cicd-mysql"
  port: "3306"
  database: "cicd"
```

```
kubectl apply -f  mysql.config.yaml
```

### 3.3.2 数据库账号 #

vi mysql-auth.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysql-auth
stringData:
  username: root
  password: root
type: Opaque
```

```
kubectl apply -f  mysql.config.yaml
```

### 3.3.3 私有仓库认证 #

```
kubectl create secret docker-registry private-registry \
--docker-username=admin \
--docker-password=admin123 \
--docker-email=admin@example.org \
--docker-server=115.28.139.92:8082
```

### 3.3.4 后台Deployment #

vi cicd-backend.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: cicd-backend
spec:
  selector:
    matchLabels:
      app: cicd-backend
  replicas: 1
  template:
    metadata:
      labels:
        app: cicd-backend
    spec:
      imagePullSecrets:
      - name: private-registry
      containers:
      - name: cicd-backend
        imagePullPolicy: Always
        image: "115.28.139.92:8082/cicd-backend:20210321202052"
        ports:
        - containerPort: 7001
        env:
        - name: MYSQL_HOST
          valueFrom:
            configMapKeyRef:
              name: mysql-config
              key: host
        - name: MYSQL_PORT
          valueFrom:
            configMapKeyRef:
              name: mysql-config
              key: port
        - name: MYSQL_DATABASE
          valueFrom:
            configMapKeyRef:
              name: mysql-config
              key: database
        - name: MYSQL_USER
          valueFrom:
            secretKeyRef:
             name: mysql-auth
             key: username
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
             name: mysql-auth
             key: password
```

```
kubectl apply -f  cicd-backend.yaml
```

### 3.3.5 后台Service #

- vi service-cicd-backend.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: service-cicd-backend
spec:
  selector:
    app: cicd-backend
  ports:
  - protocol: TCP
    port: 7001
    targetPort: 7001
  type: NodePort
```

```
kubectl apply -f  service-cicd-backend.yaml
curl http:
```

## 4 布署前端 #

### 4.1 安装编译器 #

```
yum -y install gcc gcc-c++ kernel-devel
```

### 4.1 新建jenkins项目 #

- cicd-frontend
- 设置git源码地址
- 配置git私钥
- 配置DOCKER_LOGIN_USERNAME和DOCKER_LOGIN_PASSWORD

### 4.1 配置构建步骤 #

```

time=$(date "+%Y%m%d%H%M%S")
npm install --registry=https://registry.npm.taobao.org
npm run build
docker build -t 115.28.139.92:8082/cicd-frontend:$time .
docker login -u $DOCKER_LOGIN_USERNAME -p $DOCKER_LOGIN_PASSWORD 115.28.139.92:8082
docker push 115.28.139.92:8082/cicd-frontend:$time
```

### 4.2 配置构建步骤 #

vi cicd-frontend.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: cicd-frontend
spec:
  selector:
    matchLabels:
      app: cicd-frontend
  replicas: 1
  template:
    metadata:
      labels:
        app: cicd-frontend
    spec:
      imagePullSecrets:
      - name: private-registry
      containers:
      - name: cicd-frontend
        image: 115.28.139.92:8082/cicd-frontend:20210321204724
```

```
kubectl apply -f  cicd-frontend.yaml
```

vi service-cicd-frontend.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: service-cicd-frontend
spec:
  selector:
    app: cicd-frontend
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: NodePort
```

```
kubectl apply -f  service-cicd-frontend.yaml
```

```
kubectl get svc
```

```
http:
```

## 5.集成jenkins #

### 5.1 添加全局配置文件 #

- 系统管理=>Managed files=>Add a new Config=>Custom file
- Name设置为k8s-config
- 把master上的 `~/.kube/config`拷贝到Content中

### 5.2 安装kubectl #

```
cat < /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=http:
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=http:
        http:
EOF
yum install -y kubectl
```

### 5.3 绑定配置文件 #

- 打开项目配置
- 选择绑定=>Provide Configuration files=>Target选择 `k8s-config`=>Target输入 `k8s-config.yaml`

### 5.4 shell #

- 使用 `kubectl set image` 命令快速设置镜像地址版本
- 格式为：kubectl set image deployment/[deployment名称] [容器名称]=[镜像版本]

```
#!/bin/bash
time=$(date "+%Y%m%d%H%M%S")
npm install --registry=https://registry.npm.taobao.org
docker build -t 115.28.139.92:8082/cicd-backend:$time .
docker login -u $DOCKER_LOGIN_USERNAME -p $DOCKER_LOGIN_PASSWORD 115.28.139.92:8082
docker push 115.28.139.92:8082/cicd-backend:$time
+kubectl --kubeconfig=k8s-config.yaml set image deployment/cicd-backend cicd-backend=115.28.139.92:8082/cicd-backend:$time
```

```
deployment.apps/cicd-backend image updated表示更新成功
```

## 6.推送触发构建 #

### 6.1 安装插件 #

- publish over ssh(方便操作远程的服务器)
- gitee
- Last Changes(可视化查看git文件变化)

### 6.2 构建触发器 #

- Gitee webhook触发构建,并记录 `webhook URL`地址
- 生成 Gitee WebHook 密码

## 6.3 配置WebHooks #

- 打开项目的WebHooks管理页面
- 配置 `webhookURL`和 `WebHook &#x5BC6;&#x7801;`

## 6.参考 #

- 强行删除pod

```
kubectl delete pod  cicd-mysql-84795bc9d7-fpjmp       --force --grace-period=0
```