

jenkins&&docker

3 socroot

```
version: '3'
services:
  jenkins:
    image: harbor.iovhm.com/hub/jenkins/jenkins:2.488-jdk17
    container_name: jenkins
    restart: always
    privileged: true
    user: root
    ports:
      - "8080:8080"
    volumes:
      - ./jenkins_home:/var/jenkins_home
      - ./m2:/root/.m2
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/bin/docker:/usr/bin/docker
    environment:
      - TZ=Asia/Shanghai
```

docker

> > > Add Credentials

New credentials

类型

Username with password

范围 ?

全局 (Jenkins, nodes, items, all child items, etc)

用户名 ?

☐ Treat username as secret ?

密码 ?

ID ?

这个ID很重要，后面会用到



描述 ?

Create

全局凭据 (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	名称	类型	描述
 10b8bc82-8b90-4fbb-9f55-d7e3126b35c0	这个ID很重要，后面会用到	Username with password	gitlab
 docker-huaweicloud		Username with password	docker registry huaweicloud

图标 小 中 大

springboot

sh('')('"),sh

shshsh(\$

```
pipeline {
    agent any

    //
    environment {
        // docker
        PROJECT_NAME      ="gzxfzd"
        // docker
```

```

PROJECT_NS      =" vp- park"
// docker
DOCKER_REGISTRY =" swr. cn- south- 1. myhuaweicloud. com"
//  docker
// swr. cn- south- 1. myhuaweicloud. com/vp- whdev/springboot- demo

}

// 
parameters {

    // 
    string(name: "BRANCH", defaultValue: "release", description: ")
    //  docker
    string(name: "VERSION", defaultValue: "v1", description: ")
    //  TAG docker
    string(name: "IMAGE_TAG", defaultValue: "latest", description: "tag")
    // 
    // swr. cn- south- 1. myhuaweicloud. com/vp- whdev/springboot- demo: v1- latest
    choice(
        name: ' PUSH_SERVICE',
        choices: [' ALL', ' jfast- service/jfast- auth', ' jfast- service/jfast- basic', ' jfast-
service/jfast- cloud- docking', ' jfast- devops- center', ' jfast- gateway',
        ' jfast- oa- services', ' jfast- service/jfast- research- center'],
        description: ' 
    )
}

tools {
    //  mvn
    maven "M3"
    // JDKjdk
    jdk "JDK- 8"
}

stages {
    stage(' Build') {
        steps {

```

```
// 设置credentialsId为git仓库ID
// 设置url为git仓库地址
git branch: '${BRANCH}', credentialsId: 'donglietao-vpclub-sz-git', url:
'https://g.vpclub.cn/gzxfzhdw/back.git'
```

```
// 设置pom.xml文件所在目录
// 设置mvn clean compile package -f ./source/pom.xml 命令
// 设置pom文件所在目录
```

```
dir("./"){
    // 设置命令
    // 设置命令
    // 设置命令
    sh ""
```

```
pwd
java -version
mvn clean compile package
```

```
""
```

```
}
```

```
}
```

```
}
```

```
stage("docker"){
    steps{
        script{
            def SERVICE_MAP=[
                "jfast-service/jfast-auth":"jfast-auth",
                "jfast-service/jfast-basic":"jfast-basic",
                "jfast-service/jfast-cloud-docking":"jfast-cloud-docking",
                "jfast-devops-center":"jfast-devops-center",
                "jfast-gateway":"jfast-gateway",
                "jfast-oa-services":"jfast-oa-services",
                "jfast-service/jfast-research-center":"jfast-research-center"
            ]
```

```
// 设置docker仓库credentialsId为docker仓库ID
```

```
withCredentials([usernamePassword(credentialsId: 'docker-huaweicloud',
```

```

passwordVariable: 'docker_password', usernameVariable: 'docker_username')) {
    def conditionVal= "${PUSH_SERVICE}"
    if(conditionVal=="ALL"){
        for(item in SERVICE_MAP.entrySet()){
            def serviceName = "${item.value}"
            def serverPath = "${item.key}"
            dir("./${serverPath}"){
                // ██████████
                // ████████████████████
                // ████████████████████████████████████████
                sh """
                pwd
                echo ${serviceName}

                docker login -u ${docker_username} -p ${docker_password}

                ${DOCKER_REGISTRY}

                docker build -t

                ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}/${serviceName}:${VERSION}-${IMAGE_TAG} .

                docker push

                ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}/${serviceName}:${VERSION}-${IMAGE_TAG}

                """
            }

        }

    }else{
        def serviceName =SERVICE_MAP[ conditionVal]
        def serverPath = conditionVal
        dir("./${serverPath}"){
            // ██████████
            // ████████████████████
            // ████████████████████████████████████████
            sh """
            pwd
            echo ${serviceName}

            docker login -u ${docker_username} -p ${docker_password}

            ${DOCKER_REGISTRY}

            docker build -t

            ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}/${serviceName}:${VERSION}-${IMAGE_TAG} .

            docker push

            ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}/${serviceName}:${VERSION}-${IMAGE_TAG}

            """
        }
    }
}

```



```

string(name: "IMAGE_TAG", defaultValue: "latest", description: "镜像tag")
// 镜像名称
// swr.cn-south-1.myhuaweicloud.com/vp-whdev/springboot-demo:v1-latest
}

tools {
    // Install the Maven version configured as "M3" and add it to the path.
    // 安装maven
    maven "M3"
    // jdk
    // jdk "JDK-8"
}

stages {
    stage('Build') {
        steps {
            // Get some code from a gitlab
            // 设置gitlab凭证
            // 设置url, 设置git分支
            git branch: '${BRANCH}', credentialsId: '10b8bc82-8b90-4fbb-9f55-d7e3126b35c0', url: 'https://git.whdev.vpclub.cn:10081/park-weihai/backend-api.git'

            // Run Maven on a Unix agent.
            // 设置agent
            // 设置pom.xml
            // mvn clean compile package -f ./source/pom.xml
            // 设置pom
            dir("./renren-cloud-tenant"){
                sh """

                echo \$(pwd)
                mvn clean compile package

            }
        }
    }

    stage("docker") {

```

```

steps{
  // docker build [redacted]Dockerfile
  // [redacted]Dockerfile[redacted]dir [redacted]
  // [redacted]Dockerfile[redacted]dir [redacted]
  dir('./renren-cloud-tenant/park-admin/park-admin-server/') {

    // [redacted]docker[redacted]credentialsId[redacted]docker [redacted]
    withCredentials([usernamePassword(credentialsId: 'docker-huaweicloud',
passwordVariable: 'docker_password', usernameVariable: 'docker_username')]) {

      sh """

      echo \$(pwd)
      docker login -u ${docker_username} -p ${docker_password}
      ${DOCKER_REGISTRY}

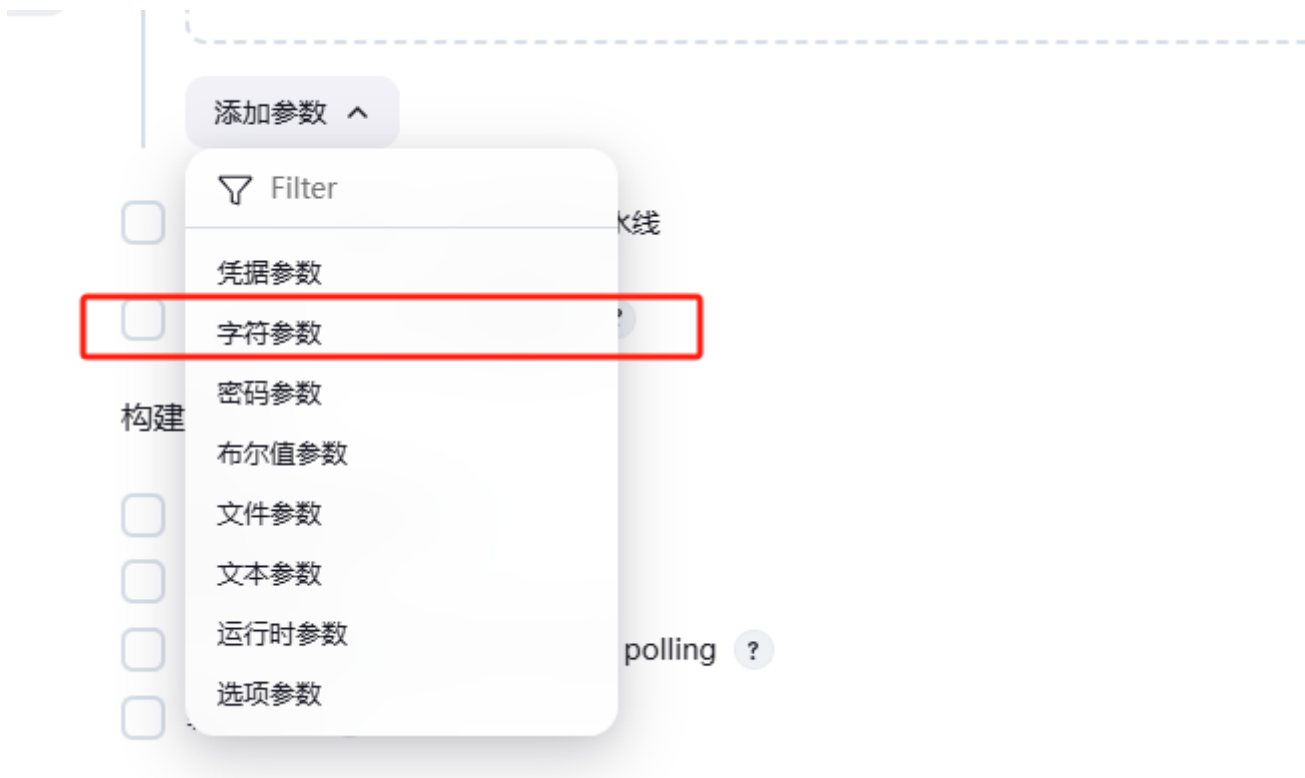
      docker build -t
      ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}:${VERSION}-${IMAGE_TAG} .
      docker push
      ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}:${VERSION}-${IMAGE_TAG}

      """
      ...
    }
  }
}
}
}
}
}
}
}
}

```

[redacted]

[redacted]Pipeline[redacted]



☒ 参数化构建过程 ? 勾选参数化构建复选框

≡ 字符参数 ?

名称 ?
BRANCH 填写与构建参数对应的环境变量

默认值 ?
dev 填入默认值

描述 ?
代码分支
填入参数描述

纯文本 预览

☐ 清除空白字符 ? 自动删除前后的空格

nodejs Pipeline

```
pipeline {  
    agent any  
  
    //   
  
    environment {  
        //   
    }  
}
```

```

PROJECT_NAME      =" nodejs- demo"
// docker
PROJECT_NS        =" vp- whdev"
// docker
DOCKER_REGISTRY  ="swr. cn- south- 1. myhuaweicloud. com"
//  docker
// swr. cn- south- 1. myhuaweicloud. com/vp- whdev/nodejs- demo
// nodejsnodejs
NODE_OPTIONS      = "--max- old- space- size=4096"

}

//
parameters {
    //
    string(name: "BRANCH", defaultValue: "develop", description: ")
    //  docker
    string(name: "VERSION", defaultValue: "v1", description: ")
    //  TAGdocker
    string(name: "IMAGE_TAG", defaultValue: "latest", description: "tag")
    //
    // swr. cn- south- 1. myhuaweicloud. com/vp- whdev/nodejs- demo: v1- latest
}

stages {
    stage('Build') {
        steps {
            // Get some code from a gitlab
            //  credentialsIdgit
            //  url, git
            git branch: '${BRANCH}', credentialsId: '10b8bc82-8b90-4fbb-9f55-
d7e3126b35c0', url: 'https://git.whdev.vpclub.cn:10081/park-weihai/front-web.git'

            // build
            //  nodejsnodejs
            nodejs('node-18.20') {
                sh '''

                echo $(pwd)

```

```

        node -v
        npm -v
        npm install --registry=https://registry.npmmirror.com
        npm run build:test

        ...
    }
}

stage("docker"){
    steps{

        // docker build [context]Dockerfile
        // [context]Dockerfile[context]dir [context]
        // [context]Dockerfile[context]dir [context]

        withCredentials([usernamePassword(credentialsId: 'docker-huaweicloud',
passwordVariable: 'docker_password', usernameVariable: 'docker_username')]) {

            sh '''

            echo $(pwd)
            docker login -u ${docker_username} -p ${docker_password}
${DOCKER_REGISTRY}
            docker build -t
${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}:${VERSION}-${IMAGE_TAG} .
            docker push ${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}:${VERSION}-${
${IMAGE_TAG}

            ...

        }

    }

}

}

}

}

```



nodejs

gyp sass jenkins docker

```
FROM harbor.iovhm.com/hub/node: 14. 21. 3 AS build
WORKDIR /data
COPY ./ /data
RUN cd /data && \
    yarn install --registry=https://registry.npmmirror.com && \
    yarn run build:prod && \
    true

FROM harbor.iovhm.com/hub/nginx: 1. 14. 2
COPY --from=build /data/dist/ /usr/share/nginx/html/management
```

- AS build
- --from=build

docker