

jenkins&&docker

```
3 socroot
```

```
version: '3'
services:
  jenkins:
    image: harbor.iovbm.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
```

```
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		Username with password	gitlab
docker-huaweicloud		Username with password	docker registry huaweicloud

这个ID很重要, 后面会用到

图标 小 中 大

pipeline

pipeline


```

// 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
// DOCKER
DOCKER_IDENTITY   ='docker-huaweicloud'

}

//
parameters {

    //
    choice(name: 'GIT_BRANCH', choices: ['development', 'release'],description: '
dev, release')
// TAGdocker
//
// swr.cn-south-1.myhuaweicloud.com/vp-whdev/springboot-demo: test
choice(name: "IMAGE_TAG", choices: ["test", "prod"], description: "tag")

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"
// JDK
jdk "JDK-8"
}

stages {
  stage('Build') {
    steps {

      // credentialsId
      // url, git
      git branch: "${GIT_BRANCH}", credentialsId: "${GIT_IDENTITY}", url:
"${GIT_URL}"

      //
      // pom.xml dir
      // mvn clean compile package -f ./source/pom.xml pom
      // pom dir

      dir("./"){
        //
        //
        //
        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"
    ]

    // docker credentialsId docker
    withCredentials([usernamePassword(credentialsId: "${DOCKER_IDENTITY}",
passwordVariable: 'docker_password', usernameVariable: 'docker_username')]) {
        def conditionVal= "${PUSH_SERVICE}"
        def branch = "${GIT_BRANCH}"
        if(conditionVal=="ALL"){
            for(item in SERVICE_MAP.entrySet()){
                def serviceName = "${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}: ${IMAGE_TAG}- ${BUILD_NUMBER} .
                        docker push
${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}/${serviceName}: ${IMAGE_TAG}- ${BUILD_NUMBER}

                        """
                }

            }
        }else{
            def serviceName =SERVICE_MAP[conditionVal]
            def serverPath = conditionVal

```



```

// docker
PROJECT_NAME = 'crabapple-drink'

// 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
// DOCKER
DOCKER_IDENTITY = 'docker-huaweicloud'

}

//
parameters {

    //
    choice(name: "GIT_BRANCH", choices: ["development", "release"], description: ")
    // TAGdocker
    //
    // swr.cn-south-1.myhuaweicloud.com/vp-park/springboot-demo: test
    choice(name: "IMAGE_TAG", choices: ["test", "prod"], description: "tag")
}

tools {

    // mvn
    maven "M3"
    // jdkjdk
    // jdk "JDK-8"
}

stages {
    stage('Build') {
        steps {

            // Get some code from a gitlab
            // credentialsIdgitIM

```

```

    // url, git
    git branch: "${GIT_BRANCH}", credentialsId: "${GIT_IDENTITY}", url:
"${GIT_URL}"

    // Run Maven on a Unix agent.
    //
    // pom.xml dir
    // mvn clean compile package -f ./source/pom.xml pom
    // pom dir

    dir("./renren-cloud-tenant"){
        sh ""

        echo \$(pwd)
        mvn clean compile package

        ""
    }
}

stage("docker"){
    steps{
        // docker build Dockerfile
        // Dockerfile dir
        // Dockerfile dir
        dir('./renren-cloud-tenant/park-admin/park-admin-server/') {

            // docker credentialsId docker
            withCredentials([usernamePassword(credentialsId: "${DOCKER_IDENTITY}",
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}/${PROJECT_NAME}:${IMAGE_TAG}-${BUILD_NUMBER} .

                docker push

```

```
${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}: ${IMAGE_TAG}- ${BUILD_NUMBER}
```

```
        ""  
    }  
  }  
}  
}  
}
```



Pipeline

添加参数 ^

Filter

凭据参数

字符参数

密码参数

布尔值参数

文件参数

文本参数

运行时参数

选项参数

构建

polling ?

☰ 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 {
    // GIT[]
    GIT_URL      = "https://g.vpclub.cn/park/sdyk/front-web-admin.git"
    // GIT []
    GIT_IDENTITY = "donglietao-vpclub-sz-git"
    // []docker[]
    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
    // DOCKER []
    DOCKER_IDENTITY = "docker-huaweicloud"
    // nodejs[]nodejs[]
    NODE_OPTIONS    = "--max-old-space-size=4096"
  }
}
```

```

// 部署
parameters {
    // 分支
    choice(name: "BRANCH", choices: ["development", "release"], description: "分支")
    // 镜像TAG部署docker部署
    choice(name: "IMAGE_TAG", choices: ["test", "prod"], description: "镜像tag")
    // 部署环境
    // swr.cn-south-1.myhuaweicloud.com/vp-whdev/nodejs-demo: test
}

stages {
    stage('Build') {
        steps {
            // Get some code from a gitlab
            // 部署凭证Id部署git部署
            // 部署url部署部署git部署
            git branch: "${BRANCH}", credentialsId: "${GIT_IDENTITY}", url: "${GIT_URL}"

            // build
            // 部署nodejs部署部署nodejs部署
            nodejs('node-18.20') {
                sh """

                echo \$(pwd)
                node -v
                npm -v

                # 部署npm install部署
                # npm cache clean --force
                # rm -rf node_modules
                # rm package-lock.json
                # rm yarn.lock

                npm add yarn --registry=https://registry.npmirror.com

                ./node_modules/.bin/yarn -v
                ./node_modules/.bin/yarn install --registry https://registry.npmirror.com
            }
        }
    }
}

```

```

# yarn
# npm install --registry=https://registry.npmirror.com

if [ "${IMAGE_TAG}" = "prod" ]; then
    ./node_modules/.bin/yarn run build:prod
elif [ "${IMAGE_TAG}" = "test" ]; then
    ./node_modules/.bin/yarn run build:test
else
    echo "IMAGE_TAG is neither 'prod' nor 'test'. No build script
executed."

fi

"""
}
}
}

stage("docker"){
    steps{

        // docker build Dockerfile
        // Dockerfiledir
        // Dockerfiledir

        withCredentials([usernamePassword(credentialsId: "${DOCKER_IDENTITY}",
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}/${PROJECT_NAME}:${IMAGE_TAG}-${BUILD_NUMBER} .
            docker push
${DOCKER_REGISTRY}/${PROJECT_NS}/${PROJECT_NAME}:${IMAGE_TAG}-${BUILD_NUMBER}

            """

        }
    }
}

```

```
    }
  }
}
```

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

#55

2024 03:06:54

2026 11:19:45