

1. 代码要符合maven编译规范
2. 代码提交至git（本例中使用[GitHub]:<https://github.com/zhangrandi/testtest>）
3. 将需要部署的静态资源上传至服务器指定目录（此例中为/home/zhang/static）
4. 进入jenkins点击Build with Parameters



Pipe

需要如下:



5. 依次输入相应参数，点击开始构建

tag: git对应分支/tag
env: 选择对应环境 (prod、test)
version: 版本号

Pipeline test

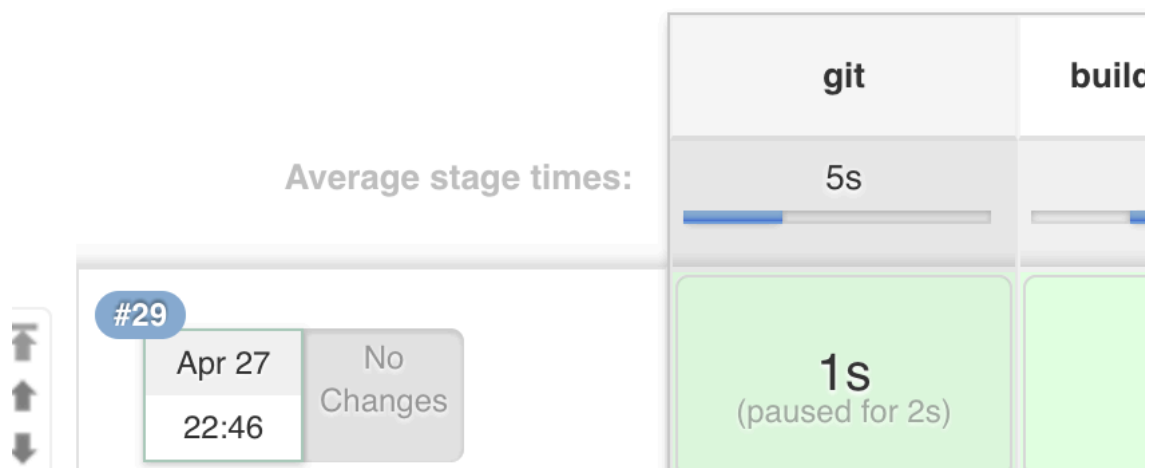
需要如下参数用于构建项目:

tag	<input type="text" value="master"/>
env	<div><div>prod</div><div>生产 : prod 测试 : test</div></div>
version	<div><div>1.0</div><div>输入版本号</div></div>

开始构建

6. 等待部署完成

Stage View



Hello World!



Back to Dashboard



Status

静态资源 (1.png)



Changes

附:

- pipeline代码

```
node("master"){
  stage 'git'
  git url:'https://github.com/zhangrandl/testtest.git', branch:'${tag}'
  input "是否开始测试? " // 这里会阻塞等待用户的输入

  stage 'build & Deploy'
  echo "开始部署编译java程序"
  sh 'cd /var/lib/jenkins/workspace/test && mvn clean package install -Dmaven.test.skip=true'
  sh "rm -rf /home/jenkins/a.war"
  sh "mv /var/lib/jenkins/workspace/test/target/a.war /home/jenkins/test"
  echo "-----"
  sh "/home/jenkins/develop.sh"
  sleep 3

  echo "开始部署静态资源"

  sh "unzip -o /home/jenkins/static/static.zip -d /home/jenkins/${version}/"
  sh "rm -rf /usr/share/nginx/html"
  sh "ln -s /home/jenkins/${version} /usr/share/nginx/html"

  echo "部署成功."
}
```

- develop代码

```
docker stop `docker ps | grep testlawk '{print $1}'`
docker rm `docker ps -a | grep testlawk '{print $1}'`
docker rmi `docker images | grep none | awk '{print $3}'`
docker build -t test:1.0 /home/jenkins/test
docker run -v /var/log/tomcat/test/:/usr/local/tomcat/logs/ -d -it -p 8081:8080 test:1.0
```

- Dockerfile

```
FROM docker.io/tomcat
COPY a.war /usr/local/tomcat/webapps/
EXPOSE 8080
CMD ["catalina.sh", "run"]
```