jenkins+docker完成自动化发布环境

1、为项目添加Dockerfile

|  |
| --- |
| FROM java:8 VOLUME /tmp ENV TZ=Asia/Shanghai RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone ADD build/libs/\*.jar app.jar ENV JAVA\_OPTS="" ENTRYPOINT exec java $JAVA\_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app.jar |

FROM java:8 拉去一个包含java8的docker镜像

ENV和RUN是设置时区

ADD 添加jar包到docker中

ENV java参数

ENTRYPOINT exec 运行这个docker实例中的jar

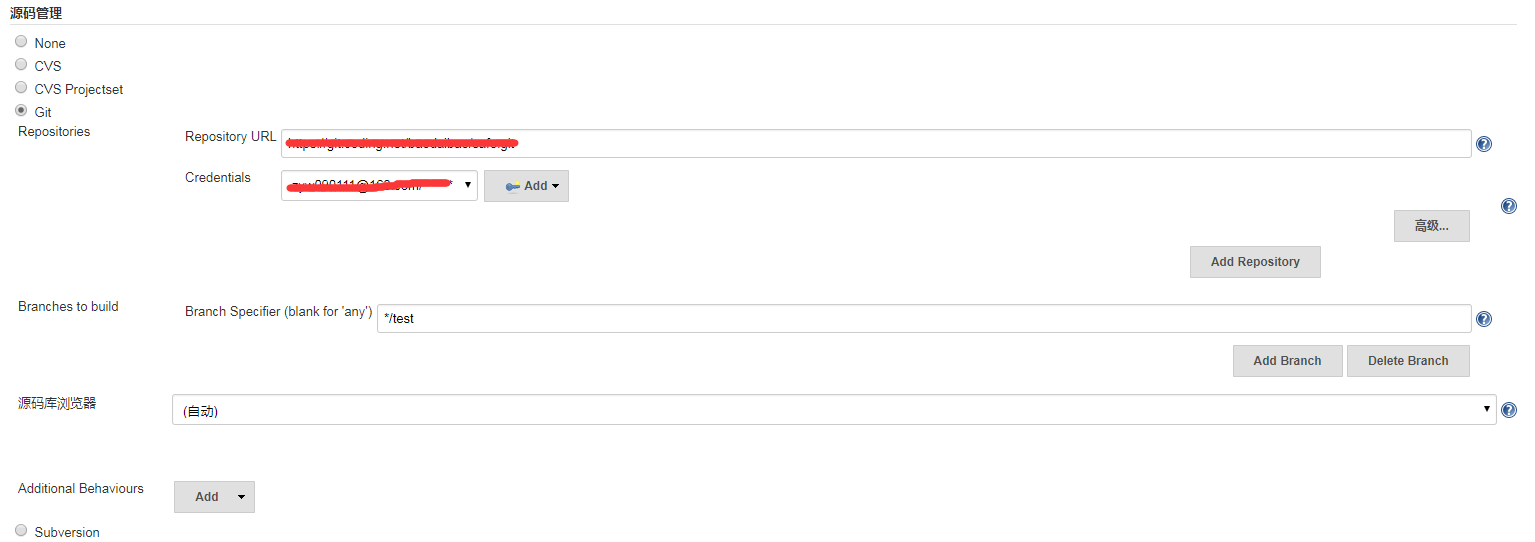
2、创建jenkins构建item



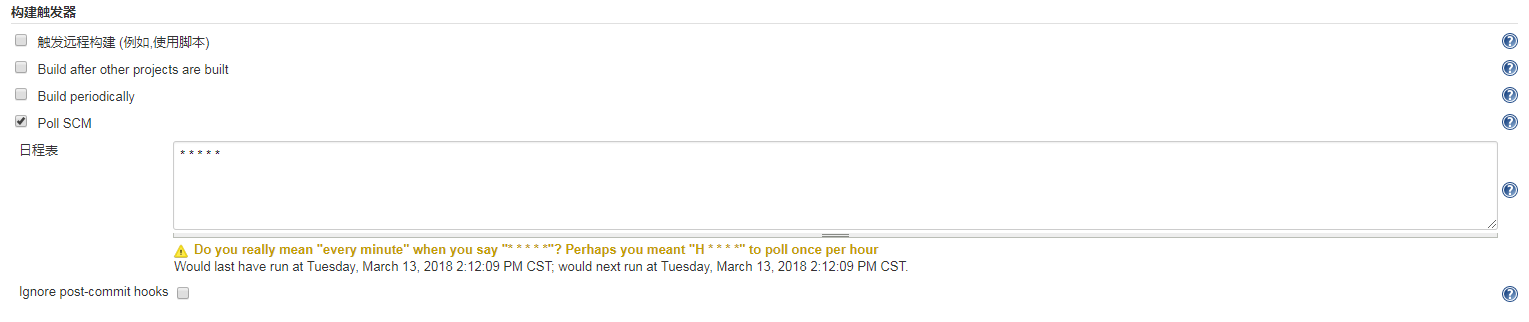
创建一个item选择自由风格的软件项目



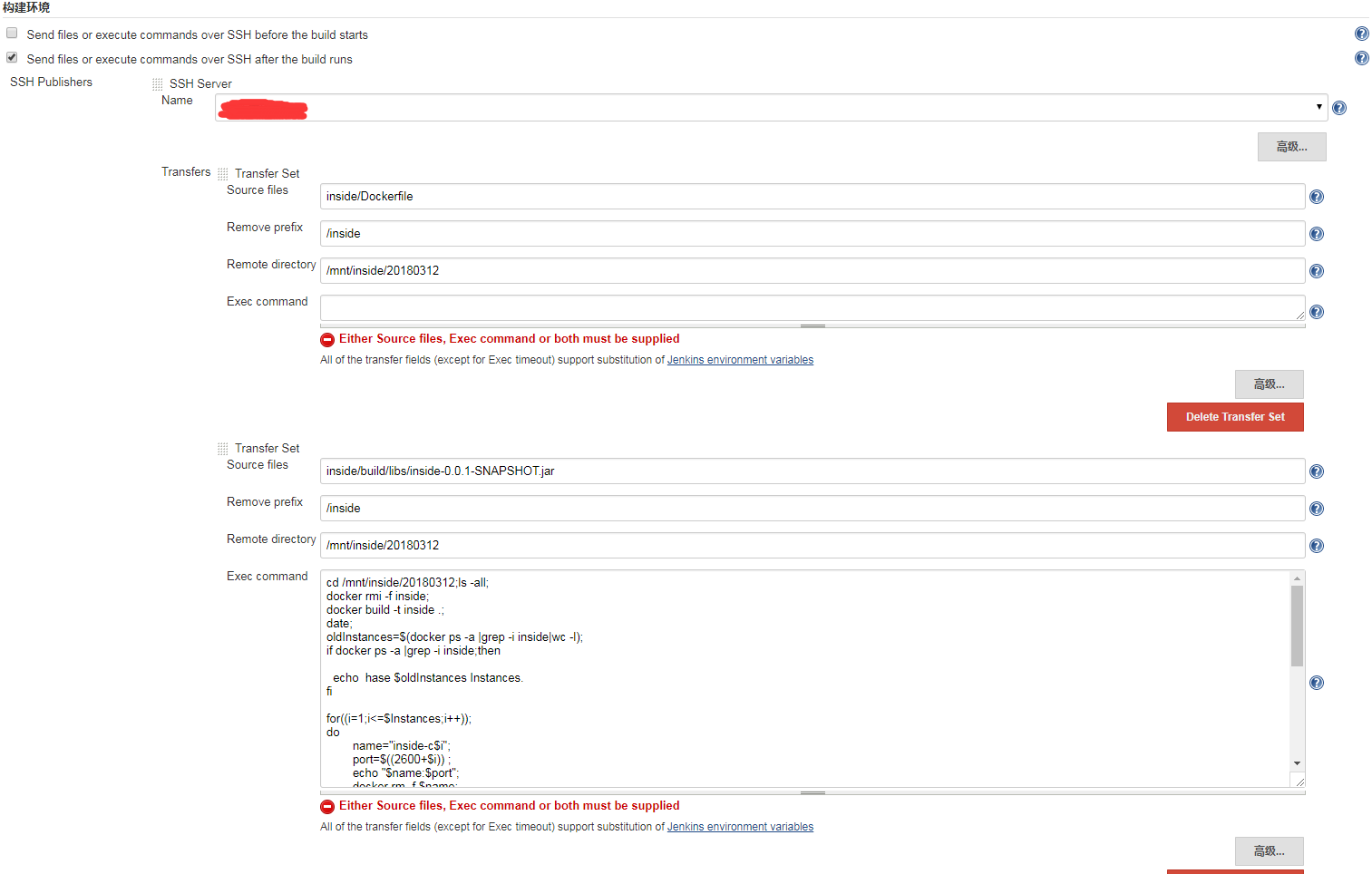
选择参数化构建，选择**String Parameter**选项输入参数名称和值，后面要用



根据自己的代码管理方式拉取代码\*/test为分支



定时任务触发器，定时到代码分支中检测代码是否有更新



配置SSH Publishers

SSH Server中配置的是你需要把打好的jar文件发送到的服务器配置

Transfers是需要发送的文件

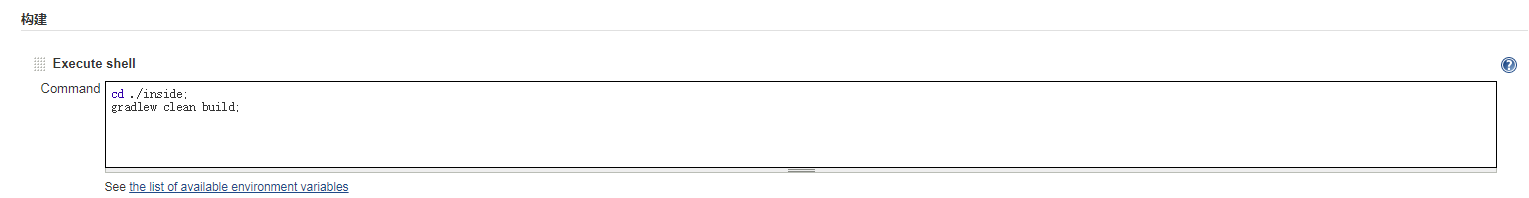
第一个是发送Dockerfile文件

第二个是发送打好的jar包

发送jar包后需要启动docker实例

shell脚本如下：

|  |
| --- |
| cd /mnt/inside/20180312;ls -all;  docker rmi -f inside;  docker build -t inside .;  date;  oldInstances=$(docker ps -a |grep -i inside|wc -l);  if docker ps -a |grep -i inside;then  echo hase $oldInstances Instances.  fi  for((i=1;i<=$Instances;i++));  do  name="inside-c$i";  port=$((2600+$i)) ;  echo "$name:$port";  docker rm -f $name;  docker run -v /mnt/inside:/mnt/inside -v /etc/localtime:/etc/localtime \  --restart=on-failure:10 \  -e "SPRING\_PROFILES\_ACTIVE=dev2" -p $port:2006 --name $name -d inside;  if [ "$i" -lt "$Instances" ];then  sleep 60;  fi  done  for((i=$Instances+1;i<=$oldInstances;i++));  do  name="inside-c$i";  echo rm $name;  docker rm -f $name;  done  docker ps; |



拉取代码完成后的构建脚本