Andrei Matveenko

1. Ansible

ansible.Dockerfile:

FROM sbeliakou/centos:7.2

RUN yum install -y python-pip python-devel gcc epel-release yum-plugin-ovl openssl-devel && \

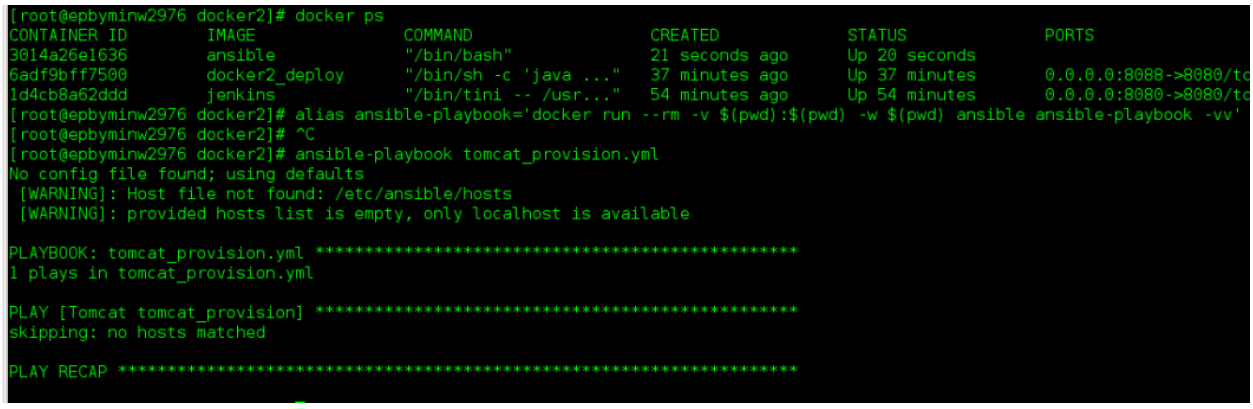
pip install ansible

Ansible container usage:

alias ansible-playbook='docker run --rm -v $(pwd):$(pwd) -w $(pwd) ansible ansible-playbook -vv'

ansible-playbook playbook.yml

Output:



Jenkins

docker run -d -p 8080:8080 -p 50000:50000 -v /home/student/docker/docker2:/var/jenkins_home jenkins

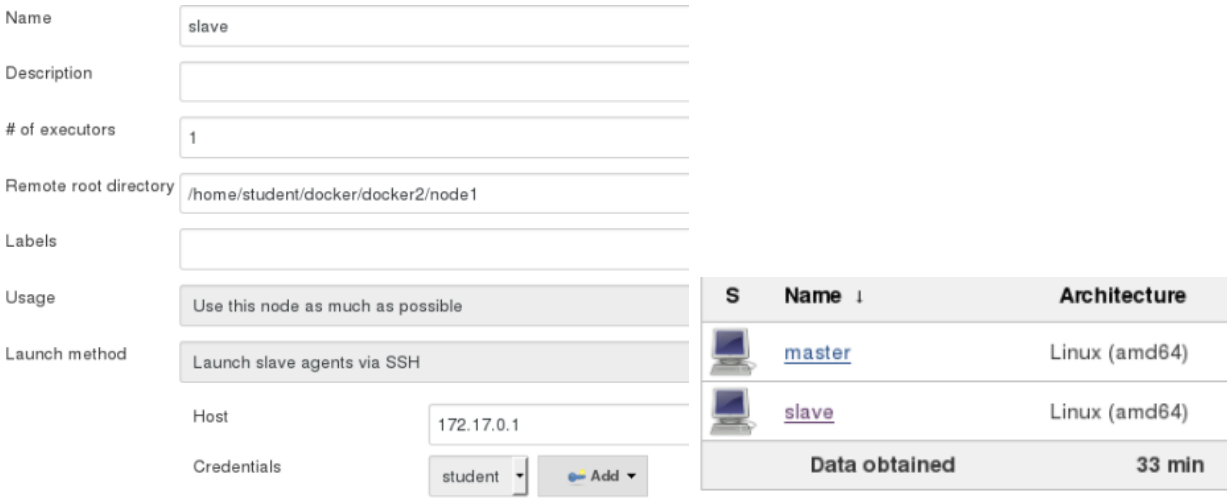
Connect host as a node.

Steps:

• create job 1

• copy public key .ssh/id_rsa.pub to ~/.ssh/authorized_keys

• connect host as node




```
[04/29/17 12:05:56] [SSH] Starting slave process: cd
<===[JENKINS REMOTING CAPACITY]===>channel started
Slave.jar version: 3.7
This is a Unix agent
Evacuated stdout
Agent successfully connected and online
```

Gradle

gradle.Dockerfile:

```
FROM sbeliakou/centos:7.2

RUN yum install -y yum-plugin-ovl && \
    yum install -y which zip unzip java-1.8.0-openjdk java-1.8.0-openjdk-devel && \
    curl -s "https://get.sdkman.io" | bash && \
    source "/root/.sdkman/bin/sdkman-init.sh" && \
    sdk install gradle 3.5

ENV JAVA_HOME /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/

ENV PATH /root/.sdkman/candidates/gradle/3.5/bin/:$PATH
```

Spring-boot application

deploy.Dockerfile:

```
FROM sbeliakou/centos:7.2

ENV APP_NAME gs-spring-boot-0.1.0.jar

ADD node1/workspace/2/initial/build/libs/$APP_NAME /opt/

RUN yum install -y yum-plugin-ovl && \
```

```
    yum -y install java
CMD java -jar /opt/$APP_NAME
EXPOSE 8080
```

Spring Docker-compose

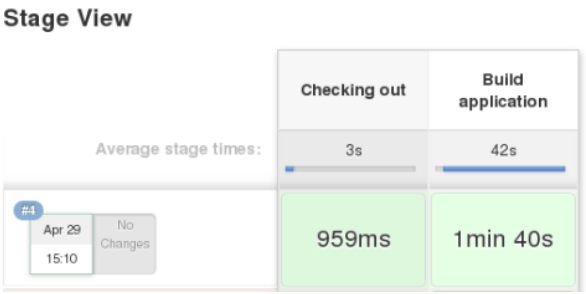
docker-compose.yml:

```
version: '2'
services:
 deploy:
    build:
      context: .
      dockerfile: deploy.Dockerfile
    expose: [ "8080" ]
    ports: [ "0.0.0.0:8088:8080" ]
```

Jenkinsfiles:

• **Gradle + spring**:

```
node('slave') {
stage('Checking out') {
git ' https://github.com/spring-guides/gs-spring-boot.git'
}
stage('Build application'){
sh '''ls -la
cd initial
alias gradle='docker run --rm -v $(pwd):$(pwd) -w $(pwd) gradle gradle'
gradle build'''
}
}
```



• **Spring compose**:

```
node('slave') {
stage ('dockercompose'){
sh ''' cd /home/student/docker/docker2/
docker-compose up -d'''
}
}
```