EPAM DevOps Summer Program 2021

Oleg Shmyrin

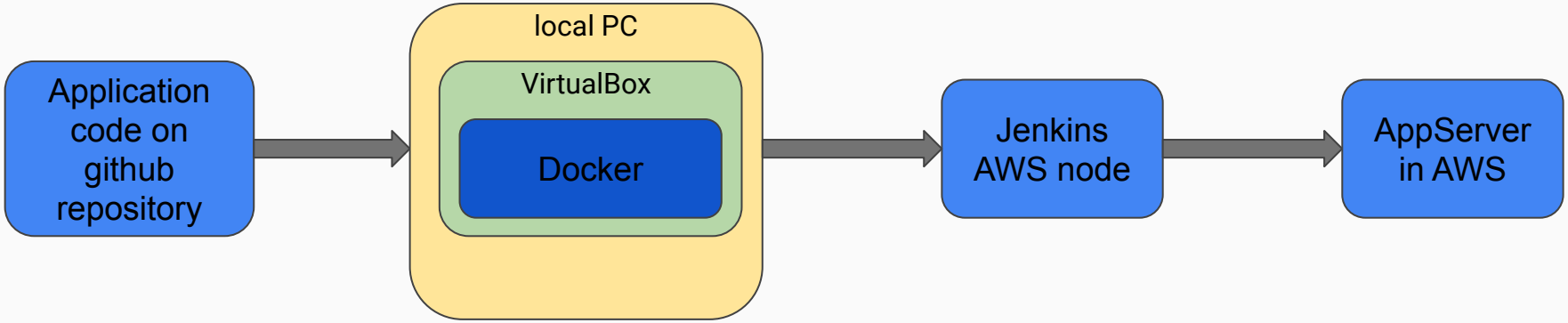
Final Project: https://github.com/OlegShmyrin/spring-petclinic

In this project I want to emulate deployment steps for Java SpringBoot application PetClinic from github CVS to “stage” environment in AWS cloud.

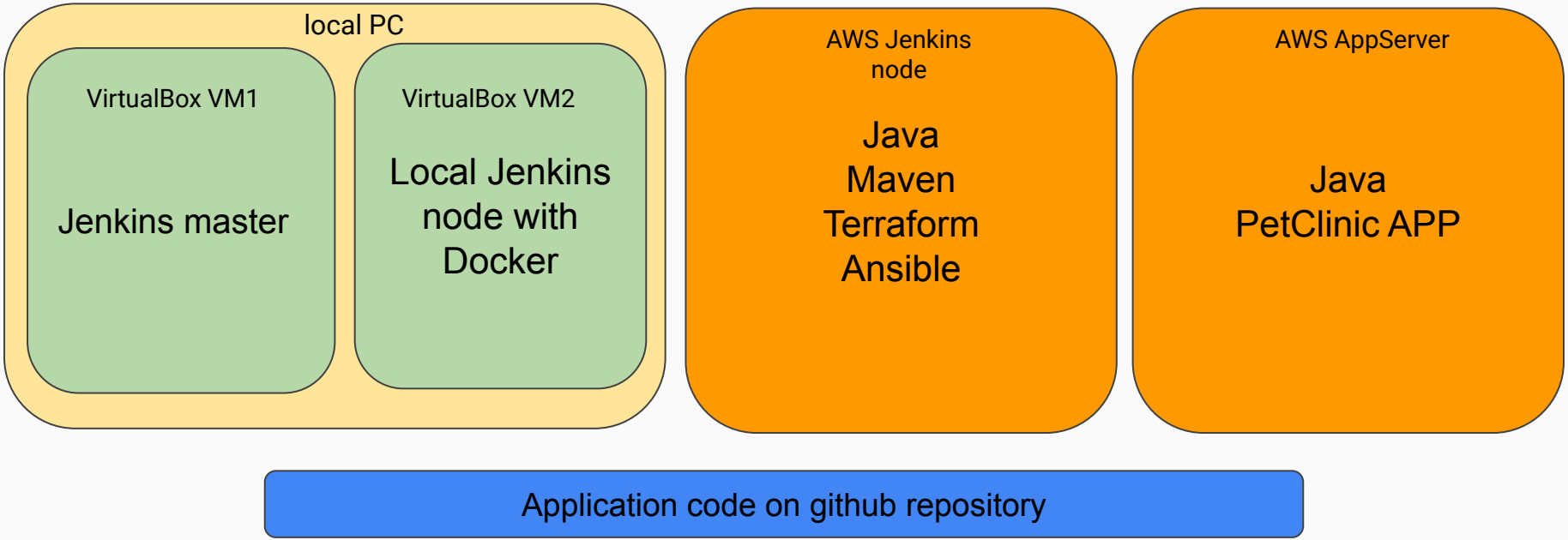
I managed to implement such steps:

- Run maven tests of application in docker container on local Jenkins node launched in VirtualBox
- Initialise AWS instance (AppServer) with Terraform from Jenkins node launched on AWS EC2
- Prepare AppServer required for application with Ansible
- Build application JAR artifact on AWS Jenkins node
- Deliver application artifact to AppServer
- Run application on AWS AppServer

Application deployment steps



Build Environment



Jenkins Pipeline configuration

PetClinic-pipe-line

General | Build Triggers | Advanced Project Options | **Pipeline**

Repositories

Repository URL

git@github.com:OlegShmyrin/spring-petclinic.git

Credentials

OlegShmyrin (github-ssh-key) | Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

Repository browser

(Auto)

Additional Behaviours

Add

Script Path

Jenkinsfile

Jenkins Pipeline configuration

Jenkinsfile contains only 3 stages:

- Test
- Build
- Deploy

```
pipeline {

    agent any

    environment{
        JAVA_TOOL_OPTION = "-Duser.home=/home/oleg/"
    }

    tools {
        terraform 'terraform-aws'
    }



    stages{
        stage("Test"){ ···
        }

        stage ("Build"){ ···
        }

        stage ("Deploy"){ ···
        }

    }
```

Test Stage

I decide run tests that contains PetClininic application in docker maven image `maven:3.6.3-openjdk-17-slim`
Test will run on local Jenkins node labeled with “docker”. To make this I call docker module in pipeline:

```
stages{
    stage("Test"){

        agent {
            dockerfile {
                label "docker"
                args "-v /tmp/maven:/home/oleg/jenkins/.m2 -e MAVEN_CONFIG=/home/oleg/jenkins/.m2"
                }
            }

        steps{
           sh "mvn test"
        }
    }
```

This module read Dockerfile from root directory of repository:

```
FROM maven:3.6.3-openjdk-17-slim

RUN useradd -m -u 1000 -s /bin/bash jenkins
RUN apt-get clean
RUN apt-get update
RUN apt install -y openssh-client
```

Test Stage

And runs docker with args that listed on screenshot.

-v -- mount local directory inside docker image, to store maven dependencies

-e -- says where to store maven config

```
+ docker build -t 91abd148c726e587bed1c6f9f203ac07728974bc -f Dockerfile .
Sending build context to Docker daemon  203.7MB

Step 1/5 : FROM maven:3.6.3-openjdk-17-slim
 ---> 850a4d5b96e4
Step 2/5 : RUN useradd -m -u 1000 -s /bin/bash jenkins
 ---> Using cache
 ---> 221f8f646dec
Step 3/5 : RUN apt-get clean
 ---> Using cache
 ---> 20ccfda35049
Step 4/5 : RUN apt-get update
 ---> Using cache
 ---> 5c0f96d4bb2e
Step 5/5 : RUN apt install -y openssh-client
 ---> Using cache
 ---> a28d193ce6bd
Successfully built a28d193ce6bd
Successfully tagged 91abd148c726e587bed1c6f9f203ac07728974bc:latest
[Pipeline] isUnix
[Pipeline] sh
+ docker inspect -f . 91abd148c726e587bed1c6f9f203ac07728974bc
.
[Pipeline] withDockerContainer
vm4-ubuntu-srv does not seem to be running inside a container
$ docker run -t -d -u 1000:1000 -v /tmp/maven:/home/oleg/jenkins/.m2 -e MAVEN_CONFIG=/home/oleg/jenkins/.m2 -w /home/oleg/jenkins/workspace/PetClinic-pipe-line -v /home/oleg/jenkins/workspace/PetClinic-pipe-
line:/home/oleg/jenkins/workspace/PetClinic-pipe-line:rw,z -v /home/oleg/jenkins/workspace/PetClinic-pipe-line@tmp:/home/oleg/jenkins/workspace/PetClinic-pipe-line@tmp:rw,z -e ******** -e ******** -e ********
-e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e
******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** 91abd148c726e587bed1c6f9f203ac07728974bc cat
$ docker top 8e166ae31bcf63032a4b7709a2cf4d3a66d785e1d75c8c38f1aec8f841697399 -eo pid,comm
[Pipeline] {
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ mvn test
[INFO] Scanning for projects...
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-starter-parent/2.5.4/spring-boot-starter-parent-2.5.4.pom
Downloading from spring-milestones: https://repo.spring.io/milestone/org/springframework/boot/spring-boot-starter-parent/2.5.4/spring-boot-starter-parent-2.5.4.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.5.4/spring-boot-starter-parent-2.5.4.pom
```

Test stage

mvn test -- start compile project and runs all test from **/src/test**

```
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
java.lang.instrument.IllegalClassFormatException: Error while instrumenting javax/sql/DataSource.
        at org.jacoco.agent.rt.internal_43f5073.CoverageTransformer.transform(CoverageTransformer.java:94)
        at java.instrument/java.lang.instrument.ClassFileTransformer.transform(ClassFileTransformer.java:244)
        at java.instrument/sun.instrument.TransformerManager.transform(TransformerManager.java:188)
        at java.instrument/sun.instrument.InstrumentationImpl.transform(InstrumentationImpl.java:540)
        at java.base/java.lang.ClassLoader.defineClass2(Native Method)
```

```
2021-10-03 16:22:08.886  INFO 1097 --- [ionShutdownHook] org.ehcache.core.EhcacheManager          : Cache 'vets' removed from EhcacheManager.
2021-10-03 16:22:08.925  INFO 1097 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2021-10-03 16:22:08.975  INFO 1097 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
[INFO]
[INFO] Results:
[INFO]
[WARNING] Tests run: 40, Failures: 0, Errors: 0, Skipped: 1
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  25:24 min
[INFO] Finished at: 2021-10-03T16:22:10Z
[INFO] ------------------------------------------------------------------------
```

Build Stage

Build stage runs on AWS Jenkins node labeled as “ansible”

```
stage ("Build"){
    agent {
        label "ansible"
    }

    steps{
        sh "terraform init"
        sh "terraform apply -auto-approve"
        sh "terraform output"
        sh "cat inventory"
        sh "chmod -x inventory"
        sh "sudo mvn package -Dmaven.test.skip"
    }


}
```

Build Stage

```
sh "terraform init"
```

```
sh "terraform apply -auto-approve"
```

This commands runs Terraform. Terraform read **main.tf** file and initiate and runs required server with ssh key generation.

```
provider "aws" {
  region = "eu-central-1"
}

2 references
variable "generated_key_name" {···
}

1 reference
resource "tls_private_key" "PetClinic_key" {···
}

resource "aws_key_pair" "generated_key" {···
}

3 references
resource "aws_instance" "AppServer" {···
}
```

Build stage

```
sh "terraform output"
```

This command generate desired outputs variables, based on outputs.tf file

```
### The Ansible inventory file
resource "local_file" "AnsibleInventory" {
 content = templatefile("inventory.tmpl",{
     AppServer-dns=aws_instance.AppServer.private_dns,
     #AppServer-ip=aws_instance.AppServer.public_ip,
     AppServer-pr-ip=aws_instance.AppServer.private_ip,
     AppServer-id=aws_instance.AppServer.id}
     #AppServer-key=aws_key_pair.generated_key.generated_key_name}
 )
 filename = "inventory"
}
```

And makes inventory file for Ansible, based on inventory.tmpl file

```
[AppServers]
${AppServer-dns} ansible_host=${AppServer-pr-ip} ansible_user=ubuntu  ansible_ssh_private_key_file=terraform-key-pair.pem # ${AppServer-id}
```

Build stage

```
sh "sudo mvn package -Dmaven.test.skip"
```

In this step we build jar artifact on AWS jenkins node.

```
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mmaven-surefire-plugin:2.22.2:test⍰[m ⍰[1m(default-test)⍰[m @ ⍰[36mspring-petclinic⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Tests are skipped.
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mjacoco-maven-plugin:0.8.5:report⍰[m ⍰[1m(report)⍰[m @ ⍰[36mspring-petclinic⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Skipping JaCoCo execution due to missing execution data file.
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mmaven-jar-plugin:3.2.0:jar⍰[m ⍰[1m(default-jar)⍰[m @ ⍰[36mspring-petclinic⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Building jar: /home/ubuntu/jenkins/workspace/PetClinic-pipe-line@2/target/spring-petclinic-2.5.0-SNAPSHOT.jar
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mspring-boot-maven-plugin:2.5.4:repackage⍰[m ⍰[1m(repackage)⍰[m @ ⍰[36mspring-petclinic⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Replacing main artifact with repackaged archive
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[⍰[1;34mINFO⍰[m] ⍰[1;32mBUILD SUCCESS⍰[m
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[⍰[1;34mINFO⍰[m] Total time:  23.746 s
[⍰[1;34mINFO⍰[m] Finished at: 2021-10-03T16:23:13Z
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[Pipeline] sh
```

Deploy stage

Deploy stage runs with command

```
sh "ansible-playbook playbook.yml -i inventory"
```

on AWS Jenkins node and configure AppServer with Ansible from **playbook.yml** file.

```
stage ("Deploy"){
    agent {
        label "ansible"
    }

    steps{
        sh "ansible-playbook playbook.yml -i inventory"
    }


}
```

```
---
- name: Prepare App Server and Build
  hosts: AppServers
  become: yes
  tasks:
  - name: Ping to servers ···

  - name: Install Java ···

  - name: Copy Builded App ···

  - name: Make a service ···

  - name: Restart systemd-sysctl ···

  - name: Start pet clinic service ···
```

Deploy stage

```yaml
- name: Copy Builded App
  become: yes
  copy:
    src: "{{ item }}"
    dest: /src/petclinic/petclinic.jar
  with_fileglob:
    - "target/*.jar"

- name: Make a service
  become: yes
  copy:
    src: "{{ item }}"
    dest: /etc/systemd/system/
    mode: '0644'
  with_fileglob:
    - "petclinic.service"

- name: Restart systemd-sysctl
  become: yes
  ansible.builtin.service:
    name: systemd-sysctl
    state: restarted

- name: Start pet clinic service
  become: yes
  ansible.builtin.service:
    name: petclinic
    state: started
```

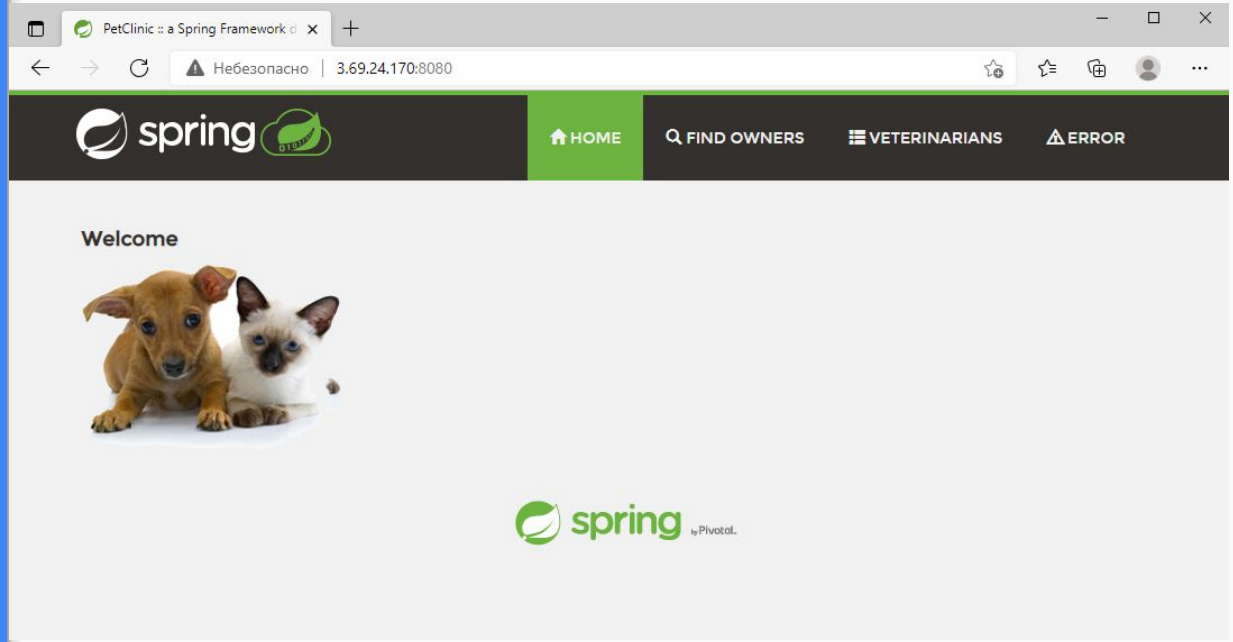
```ini
[Unit]
Description=Petclinic Java service

[Service]
WorkingDirectory=/src/petclinic
ExecStart=/bin/java -jar /src/petclinic/petclinic.jar
User=ubuntu
Type=simple
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

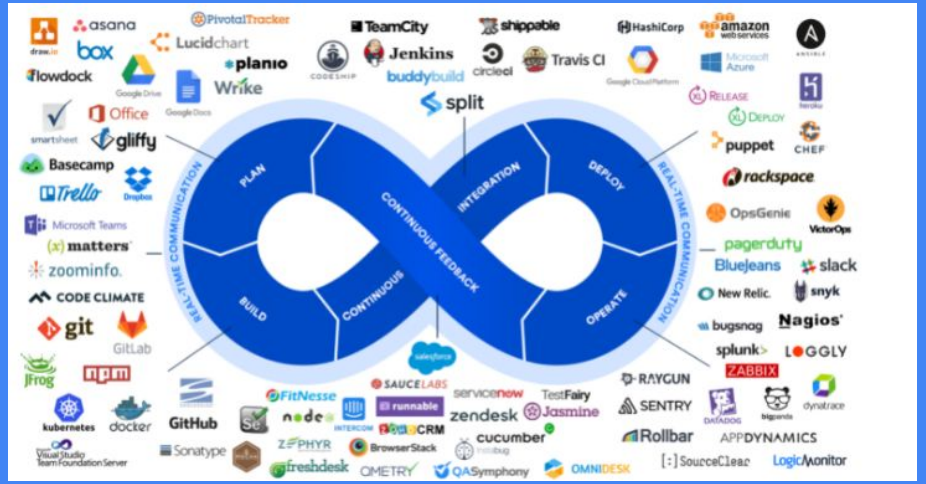
Results

We receive running application on AWS cloud server.



TODO steps

1. Use mysql database instead H2
2. Add nginx proxy before app
3. Make one more AppServer in another availability zone and use balancer
4. …



Thank You for lessons!