Chapter 4

Lab 4.4

Objective: Deploying a Java application with Vagrant

Prerequisites

This lab requires Vagrant and VirtualBox to be installed on your workstation. Vagrant is a virtual machine management tool that allows for the creation and management of local virtual machines on different virtualization platforms, including VirtualBox and VMware. Information and installation directions for Vagrant are available in Chapter 2 of this course (Lab 2.1). Additional information is also provided by HashiCorp, the company behind Vagrant.

Note: This lab assumes that there is a running Artifactory instance with a deployed `petclinic.war` file on a separate Vagrant box. It also assumes that this Vagrant box has the IP address of `192.168.50.5` set in the `Vagrantfile`.

Using Vagrant

Step 1. Create and initialize a `vagrant` directory for a new Ubuntu virtual machine

1. Create a new directory for the `Vagrantfile`:

   ```
   $ mkdir petclinic-vagrant
   ```

2. Navigate to the new directory:

   ```
   $ cd petclinic-vagrant
   ```

3. Download the Ubuntu 16.04 box:

   ```
   $ vagrant box add ubuntu/trusty64
   ```

4. Create a `Vagrantfile` in the `petclinic-vagrant` directory:

   ```
   $ vagrant init .
   ```

   The last command created a `Vagrantfile`, which provides Vagrant with instructions detailing the steps required for launching and configuring a virtual machine.

Before launching the virtual machine, edit the **`Vagrantfile`**, and ensure it contains the following:

```
$ Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.network "forwarded_port", guest: 8080, host: 8082
  config.vm.boot_timeout = 600
  config.vm.provider "virtualbox" do |vb|
    vb.memory = "2048"
  end

config.vm.provision "shell", inline: <<-SHELL
  # Install open-jdk-8
  add-apt-repository ppa:openjdk-r/ppa -y
  apt-get update
  apt-get -y install apache2 openjdk-8-jdk
  update-alternatives --config java
  echo "JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre" | \
  tee --append /etc/environment \
  > /dev/null
  source /etc/environment
  export JAVA_HOME

  # Install Tomcat 7
  apt-get install --show-progress -y default-jre tomcat7 tomcat7-docs
  tomcat7-admin tomcat7-examples
  chown tomcat7.tomcat7 /usr/share/tomcat7
  chmod 0755 /usr/share/tomcat7

  # Configure the port that Tomcat runs on
  sed -i -e 's/8080/8082/g' /var/lib/tomcat7/conf/server.xml

  # Create a login user with username admin and password admin
  sed -i "s#</tomcat-users>##g" /etc/tomcat7/tomcat-users.xml; \
  echo '  <role rolename="manager-gui"/>' >>  /etc/tomcat7/tomcat-users.xml;
  \
  echo '  <role rolename="manager-script"/>' >>
  /etc/tomcat7/tomcat-users.xml; \
  echo '  <role rolename="manager-jmx"/>' >>  /etc/tomcat7/tomcat-users.xml;
  \
  echo '  <role rolename="manager-status"/>' >>
  /etc/tomcat7/tomcat-users.xml; \
  echo '  <role rolename="admin-gui"/>' >>  /etc/tomcat7/tomcat-users.xml; \
  echo '  <role rolename="admin-script"/>' >>  /etc/tomcat7/tomcat-users.xml;
  \
  echo '  <user username="admin" password="admin" roles="manager-gui,
  manager-script, manager-jmx, manager-status, admin-g
```

```
ui, admin-script"/>' >>  /etc/tomcat7/tomcat-users.xml; \
echo '</tomcat-users>' >> /etc/tomcat7/tomcat-users.xml

# restart tomcat
sudo service tomcat7 restart

 # Place spring-petclinic.war into Tomcat's webapps directory
 # The path to your artifact may need to be altered
 curl -o /var/lib/tomcat7/webapps/petclinic.war
'http://192.168.50.5:8081/artifactory/<repository_path>'

   SHELL
end
```

Step 2. Launch VM for testing

1. In the `petclinic-vagrant` directory enter:

   ```
   $ vagrant up
   ```

   For the `ssh` access to your virtual machine, simply enter `vagrant ssh` in the `petclinic-vagrant` directory.

   To see the running `petclinic`, browse to `http://localhost:8082/petclinic/`.