## Step by step instructions:

Like in lab1. Launch a VM running Ubuntu 20.01 with ansible and Jenkins.
 Here is the content of my Vagrant file. I used docker-compose to install Jenkins and used provision.sh to install ansible.

```
Vagrant.configure("2") do |config|
config.vm.box = "ubuntu/focal64"
config.vm.hostname = "sonarqube.box"
config.vm.provision "docker"

config.vm.provider "virtualbox" do |v|
    v.memory = 7792
    v.cpus = 2
end

config.vm.provision :docker_compose, yml: "/vagrant/docker-compose.yml", run: "always"

config.vm.provision "shell" do |shell|
    shell.path = "provision.sh"
end

config.vm.network "forwarded_port", guest: 9000, host: 9000
config.vm.network "forwarded_port", guest: 9092, host: 9092

config.vm.network "forwarded_port", guest: 8080, host: 8080
config.vm.network "forwarded_port", guest: 50000, host: 50000
config.vm.network "public_network"
end
```

I used public network here for convenience. The content in provision.sh is:

```
sysctl -w vm.max_map_count=262144
echo "vm.max_map_count=262144" >> /etc/sysctl.conf
sysctl -w fs.file-max=131072
echo "fs.file-max=131072" >> /etc/sysctl.conf
ulimit -n 65536
ulimit -u 4096
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install wget unzip -y
sudo apt-get install openjdk-11-jdk -y
sudo apt-get install openjdk-11-jre -y
sudo apt-get update
sudo apt-get install -y python3-pip
sudo pip install ansible
sudo cp ./ansible.cfg /etc/ansible/ansible.cfg
\verb"sudo" cp ./hosts /etc/ansible/hosts"
sudo apt install net-tools
sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config; sudo systemctl restart
sshd;
```

This installs ansible as well as sshpass and jdk.

2. I then used vagrant to set up a second VM as the web-server. I only need to install java and set some ssh parameters. So the vagrant file and provision.sh is very simple:

```
Vagrant.configure("2") do |config|
config.vm.box = "ubuntu/focal64"
config.vm.hostname = "ansible.box"

config.vm.provision "shell" do |shell|
    shell.path = "provision.sh"
    end

config.vm.network "forwarded_port", guest: 8080, host: 8081, id: "pet"
    config.vm.network "forwarded_port", guest: 80, host: 8082, id: "nginx2"
    config.vm.network "public_network"
end

sudo apt-get update
sudo apt-get upgrade
sudo apt-get install wget unzip -y
sudo apt-get install openjdk-11-jdk -y
sudo apt-get install openjdk-11-jre -y

sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config; sudo systemctl restart sshd;
```

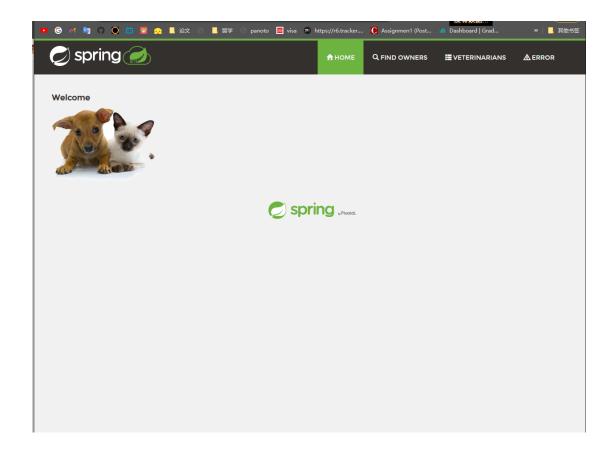
This launches a second VM with java.

 Then I used vagrant ssh to enter the first VM, where I edited the ansible.cfg and hosts in the ansible config directory. The modified version is here: <a href="https://github.com/Dedsec-Xu/devops/blob/main/hw2/vagrant/ansible.cfg">https://github.com/Dedsec-Xu/devops/blob/main/hw2/vagrant/ansible.cfg</a> https://github.com/Dedsec-Xu/devops/blob/main/hw2/vagrant/hosts

Because I used public\_network for both VMs. I can easily setup the ipaddress in hosts file. Then I ping the webserver from the host VM to see if it works.

4. Then I wrote an ansible playbook yml file to send the jar to the webserver and run it there.

After a while, pet-clinic is successfully running on the web-server VM. And I can open it on port 8081(Because what I set in Vagrant file)



5. I have automated most of these. You can run auto.bat, which can automatically setup both webserver and the host VM. And install java on both and setup ansible and ansible configuration.

Then you only need to run

ansible-playbook play.yml

in the host machine and it will automatically deploy the app to the webserver.

Here is a screenshot of running the full auto script.

```
| Elytranical processing of the control of the cont
```