Week 12: Creation of virtual machine for Ubuntu OS and Deploying the web application

1. Creation of virtual machine
2. Deploying the web application
3. Accessing it publicly

Deploying an application into cloud

Steps for Deploying application into the cloud

I. Create application and Push into github
II. Create the virtual machine and connect to it.
III. Clone the application from github, Write the Dockerfile
IV. Create the image
V. Run the image and access it public ip of virtual machine

I. Create Maven-web-java project in eclipse & push into github

II. Create the virtual machine (EC2--instance) in aws and connect to

Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.

Ex : **Launch ubuntu instnace**

Step 1: Login to AWS /canvas account

Step 2: Services -- EC2

Step 3: Choose region which is near ?

Services -- EC2 --- Launch Instance

Stage 1 --Name (Giving name to the machine) ubuntu

Stage 2 -- Select AMI (Note: Select free tier eligible) ubuntu server

Stage 3 -- Architecture as 64-bit

Stage 4 -- **Instance type** ---- t2.micro(default 1 CPU,1 GB RAM)

Stage 5 -- Create a new keypair---a keypair will downloaded with extension .pem

Store key in folder AWS

Stage 6 -- Network Setting ----Create Security group -- (It deals with ports)

(Note for understanding We have 0 to 65535 ports. Every port is dedicated to special purpose)

Do this step : HERE select http and https

Stage 7 -- Storage - 8GB (Observation - we have root - it is same as C Drive)

Stage 8 --- click on launch instance

Stage 9: Number of instances ---1

+++++++++++++++++++++++++++++++

Observation - One machines created

Do this step:---once it is created select that instance and click on connect

Here copy the ssh – i command from SSH client connect tab

We can use powershell /gitbash /webconsole , to connect to ubuntu machine.

NOTE:- cd path of AWS folder // change path

To connect to above terminals we need to go into the path of the keypair.and

paste the

ssh -i command from the aws console

III. Clone the application from github, Write the Dockerfile

once connected to instance

Step 1:- Run the following commands to install s/w

1. Update all softwares in Ubuntu by command
   sudo apt update
2. Install docker by command
   sudo apt-get install docker.io

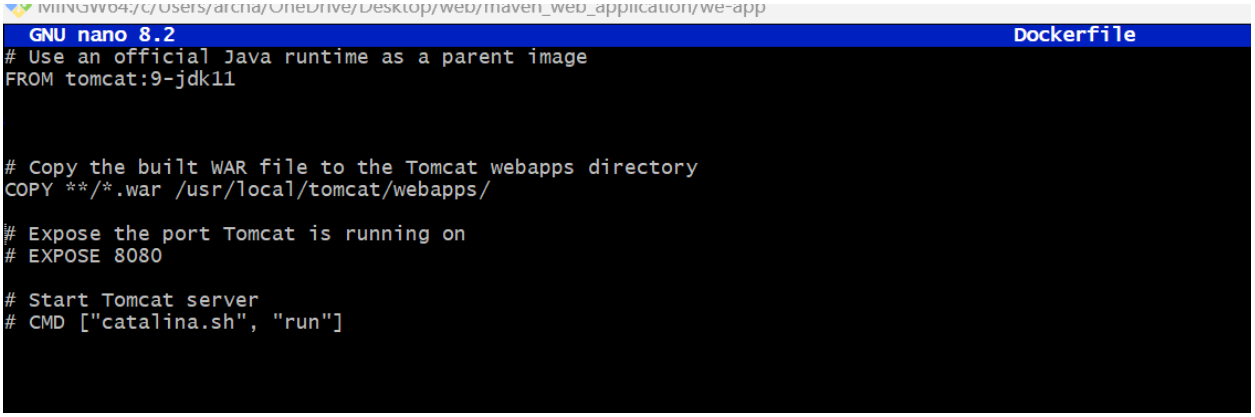
3. Install git by command
   sudo apt install git
4. Install nano(text editor) by command
   Sudo apt install nano

step 2:- git clone <**paste the github link of maven-web-java project**>

step 3:- navigate to the maven-web-java project

VI. Create the image

Step 4:- nano Dockerfile


MINGW64:/c/Users/archa/OneDrive/Desktop/web/maven_web_application/we-app
GNU nano 8.2
Dockerfile

```
# Use an official Java runtime as a parent image
FROM tomcat:9-jdk11



# Copy the built WAR file to the Tomcat webapps directory
COPY **/*.war /usr/local/tomcat/webapps/

# Expose the port Tomcat is running on
# EXPOSE 8080

# Start Tomcat server
# CMD ["catalina.sh", "run"]
```

V. Run the image and access it with public ip of virtual machine

Step 1:- build your image

    docker build –t <imagename> .(dot)

Step 2:- check for images

Step 3:- run image

    docker run –d --name app-demo –p 6060:8080 <image name>

Step:-4 **Accessing the app by public ip of virtual machine**

Note:-if your are not able to connect change the inbound rules..

Schenerio based questions

1. You have a simple index.html file on your laptop and you launched an EC2 instance with Amazon Linux 2. The instance is running but when you open the public IP in browser, the page doesn’t load.

What steps will you take to host the index.html?

2. You deployed your index.html to /var/www/html/ directory on EC2, but the web page still isn't loading.What are two possible issues you would check?

3. You installed Apache HTTP server on EC2 to host index.html, but the service stops after instance reboot. What command should you run to ensure it auto-starts on boot?

4. You are deploying a Maven web application onto an EC2 instance.Maven is not installed on the instance. What commands or steps will you follow to install Maven on Amazon Linux/Ubuntu EC2?

5. You built a Maven project on EC2, and a .war file generated inside target/.

You want to deploy it using Tomcat. Where will you place the .war file and why?

6. Your Maven web app is deployed to Tomcat on EC2, but accessing it via browser gives 404 error. What configuration or path issues will you check?

7. You can access your web application locally on the EC2 instance using curl localhost:8080 but not from your browser. What AWS setting is likely missing or misconfigured?

8. You have deployed your web app on EC2 successfully, but after shutting down your local Wi-Fi and reconnecting, the public IP changed and the app is not opening. What AWS feature helps avoid this issue?

9. You want to automate deployment of your index.html whenever you restart the EC2 instance. Which EC2 feature can be used to run commands automatically during instance setup?

10. Your Maven application needs external dependencies during build but EC2 has no internet.

What AWS service or change can allow the EC2 instance to download dependencies securely?