WEEK-1

1. Go to https://www.atlassian.com/software/jira

2. Signup with Google account → Click on Google

3. As you have already account so it will open. Let's name your site → Click on continue

4. It'll ask to choose the Project type then choose Scrum

5. Then name your project as "Project1" → Click on Get Started

6. Go to Projects → Backlog session

7. In Backlog click on EPIC → Click on Show EPIC

8. Click on create epic → Name it "Epic: Improve Service"

9. In Backlog click on Create → Name it as BY DROPDOWN Story

→ Select Story → CLICK ENTER

→ Like this create the Story & few tasks

10. Select the Scrum you have created → Click on Add epic → Add this to the epic which you have created. It will Show down.

11. After completing click on … → move work items → Scrum/Sprint

12. Click on (…) beside Start Sprint → Keep the Start date & End date & Click on update

13. Click on Start Sprint → Click Start

14. Click on To Do → Scrum → Change To Do to In Progress

15. To Review → To Done

16. Continue for all tasks & Sprint Scrum

17. Click on Complete Sprint

18. Go to Summary

WEEK-3

1. Log in into your AWS free tier account

2. Open the GitHub account: https://github.com/cmr73

3. Click on Launch Instance in EC2

After clicking on Launch Instance:

Name – Devops Jenkins

Select-Ubuntu

Instance Type – t2.micro → Enable All generations

Click on Create new key pair

Name the keypair “jenkins” → Click on Create Key Pair

The key pair will get downloaded

4. In Network Settings:

Select the "Allow HTTP traffic from the internet"

Select the "Allow SSH traffic from"

Click on Edit

Add click on Add security group rule

Type – Custom TCP

Protocol – TCP

Port range – 8080

Source type – Custom

Source – 0.0.0.0/0

Click on Launch Instance

5. Select the Jenkins instance → Click on: Connect → SSH Client

Copy the “Example” ssh -i “devvm.pem” ubuntu@ec2-56-15x-x-

6. Go to the Downloads → Right Click on the empty space

Click on Show more options → Click on Open GitBash

Paste the command there → Click Enter → Type “yes” → Enter it

7. Go to the GitHub & copy the step-1 | full & paste it in GitBash → Click Enter

In middle if it asks for type Y then type “Y” & Enter

8. Go to the GitHub & copy the Step – 2

First command & paste it → Enter

Copy second command – echo & paste it → Enter

Copy third command – sudo apt-get update & paste it → Enter

Copy the “sudo apt-get install Jenkins” → paste it → Enter

Type “Y”

Copy the “sudo systemctl start Jenkins” & paste it

9. Copy the Step – 3 command & paste it to GitBash

10. Copy the instance public IP address & paste it in browser tab:

IPaddress:8080 → Enter

It will show Unlock Jenkins

Enter the Administrator password

To know the password: sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Copy the password & paste it in the Unlock Jenkins

Click on Continue → Click on Install suggested plugins

Enter admin → username, password, confirm password, full name, email – admin@gmail.com → Save & continue

Click Save & Finish → Click on Start using Jenkins

WEEK-4

While creating instance for week-3 at that time

→ Create 2 instances & launch it

- One instance name as: devm

- Second instance name as: setm

Complete doing the week-3 full experiment

➞ Go to the Gitbash which you have used for week-3

➞ Go to the GitHub com73 → In Maven Setup on devm → Step-1

- Copy the full command & paste it in Gitbash → Enter → To see list type-ls

➞ Go to GitHub copy the Step-2 first Command & paste it

- Bring the cursor down to it → Press i

- Copy the second command from exports to (PATH) → paste it in Gitbash

For existing press esc then !wq or (:wq) → Press Enter

➞ Go to GitHub copy the step-2 Reload command & paste it in Gitbash

➞ Open Jenkins page

➞ Click on Manage Jenkins → Plugins → Available Plugins

- In search box type – "Maven Integration"

- Select the Maven Integration → Click on Install

- After successful install

➞ Go to Manage Jenkins → Tools → Click on Add JDK

- Name – java-21

- JDK Path: /usr/lib/jvm/java-21-openjdk-amd64

- Paste in JAVA-HOME

➞ Scroll down click on Add Maven

- Uncheck the install automatically

- Name – maven

- MAVEN-HOME = /opt/maven

➞ Click on Apply & Save

➞ Click on New item → project

- Select the maven project → Click OK

- Check the GitHub project → In project URL paste GitHub pom.xml URL

- Login to GitHub account & open the practice repository → Click on code → Copy the HTTPS link & paste it in project URL

➞ In Goals & Options → type clean install

➞ In Source code management → select Git

➞ In Repository URL – paste your GitHub pom.xml URL

➞ Click on Apply & Save

➞ Click on Build Now

➞ If the build takes a lot of time to build then logout from Jenkins

➞ Select the instance & reboot the instance

➞ Copy the public IP of the instance & again paste it in tab with :8080 & login into Sign in to Jenkins

- Username – admin

- Password – admin

➡ Click on Sign in

CLICK ON BUILD NOW

WE SEE THE PROJECT IS BUILD IN DASHBOARD

WEEK-6

→ After completing the Week-4 experiment

→ Select the same instance → Click on Connect →

EC2 instance connect → Click on Connect

→ Go to GitHub to setup Tomcat on setm

→ Copy the step-1 command & paste it into setm

EC2 instance connect & Enter

→ If tomcat9 is not installing means uninstall tomcat10

sudo apt update

sudo apt uninstall -y tomcat10 tomcat10-admin

→ After installing tomcat9

→ Copy the step 2 command - cd /etc/tomcat9

→ If tomcat10 is installed then type cd /etc/tomcat10

→ Copy the next command, step-2 in GitHub

“sudo nano tomcat-users.xml” and paste it

Bring cursor to last above "tomcat-users" → paste the

next command & role → ctrl + o, Enter, ctrl + x

→ Copy the setm instance public IP address & paste

it in tab - IP address:8080 → It shows It works

page

→ Copy the next step-3 command to Restart tomcat

& paste it

→ If you installed tomcat10 then change command

to tomcat9 to tomcat10

→ Go to GitHub & copy the step-4 command

paste it in new tab in place of “IP address”

Keep same IP address & Enter

→ Go to Jenkins page

Click on Dashboard → Manage Jenkins → Plugins

→ Available Plugins → search for “Deploy to

Containers”

Select & Install

→ After successful install

→ Click on Project → Configure

Scroll down → Click on Add post-build action

Select the Deploy war/ear to a container

→ In war/ear files : */.war

Context path : sit

→ Click on Add container → select tomcat9x. Remote

→ In Credentials click on Add → Jenkins

→ Username - admin

→ Password - admin → click on Add

→ Click on Credentials → Click on none then

Select admin /***

→ Click on tomcat URL & paste the http://setmipaddress:8080/

→ Click on Apply & Save

→ Click on Build Now. After successful Build

→ Copy the step-4 command from GitHub & paste

it in Browser → http://setm ip address:8080/sit

WEEK-7

→ Login into the GitHub

→ Click on practice → Click on src/main/webapp → click on index.jsp

→ Go to Jenkins page → Click on project → Click on configure → Click on triggers

Check the Poll SCM → In schedule → Enter * * * * *

Click on Apply & Save

→ Go to the index.jsp → Click on edit

Do some changes in the file

→ Click on commit changes → Click on commit changes

→ Reload the index.jsp page

→ Go to Jenkins page & reload the page

Wait, a build will be building

After the completion of build file

→ Reload the setmipaddress:8080/sit page

→ The changes we made in index will be represented in this page

WEEK-5

Open GitHub link cmr73

Login into the AWS-free tier account

Click on EC2 → Click on Launch Instance

Name – Ansible

Select – Ubuntu

Instance type – t2.micro, Enable All generations

Click on Create New Key Pair

Name – Ansible → Click on Create keypair

Select the Allow HTTP traffic from Internet

In Summary → Keep no. of instances – 3

Click on Launch Instance

Choose the 2nd & 3rd as instances as server1 and server2

Select the first instance Ansible & Click on Connect

SSH Client → Copy the example command

Go to Downloads → Right click on empty space → Click on "Git Bash Here"

Paste the command in Git Bash → Enter

Type – yes

Go to GitHub & copy the first command of Step 2 & paste it in Git Bash

sudo su → Enter

copy apt update -y & paste

copy apt-add-repository ppa:ansible/ansible & paste

copy apt update & paste

copy apt install ansible -y & paste

copy ansible --version & paste

Go to AWS & select Server1 → Click on Connect → EC2 Instance → Click on Connect

Select Server2 → Click on Connect → EC2 Instance → Click on Connect

Go to GitHub & go to Step 3

Copy the first command → nano /etc/hosts

Add the server1 IP address server1

Add the server2 IP server2

Save by Ctrl+O → Enter, Ctrl+X

Go to GitHub step 4, copy the first command, paste it in Git Bash → Enter

Press Enter, Enter, Enter for Enter file, enter passphrase, Enter (same passphrase)

Copy the content from ssh-rsa to end & paste it into a notepad

Copy the step 4 second command & paste it in server1 & server2 Ubuntu

Copy the content key from notepad & paste it in server1 (at starting) & Ctrl+O → Enter + Ctrl+X

Do same for server2

Go to GitHub & paste the Step 5 (server1) command

Type – yes → Enter

After that type exit

Paste SSH ubuntu@server2 command

Type – yes → Enter

After that exit

Copy the step 6 first command & paste it in Git Bash

Copy the nano inventory command & paste it

In the file add →

[webservers]

server1

server2

Ctrl+O → Enter, Ctrl+X

Copy nano ansible.cfg & paste

In this file add [defaults] → false

Ctrl+O → Enter, Ctrl+X

Copy the step 7 first command & paste it

In the file paste the second command content of step 7 → Ctrl+O → Enter → Ctrl+X

Copy the Step-8 command & paste it in Git Bash

Copy the server2 IP address in tab

We can see Ubuntu

Copy the server1 IP address in tab

We can see nginx