**Program 1**

1.git clone the repo

**Git init**

**Git clone url**

In eclipse import then maven then existing then blouse ur folder inside a created folder

[2.in](http://2.in) case gradle not downloaded ,go to browser type gradle download,go to gradle org ,installing manually,download link,releases,binary only then gets downloaded,open zip extract it ,now open it and open bin folder,copy link of gradle bin like after clicking inside the bin copy path from top,

Edit environment var then system var ,down side select path then edit then new add the path next then complete

Check gradle -v in cmd

3. Pom.xml maven update

Right click project then maven test or clean and install

Right click project then run as test ng

Right click project go to prop copy path cd in cmd the gradle init u can see the gradle

4. In case test ng not working

Help eclipse marketplace ,search testng,install next next

Else

Open testng org go to locations of project

Eclipse plugin,copy release top url

In eclipse go to help install new soft,paste url once loaded click tick and finish

harsh1n1vg

**Program 2**

Jenkins 8080

Username: keertana

Password: $SRIsri6

TOMCAT 8085

U:kskeertana

P:$SRIsri6

1.Dont forget to open tomcat

cd "C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf"

notepad tomcat-users.xml

And add this in the notepad by opening in notepad

<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<role rolename="manager-script"/>

<user username="kskeertana" password="$SRIsri6" roles="manager-gui,admin-gui,manager-jmx,manager-status,manager-script"/>

This username and pwd to be given when asked

In lab both username and password is tomcat

**🔁 How to Restart Tomcat (Windows):**

1. Press Windows + R, type services.msc, hit Enter.

Scroll down to find something like:  
  
 nginx  
CopyEdit  
Apache Tomcat 9.0 Tomcat9

1. Right-click on it → Click **Restart**.

Alternatively, if you’re running Tomcat via command line (startup.bat or catalina.bat), just close the console window and re-run the startup.bat.

1.install jenkins and tomcat

2.create war file

3. Push to git

4.deploy war file

Git clone,then import existing maven project ,then open that inside folder of cloned file

Then right click project maven update project

Then right click project run as maven build goals compile run ,shd see success

In cmd cd the project path then git init

In eclipse right click main project then share project,tick box,create repository then that master will come next to the name of proj

To push to git right click team then commit,select all files and drag to staged changes and give a commit msg then commit and push

Error so click push head give github url,auth give username and password,preview again preview ,enter login details still not working

In github go to profile ,settimgs,developer settings then personal access token ,token classic,gen new one ,new token classic select all chk box,gen token

Copy it now in auth give that token as pwd and username github name

Next push again put token as pass ghp\_Pz1DoPRCubNXezPftK51CIyr1UWo263KOIUZ

While u do create new remote and push (remember to create a new repo)

Jenkins chk maven and git installed,deploy to container all shd be installed

Next jenkins login,then new item ,give name select free style

Select git give url of that new repo u made ,select credentials,

Branch master

Add timestamps to the Console Outp tick

Build station-invoke top level

Select maven version

Goals clean install

Add post b… deploy war/…to container

War file \*\*/\*.war

Containers tomcat 7x remote

Credentials add

U and p tomcat tomcat

Id give tomcat\_u\_p

Add , then u can select that

Next give tomcat url [http://……..8080/](about:blank)

Then apply

Then go to dashboard and build now ,go check down logs

Refresh tomcat ,go to tomcat ,manager app then login with cred there select war file u see output

**PROGRAM 3**

**1.go to docker desktop for windows and download**

**2.open and run and install**

**3.go to search search nginx then run**

**Do same for tomcat**

**4.click somewhere go to containers u shd see them else go to images and again click on download now u can see them in the containers**

**And they will be running**

**5.follow mams notes from prerequisites make sure………….**

**Prerequisites:**

Make sure you have the following installed:

1. Docker Desktop (running)

2. VS Code

3. Docker extension for VS Code (optional but helpful)

4. A .war file ready to use (e.g., sample.war)

1: Place your .war File on VS code Terminal

Right-click the project folder in Eclipse select properties→ Copy the project path

Open VS code terminal type cd type project path in “ “

Go to File and Open Folder

Select the eclipse project folder you copied

VS Code will load the project

**then follow next steps in mams notes rember mine is 8082 tomcat but in code keep it as 8080 only**

**7.the docker shd always be kept open throughout**

**8.commands follow the obser and notes**

**Imp make sure to change the path or name of war with urs instead of what mam has given**

**FROM tomcat:9.0**

**RUN rm -rf /usr/local/tomcat/webapps/\***

**COPY target/program2-0.0.1-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war**

**EXPOSE 8080**

**Dont copy paste remove the bracket**

**docker run -d -p 8095:8080 sample-app**

**If did some mistake use different ports and try it**

**Now in localhost:8095 if i open i see hello world thats the output**

**3b** Execute given commands in terminal

docker-compose build

docker-compose up

Remember to name as given in the docs not how u wrote in obs bcz code has it ,will get the output(see desktop devops3b file for ref)

In lab you should login to docker both in the vs and in docker

If not opening in browser just type docker and sign in

For signin

Username:kskeertana

pwd:$Keertana99 only in case docker-compose build doesnt work

Login in google or desktop docker and in vs type

docker login --username keertana246@gmail.com --password $Keertana99

docker login --username [keertana246@gmail.com](mailto:keertana246@gmail.com)

Then give password

**Chk output in** [**http://localhost:5000**](http://localhost:5000)

**PROGRAM 4**

Commands (keep docker open )

minikube start --driver=docker

minikube status

kubectl get nodes

minikube start

minikube status

Go to vs make a new folder on desktop and open in vs

Create a file pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: my-nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

In terminal run kubectl apply -f pod.yaml

Check if its running:

kubectl get pods

You should see:

NAME READY STATUS RESTARTS AGE

my-nginx 1/1 Running 0 &lt;time&gt;

Only if u dont give gap and do u see 1/1

Else 0/1

kubectl get pods -o wide (it displays complete information about

the each running pods)

minikube ssh

Then **curl <nginx ip-address>** to See the NGINX welcome page inside the cluster.

Here remove <> and copy ip from above like 10.244.0.5

Do all files then

**Take new terminal**

Docker build . (do this only better)

Or **docker build -t keertana/app1-k8s:latest**

**docker push keertana/app1-k8s:latest**

**If error login to docker see up**

In case any error create new files and follow manual

next

**kubectl apply -f deployment.yaml**

**Should see hw-deployment and next command hello world**

* **kubectl scale deployment/hw-deployment --replicas=3**

**Scaled**

To see how many replicas are running:

* **kubectl get deployment**
* **kubectl get pods**

Forwards container port 5000 to host port 5000

* **kubectl port-forward svc/hello-world 5000:5000**
* **Goto browser and type http://localhost:5000**

**4b**

Clone or do all files in manual

Simply docker build .

Apply Everything

* + **kubectl apply -f configmap.yaml**
  + **kubectl apply -f secret.yaml**
  + **kubectl apply -f deployment.yaml**
  + **kubectl apply -f service.yaml**

Check the Pod and Service Status

* **kubectl get pods**
* **kubectl get svc**

**kubectl port-forward svc/python-service 8095:80**

* + - [**http://localhost:8095**](http://localhost:8095)

**Note:**

## Delete All Pods in the Current Namespace (usually default):

**kubectl delete pods --all**

## Delete Everything (Pods, Deployments, Services, etc.)

**kubectl delete all --all**

* Prevent Pods from Coming Back

**kubectl delete deployment <deployment-name> /**

**kubectl delete deployments --all**

**DOCKER KEEP OPEN ALWAYS**