**RDS Assignment : 2**

**With the help of ci/cd pipeline building war and deploying inside a tomcat container & creating Mysql container as well. & then manually making connection and creating database.**

Step-1: Launching ec2 instance ,installing java ,downloading tomcat and jenkins.war in tomcat webapps

* Accessing the jenkins publicip:8080/Jenkins
* Creating a Jenkins ci/cd pipeline

For downloading docker for creating network & container, git to clone, maven to build war

pipeline {

agent any

stages{

stage("Required Installation"){

steps{

sh "yum install maven git docker -y"

}

}

stage ("Git clone"){

steps{

dir ("/root/build") {

git 'https://github.com/Shantanumajan6/project.git'

}

}

}

stage ("Build war"){

steps{

dir ("/root/build/") {

sh "mvn clean install"

}

}

}

stage ("Creating Docker Network"){

steps {

sh "service docker start"

sh "docker network create mynetwork"

}

}

stage("Tomcat container"){

steps {

sh "docker run -itd --name tomcat\_cont -p 80:8080 --network mynetwork -v /root/build/target/:/usr/local/tomcat/webapps/ tomcat:9"

}

}

stage("Mysql container"){

steps {

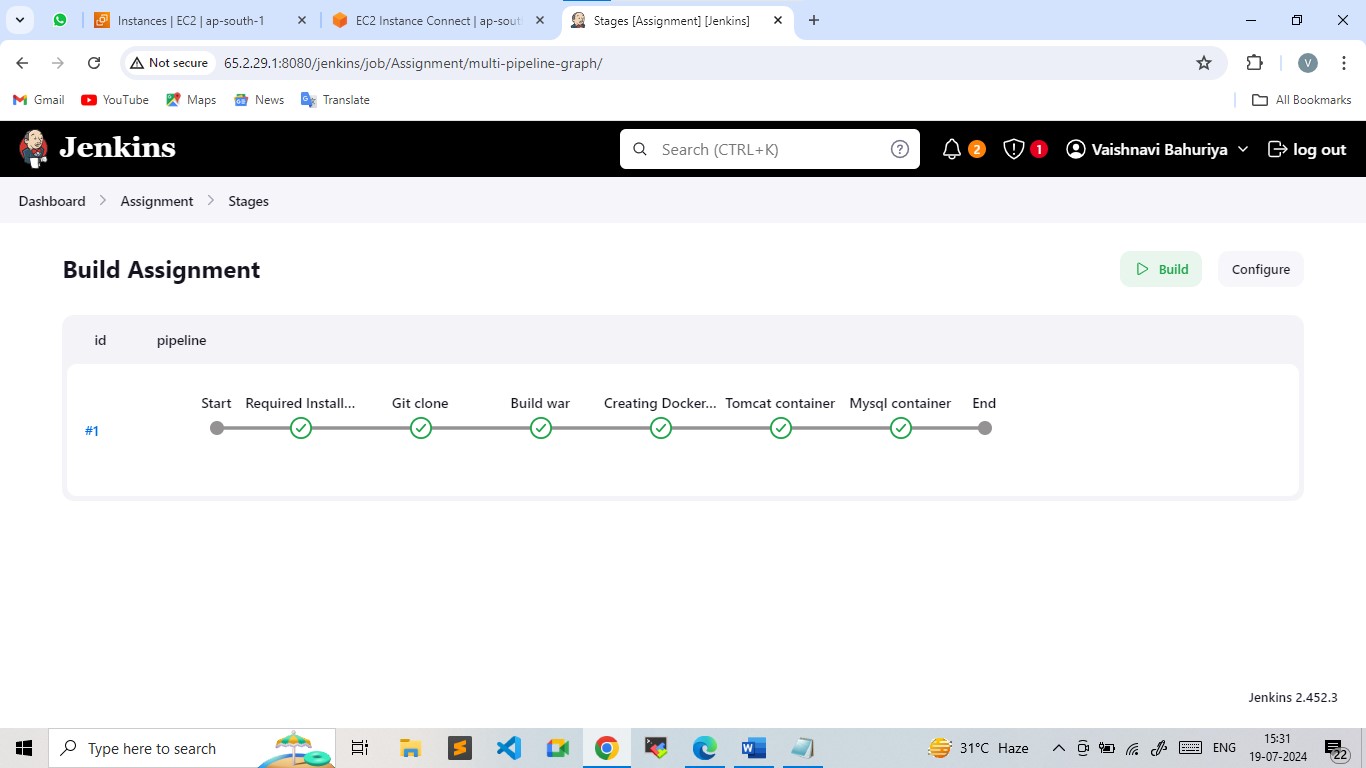
sh "docker run --name mysql --network mynetwork -e MYSQL\_ROOT\_PASSWORD=root -d mysql:latest "

}

}

}

}



Step-2: Login to Mysql container and creating database inside it

* docker exec -it sql mysql -u root -p
* create database test;
* use test;

CREATE TABLE USER (

id int(10) unsigned NOT NULL auto\_increment,

first\_name varchar(45) NOT NULL,

last\_name varchar(45) NOT NULL,

email varchar(45) NOT NULL,

username varchar(45) NOT NULL,

password varchar(45) NOT NULL,

regdate date NOT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Step-3: Login to the tomcat container and made changes in userRegistration.jsp file for connection

* docker exec -it tomcat\_cont bash
* cd webapps/
* cd LoginWebApp
* cat userRegistration.jsp

Connection con = DriverManager.getConnection("jdbc:mysql://172.18.0.3:3306/test",

"root", "root");

* To get the ip address
* docker network inspect mynetwork

(Here you can see the ip address of the sqlcontainer add that ip in LoginWebapp userRegistration )

* Access the application

<http://public:8080/LoginWebApp>

* Add the details in registration form
* Now check in Mysql container if the data is stored in database or not
* docker exec -it sql mysql -u root -p
* use test;
* select \* from USER;