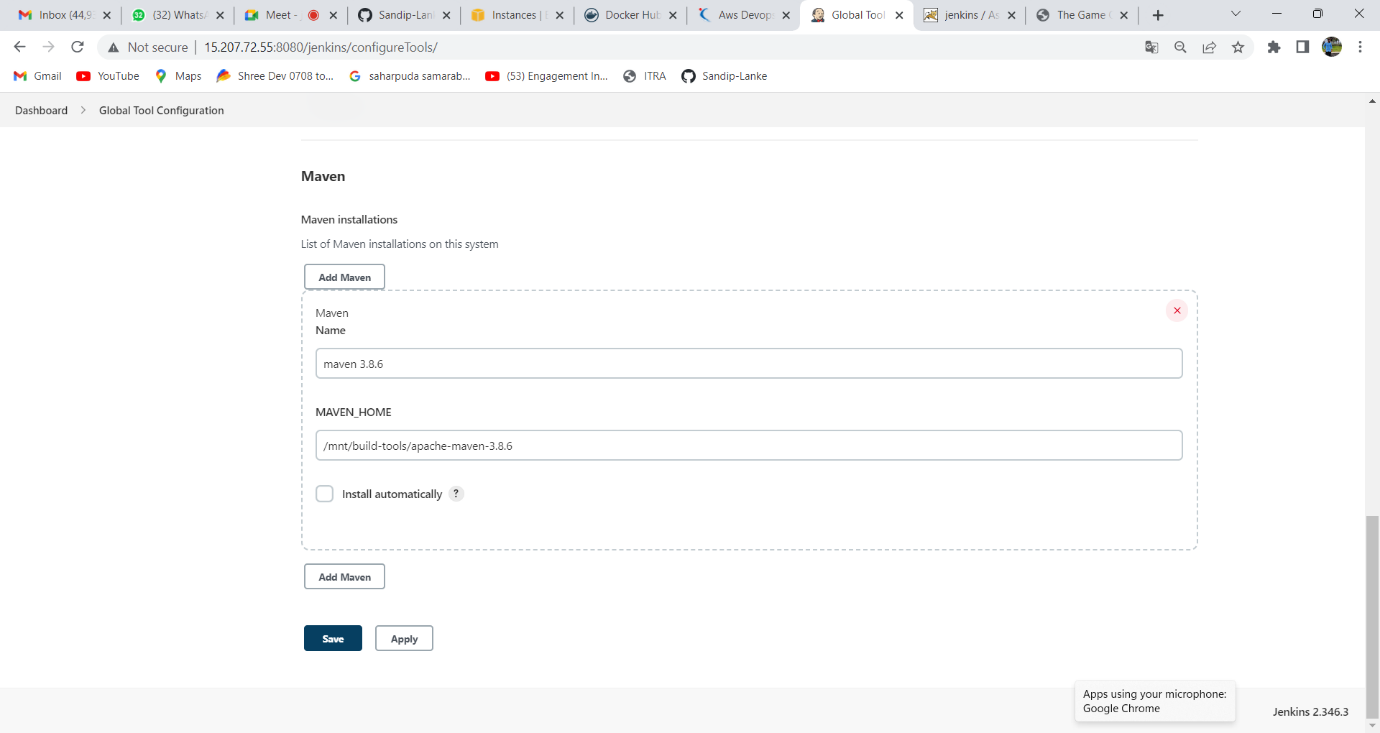
Assignment -7

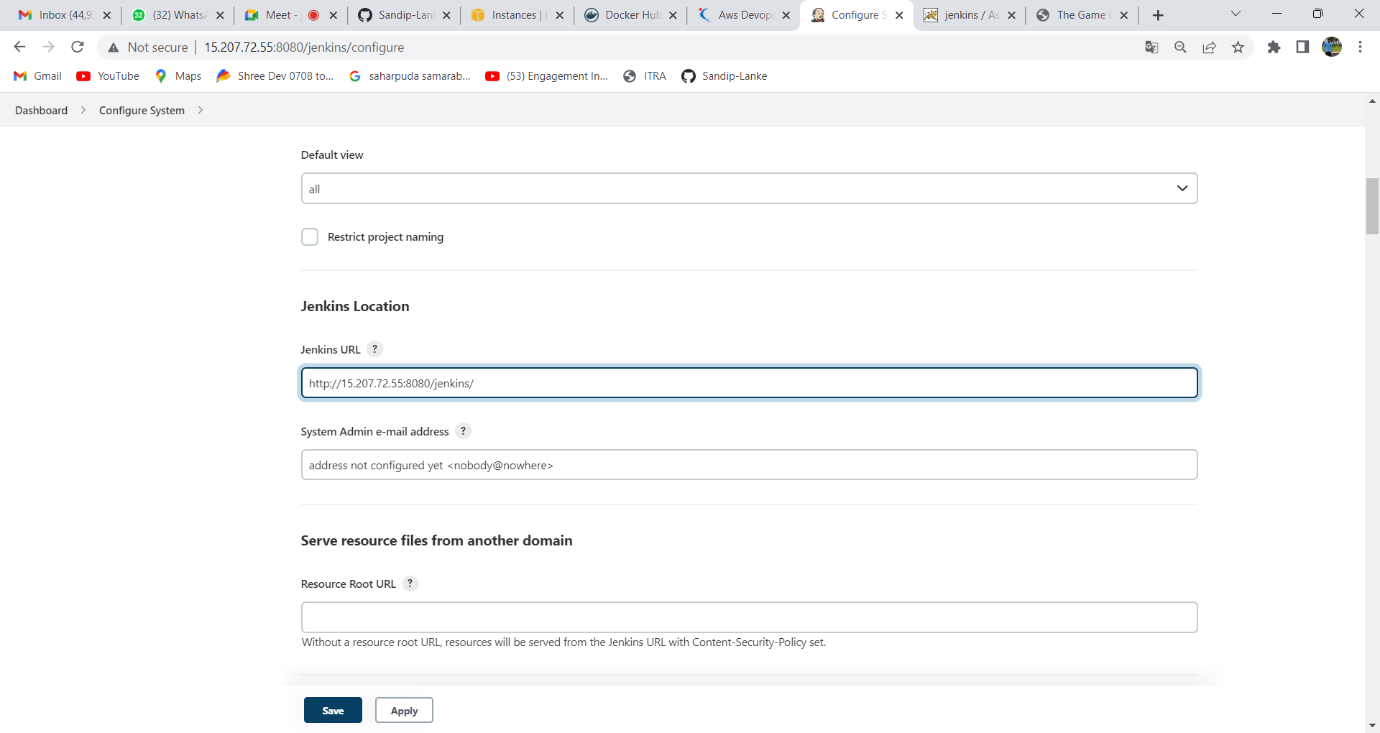
**Question** - Clone game of life in master and build, then create custom image for Apache tomcat using dockerfile and deploy war file in tomcat container using docker compose.

**Step -1 :** configure Jenkins master –

1. Yum update -y
2. Install java 8.
3. Install Apache tomcat, start tomcat – deploy Jenkins in tomcat webapps.
4. Install maven. Set environment variable.

**Step-2 :**  Jenkins console :

1. Setup maven tool in global tool configuration.
2. Set url of Jenkins



3. Create Dockerfile /mnt/assignment7/war/

FROM tomcat:9

MAINTAINER sandip

COPY gameoflife.war /usr/local/tomcat/webapps/

EXPOSE 8080

CMD ["catalina.sh", "run"]

4. Create docker-compose.yaml /mnt/assignment7/

version: '3'

services:

service-tomcat:

build:

context: /mnt/assignment7/war

image: tomcat-server

ports:

- "8085:8080"

5.Create job -> select pipeline job 🡪

Pipelinescript:

pipeline{

agent{

label{

label'built-in'

customWorkspace'/mnt/assignment7'

}

}

stages{

stage('stage-1-git-clone'){

steps{

sh "rm -rf game\*"

sh "git clone https://github.com/Sandip-Lanke/game-of-life.git"

}

}

stage('stage-2-maven'){

steps{

sh "rm -rf /root/.m2/repository"

sh " cd /mnt/assignment7/game-of-life && mvn clean install"

}

}

stage('stage-3-copy war-permission'){

steps{

sh "cp -r /mnt/assignment7/game-of-life/gameoflife-web/target/gameoflife.war /mnt/assignment7/war"

sh "chmod -R 777 /mnt/assignment7/war/gameoflife.war"

}

}

stage('stage-4-docker'){

steps{

sh "yum install docker -y"

sh " systemctl start docker"

sh "curl -SL https://github.com/docker/compose/releases/download/v2.11.0/docker-compose-linux-x86\_64 -o /usr/local/bin/docker-compose"

sh "sudo chmod +x /usr/local/bin/docker-compose"

sh "sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose"

sh "sudo chmod +x /usr/bin/docker-compose"

sh " sleep 3"

sh "docker-compose up -d "

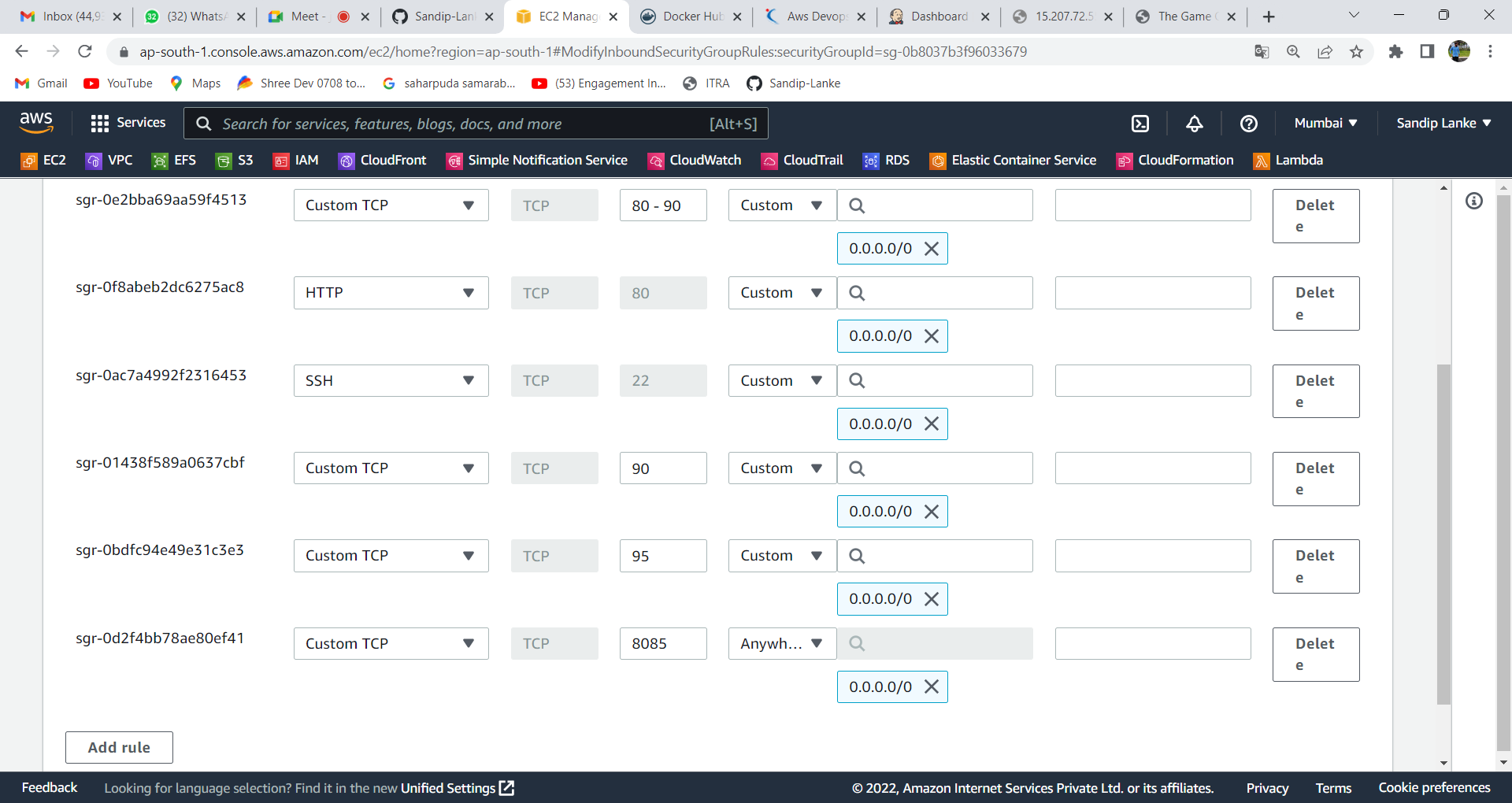
}

}

}

}

Step 5: open port 8085 in SG



Step 6: Build the project and check console output and verify the result.

