## Assignment 1

Deploy gameoflife.war file on a tomcat 9 container using Dockerfile.

Put the Dockerfile in Git, then clone it and clone the gameoflife project from Git as well.

## Pipeline

**pipeline {**

**agent {**

**label {**

**label "built-in"**

**customWorkspace "/mnt/jenkins/"**

**}**

**}**

**stages {**

**stage ('install-git') {**

**steps {**

**sh "yum install git -y"**

**}**

**}**

**stage ('clone-project') {**

**steps {**

**dir ('/mnt/project/') {**

**sh ("rm -rf \*")**

**sh "git clone https://github.com/aakashchavhan/game-of-life.git"**

**}**

**}**

**}**

**stage ('download-maven') {**

**steps {**

**dir ('/mnt/build-tool/') {**

**sh "rm -rf \*"**

**sh "wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.zip"**

**sh "unzip apache-maven-3.8.6-bin.zip"**

**}**

**}**

**}**

**stage ('build-app') {**

**steps {**

**dir ('/mnt/project/game-of-life/') {**

**sh "yum install maven -y"**

**sh "mvn install"**

**}**

**}**

**}**

**stage ('copy-app') {**

**steps {**

**dir ('/mnt/project/game-of-life/gameoflife-web/target/') {**

**sh "cp -r gameoflife.war /mnt"**

**}**

**}**

**}**

**stage ('install-start-docker') {**

**steps {**

**sh "yum install docker -y"**

**sh "systemctl start docker"**

**}**

**}**

**stage ('cloning-dockerfile') {**

**steps {**

**dir ('/mnt') {**

**sh "rm -rf Docker"**

**sh "git clone https://github.com/aakashchavhan/Docker.git"**

**sh "cp -r gameoflife.war /mnt/Docker"**

**}**

**}**

**}**

**stage ('creating-container') {**

**steps {**

**dir ('/mnt/Docker') {**

**sh "docker build -t tom ."**

**sh "docker run -itdp 8085:8080 tom"**

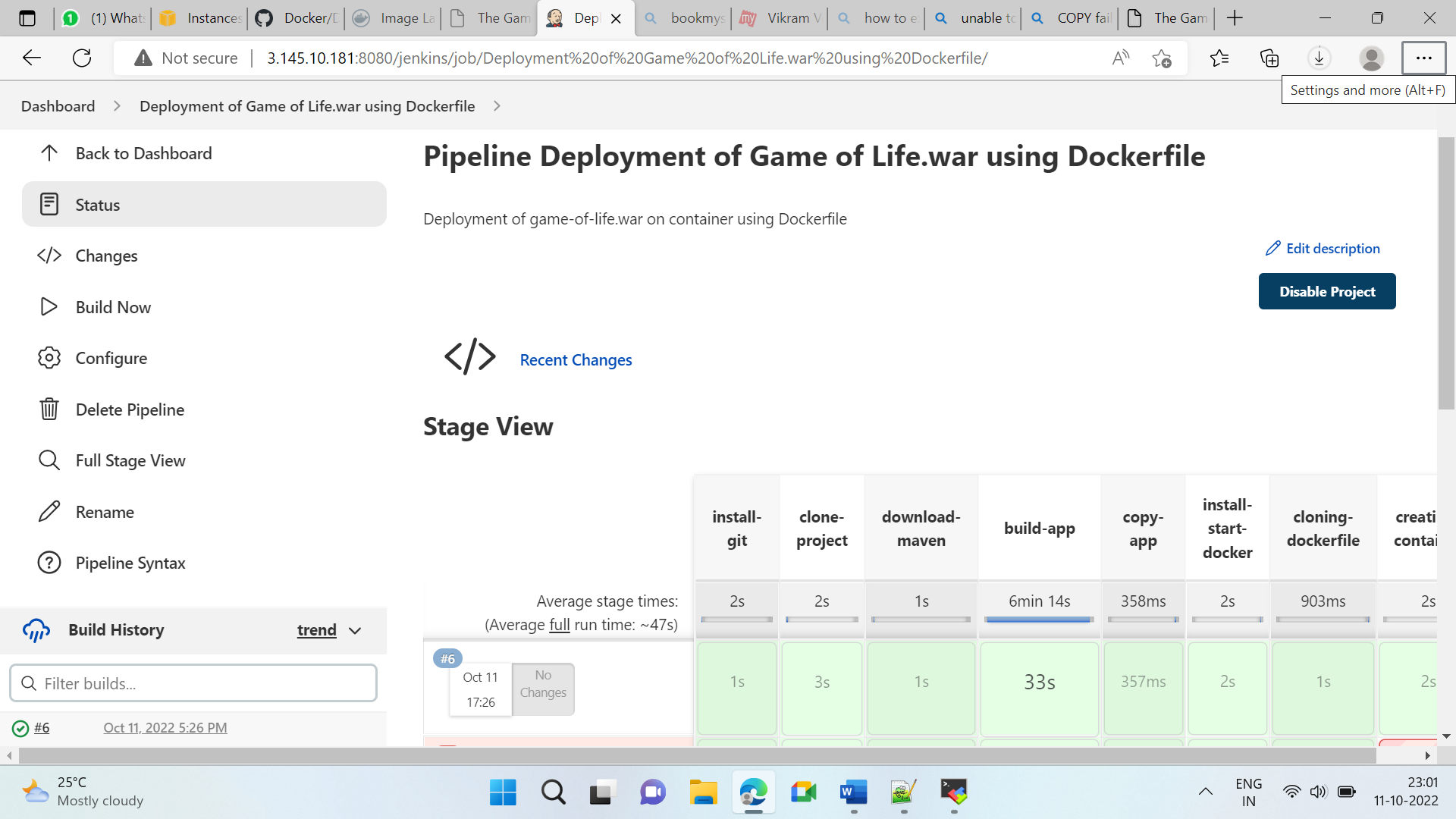
**}**

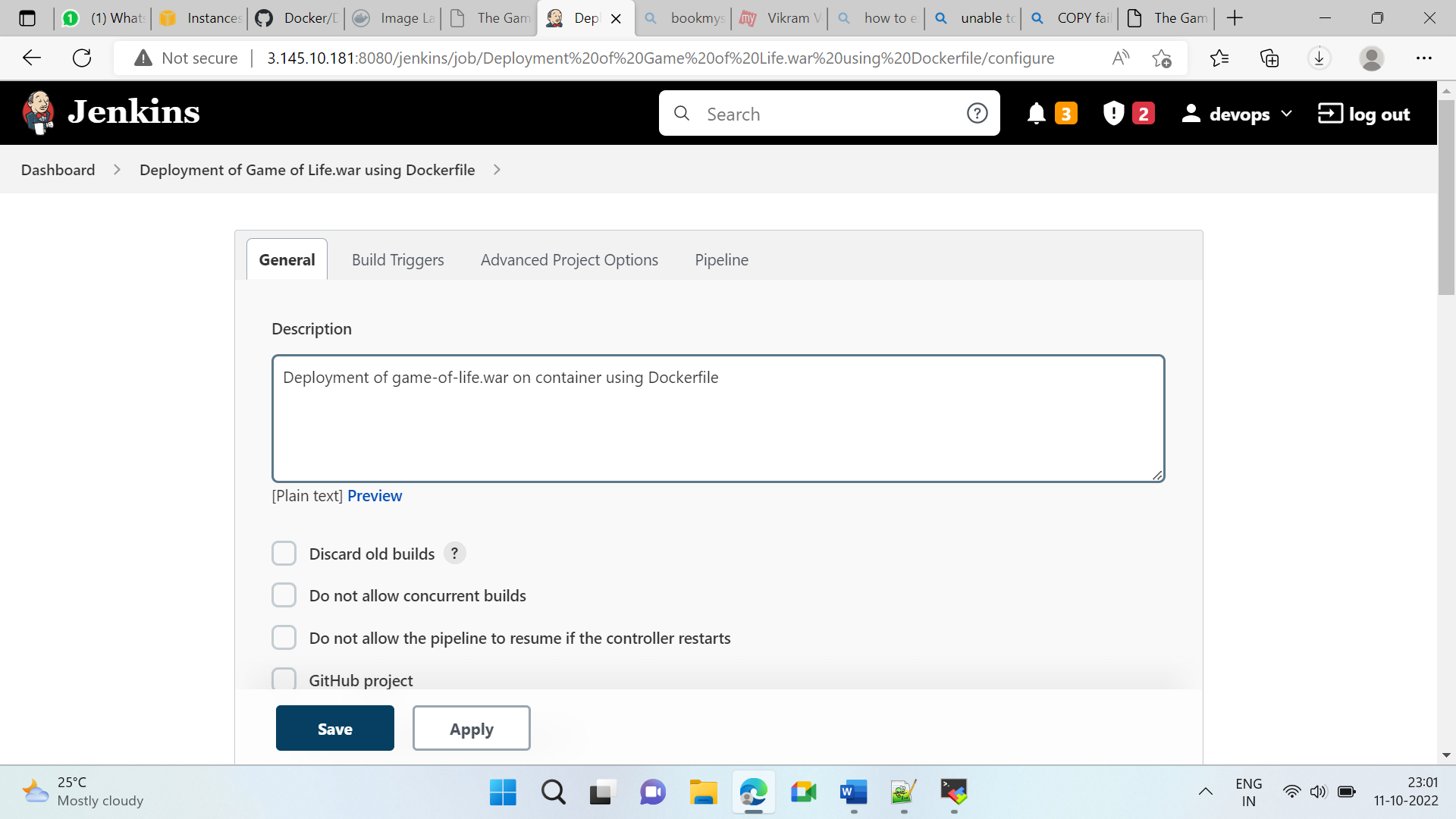
**}**

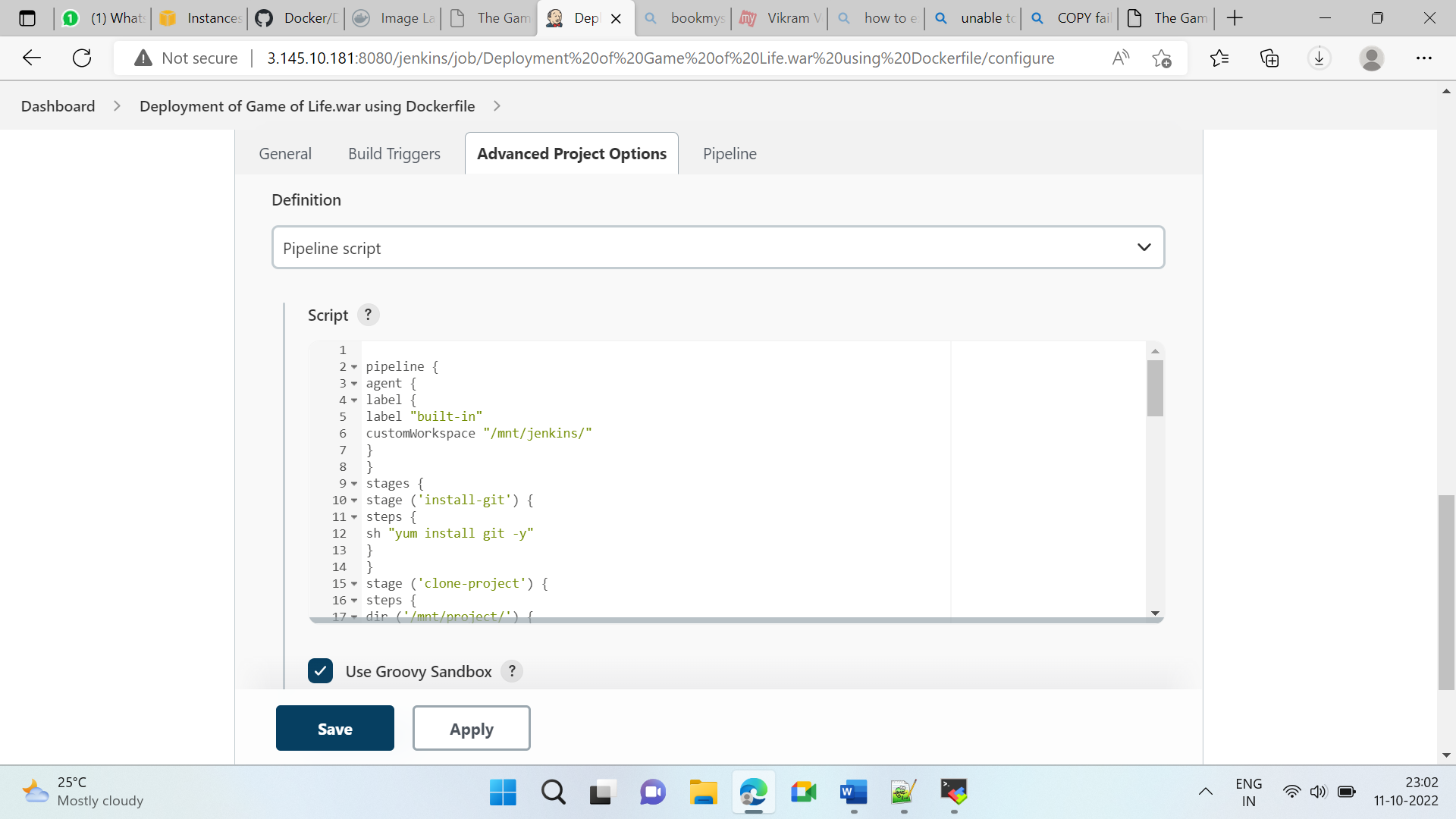
**}**

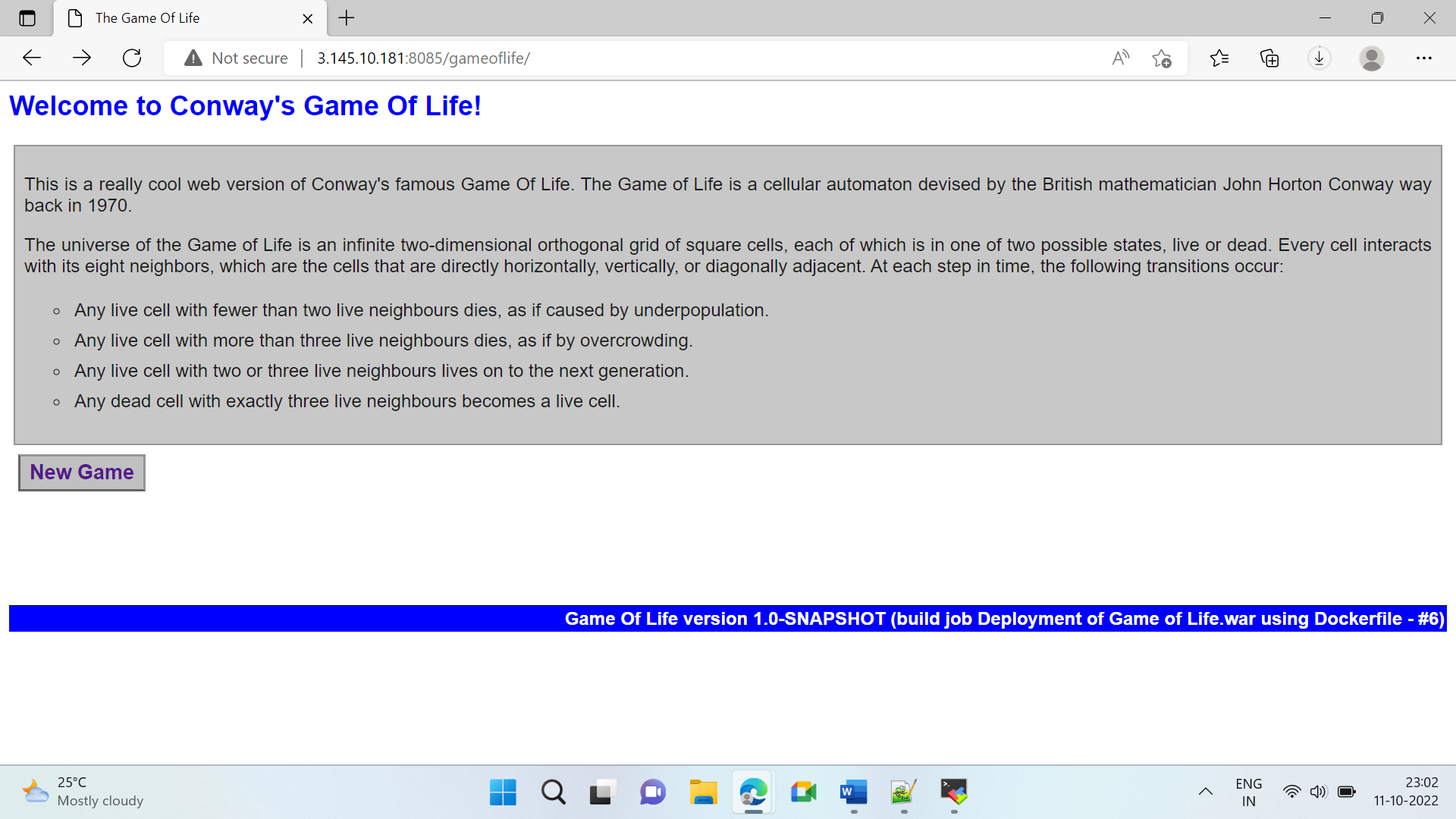
**}**

**}**





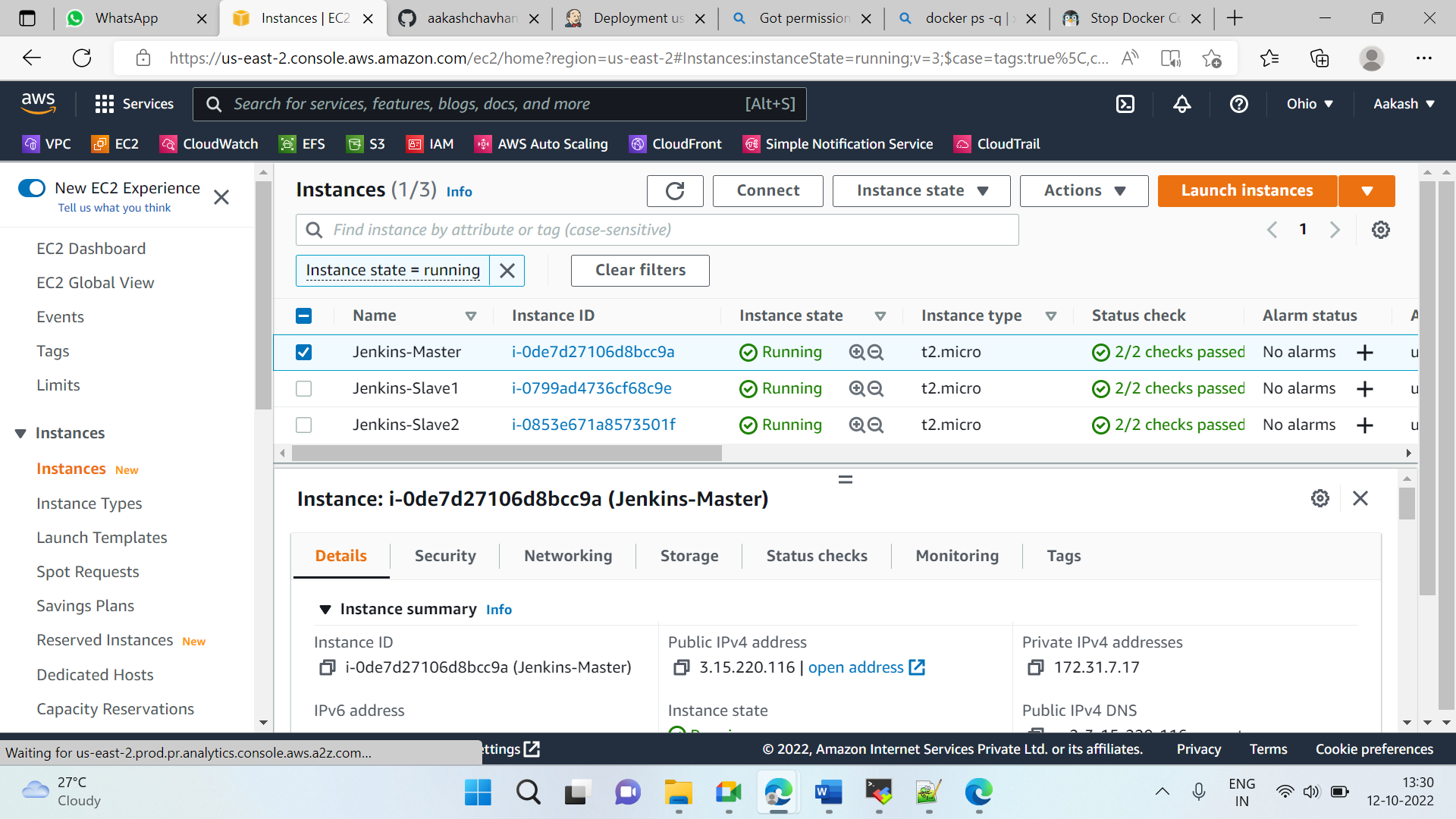


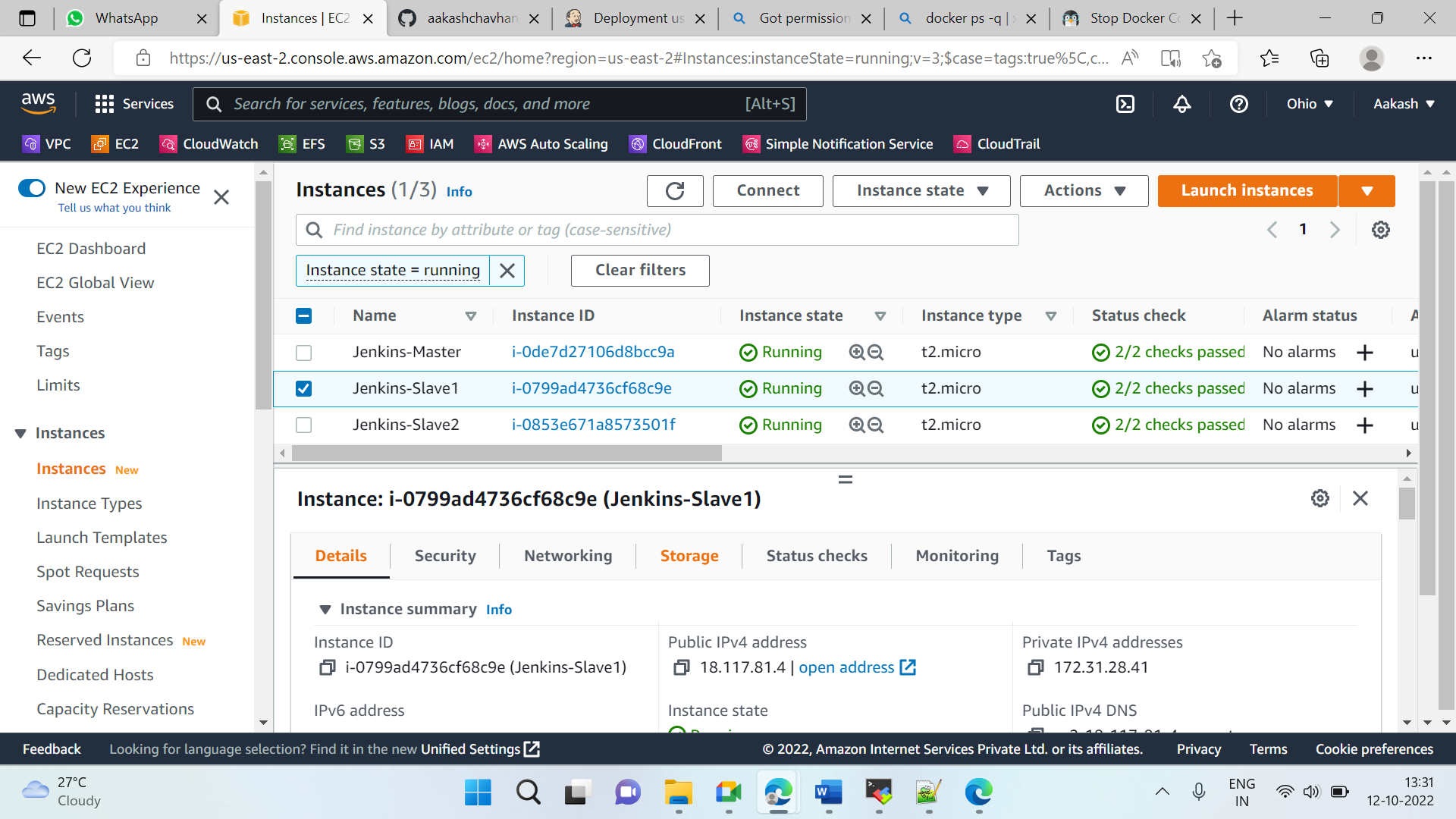


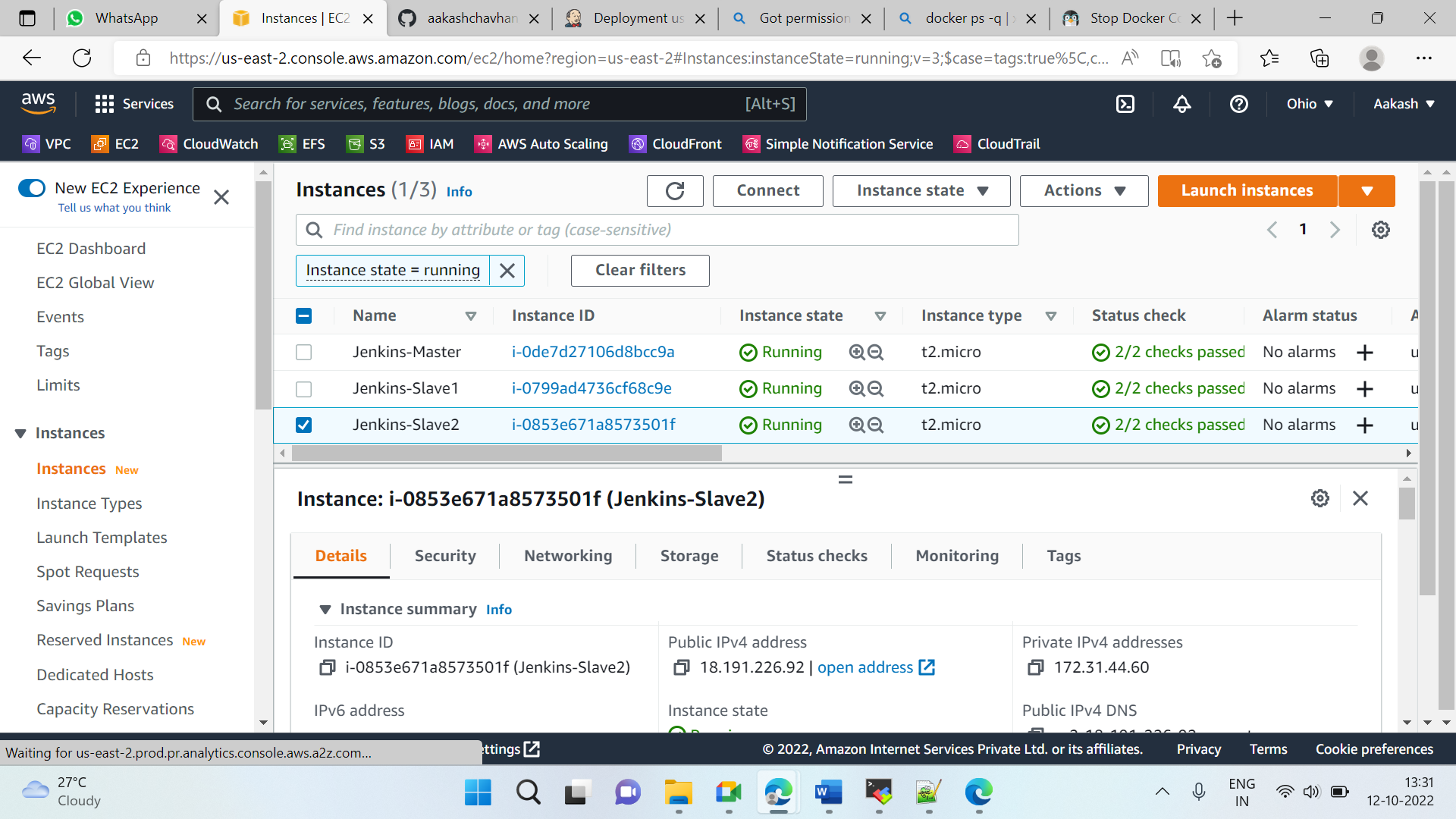
# Assignment 2

## Step 1: Created 1 JM with 8081 and 8082 port open and 2 Slaves.

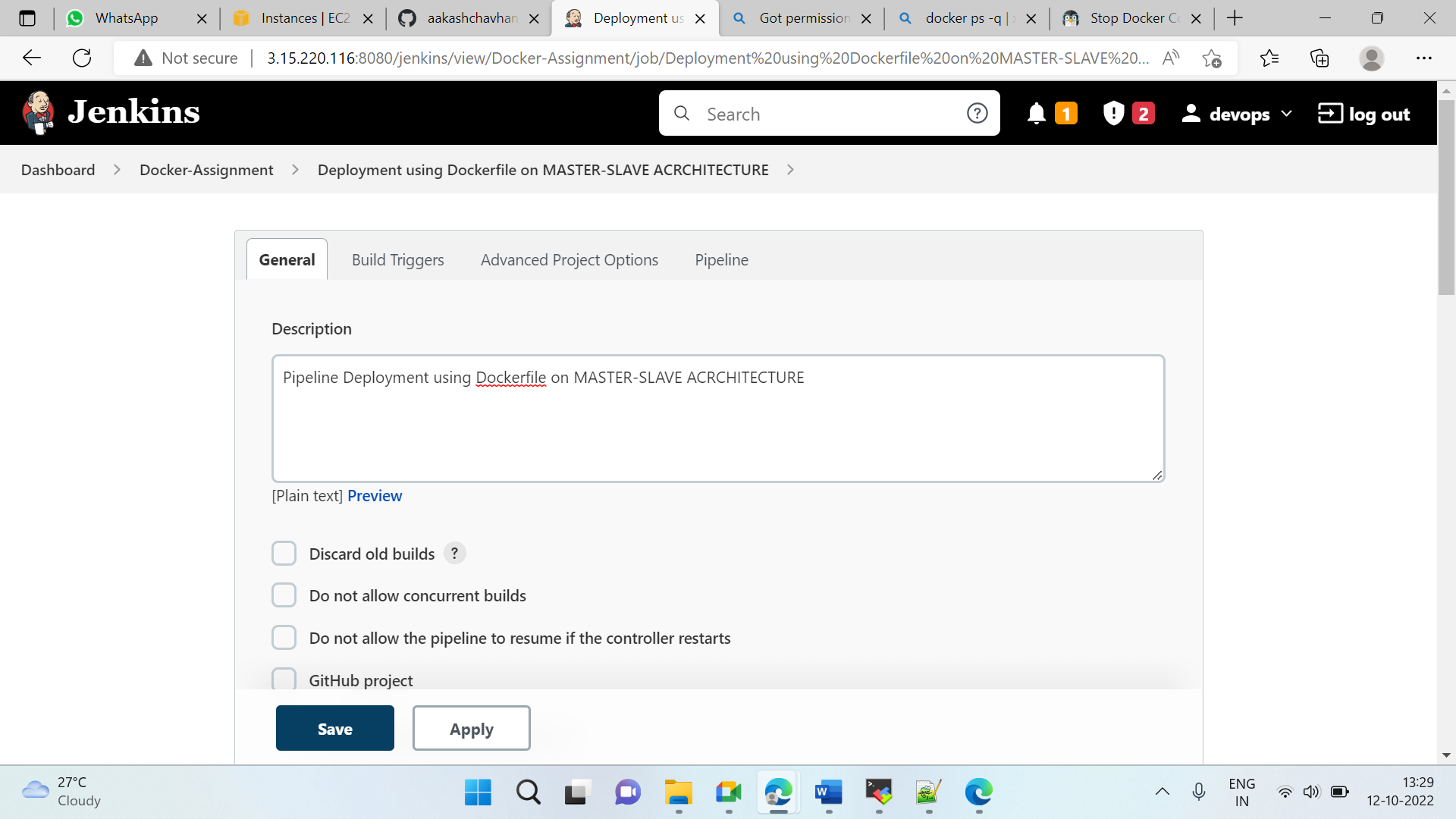
## Step 2: Install Maven, Git, Docker, Java, Tomcat on JM and Slaves except Tomcat.







## Step 3: Created Job



## Dockerfile

FROM tomcat:9.0.68-jre8-temurin-jammy

MAINTAINER aakash\_90

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

EXPOSE 8080

WORKDIR /usr/local/tomcat/webapps

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

## Step 4: Added Pipeline

pipeline {

agent {

label {

label ('built-in')

customWorkspace '/mnt/project'

}

}

stages {

stage ('clone project') {

steps {

sh "rm -rf \*"

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

}

}

stage ('build-app') {

steps {

dir ('/mnt/project/game-of-life') {

sh "yum install maven -y"

sh "mvn install"

}

}

}

stage ('copy-app-slave') {

steps {

dir ('/mnt/') {

sh "scp -i 'ohio-kp.pem' /mnt/project/game-of-life/gameoflife-web/target/gameoflife.war ec2-user@172.31.28.41:/mnt/jenkins-slave1/"

sh "scp -i 'ohio-kp.pem' /mnt/project/game-of-life/gameoflife-web/target/gameoflife.war ec2-user@172.31.44.60:/mnt/jenkins-slave2/"

}

}

}

stage ('clone dockerfile') {

agent {

label {

label ('Slave-1')

customWorkspace '/mnt/jenkins-slave1/'

}

}

steps {

sh "sudo chmod -R 777 /mnt"

sh "rm -rf Docker\*"

sh "sudo git clone https://github.com/aakashchavhan/Docker.git"

sh "sudo cp -r gameoflife.war /mnt/jenkins-slave1/Docker/"

}

}

stage ('creating container-1') {

agent {

label {

label ('Slave-1')

}

}

steps {

dir ('/mnt/jenkins-slave1/Docker') {

sh "sudo docker build -t tom ."

sh "sudo docker run -itdp 8081:8080 tom"

}

}

}

stage ('clone dockerfile 2') {

agent {

label {

label ('Slave-2')

customWorkspace '/mnt/jenkins-slave2/'

}

}

steps {

sh "sudo chmod -R 777 /mnt"

sh "sudo rm -rf Docker\*"

sh "sudo git clone https://github.com/aakashchavhan/Docker.git"

sh "sudo cp -r gameoflife.war /mnt/jenkins-slave2/Docker/"

}

}

stage ('creating container-2') {

agent {

label {

label ('Slave-2')

}

}

steps {

dir ('/mnt/jenkins-slave2/Docker') {

sh "sudo docker build -t tom ."

sh "sudo docker run -itdp 8082:8080 tom"

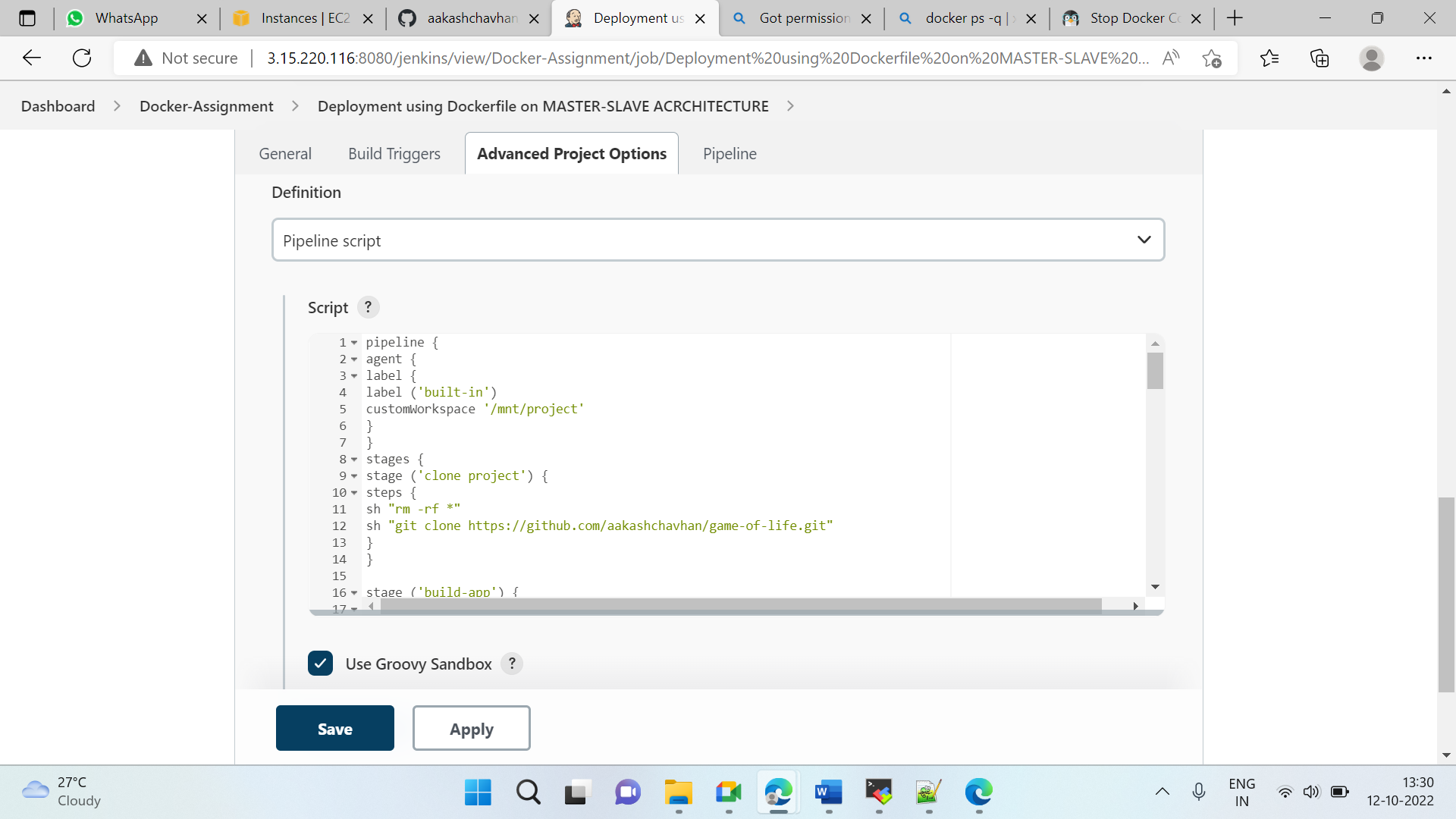
}

}

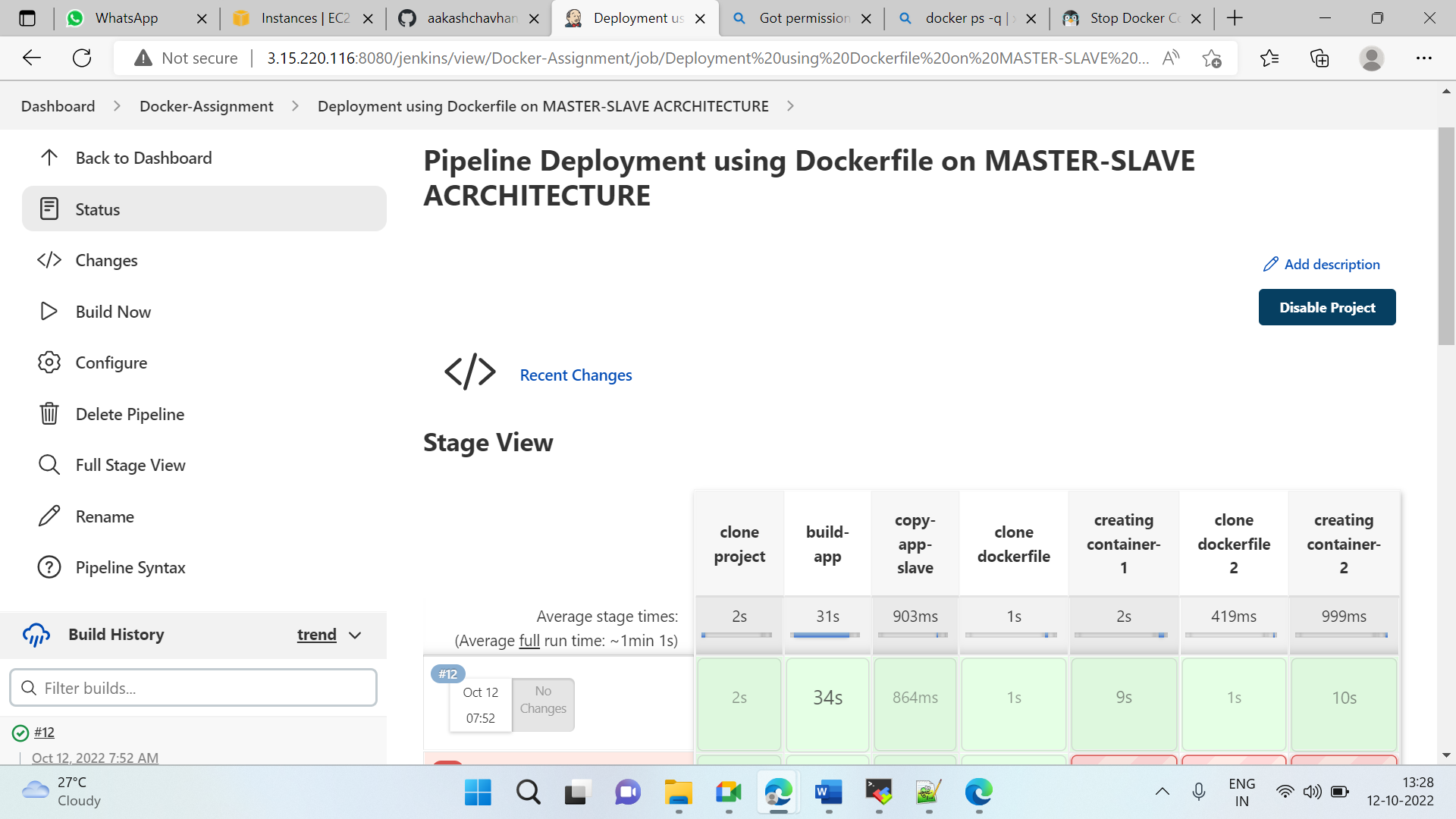
}

}

}



## Step 5: Build Now



### Step 6: Copy paste public ip:port/gameoflife

