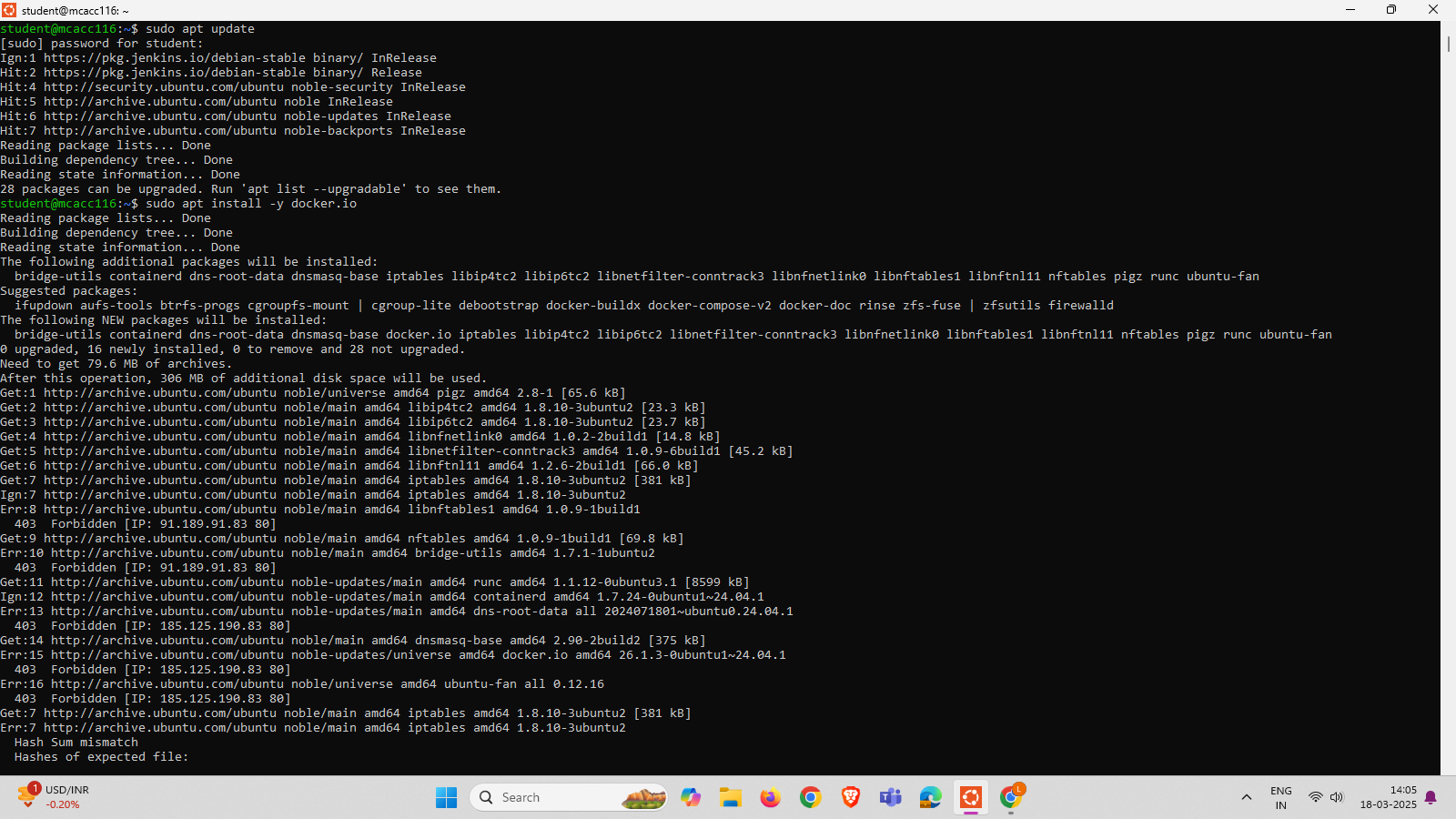
**DAY-2**

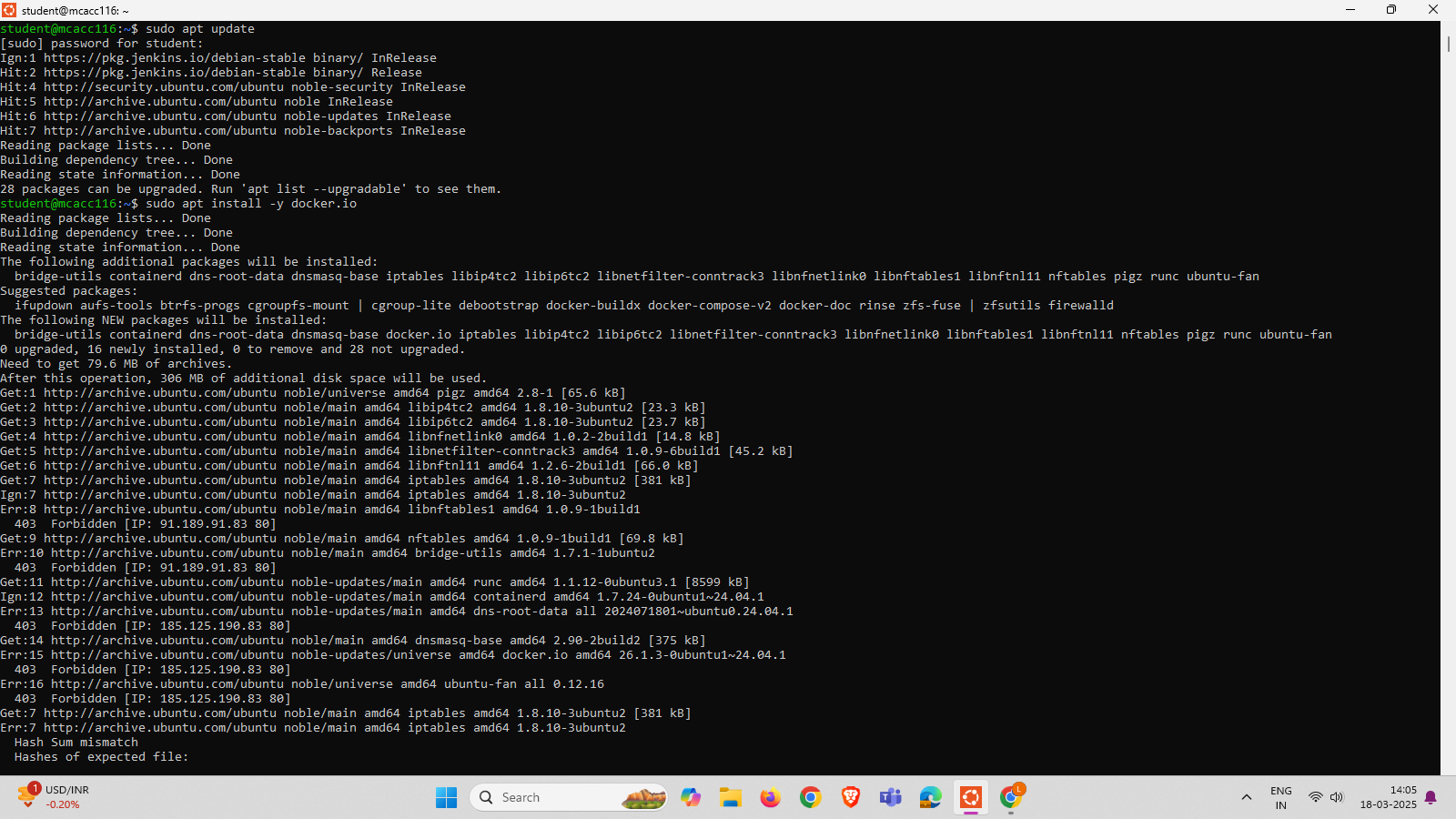
**DEVOPS**

STEP -1 :INSTALL DOCKER

sudo apt update

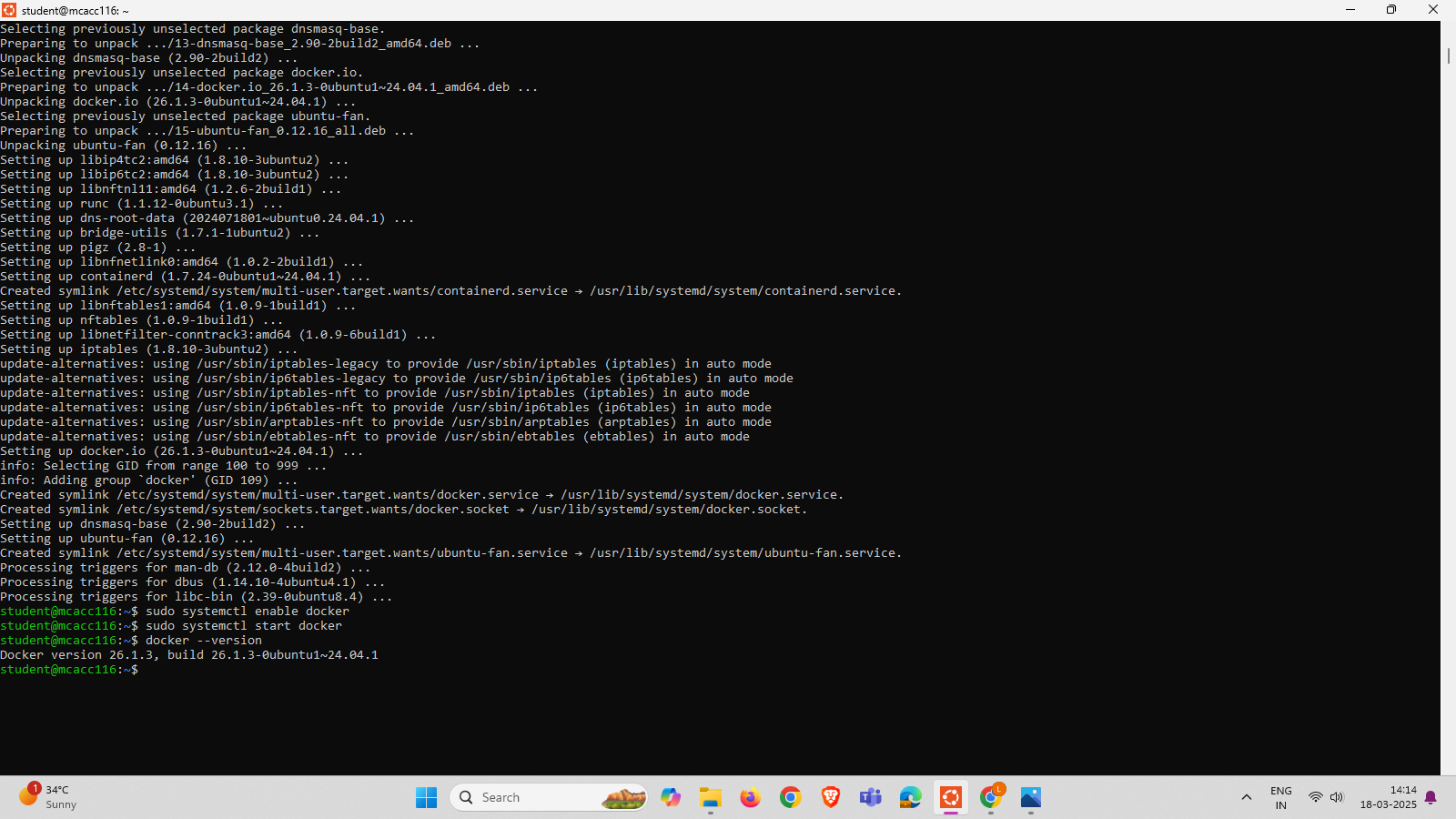


sudo apt install -y docker.io



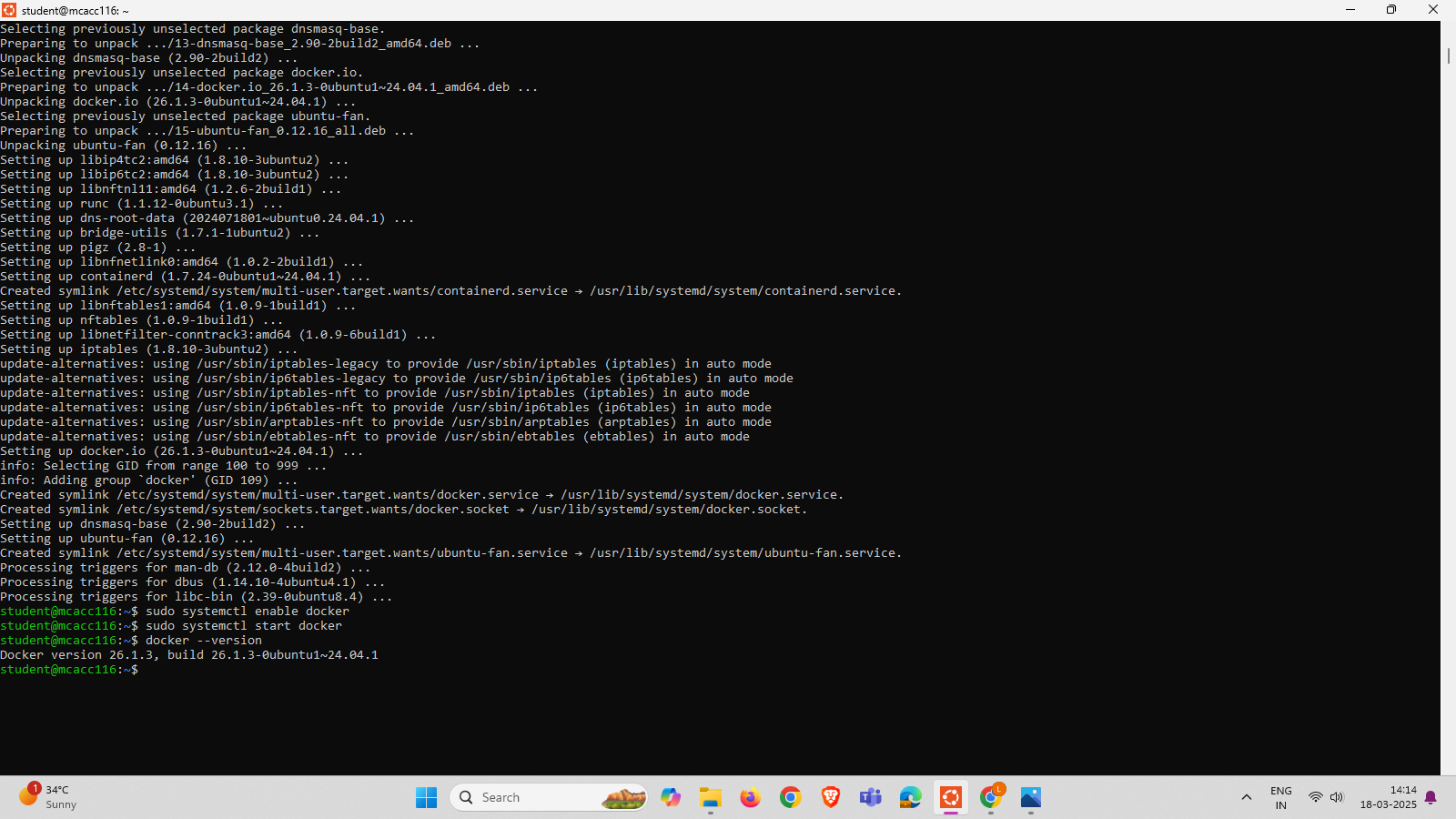
STEP 2: ENABLE AND DISABLE

1. sudo systemctl enable docker
2. sudo systemctl start docker



STEP 3:VERIFY THE INSTALLATION:

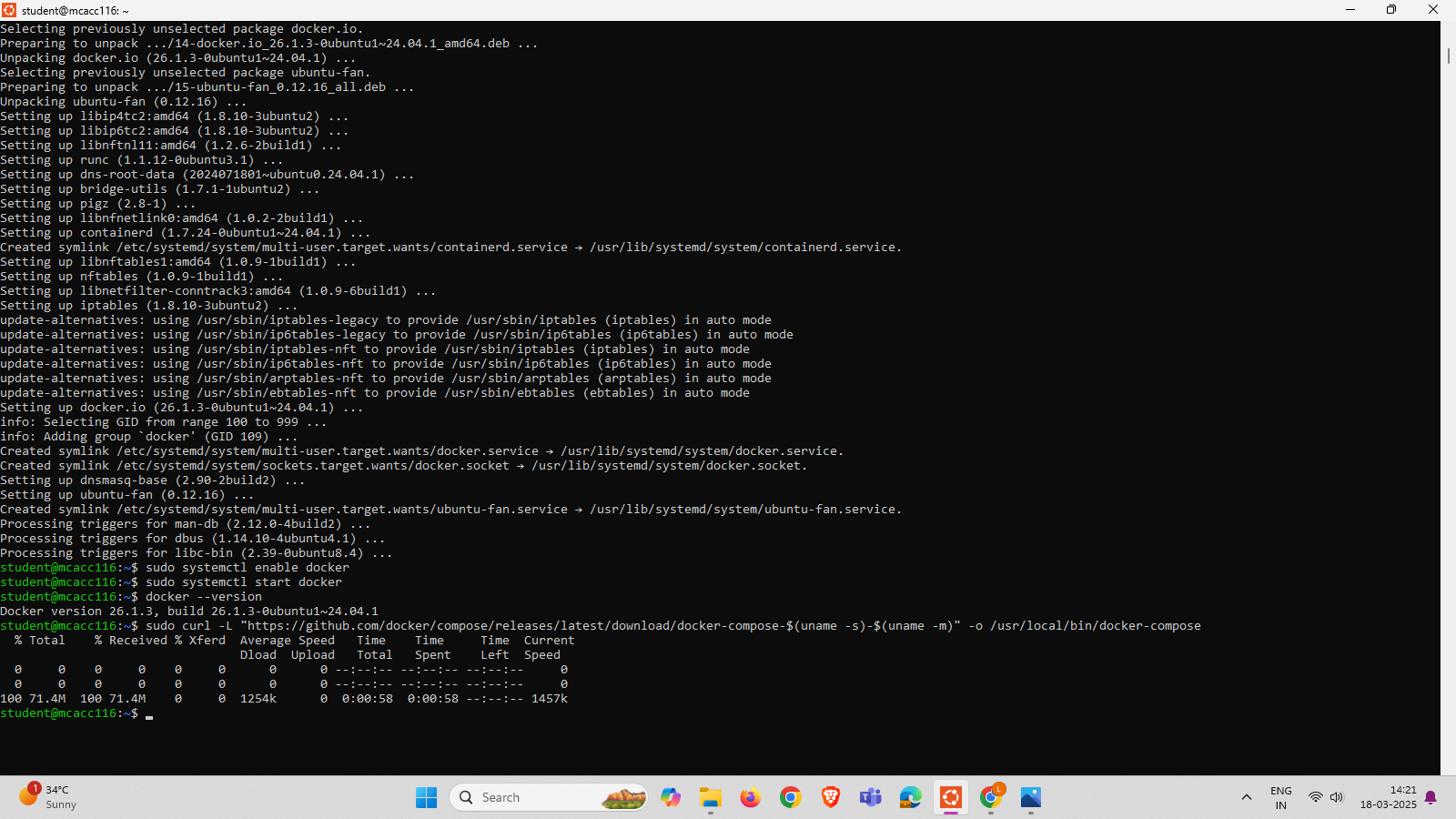
docker –version



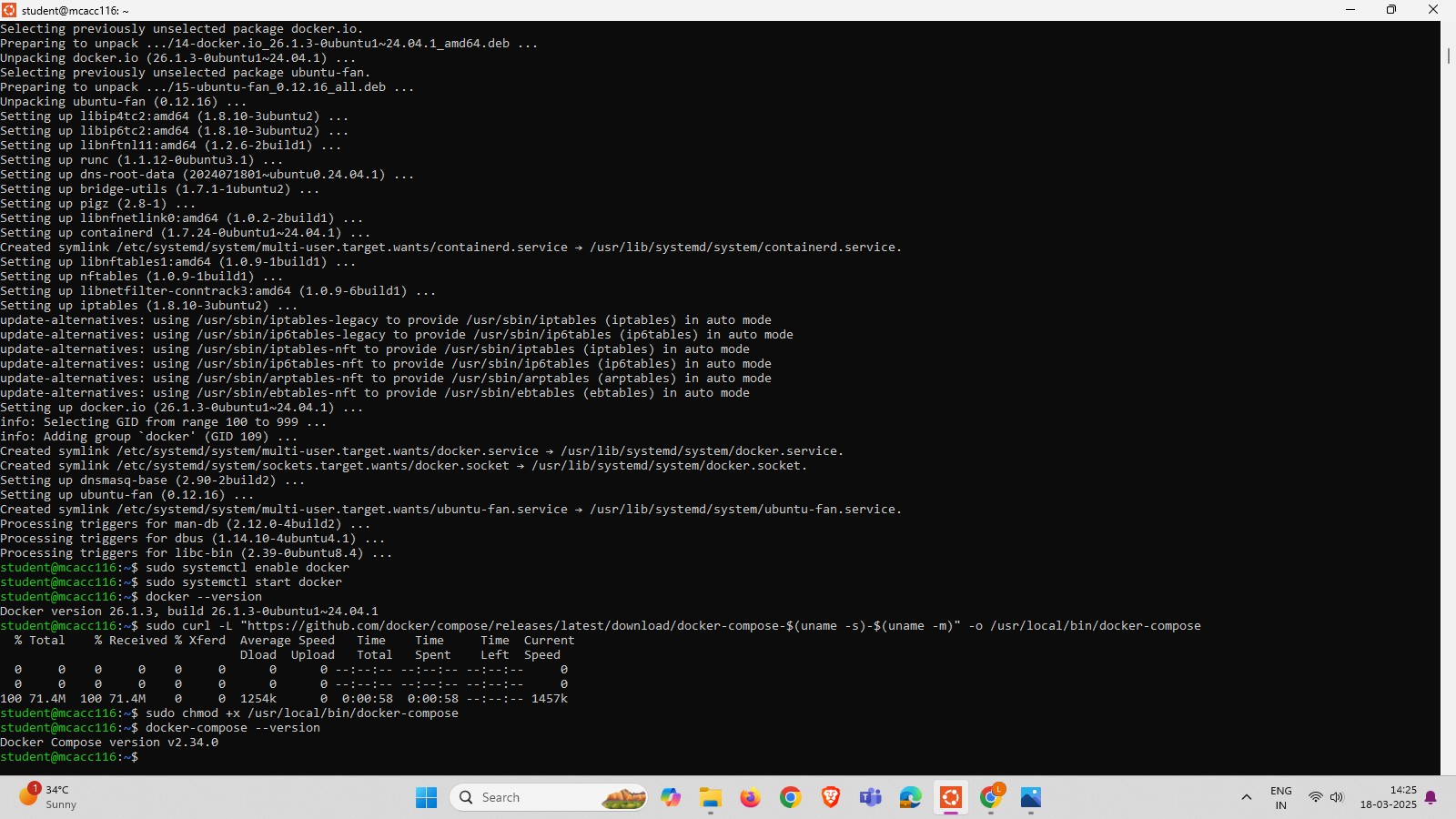
STEP 4:INSTALL DOCKER COMPOSE

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-

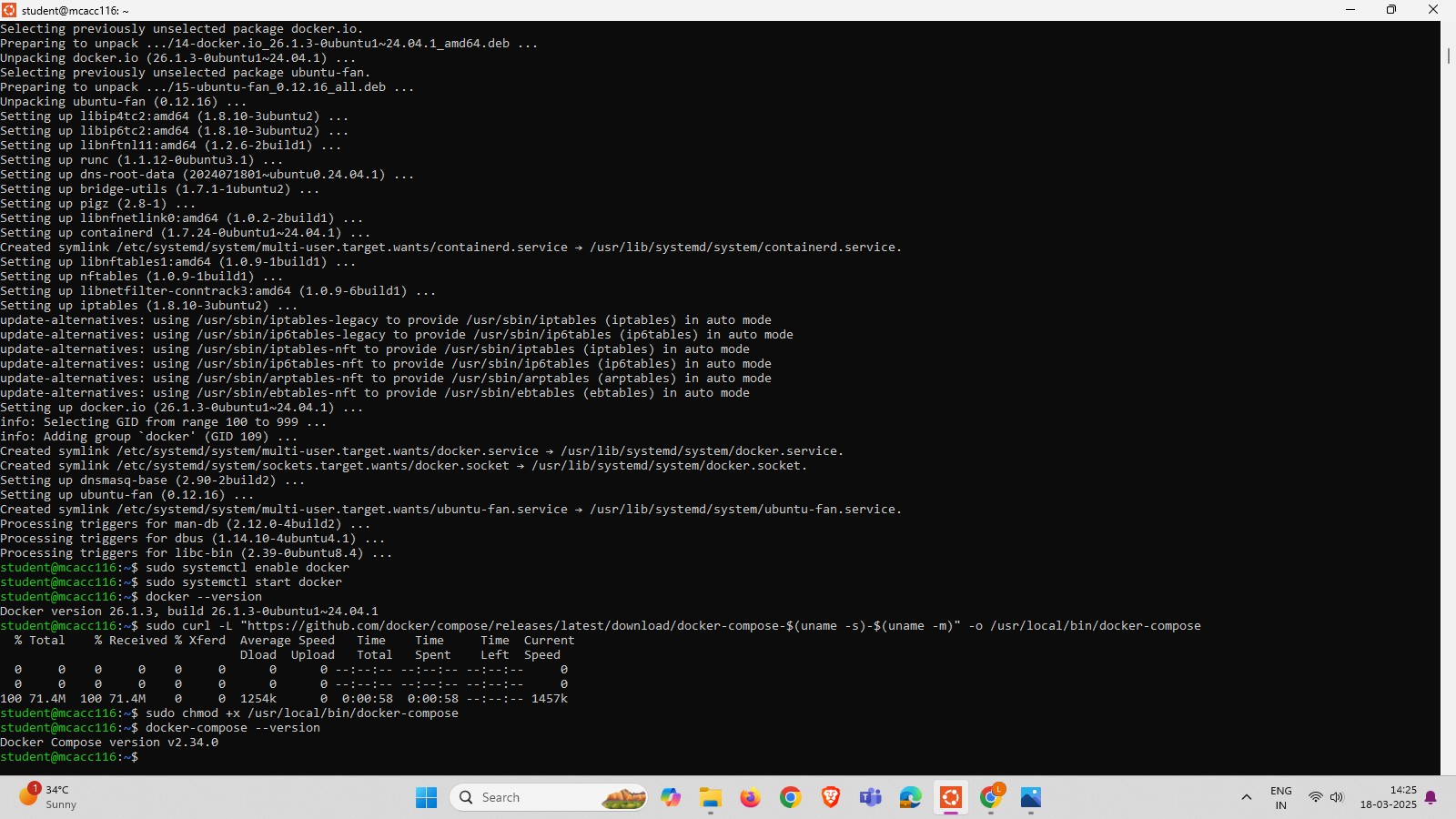
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose



Give execution permission:

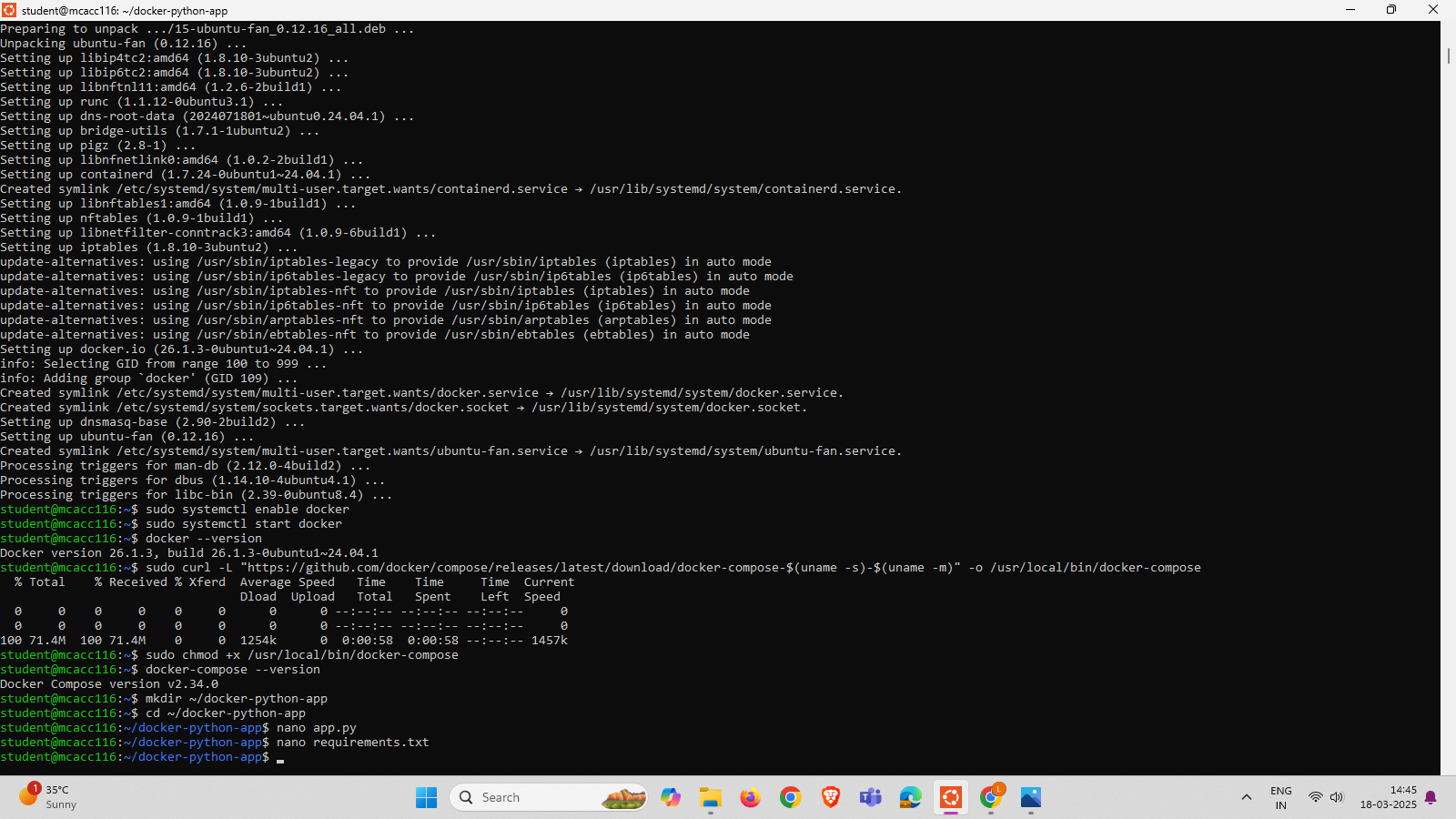


VERIFY INSTALLATION



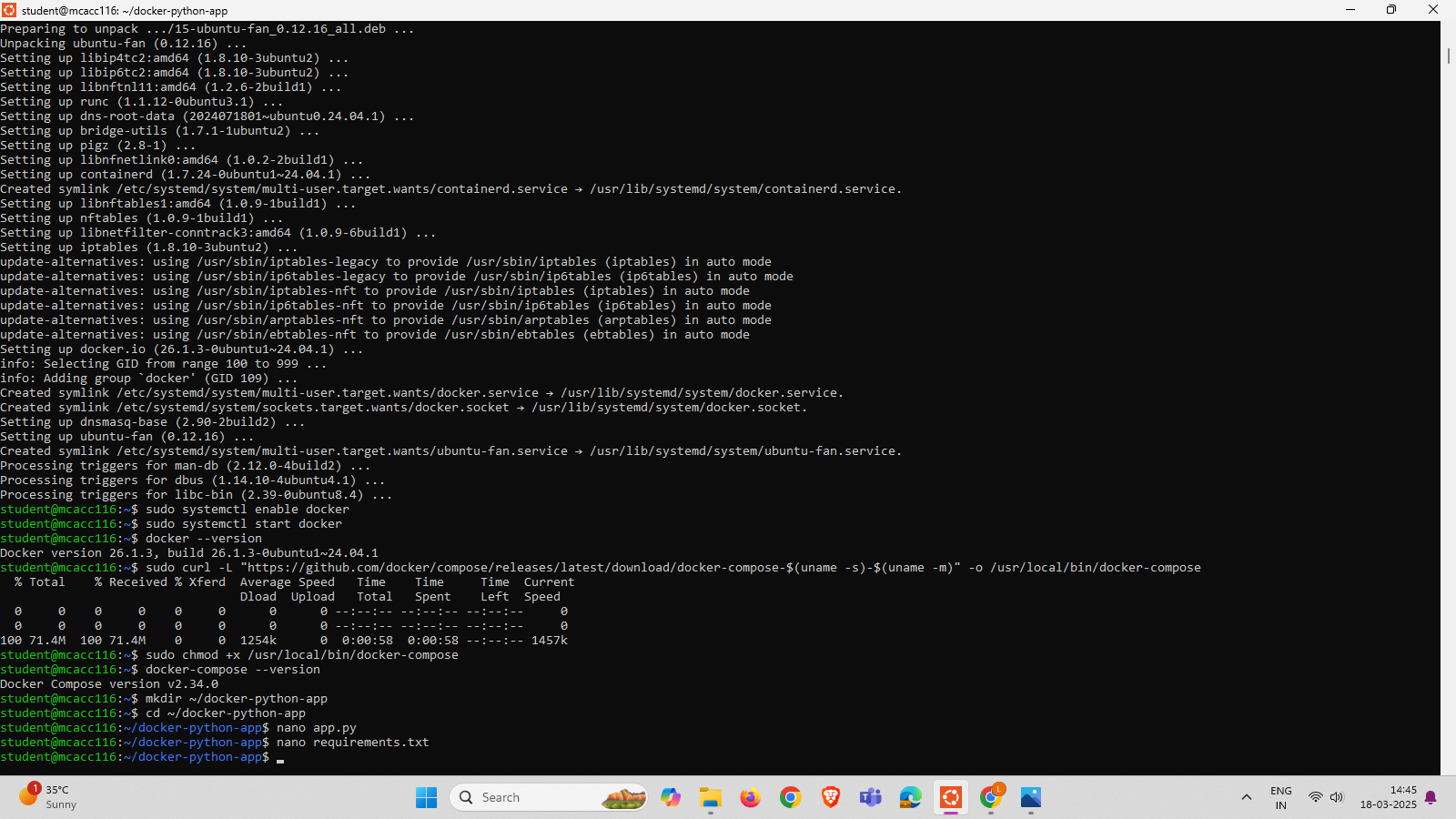
CREATE AN “HELLO WOLRD: APPLICATION

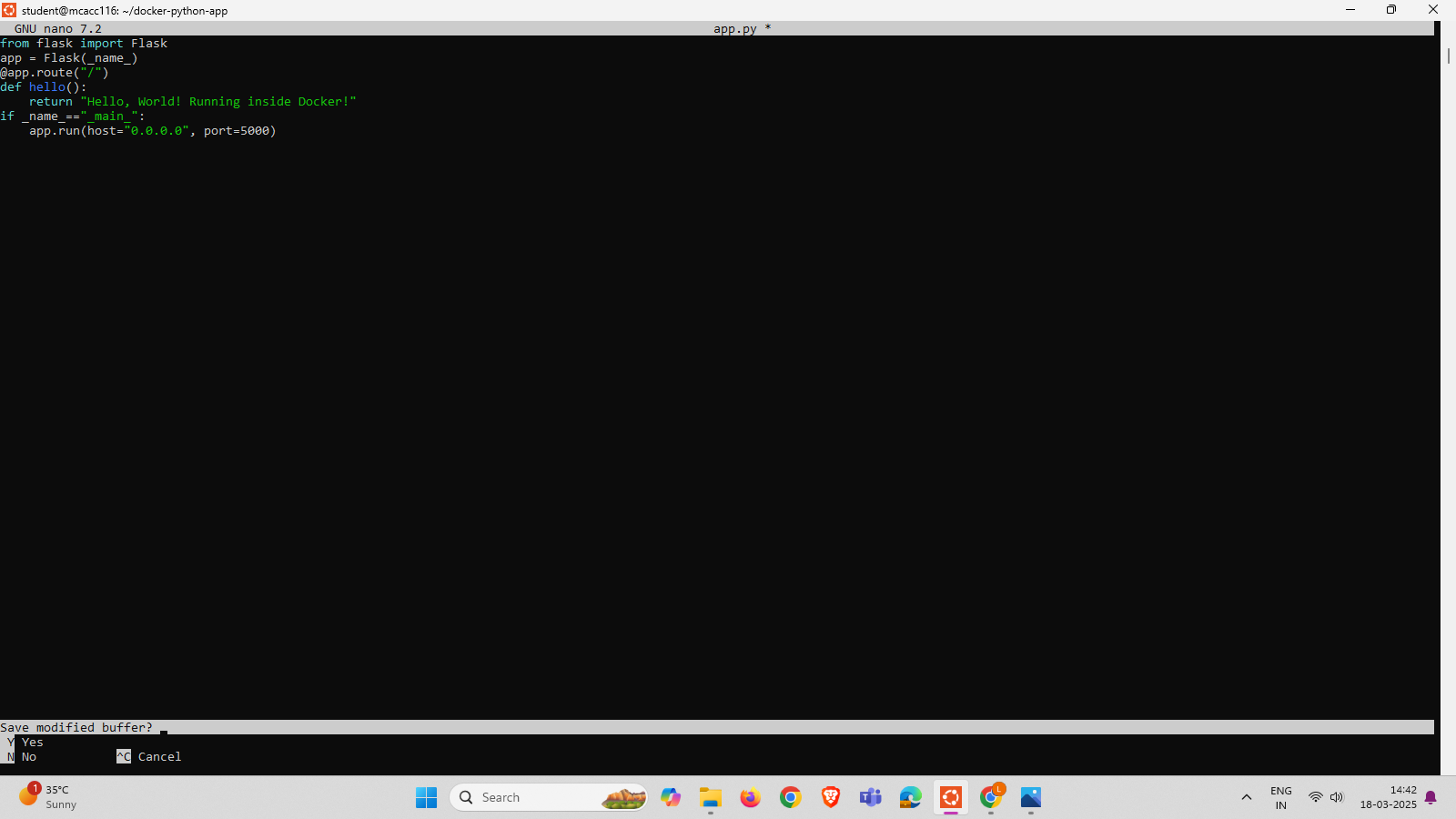
Create a project directory



Create the python Application File

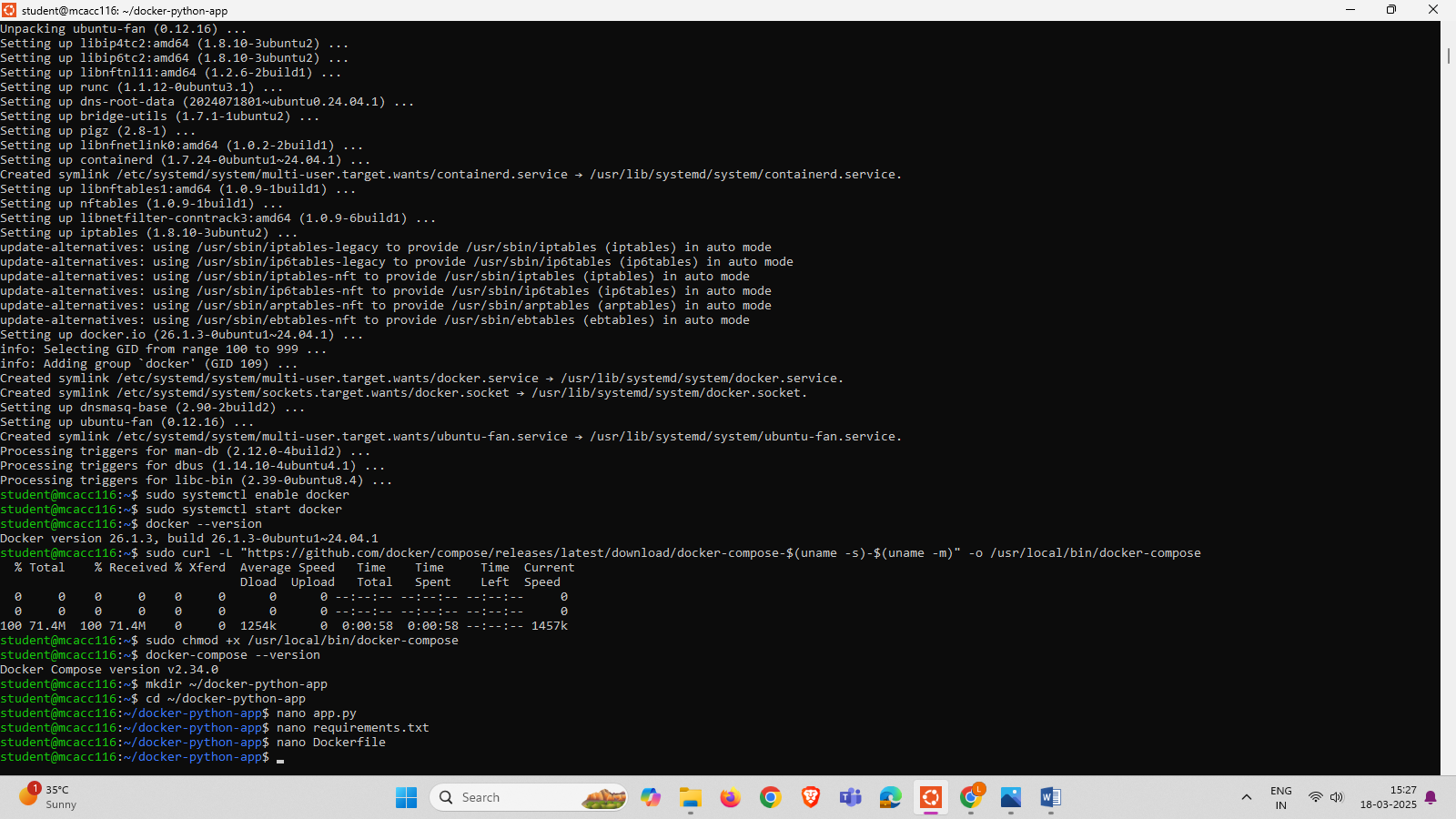
Create a file

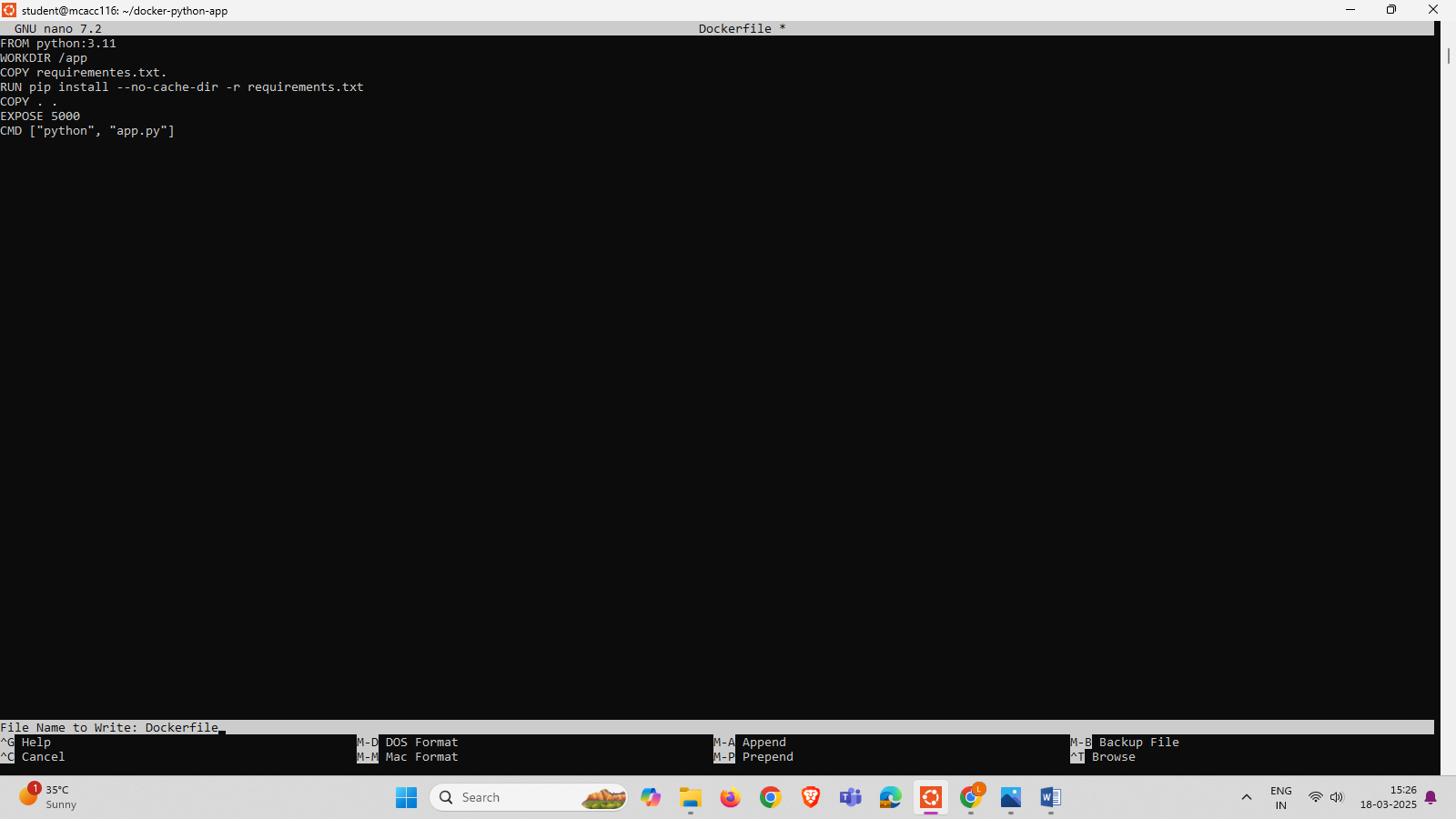


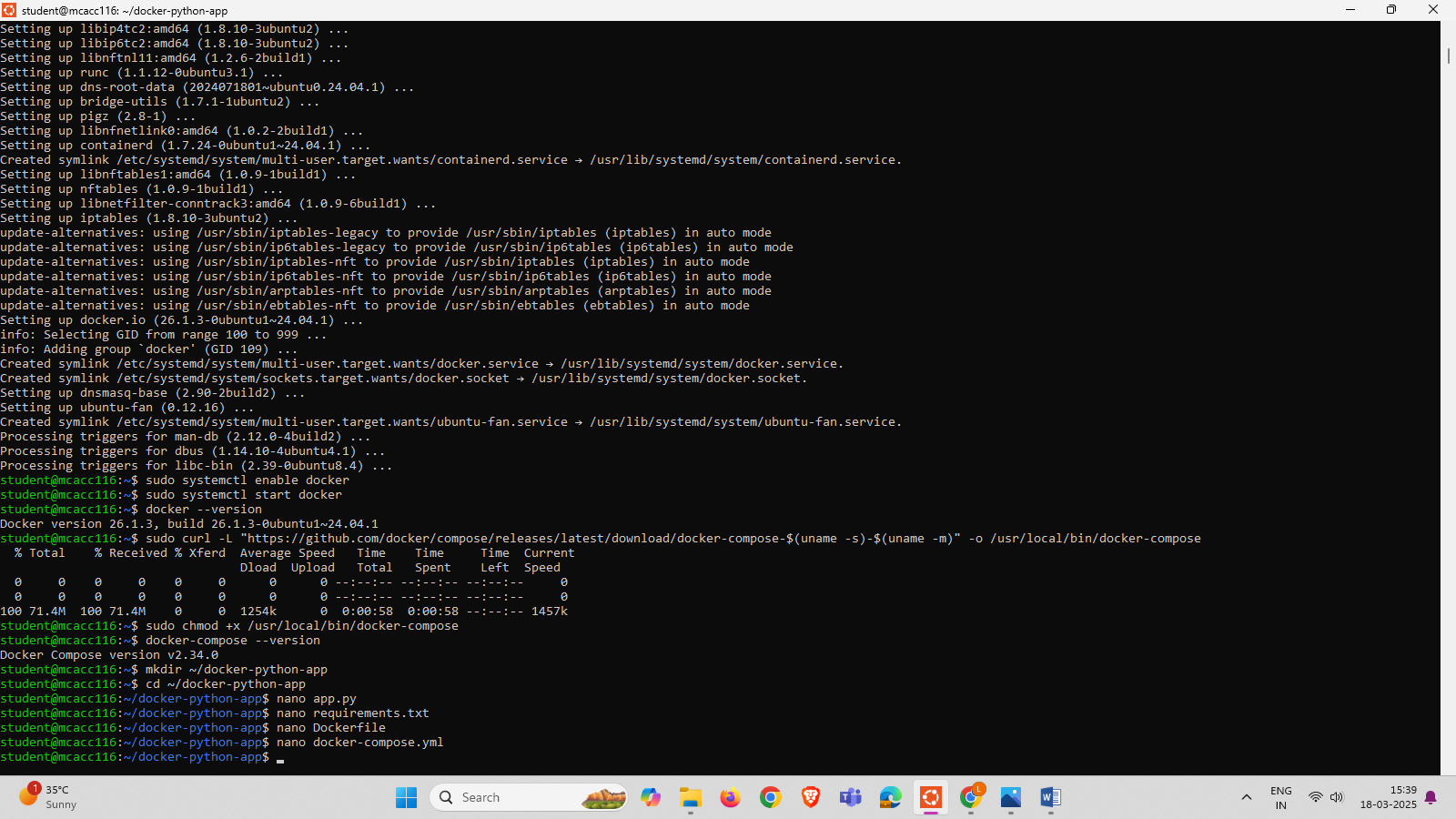


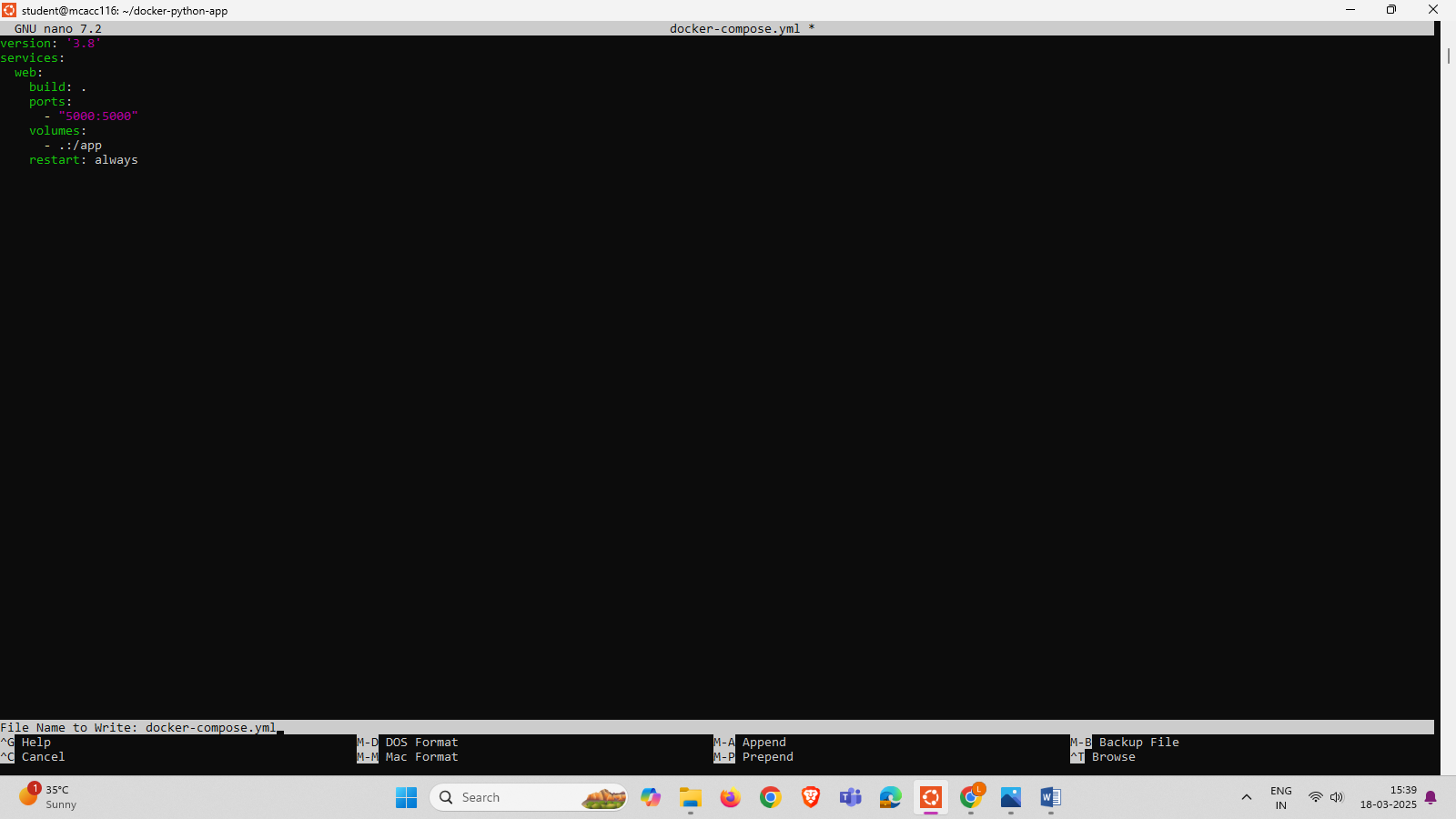
IN REQUIREMENTS.TXT TERMINAL WILL BE OPEN TYPE flask AND SAVE THE FILE BY CTRL+X,YES,ENTER.

**STEP -5 : CREATE A DOCKER FILE**

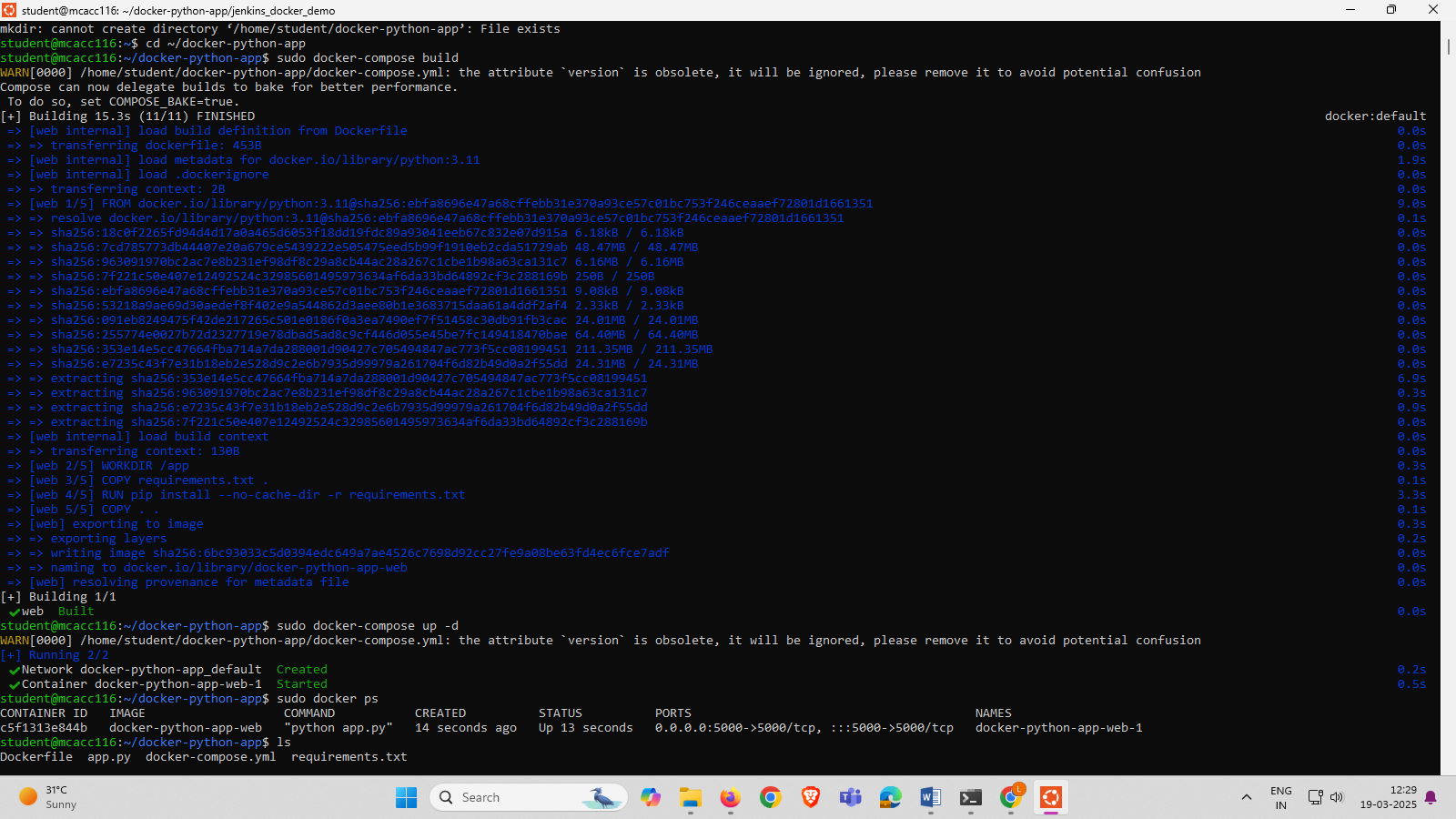


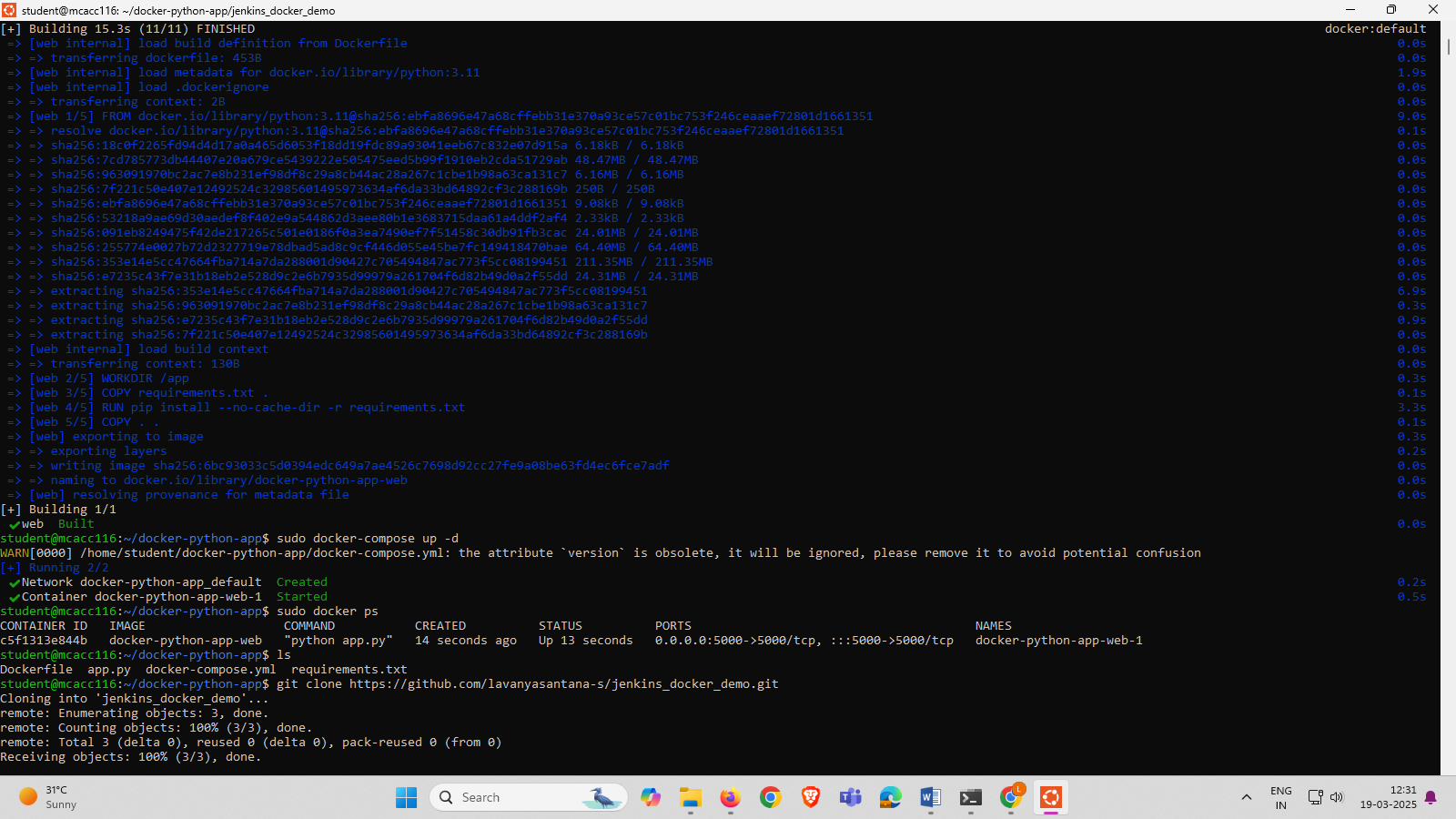


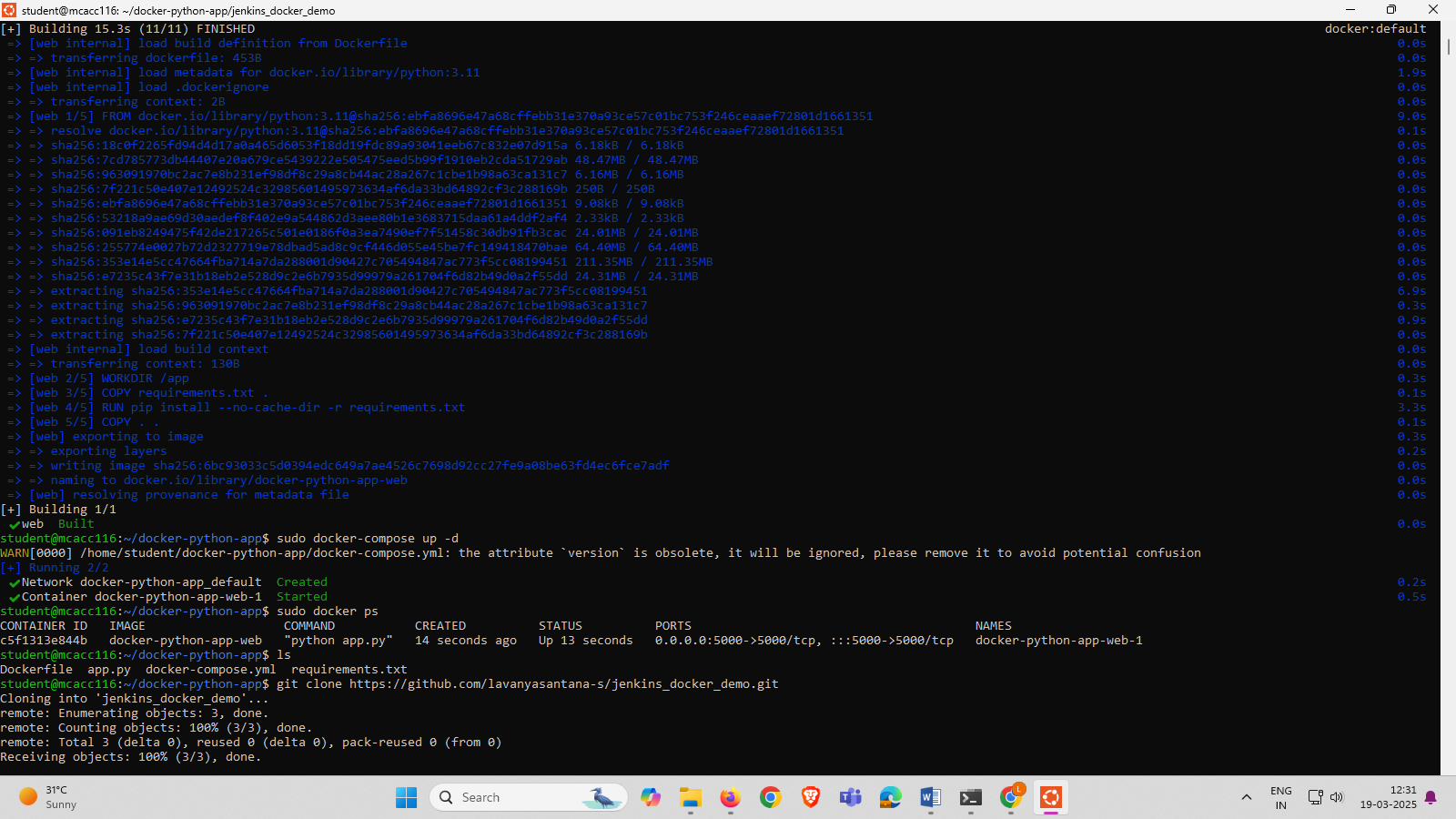
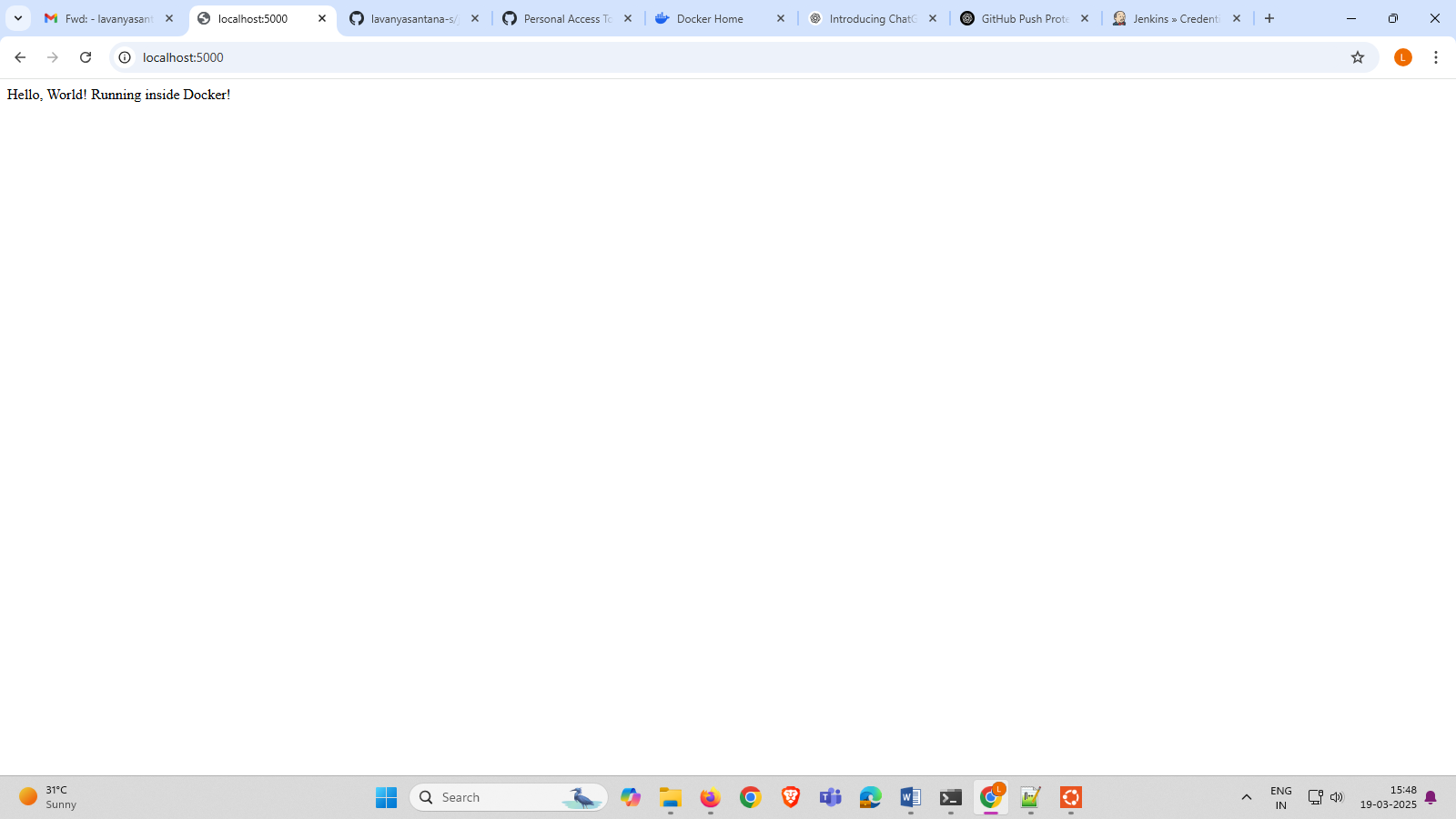
CREATE A DOCKER COMPOSE



STEP 6:BUILD AND RUN THE DOCKER CONTAINER





OPEN THE LOCALHOST:5000 IT WILL DISPLAY OUTPUT OF CODE 

STEP-7 CREATE A NEW REPO IN GITHUB

<https://github.com/lavanyasantana-s/jenkins_docker_demo/tree/main>

STEP-8 GO TO THE LINK

STEP-9 CLICK TOKEN CLASSIC AND GENERARTE TOKEN CLASSIC

STEP-10 IN GENERATE TOKEN CLASSIC GIVE THE NAME AND CLICK THE WORKFLOW AND ADMIN HOOK REPO

STEP-11 GENERATE TOKEN

STEP-12 START THE JENKINS

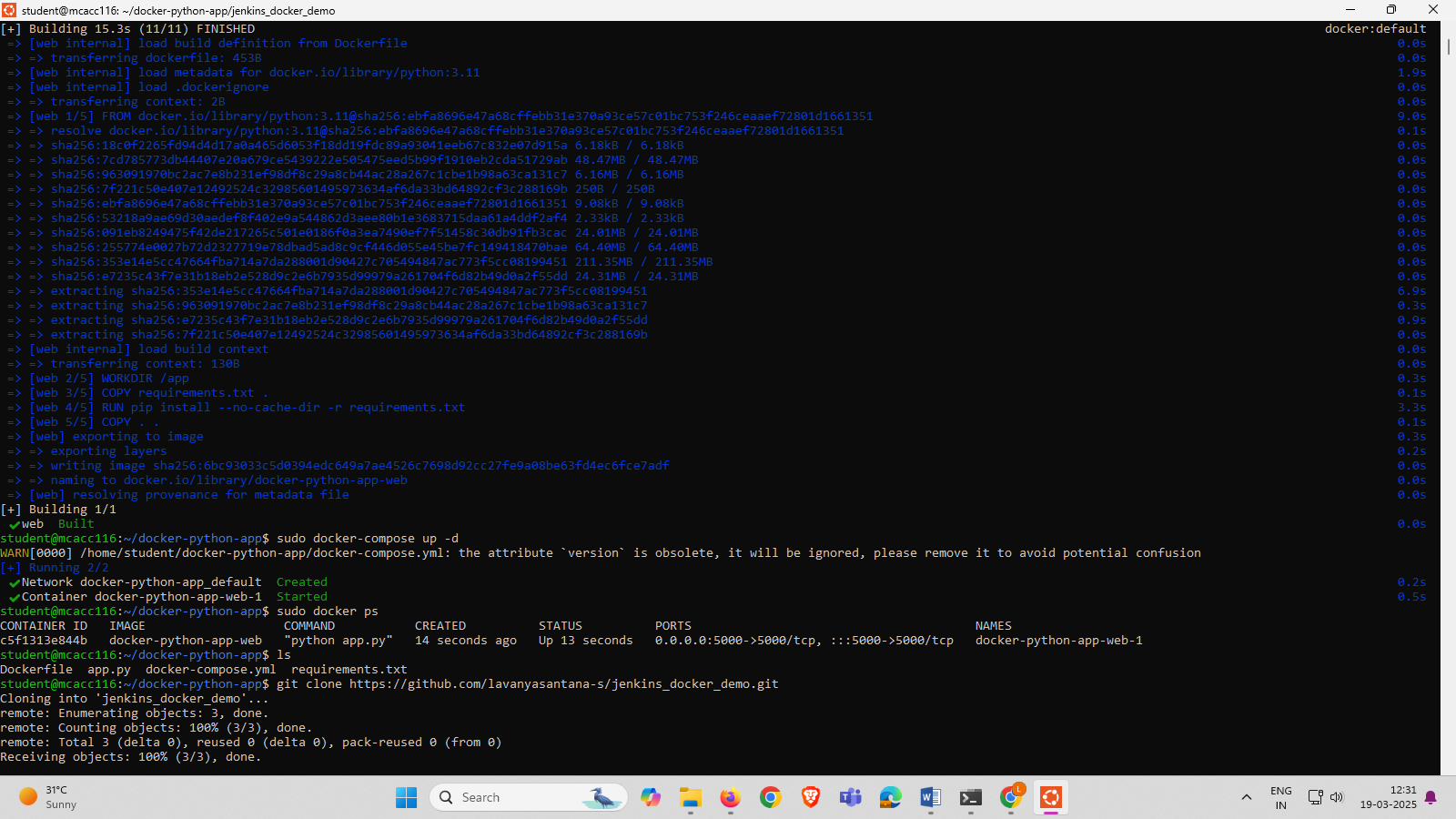
STEP-13 Create a new ITEM AND PIPELINE THEN CLICK THE PIPLELINE THEN PILELINE SCM THEN GIT

STEP-14 PASTE THE GITHUB LINK AND IN CREDITIONALS ADD OPTIONS THEN JENKIN.

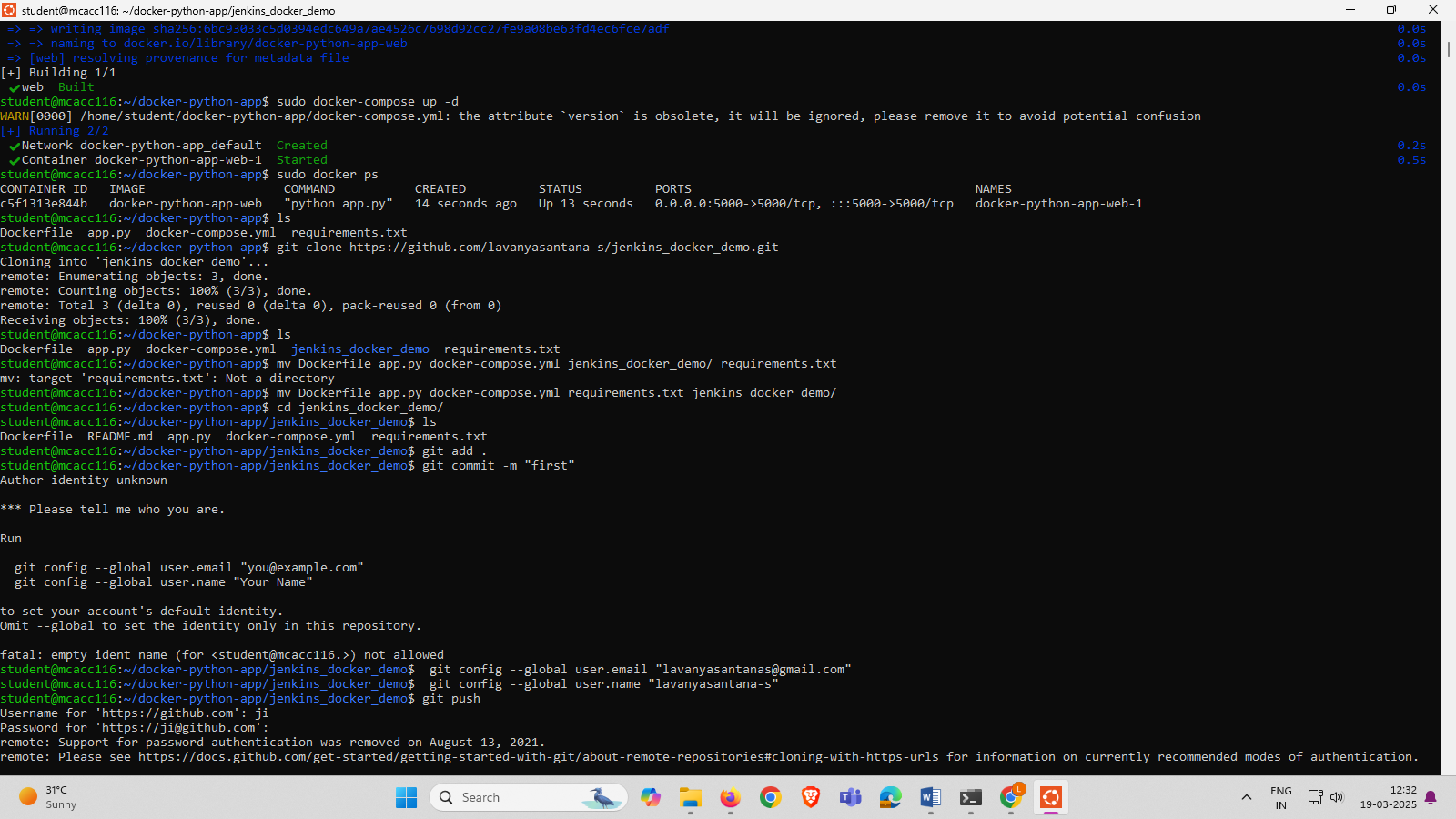
STEP-15 PROVIDE GITHUB USERNAME AND GENERATED TOKEN PASSWORD IN PASSWORD THEN GIVE THE ID AS YOUR PREFERENCE AFTER COMPLETING CLICK ADD

STEP-16 TO PUSH INTO GITHUB

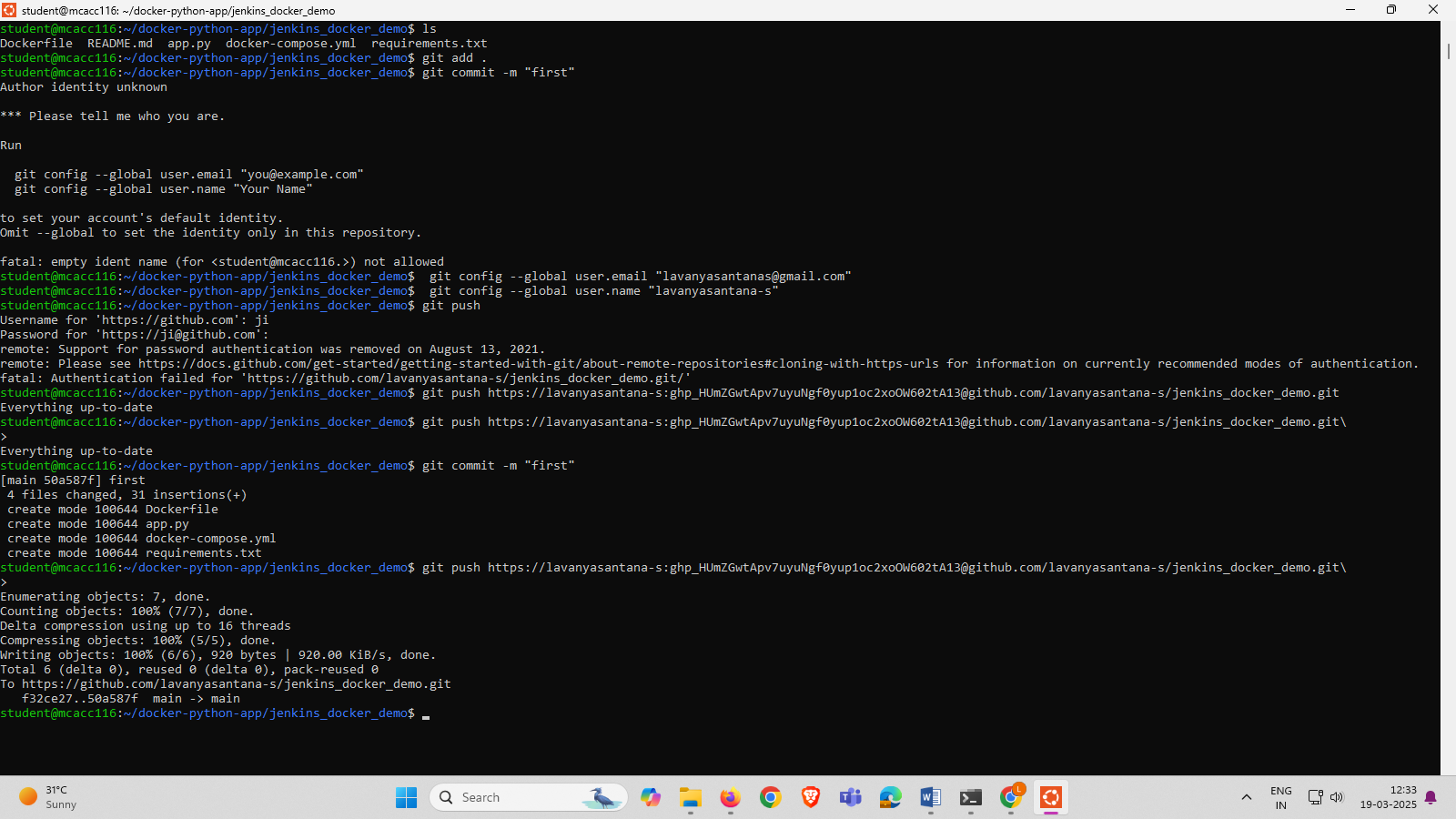
Clone

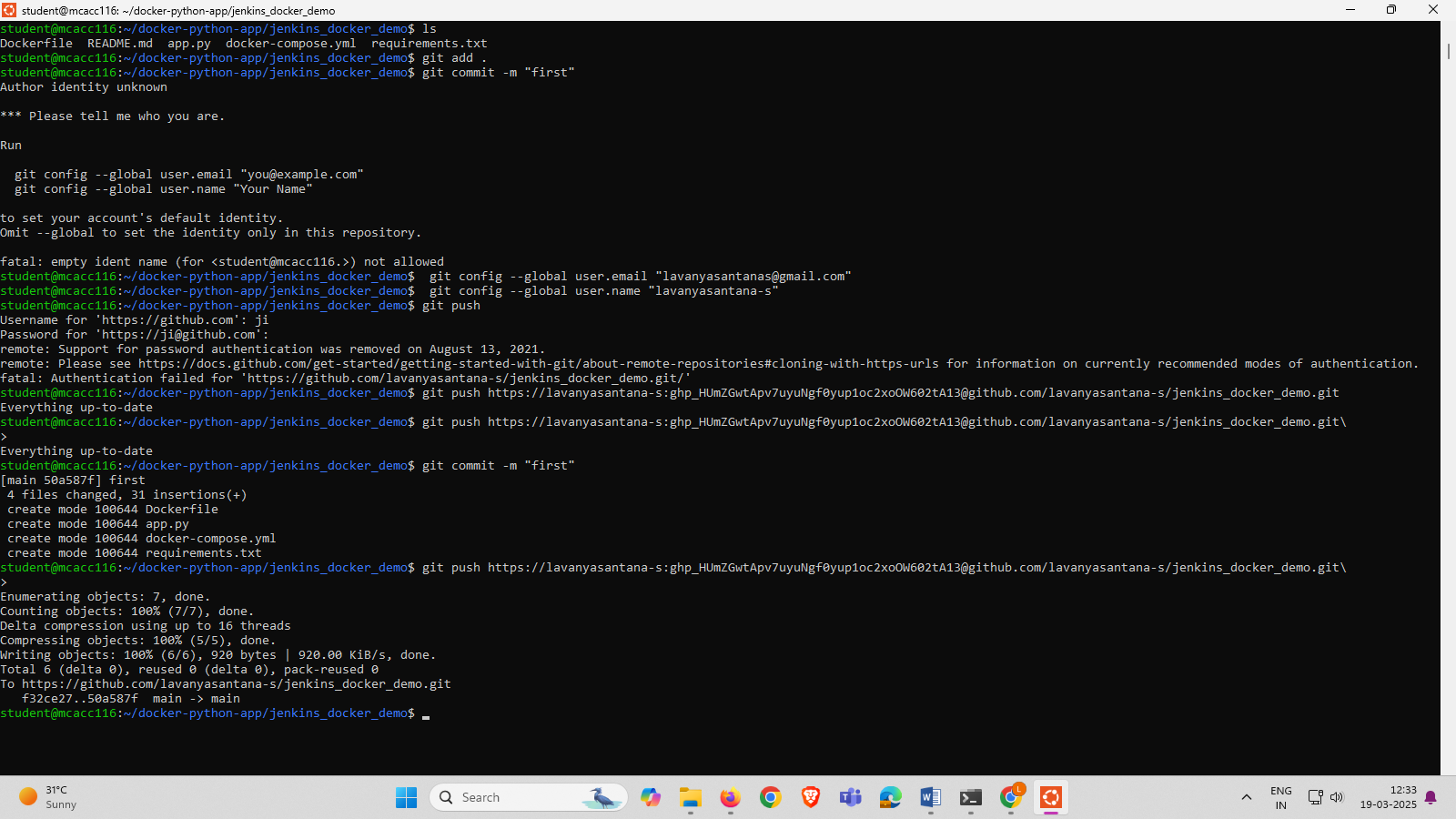


check file

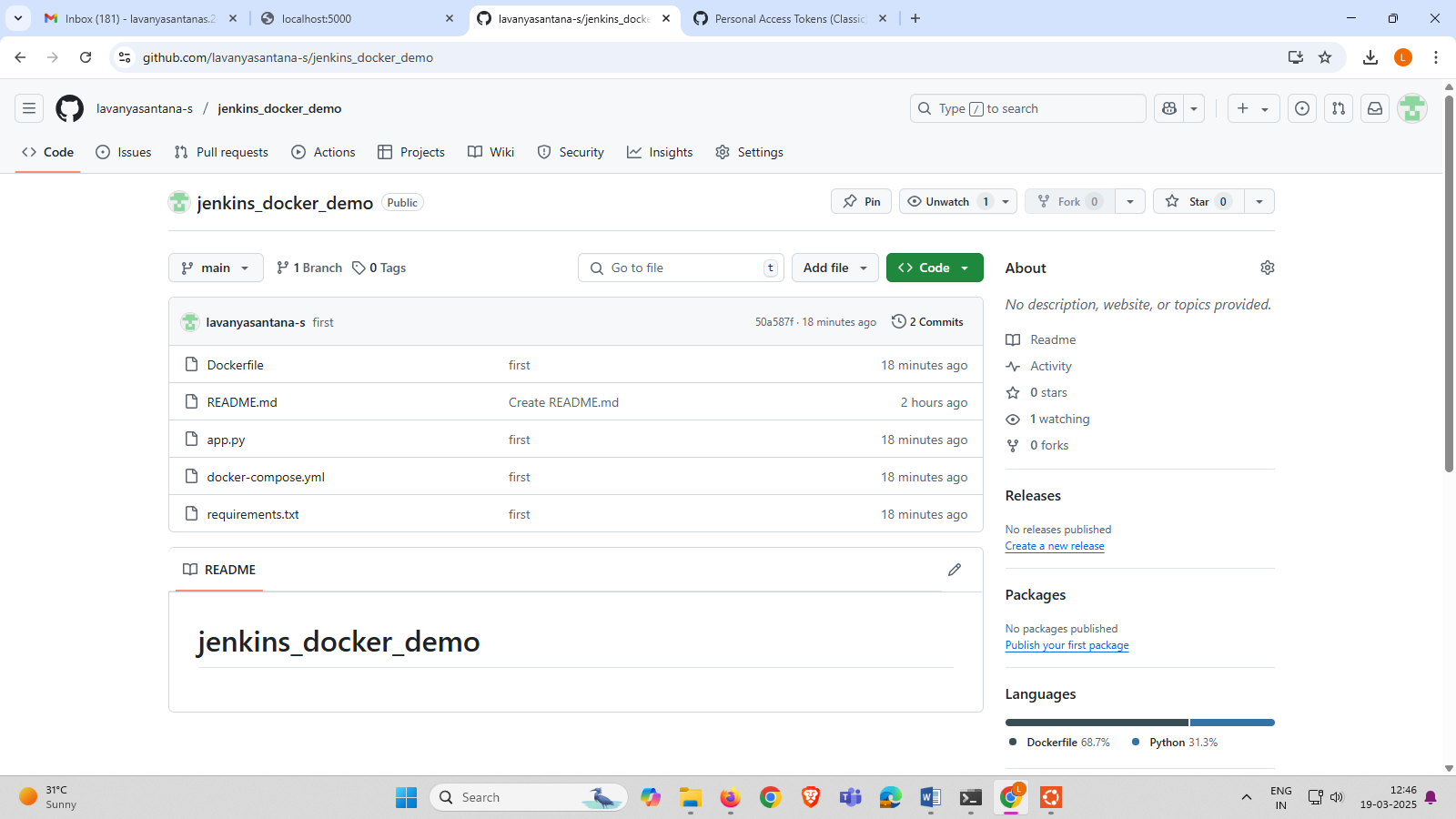


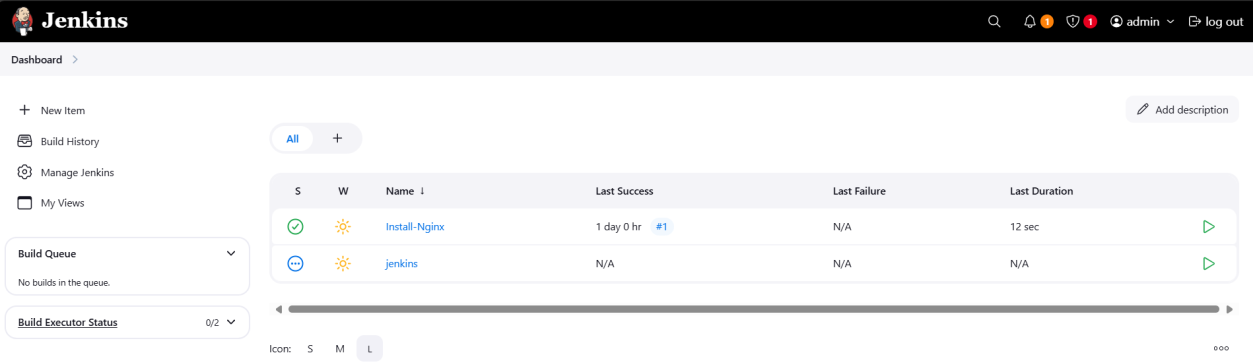
Add to the repository



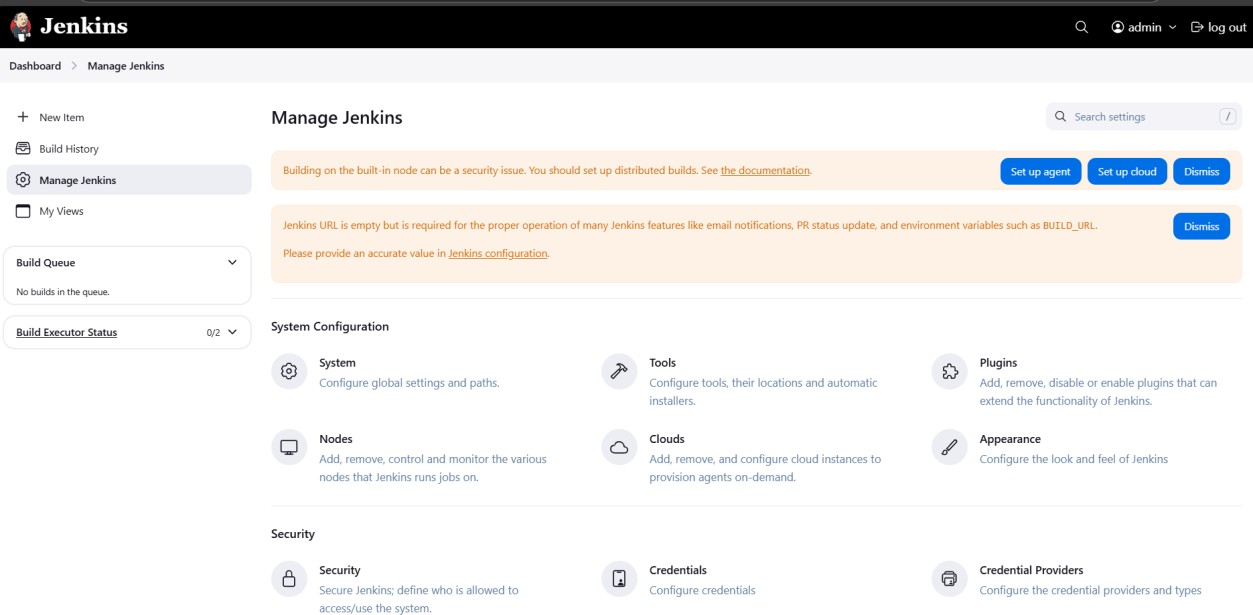


STEP-17 Open docker image app.docker.com

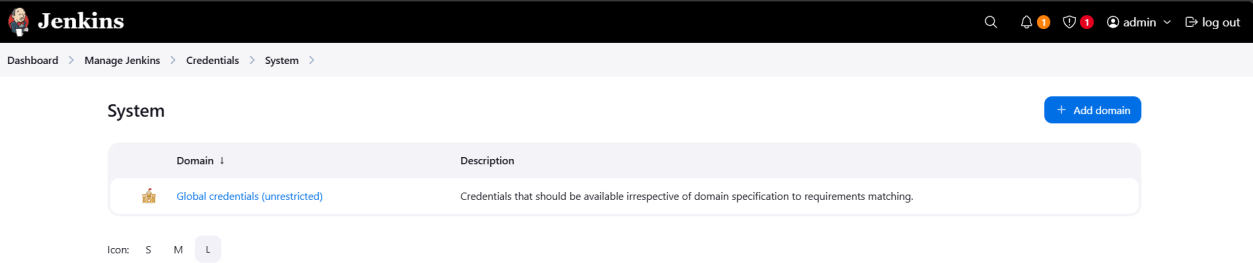




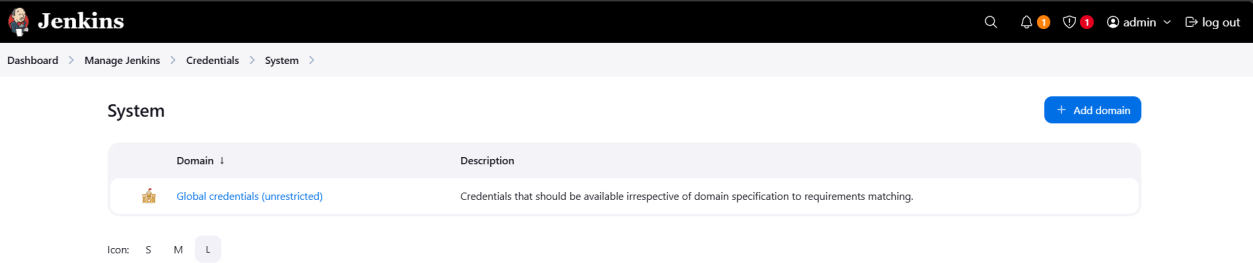
STEP-18 OPEN MANAGE JENKINS



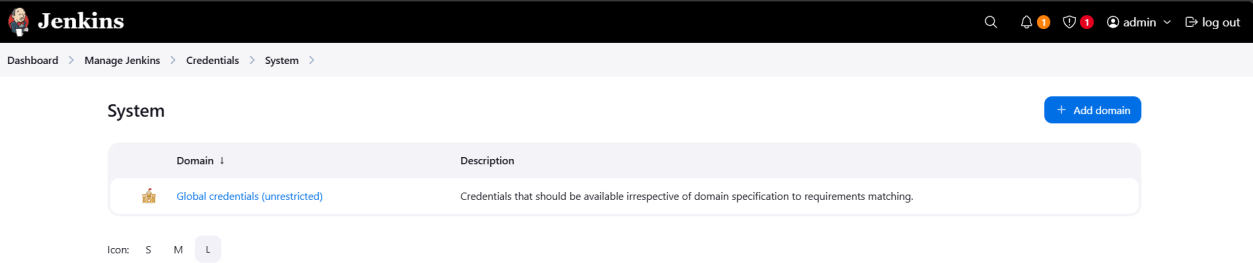
STEP-19 OPEN CREDENTIALS



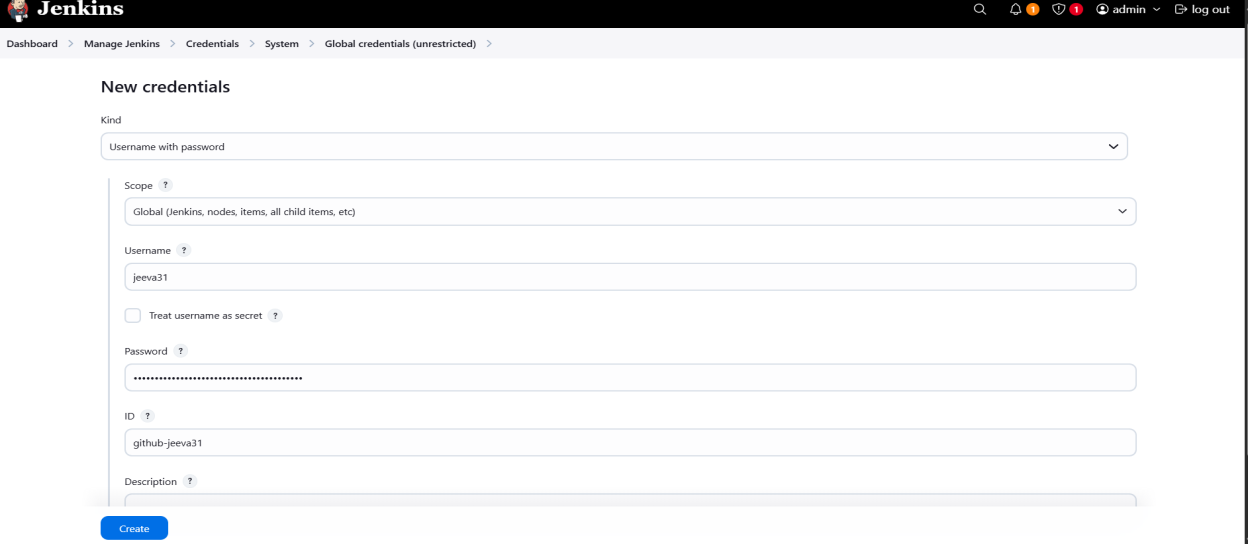
STEP-20 OPEN SYSTEM



STEP-21 OPEN MANAGE JENKINS

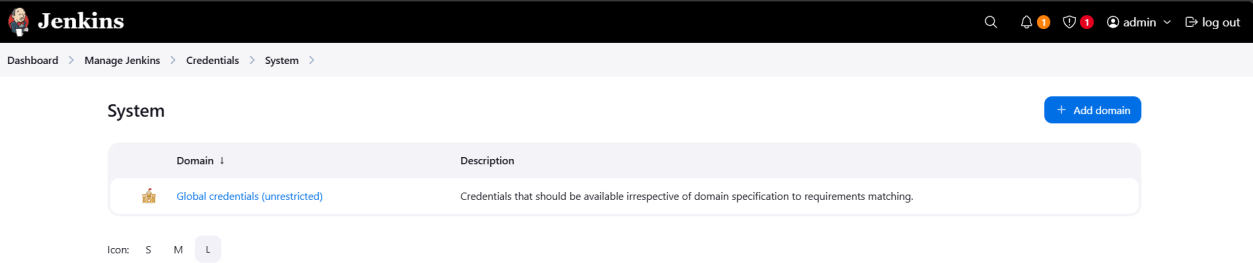


STEP-22 OPEN GLOBAL CREDENTIALS



STEP-23 OPEN MANAGE JENKINS

=>IT DISPLAYS THE GITHUB CREDENTIALS ID AND DOCKER CREDENTIALS ID



STEP 24:OPEN UBUNTU AND CREATE nano Jenkinsfile

STEP 25:

if [ "$(docker ps -aq -f name=$CONTAINER\_NAME)" ]; then docker stop $CONTAINER\_NAME || true

docker rm $CONTAINER\_NAME || true

fi '''

}

}

}

stage('Run Docker Container') { steps {

sh 'docker run -d -p 5001:5000 --name $CONTAINER\_NAME

$DOCKER\_IMAGE'

}

}

}

post {

success {

echo "Build, push, and container execution successful!"

}

failure {

echo "Build or container execution failed."

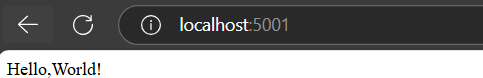
}

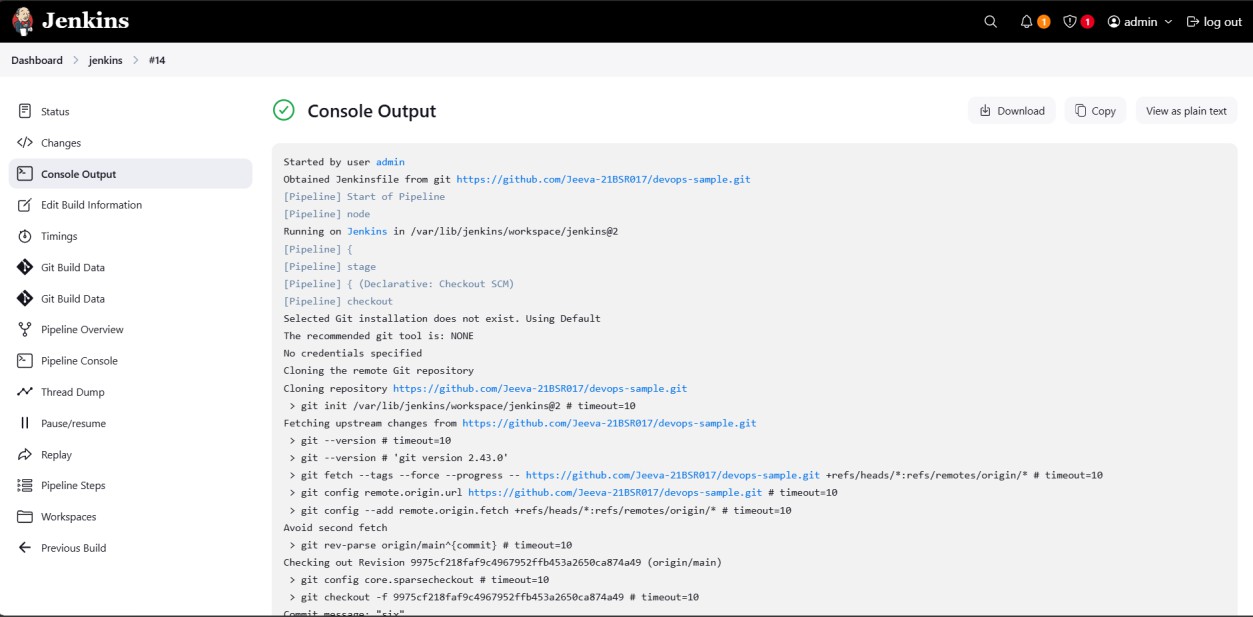
}

}

STEP 26:IN JENKINS BUILD NOW THE Jenkins

STEP 27:IN LOCALHOST:5001 IT DISPLAYS THE OUTPUT.





STEP-28 : IN DOCKERHUB THE PROCESS ARE TO DONE AND THEN THE LINUX IMAGE HAS TO BE DISPLAY.

