1. Jenkins CI CD pipeline for flask application

Objective:

Set up a Jenkins pipeline that automates the testing and deployment of a simple Python web application.

Requirements:

1. Setup:

- Install Jenkins on a virtual machine or use a cloud-based Jenkins service.

- Configure Jenkins with Python and any necessary libraries.

2. Source Code:

- Fork the provided Python web application repository on GitHub (provide a link to a sample Python web application repository).

- Clone the forked repository into your Jenkins server.

3. Jenkins Pipeline:

- Create a Jenkinsfile in the root of your Python application repository.

- Define a pipeline with the following stages:

- Build: Install dependencies using pip.

- Test: Run unit tests using a testing framework like pytest.

- Deploy: If tests pass, deploy the application to a staging environment.

4. Triggers:

- Configure the pipeline to trigger a new build whenever changes are pushed to the main branch of the repository.

5. Notifications:

- Set up a notification system to alert via email when the build process fails or succeeds.

6. Documentation:

- Document the pipeline process and any prerequisites needed for the setup in a README.md file in the repository.

7. Submission:

- Provide the URL to the GitHub repository with the Jenkinsfile and updated README.md.

- Include screenshots of the Jenkins pipeline showing the build, test, and deployment stages.

Deliverables:

- Forked GitHub repository with Jenkinsfile.

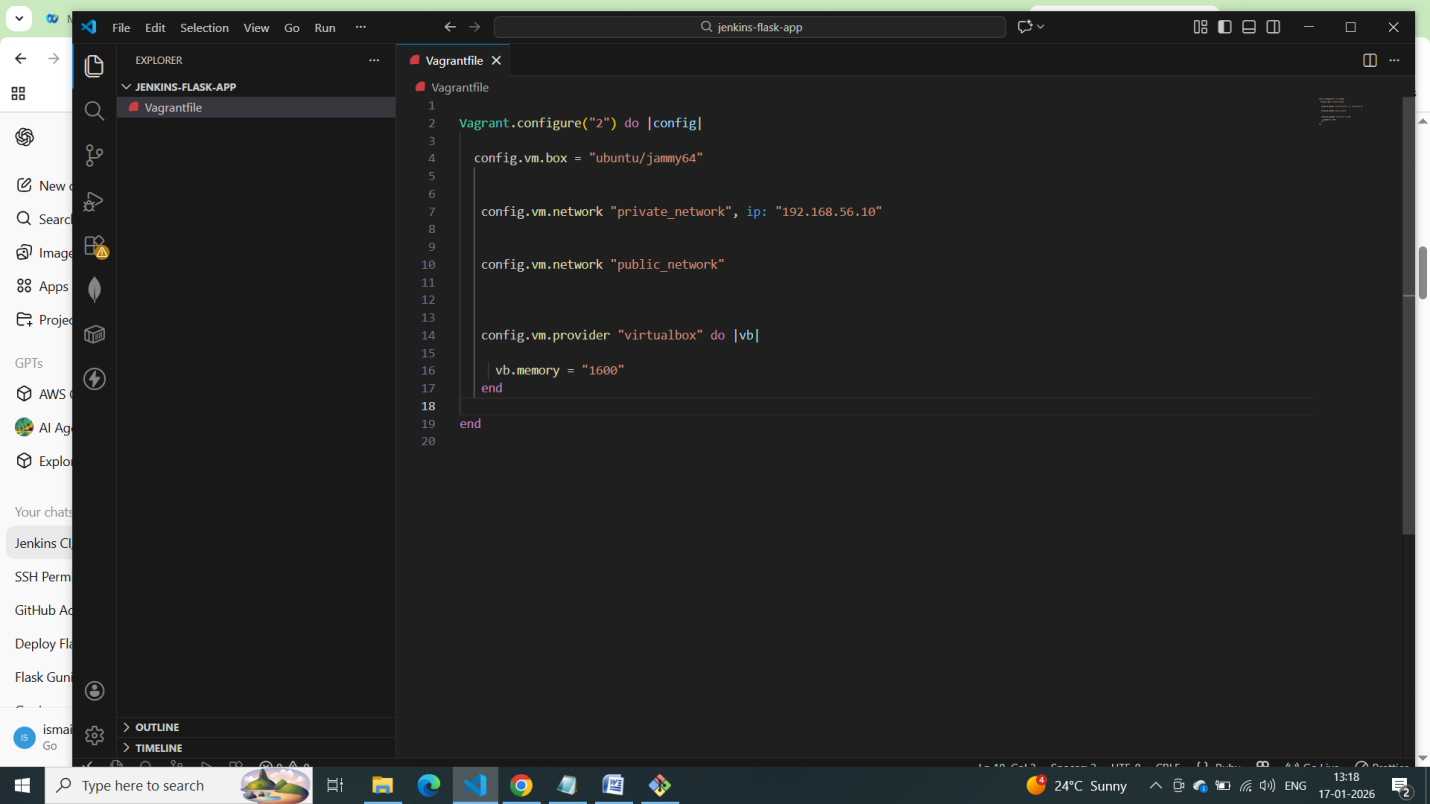
- Documentation in README.md.

- Screenshots of the Jenkins pipeline execution.

Project solution:

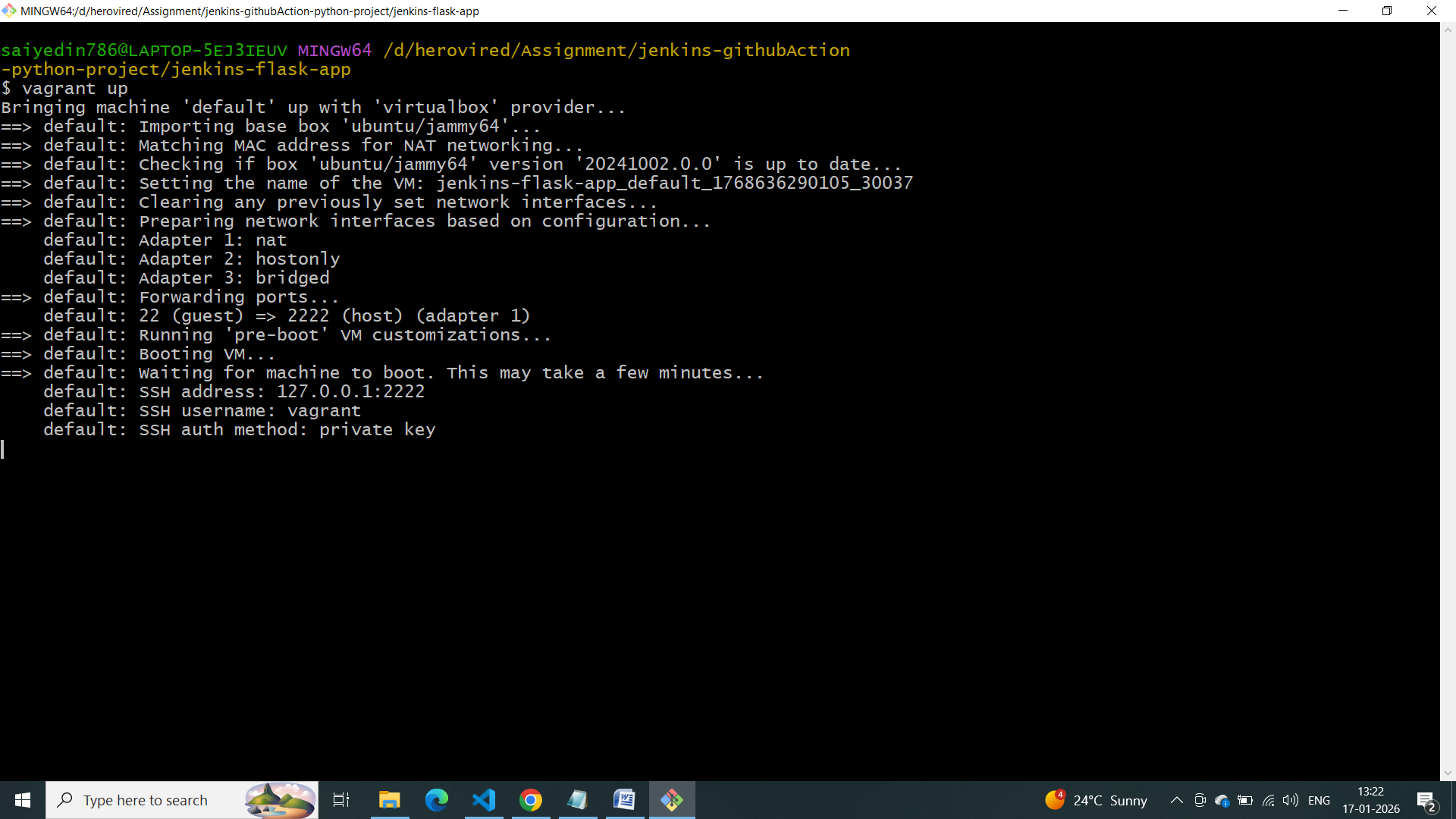
Setting up local vm through vagrant

Downloaded vagrant file from vagrant cloud and than setting up locally ubuntu linux



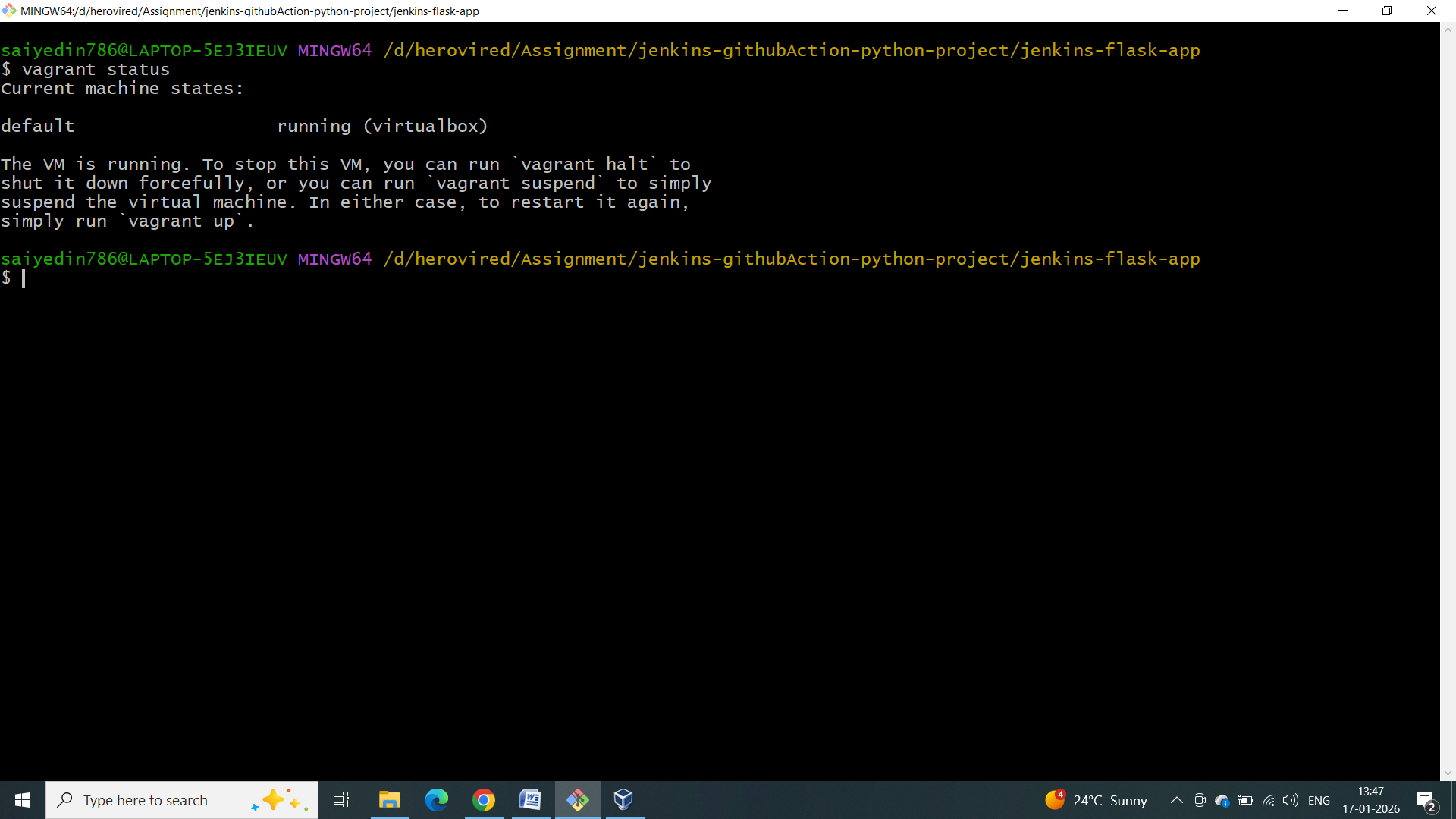
Starting ubuntu vm:

Vagrant up



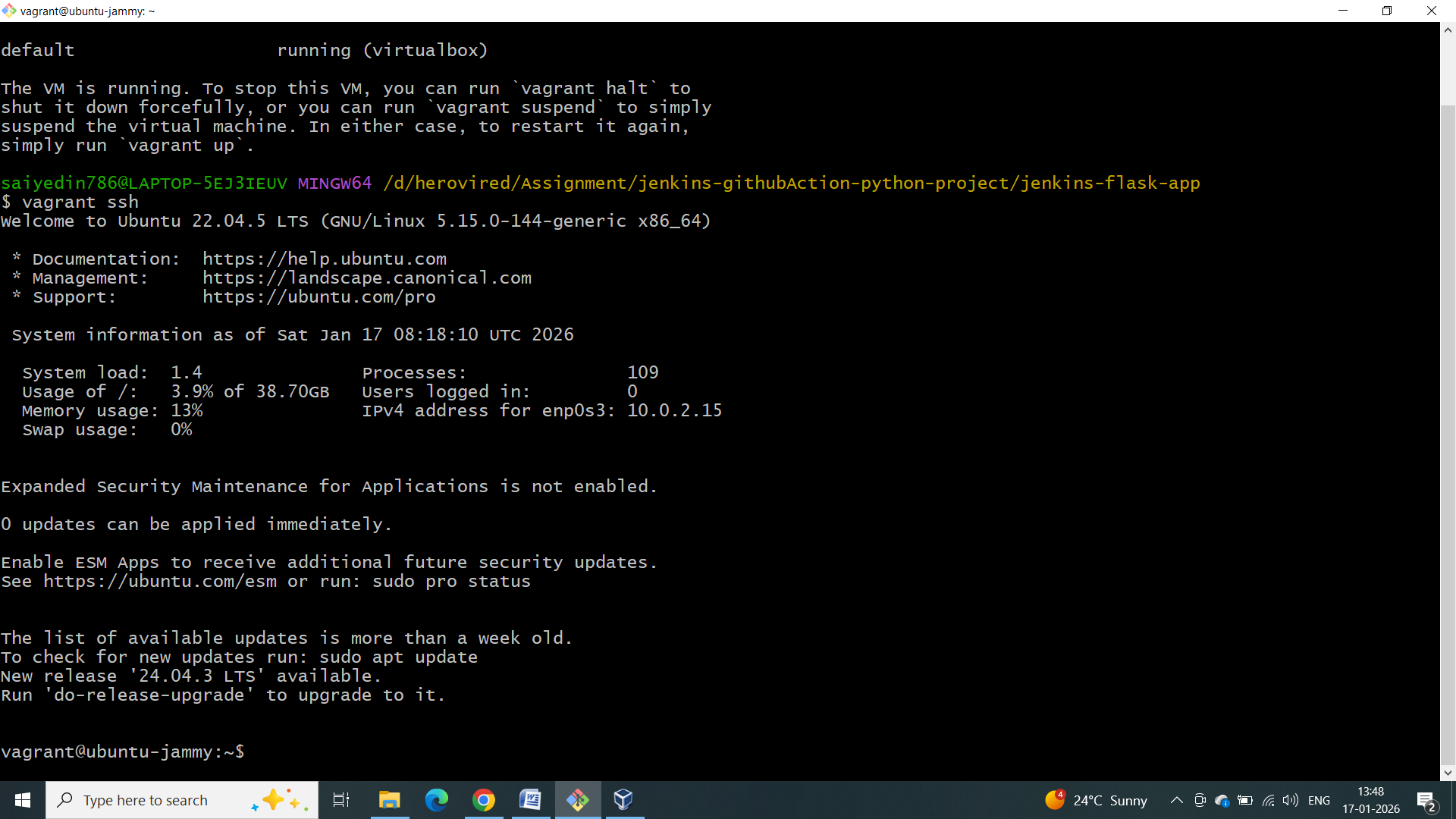
Checking status of ubuntu vm:

Vagrant status



Ssh into the vm:

Vagrant ssh



Installing Jenkins into vm:

sudo wget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo dnf upgrade

*# Add required dependencies for the Jenkins package*

sudo dnf install fontconfig java-21-openjdk

sudo dnf install jenkins

sudo systemctl daemon-reload

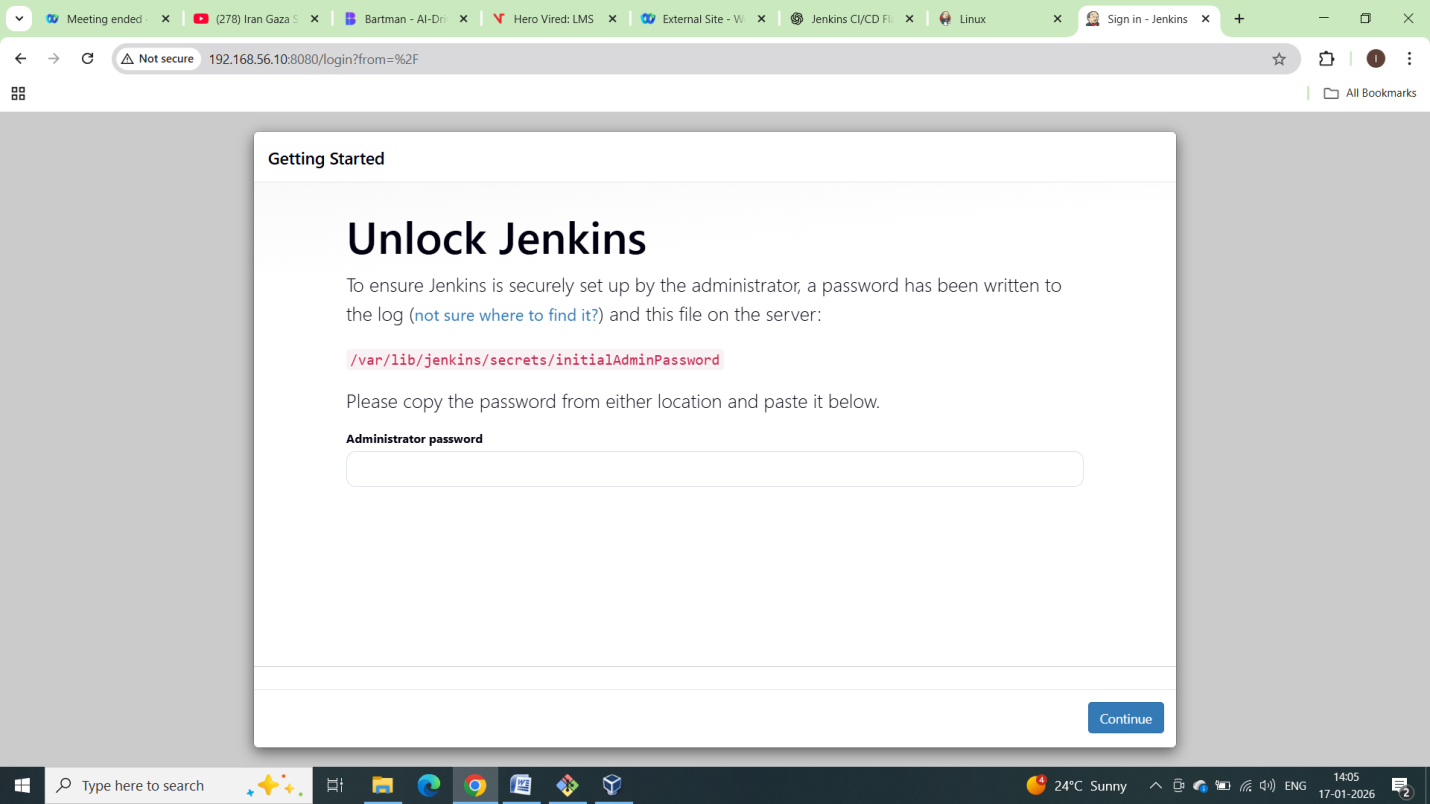
sudo apt update

sudo apt install -y jenkins

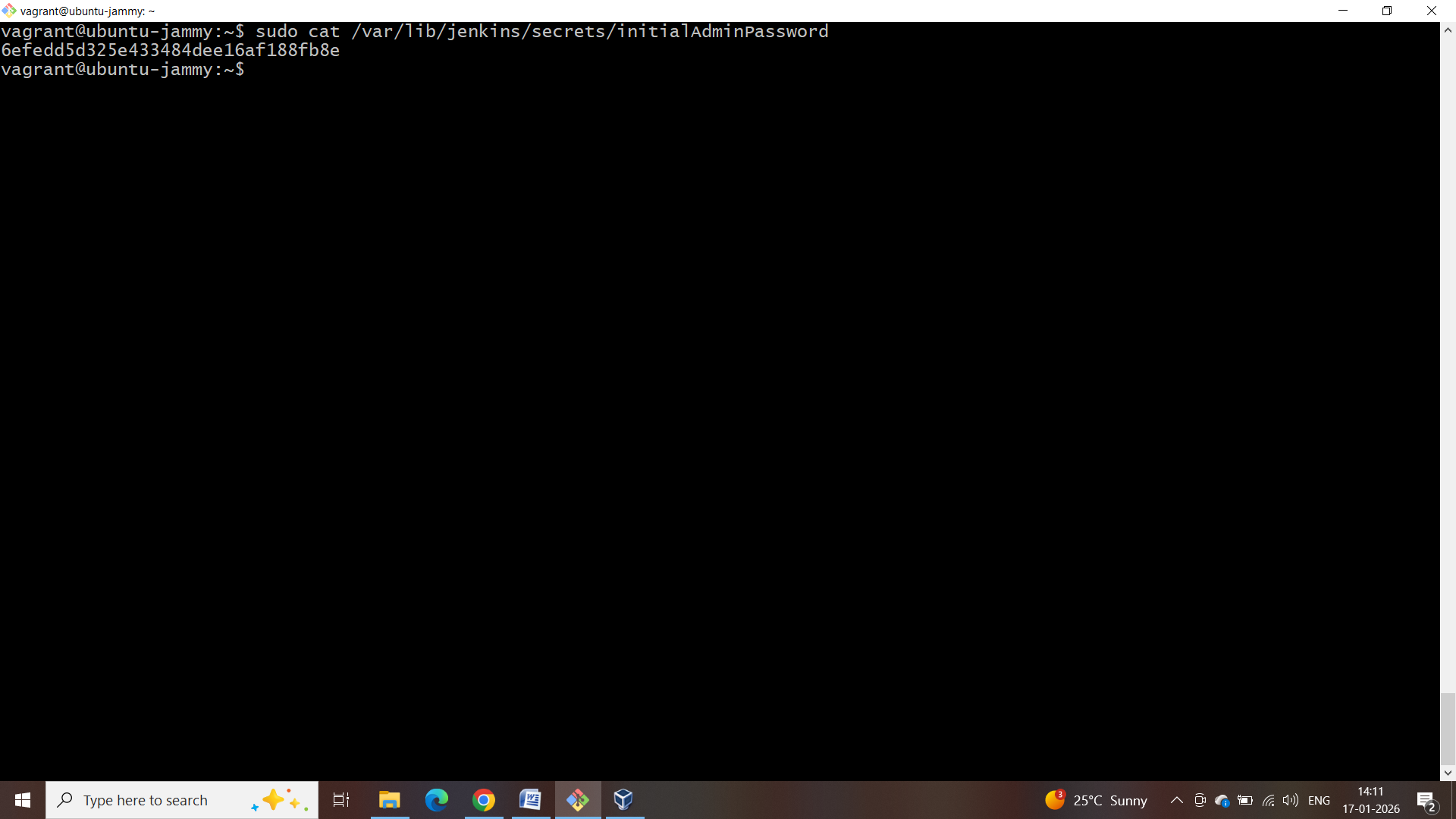
sudo systemctl start jenkins

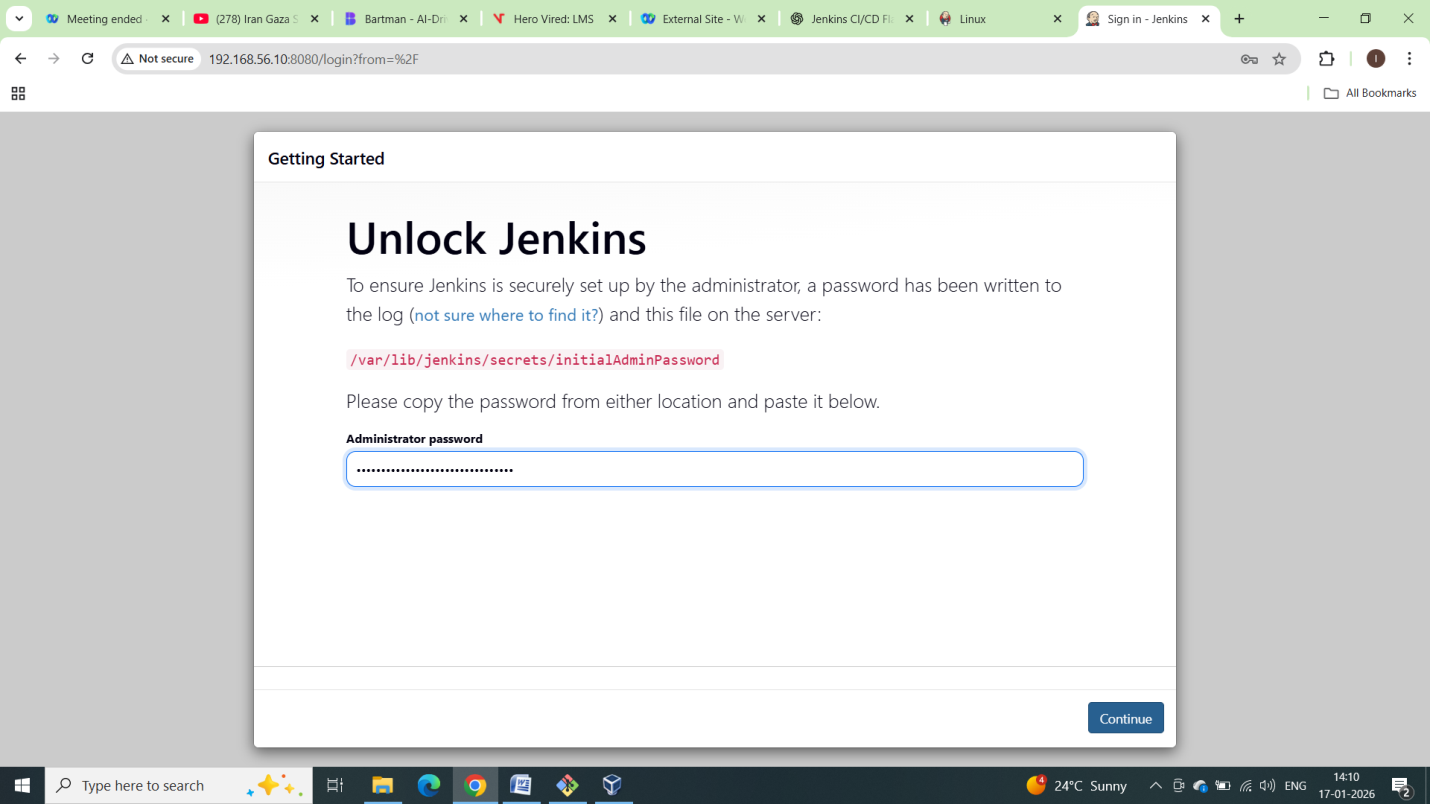
sudo systemctl enable Jenkins

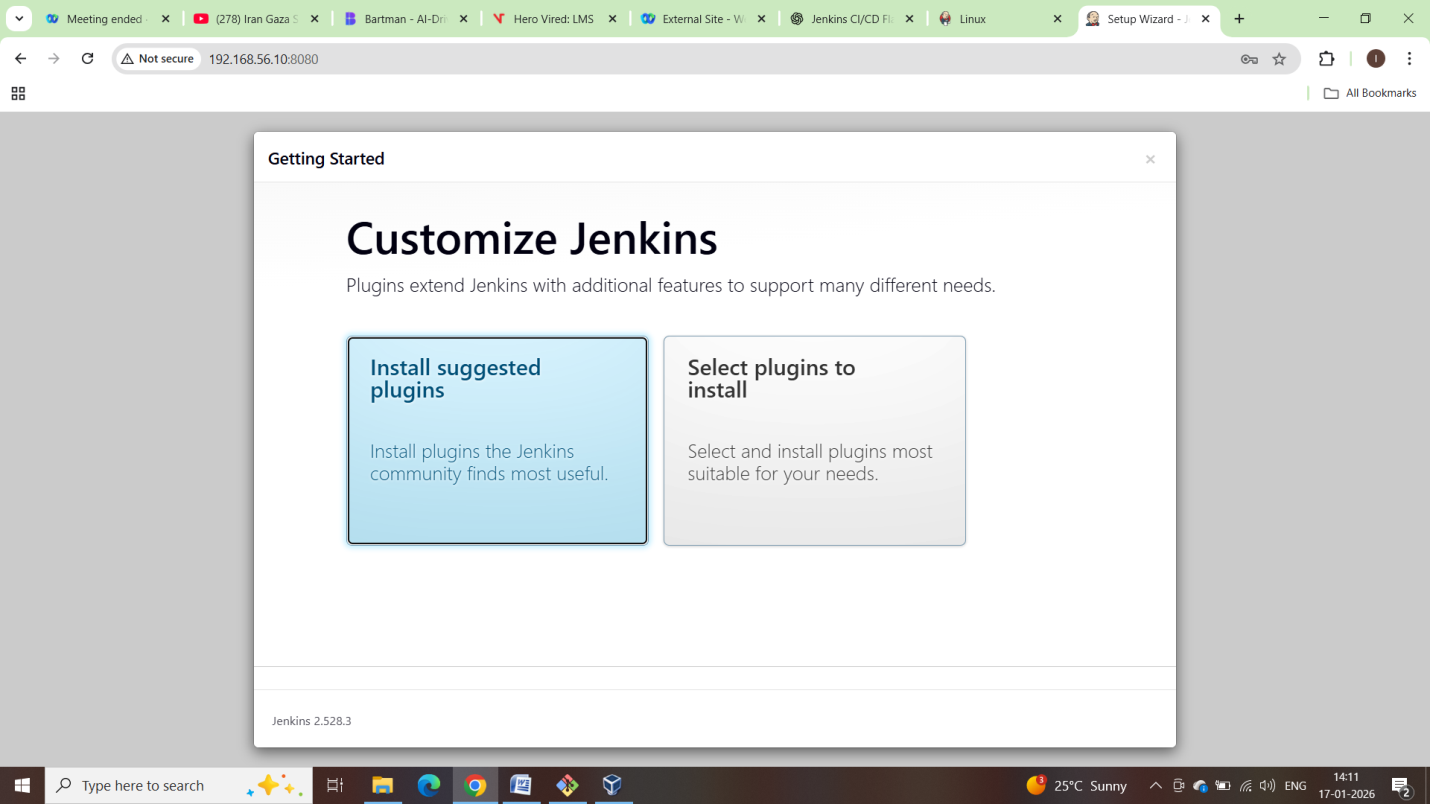
accessing Jenkins server in browser

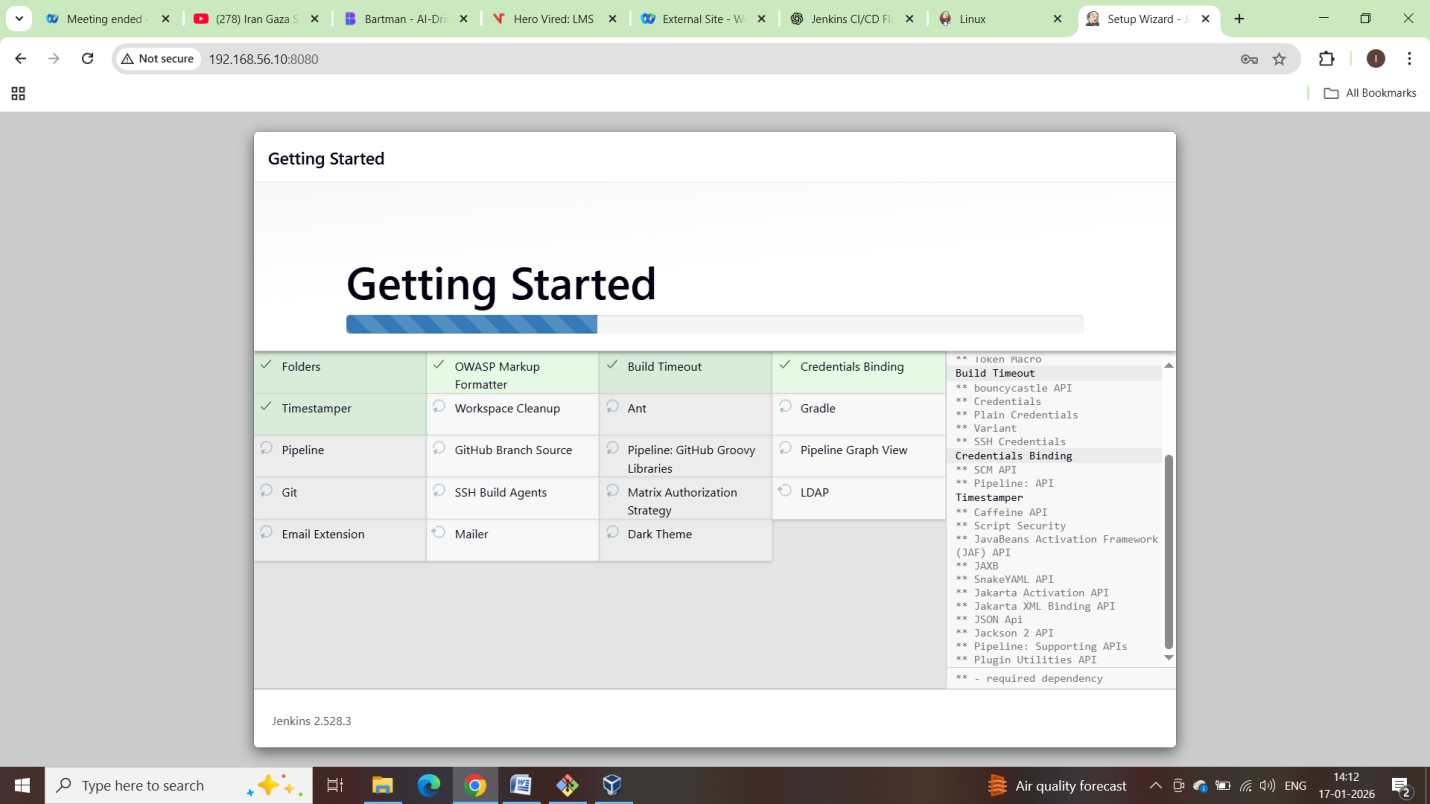


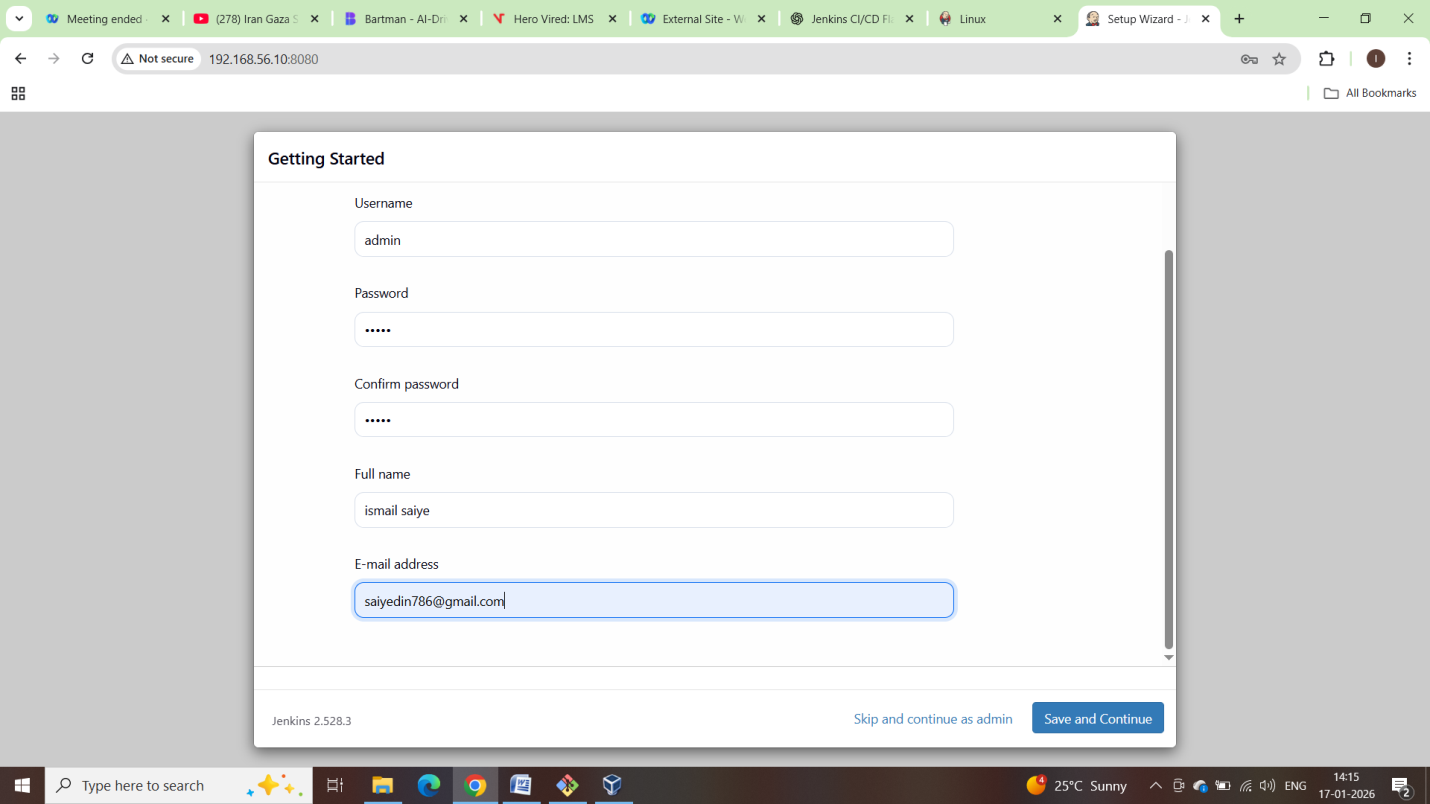
Setting password:

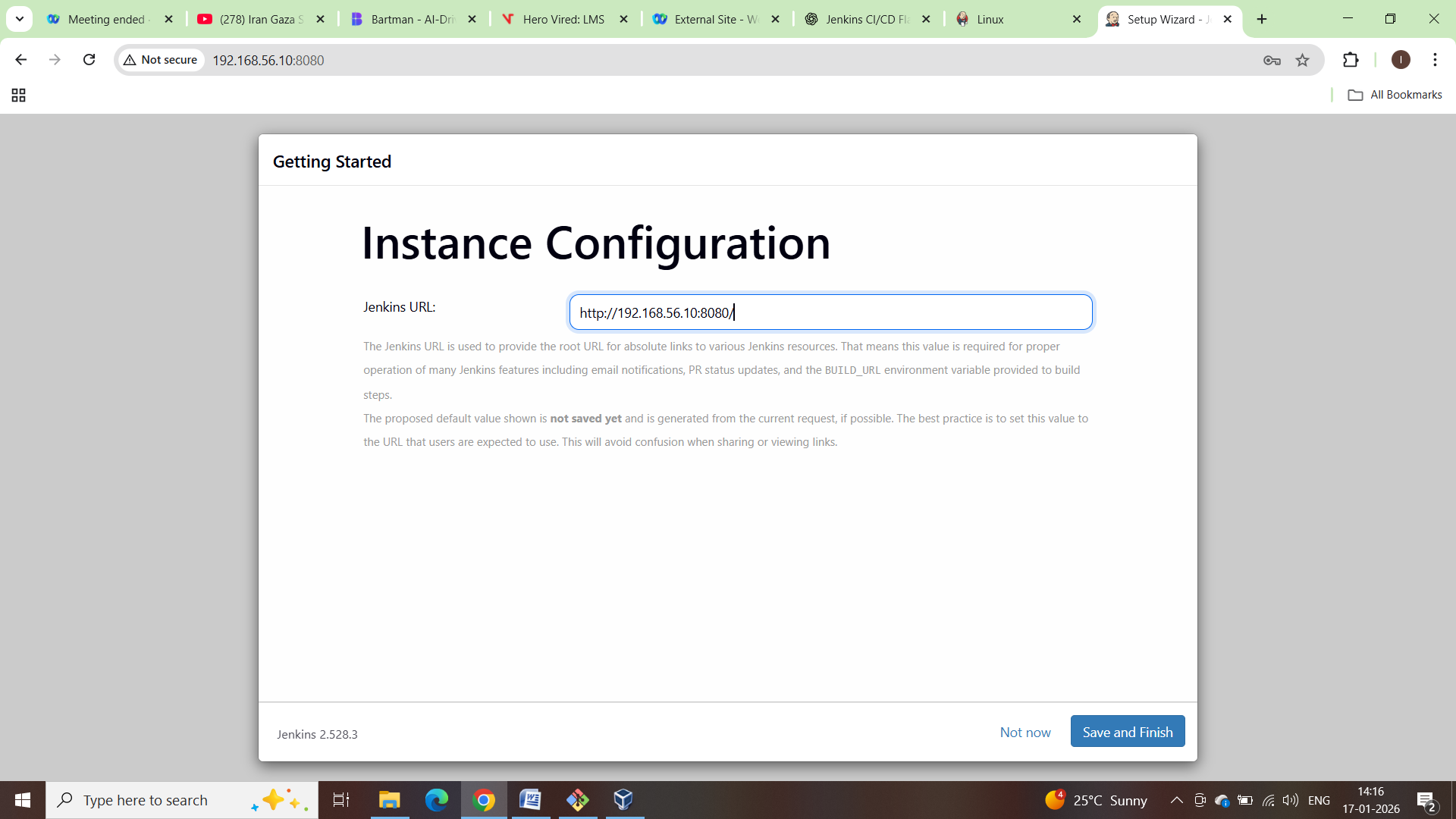


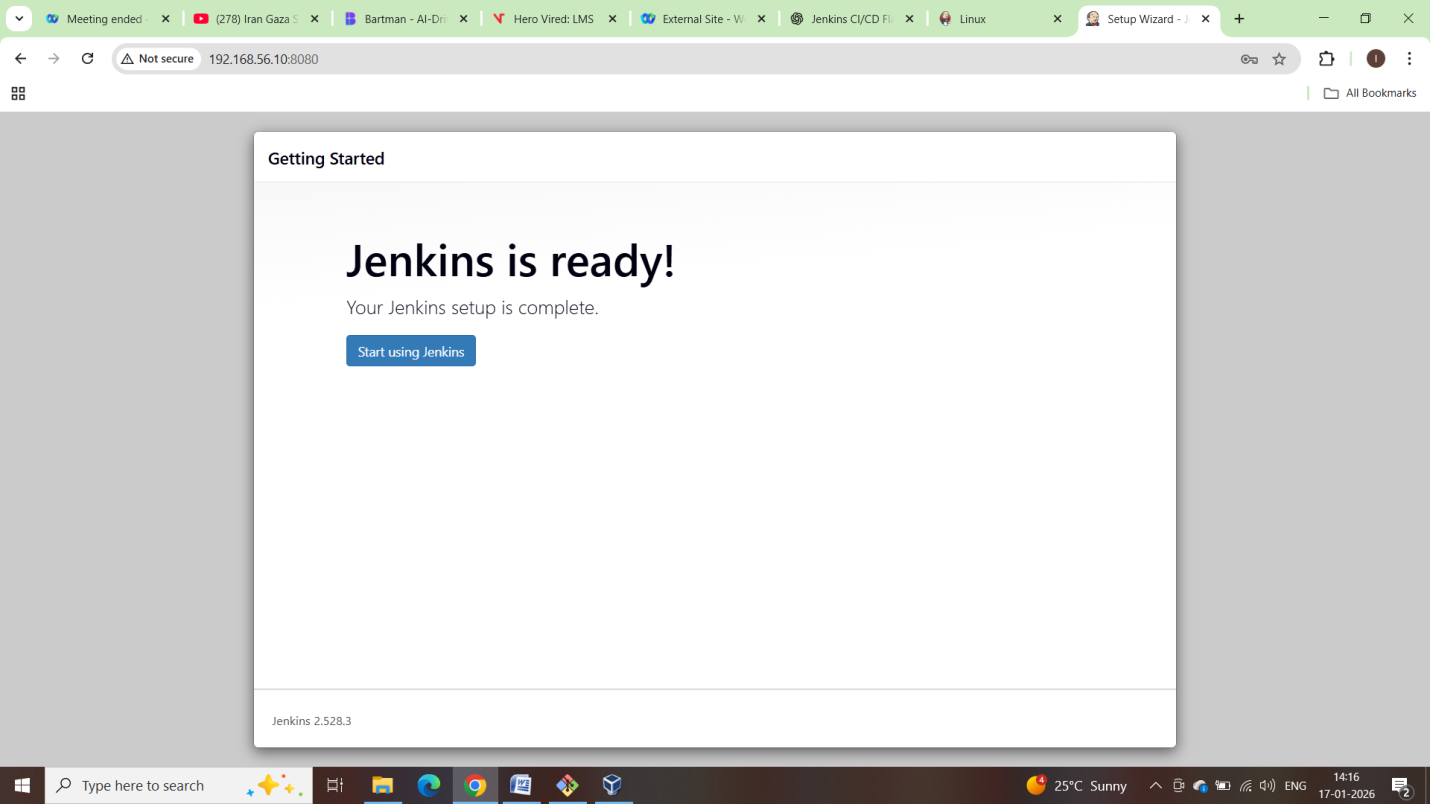


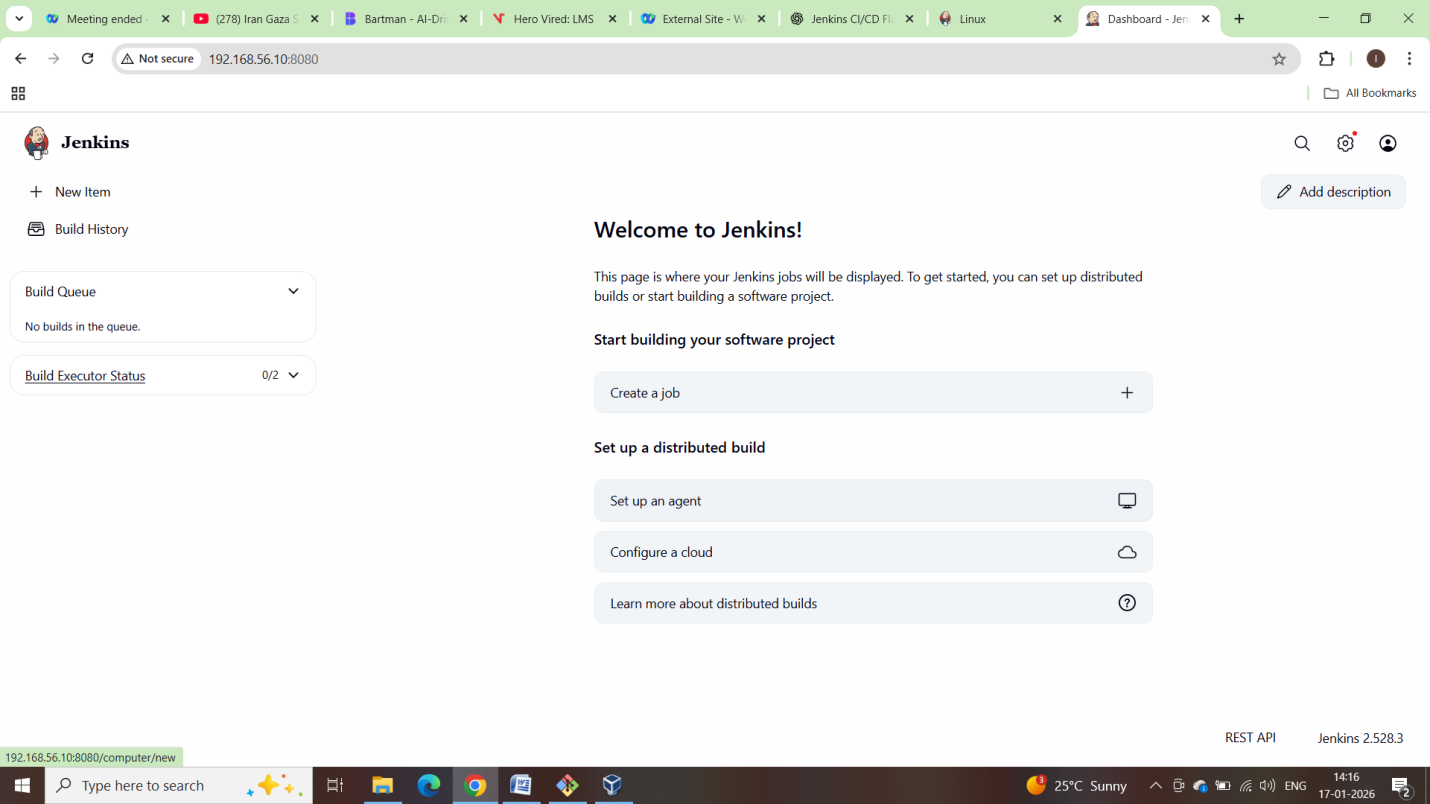






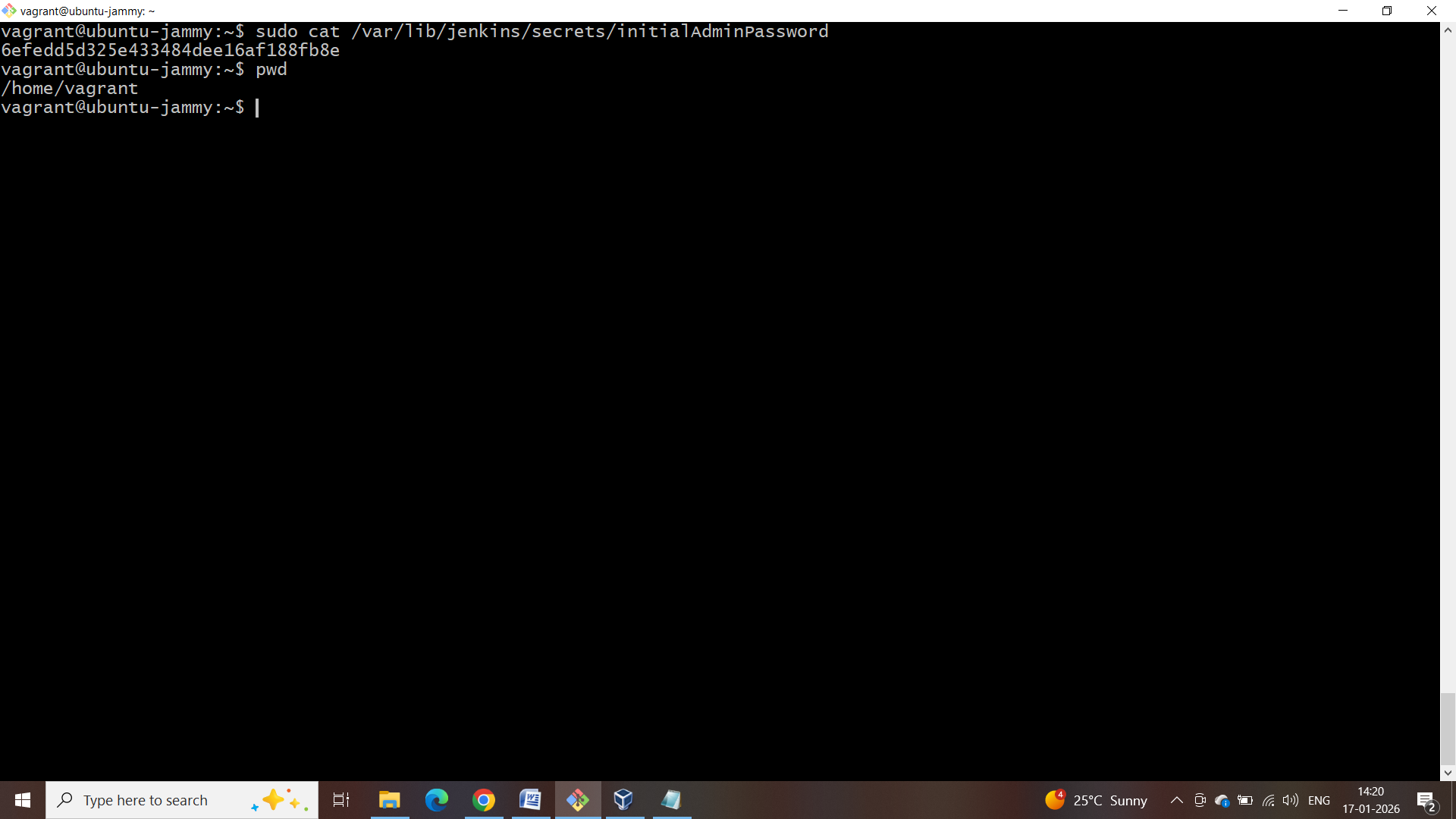




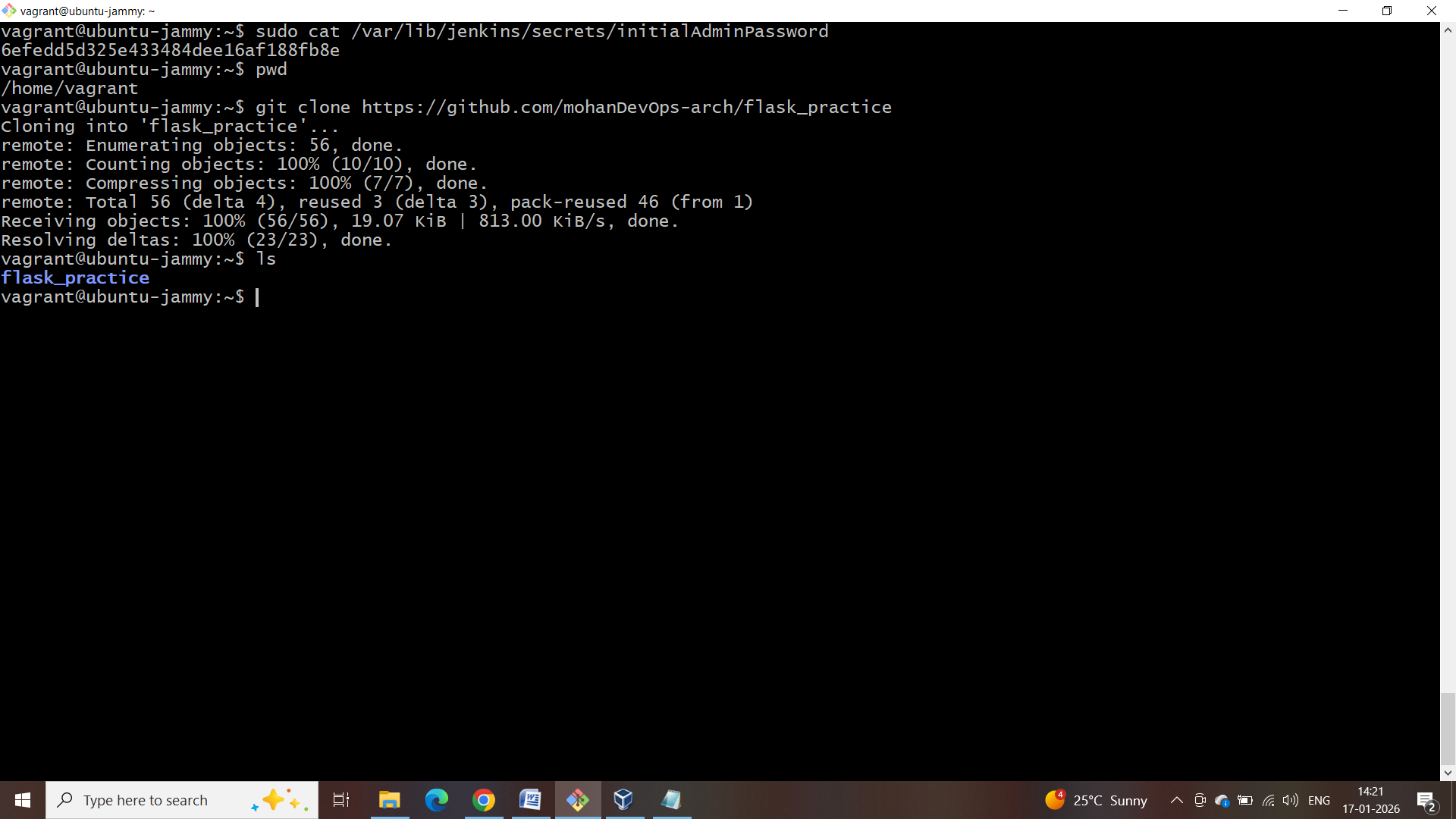


Clone Repository on Jenkins Server:

cd /home/vagrant



git clone <https://github.com/mohanDevOps-arch/flask_practice>



Creating Jenkinsfile pipeline {

pipeline {

agent any

environment {

VENV = "venv"

MONGO\_URI = "mongodb+srv://admin:admin@cluster0.stznivp.mongodb.net/sms"

}

stages {

stage('Build') {

steps {

sh '''

python3 -m venv $VENV

. $VENV/bin/activate

pip install --upgrade pip

pip install -r requirements.txt

'''

}

}

stage('Test') {

steps {

sh '''

. $VENV/bin/activate

pytest

'''

}

}

stage('Deploy to Staging') {

steps {

sh '''

echo "Deploying Flask app to staging..."

pkill -f app.py || true

nohup venv/bin/python app.py > app.log 2>&1 &

'''

}

}

}

post {

success {

mail to: 'saiyedin786@gmail.com',

subject: 'Jenkins Build SUCCESS',

body: 'The Flask pipeline completed successfully.'

}

failure {

mail to: 'saiyedin786@example.com',

subject: 'Jenkins Build FAILED',

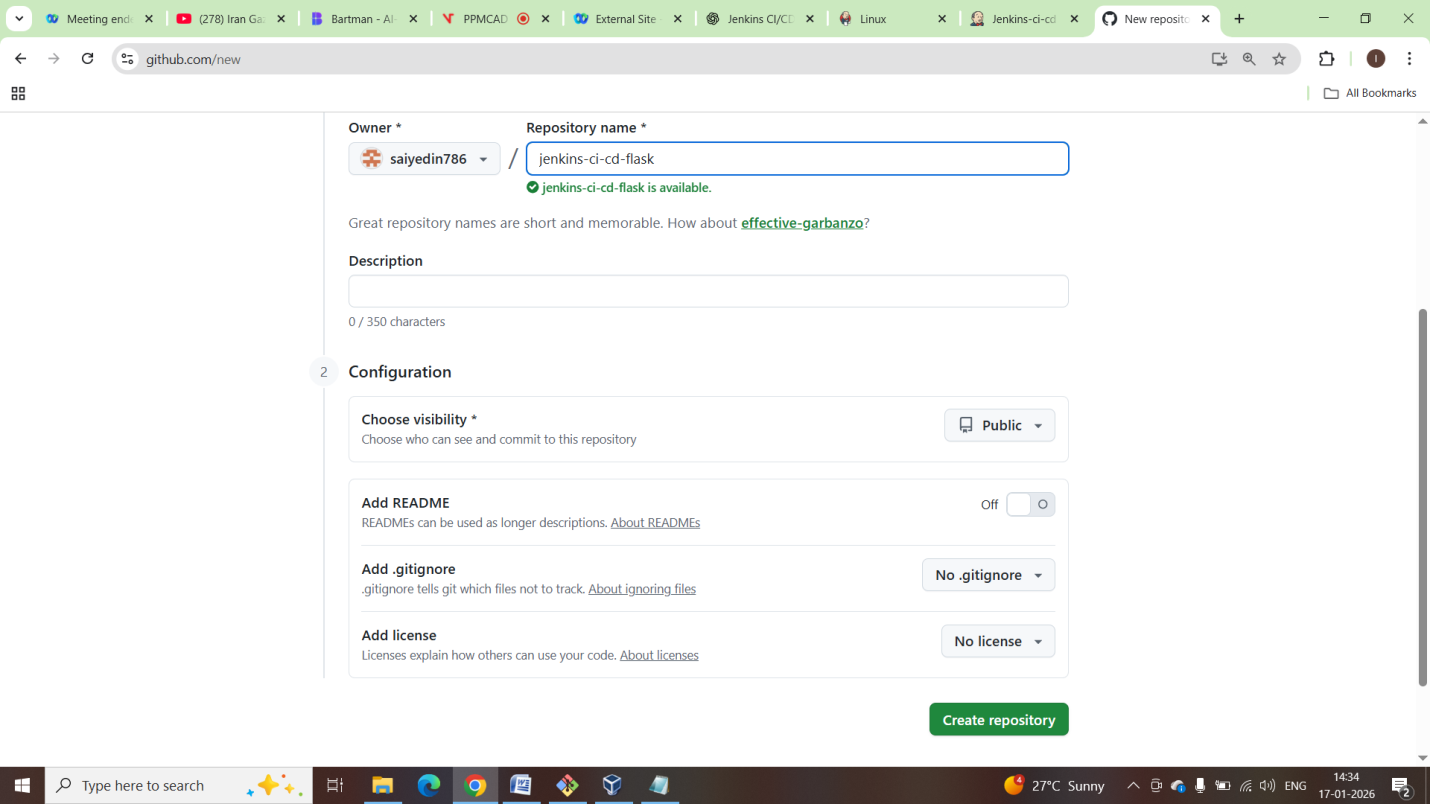
body: 'The Flask pipeline failed. Please check Jenkins.'

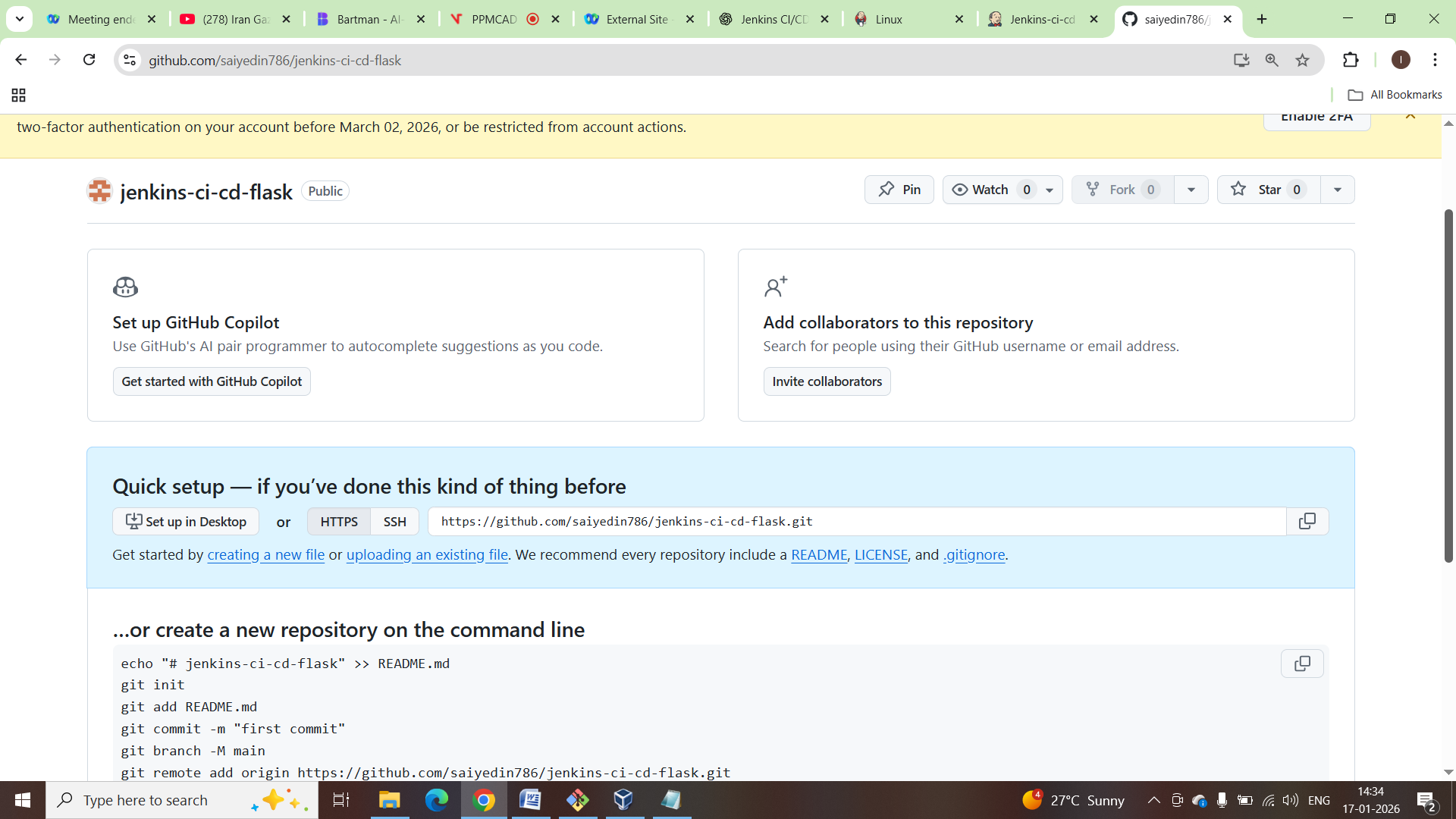
}

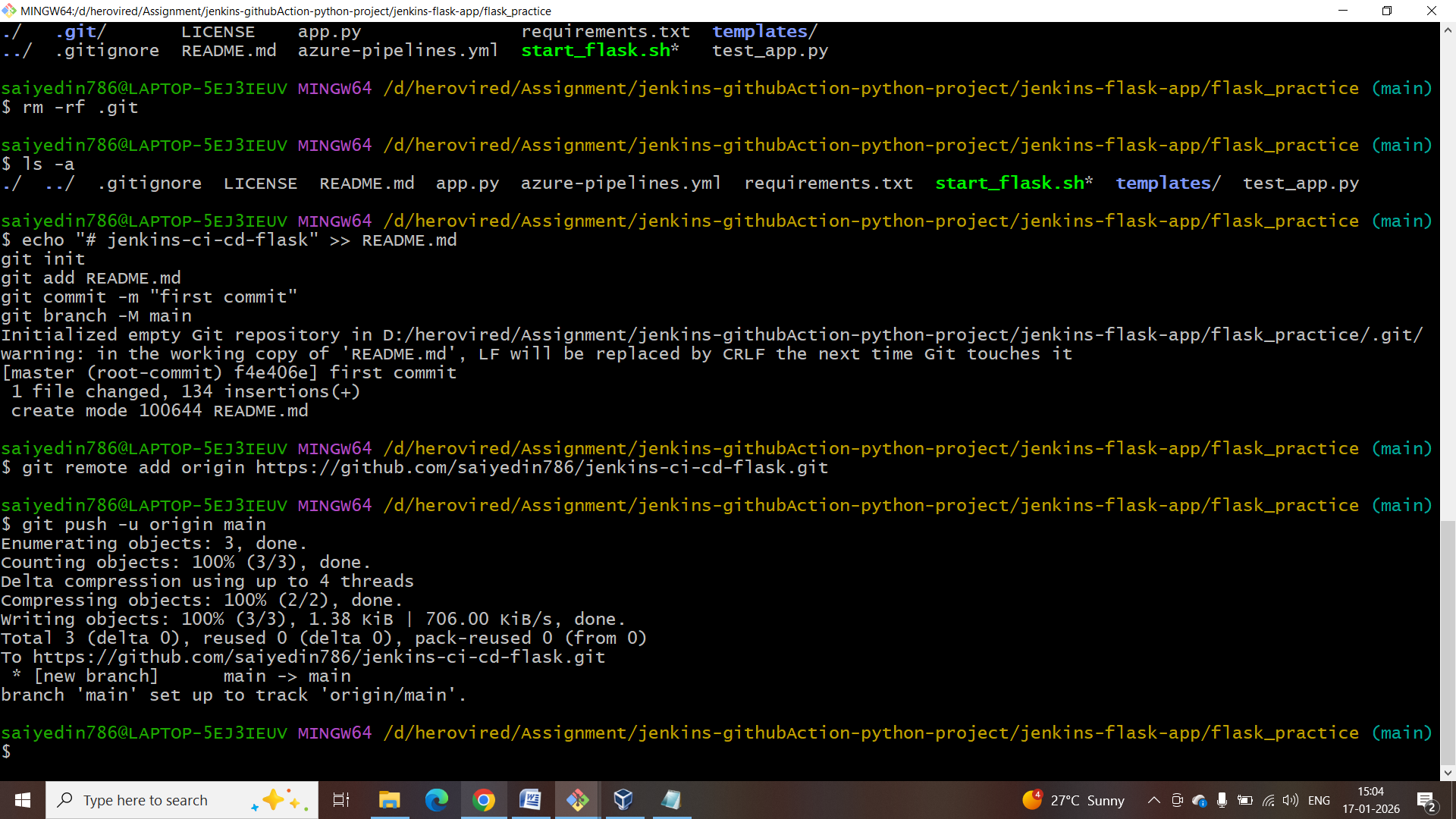
}

}

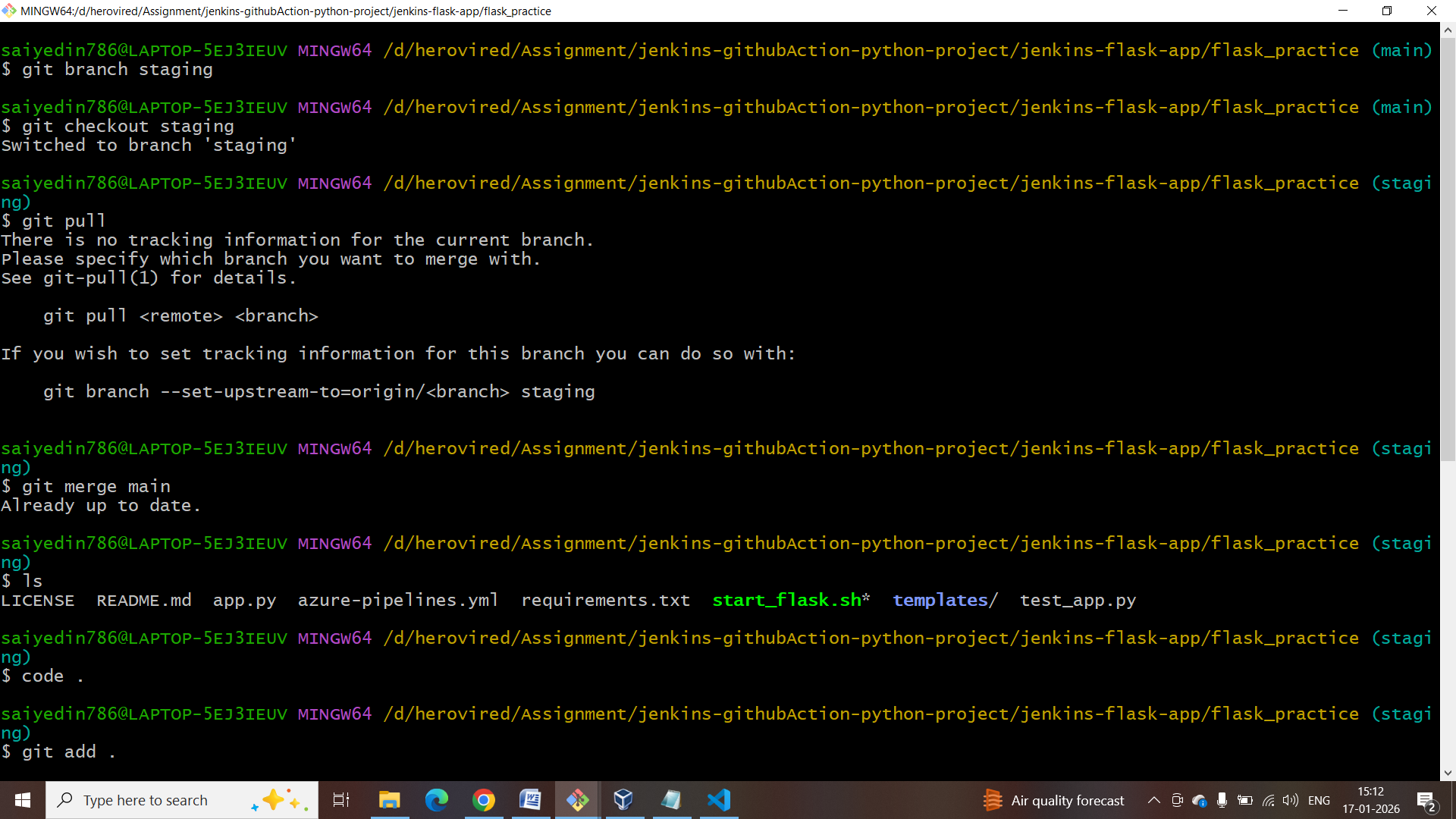
Creting an github repo:

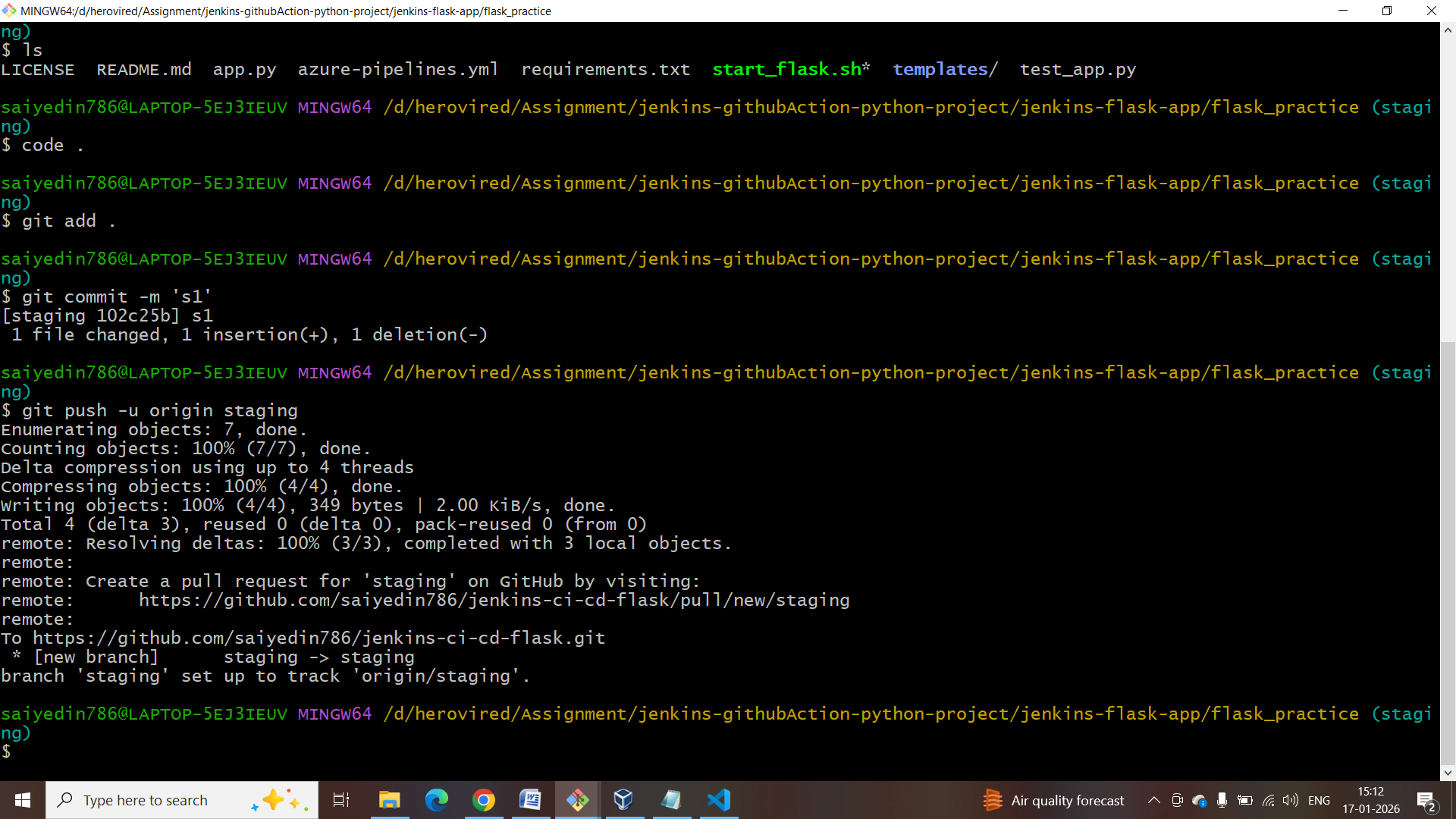






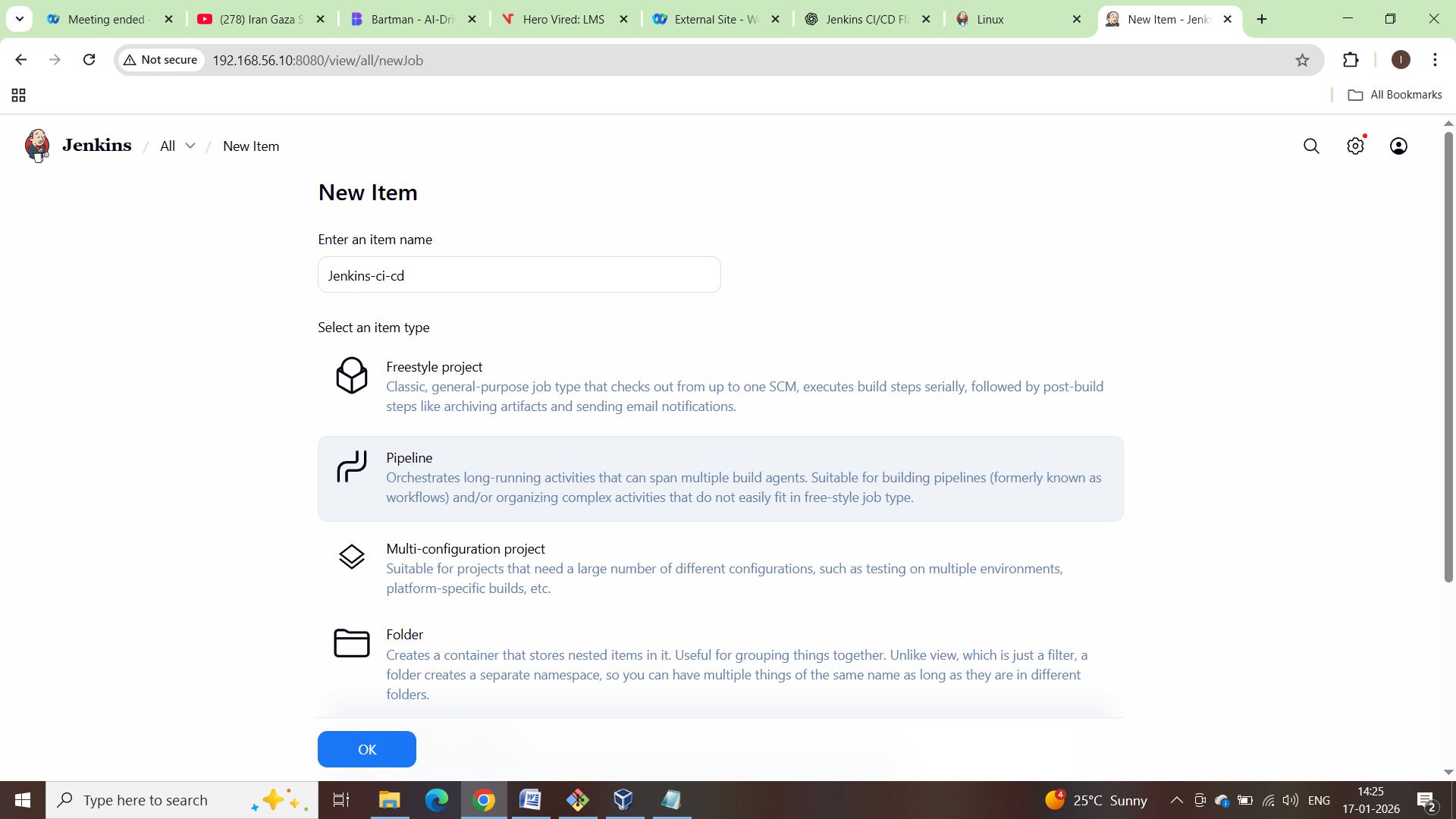
Creating staging repo:

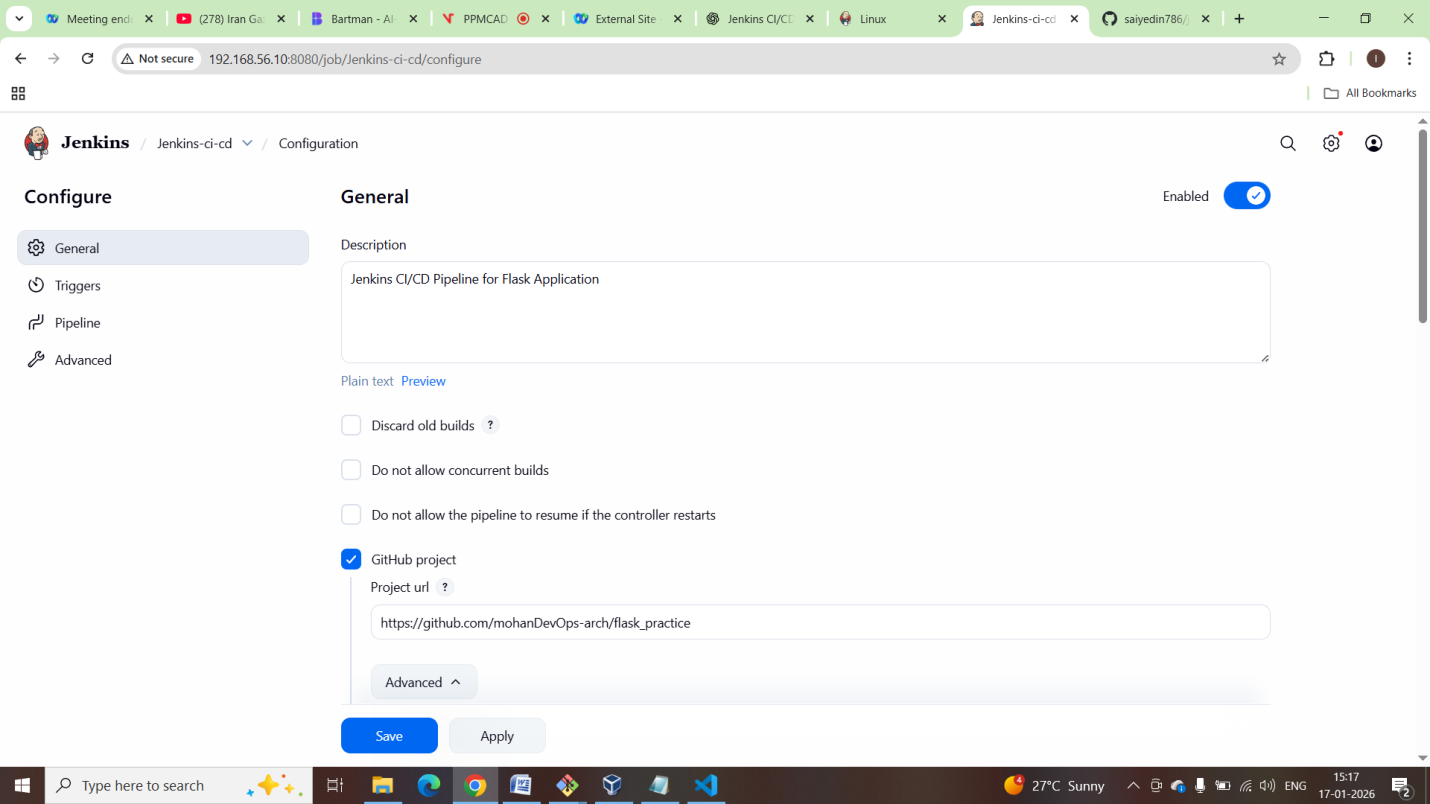


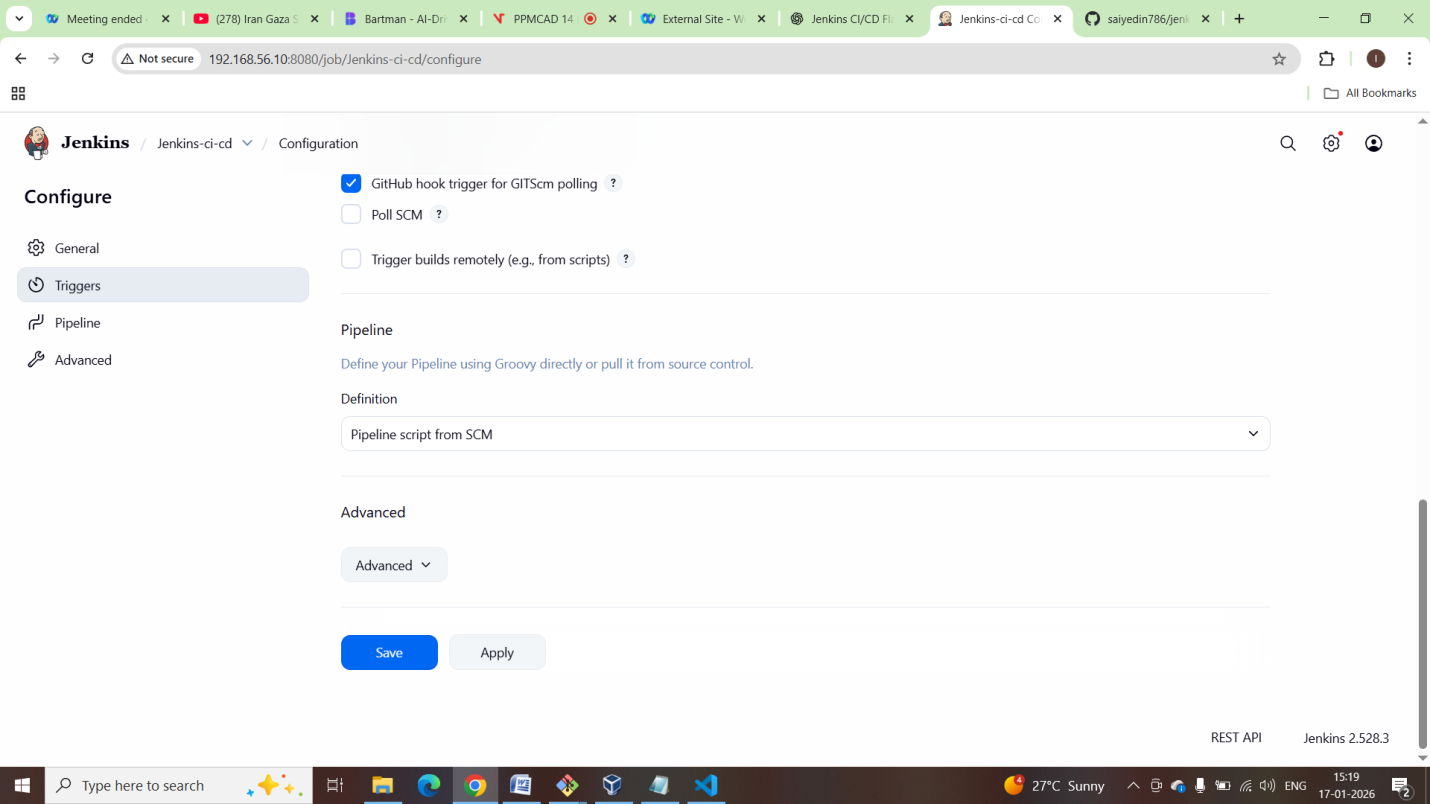


**Configure Jenkins Pipeline Job**

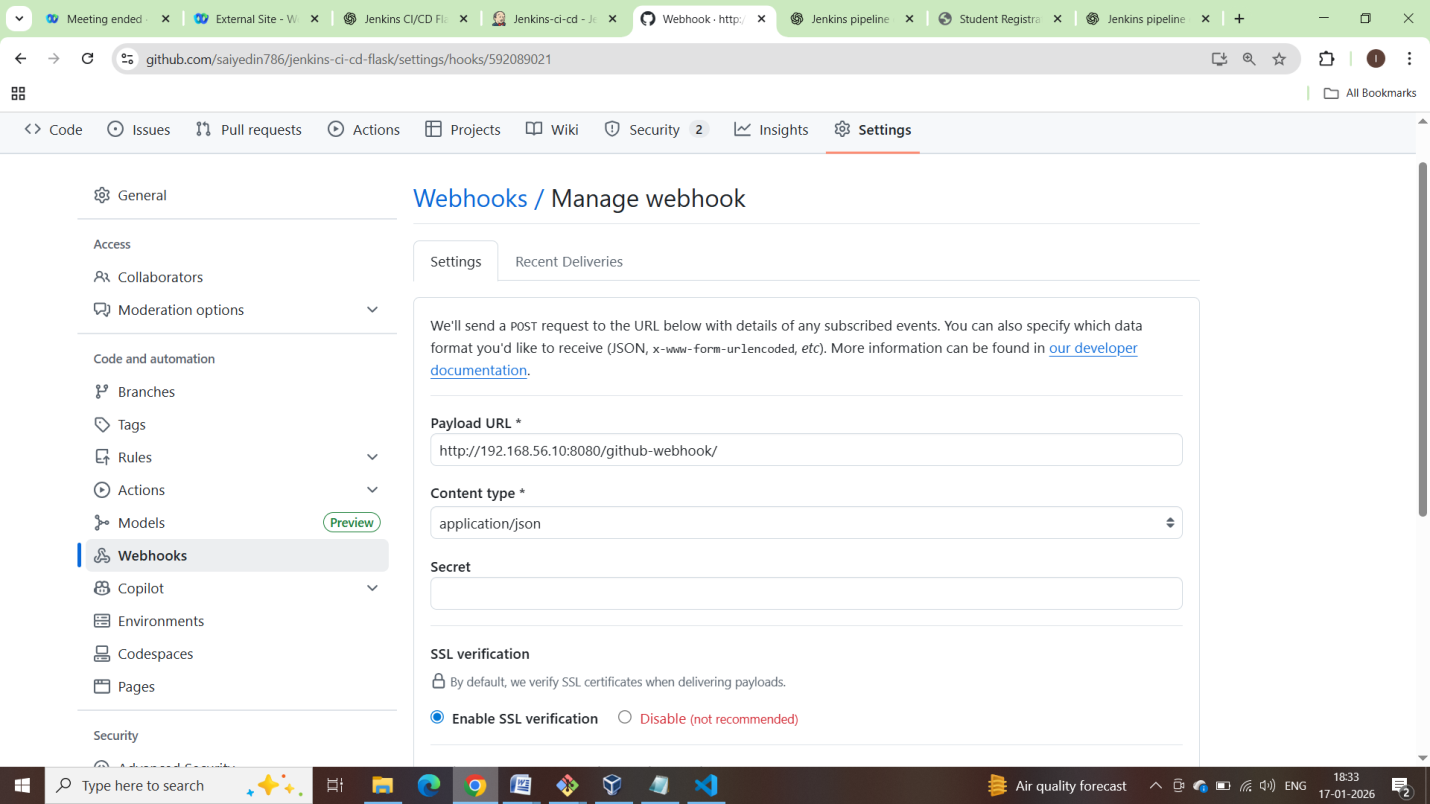
1. Jenkins Dashboard → **New Item**
2. Name: Jenkins-ci-cd
3. Type: **Pipeline**
4. Under **Pipeline**:
   * Definition: Pipeline script from SCM
   * SCM: Git
   * Repository URL:
   * <https://github.com/mohanDevOps-arch/flask_practice>
   * Branch: \*/main
   * Script Path: Jenkinsfile
5. Save

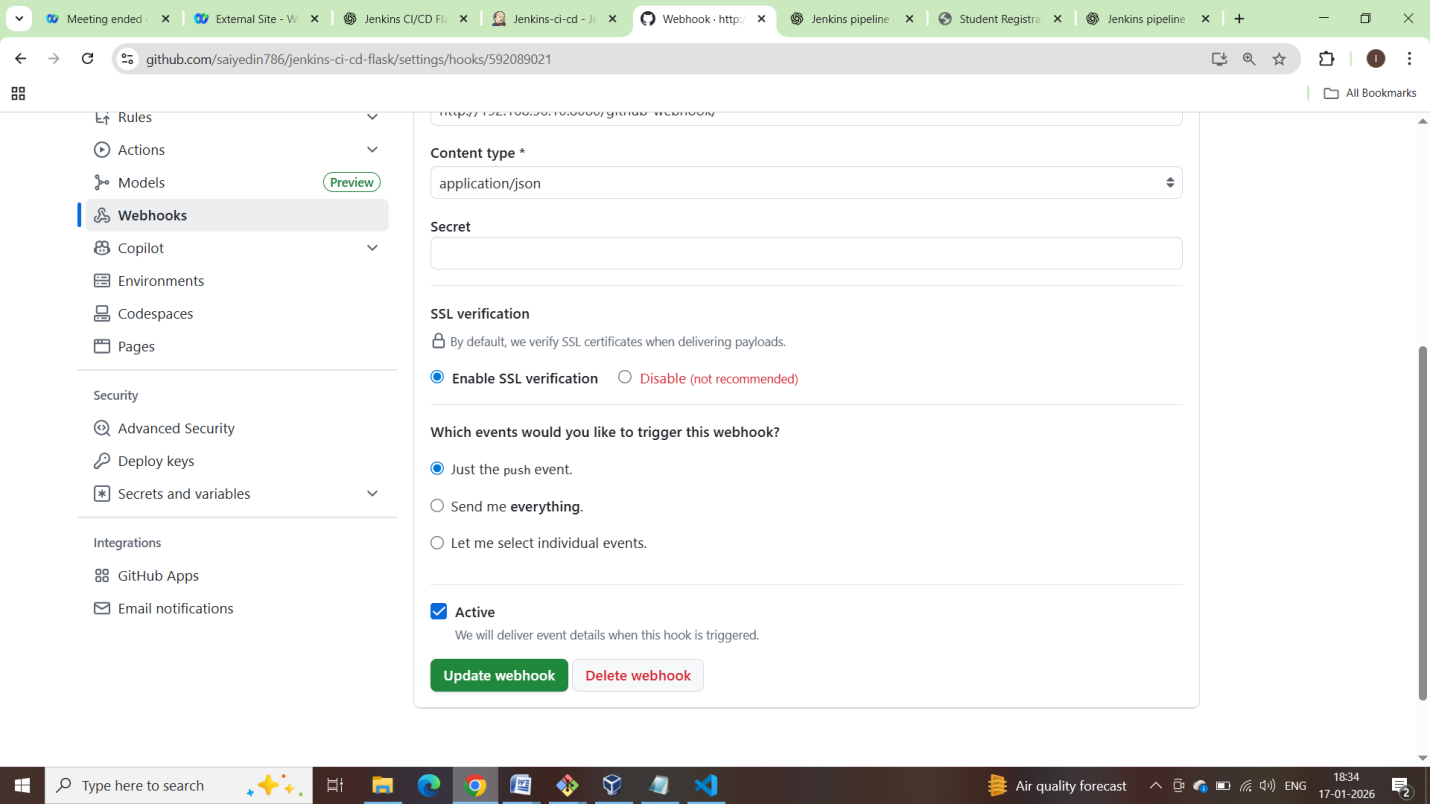




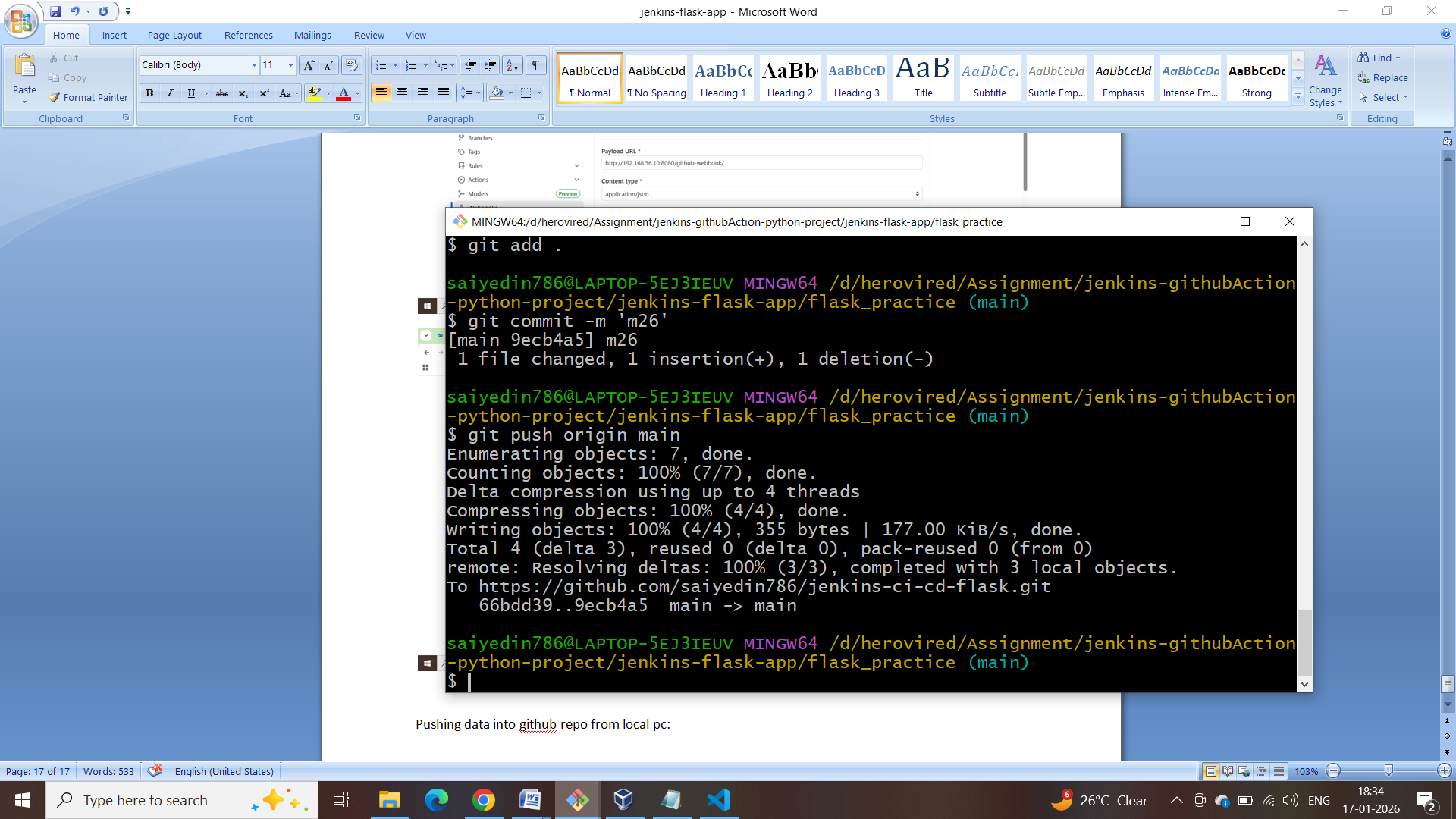


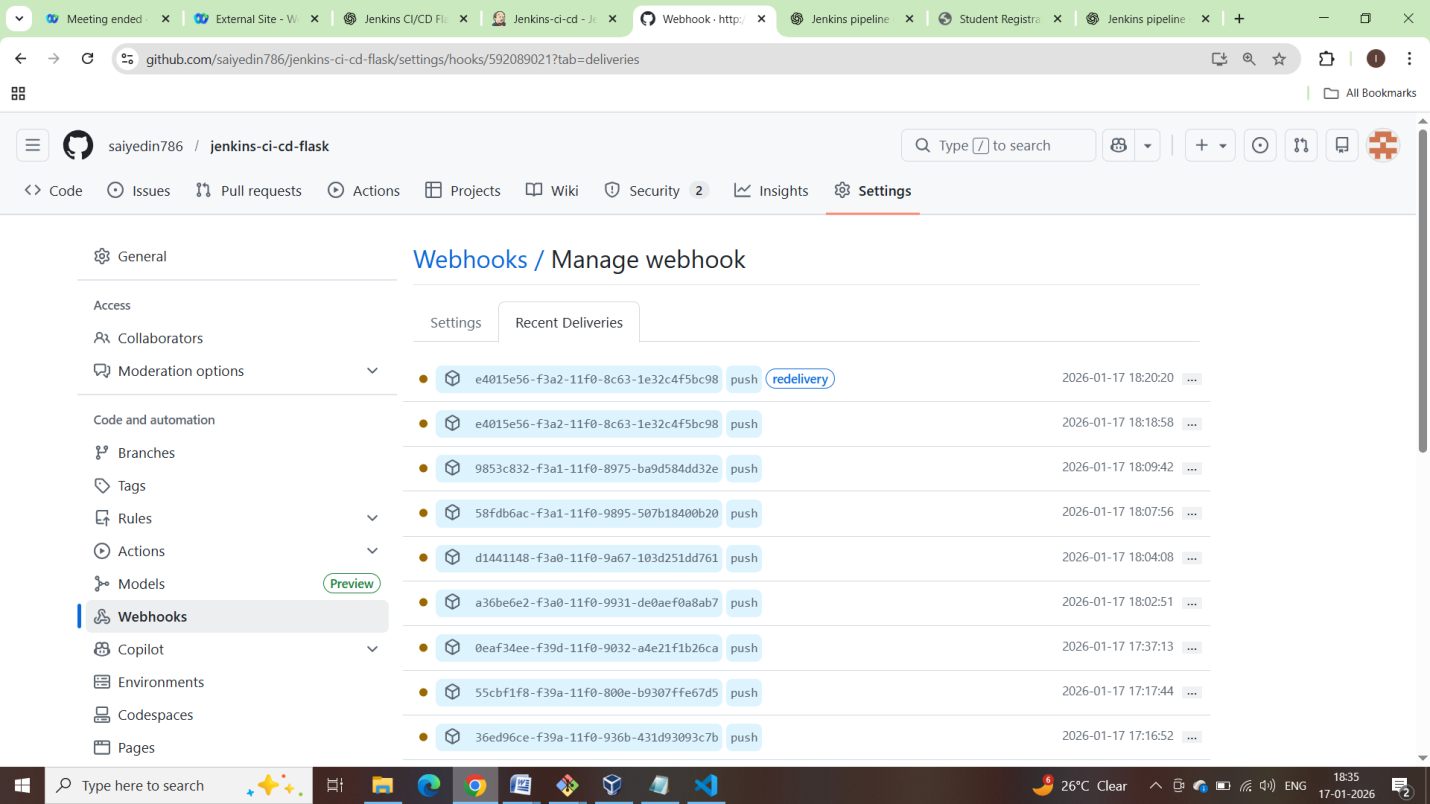
Configuring github webhook:



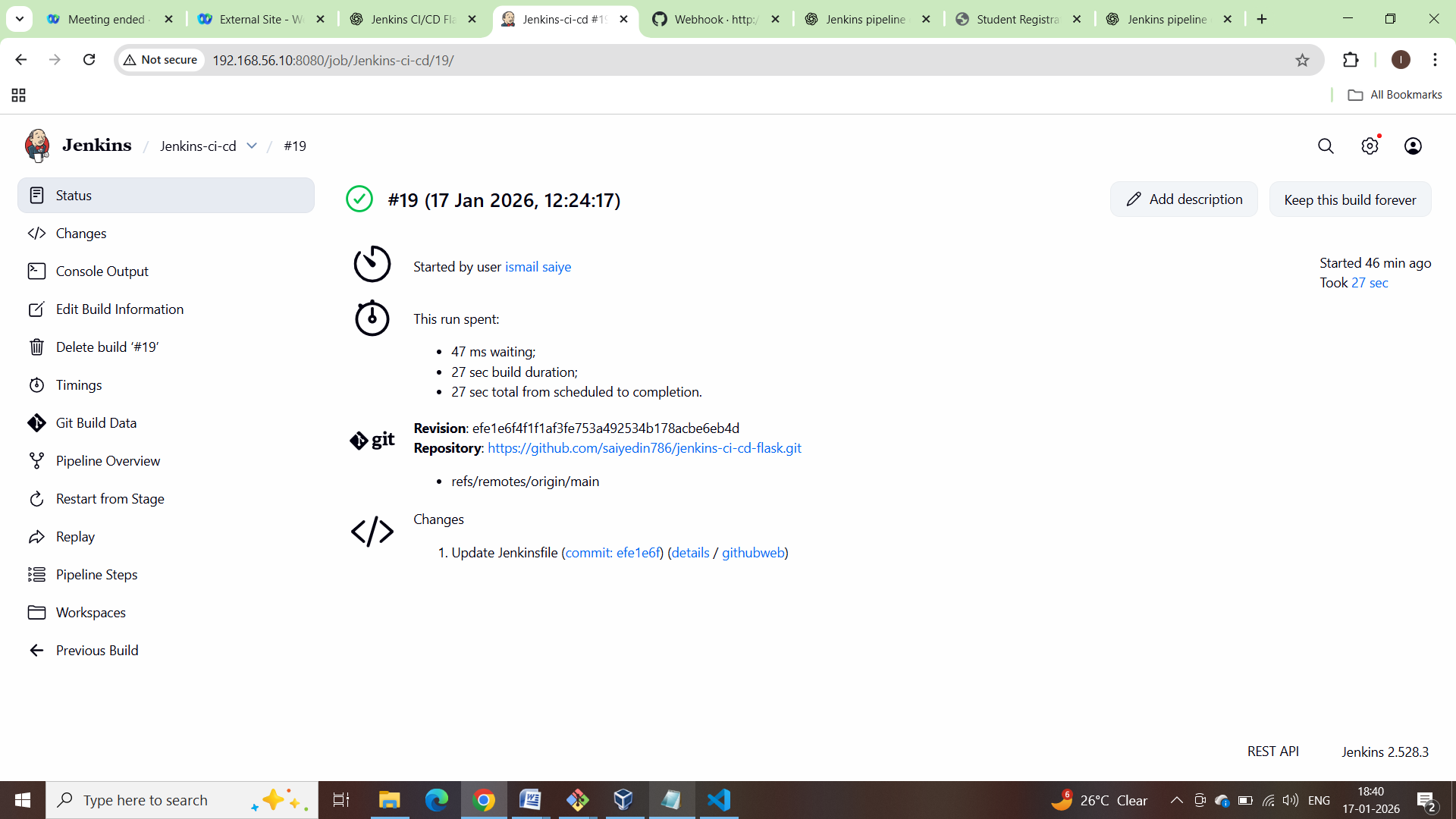


Pushing data into github repo from local pc:

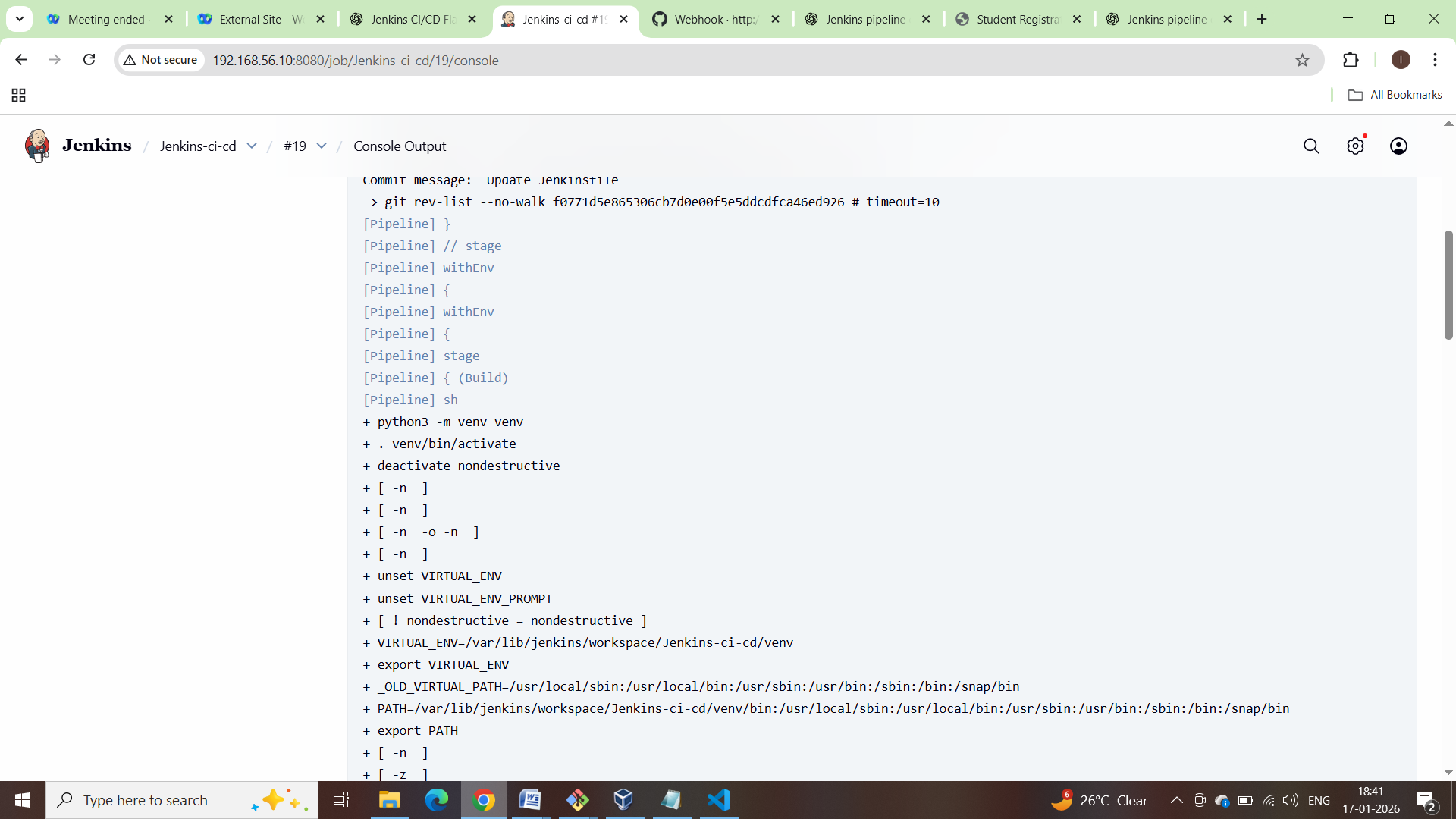


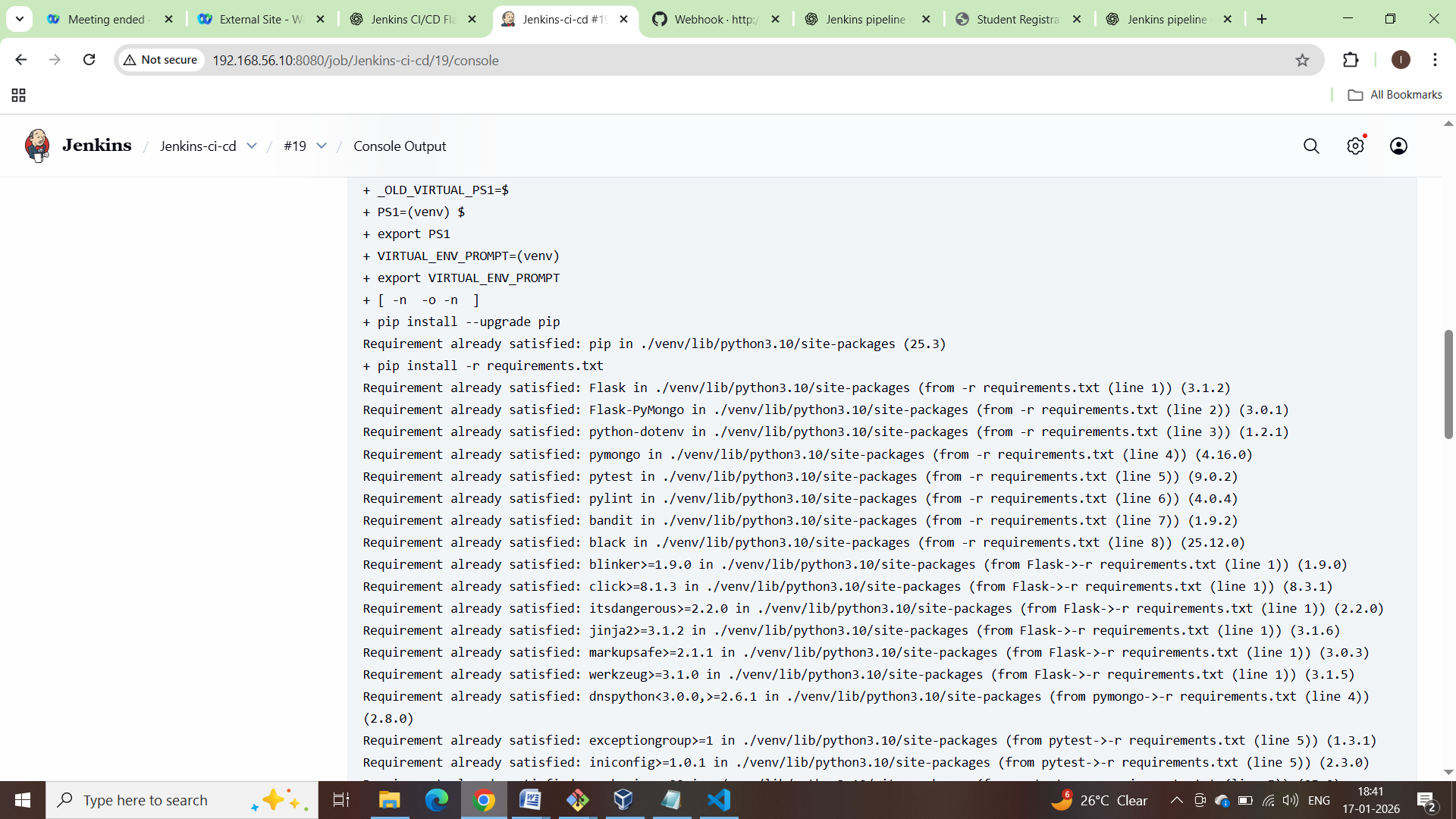


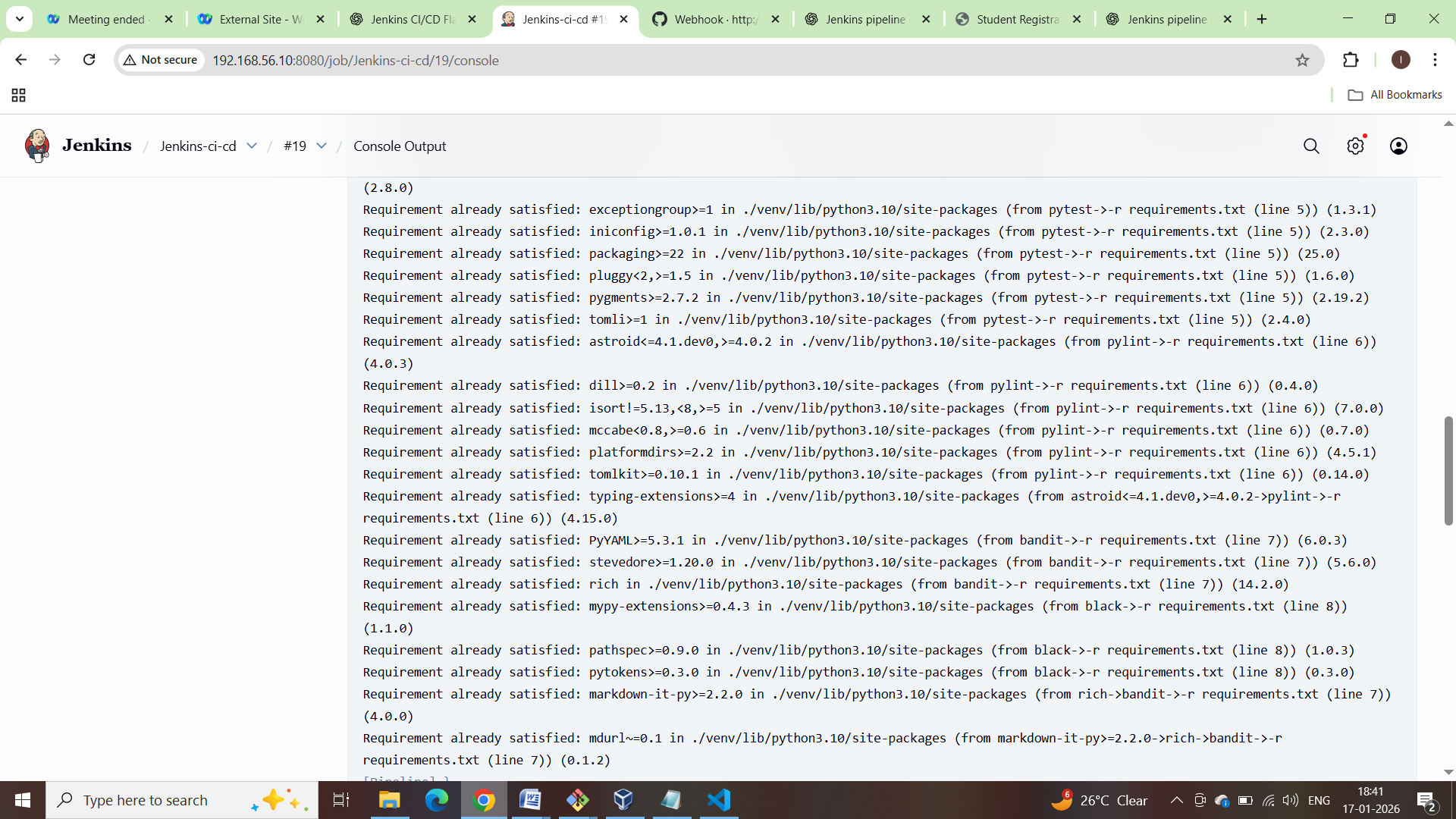
Jenkinfile runs code is deployed on ubuntu vm



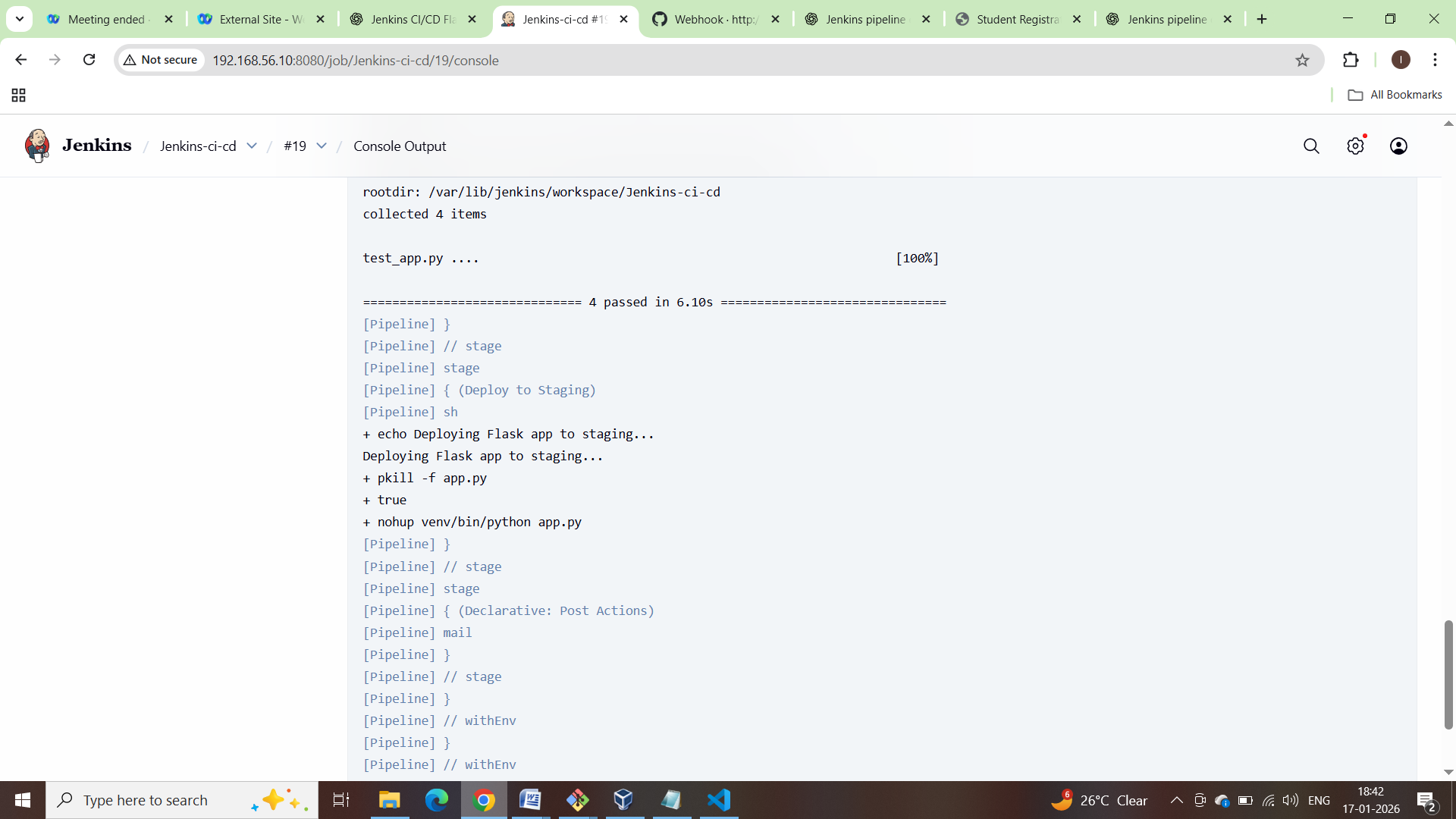


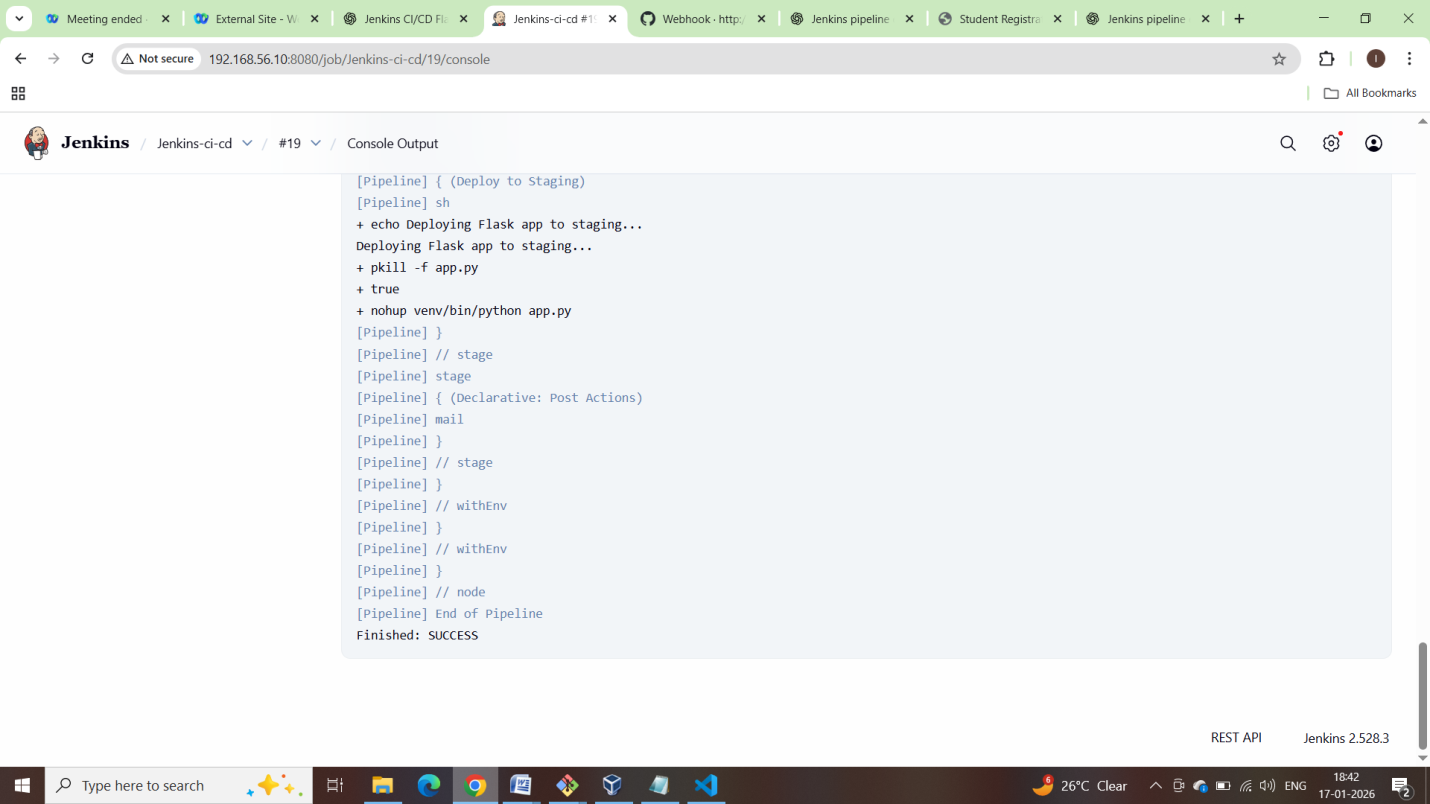


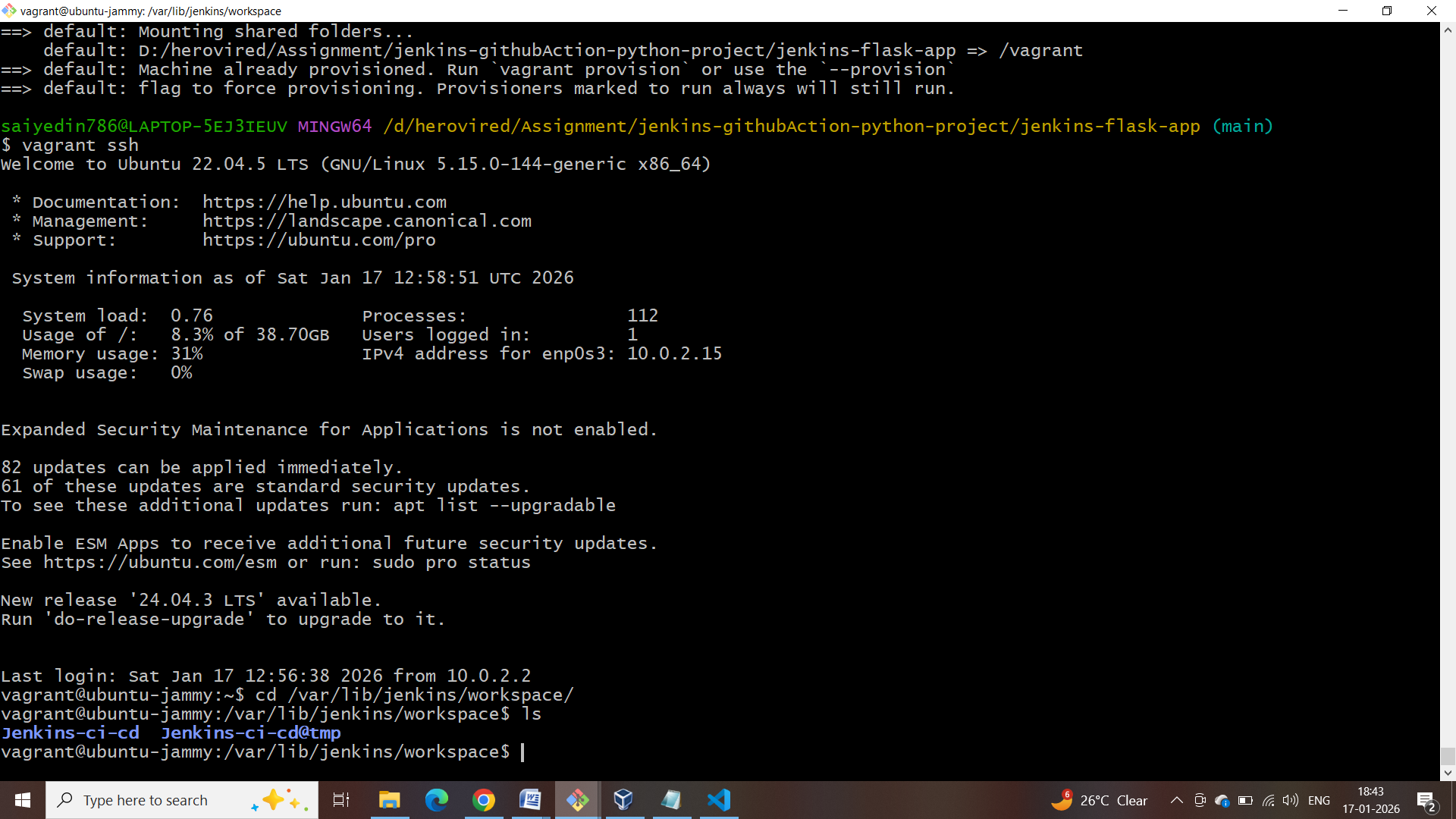


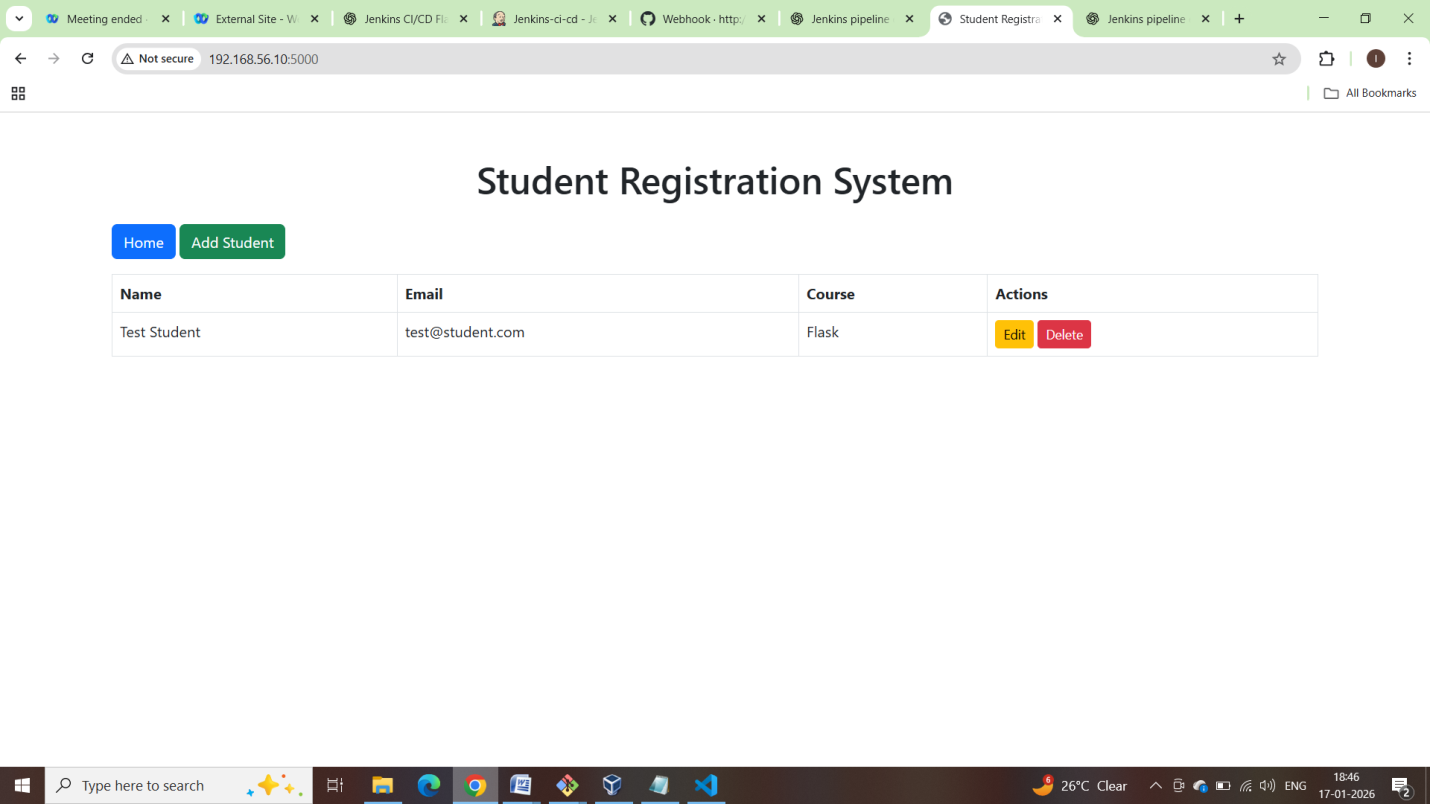












Email notification received in my email id:

