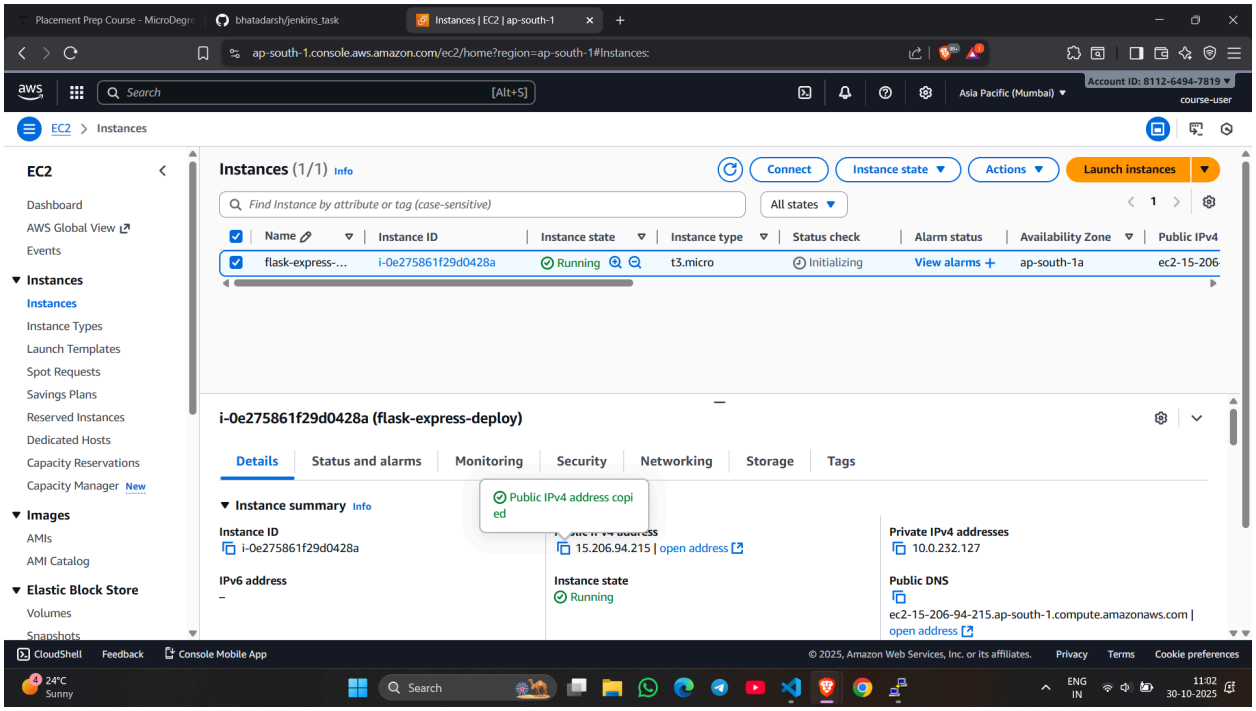
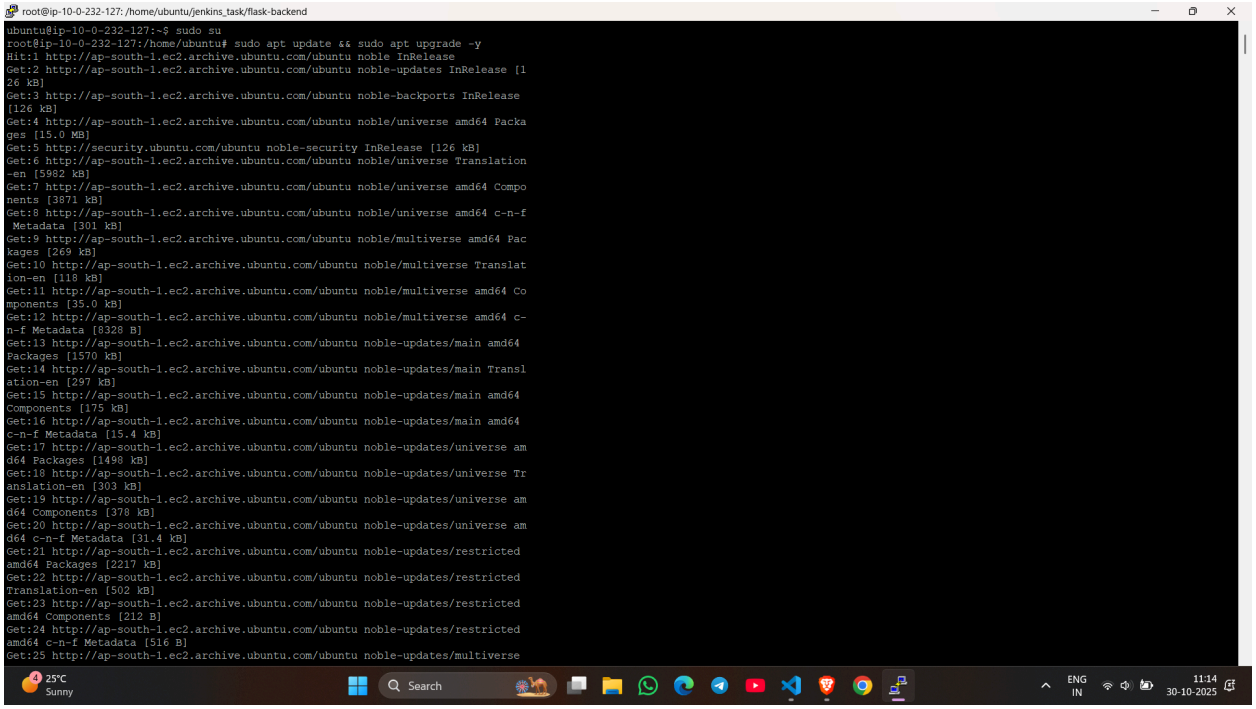
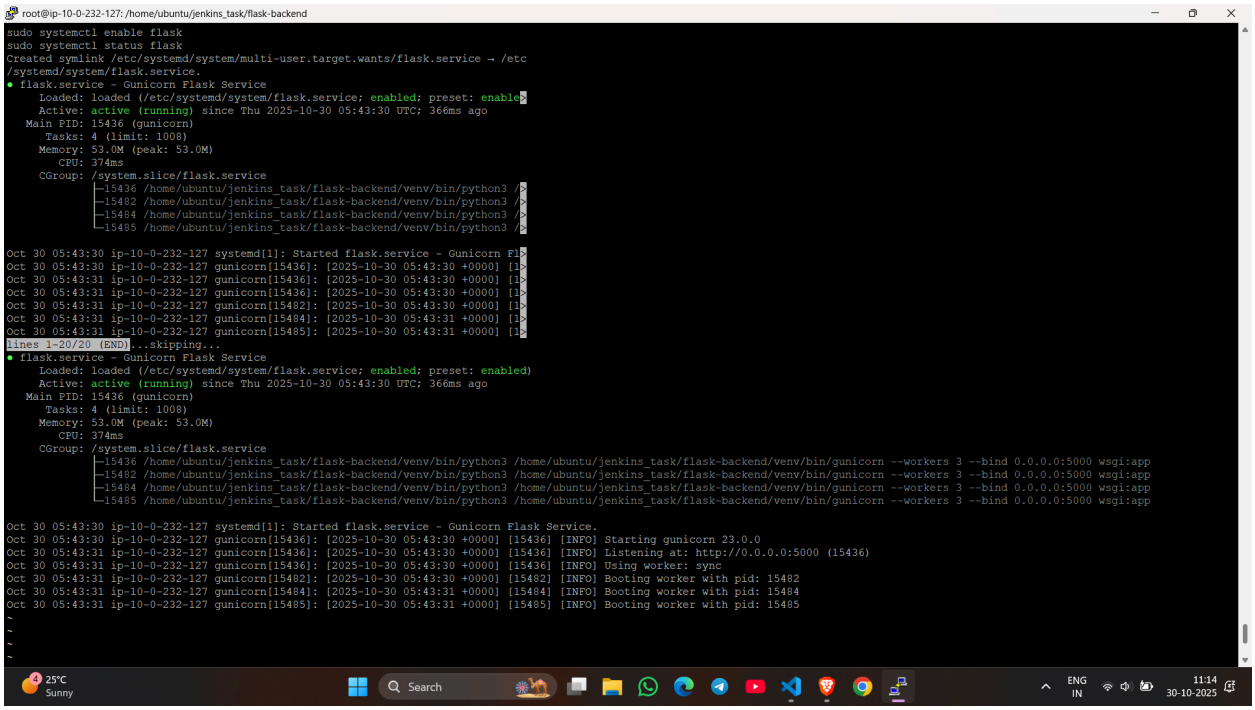
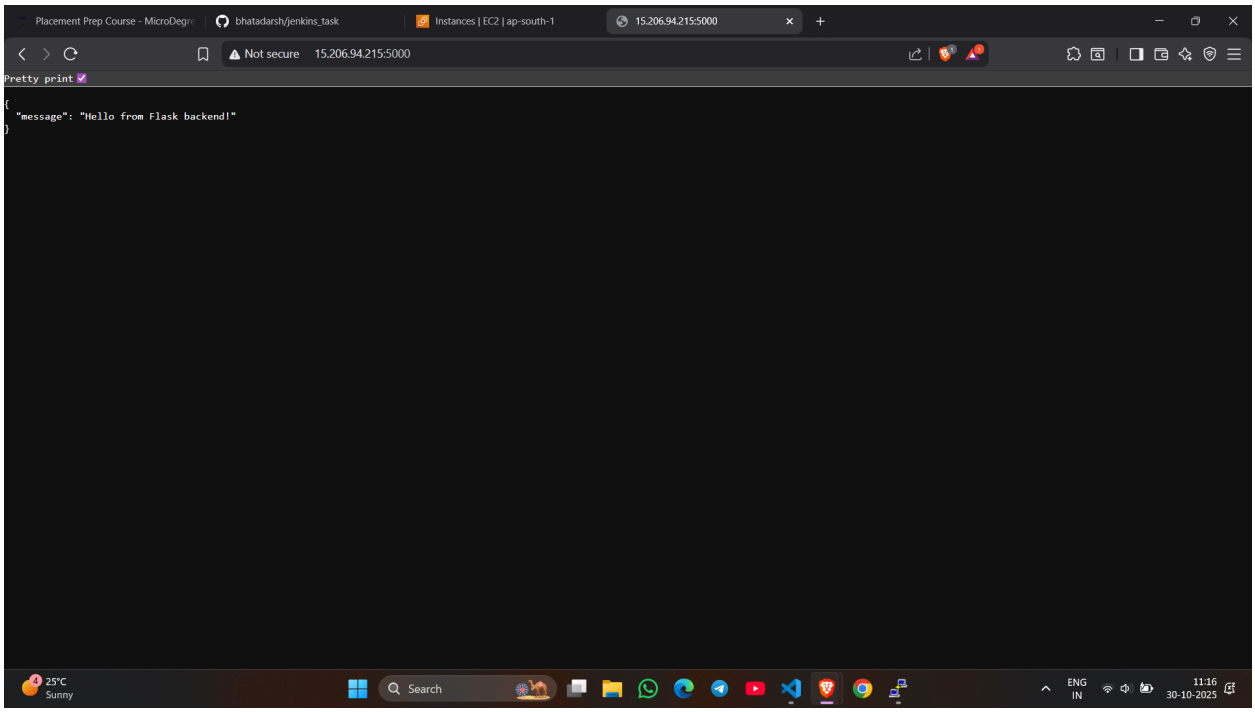
GitHub Repo: https://github.com/bhatadarsh/jenkins_task

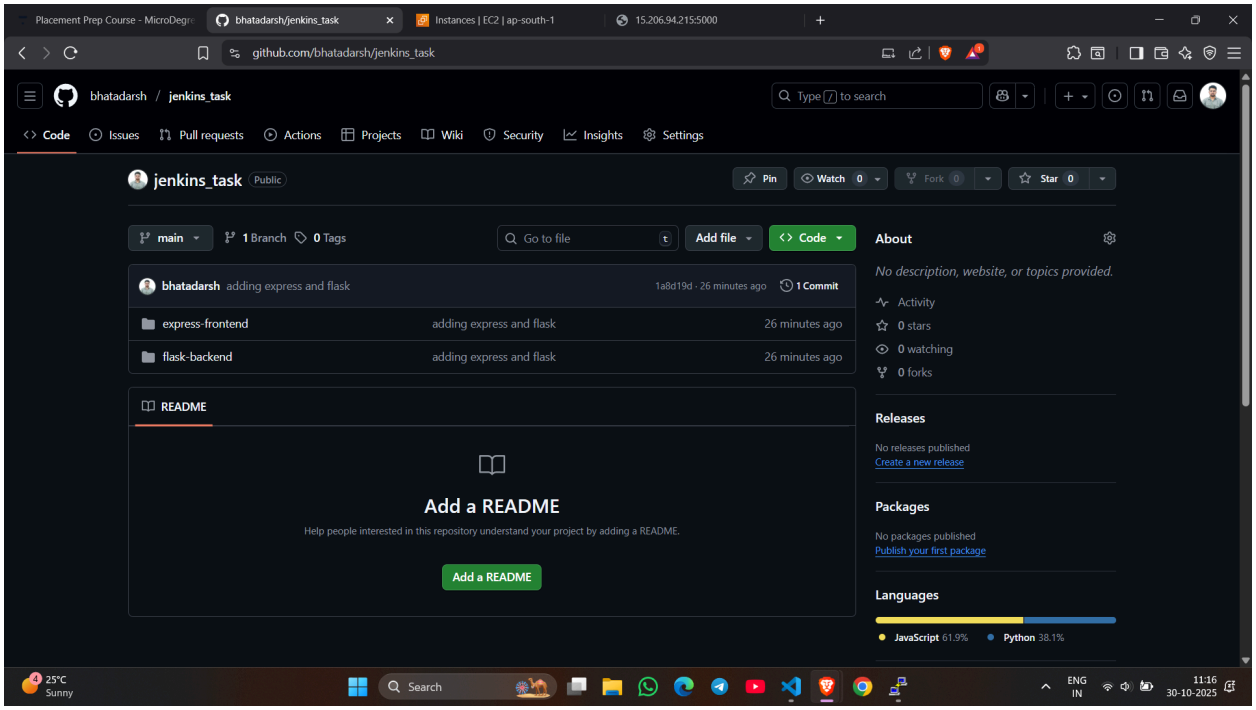
Jenkins_Task1:

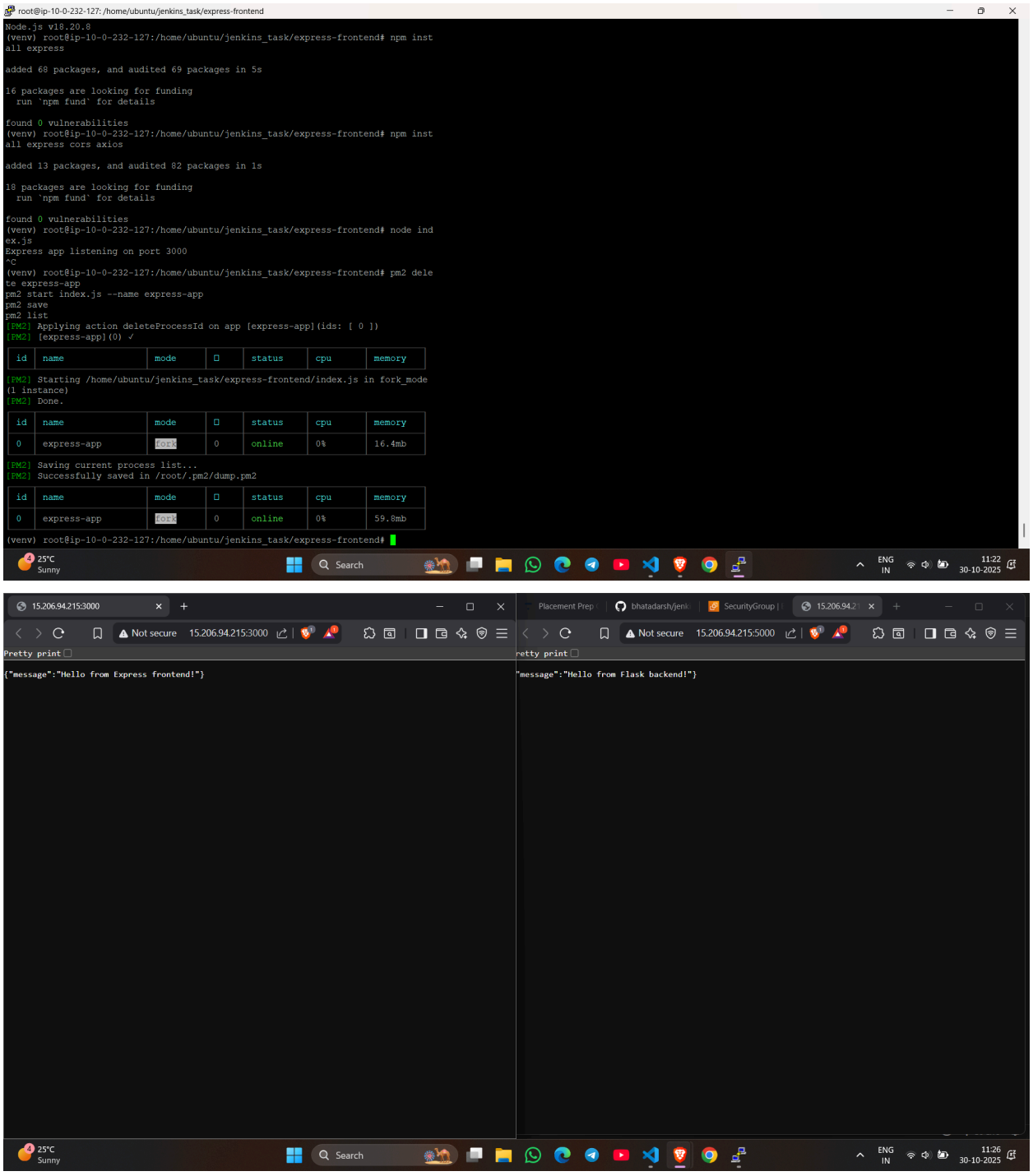





root@ip-10-0-232-127: /home/ubuntu/jenkins_task/flask-backend
sudo systemctl enable flask
sudo systemctl status flask
Created symlink /etc/systemd/system/multi-user.target.wants/flask.service → /etc/systemd/system/flask.service.
● flask.service - Gunicorn Flask Service
Loaded: loaded (/etc/systemd/system/flask.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-10-30 05:43:30 UTC; 366ms ago
Main PID: 15436 (gunicorn)
Tasks: 4 (limit: 1008)
Memory: 53.0M (peak: 53.0M)
CPU: 374ms
CGroup: /system.slice/flask.service
├─15436 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /
├─15482 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /
├─15484 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /
└─15485 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /
Oct 30 05:43:30 ip-10-0-232-127 systemd[1]: Started flask.service - Gunicorn Fl
Oct 30 05:43:30 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [1
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [1
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [1
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15482]: [2025-10-30 05:43:30 +0000] [1
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15484]: [2025-10-30 05:43:31 +0000] [1
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15485]: [2025-10-30 05:43:31 +0000] [1
lines 1-20/20 (END)...skipping...
● flask.service - Gunicorn Flask Service
Loaded: loaded (/etc/systemd/system/flask.service; enabled; preset: enabled)
Active: active (running) since Thu 2025-10-30 05:43:30 UTC; 366ms ago
Main PID: 15436 (gunicorn)
Tasks: 4 (limit: 1008)
Memory: 53.0M (peak: 53.0M)
CPU: 374ms
CGroup: /system.slice/flask.service
├─15436 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /home/ubuntu/jenkins_task/flask-backend/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 wsgi:app
├─15482 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /home/ubuntu/jenkins_task/flask-backend/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 wsgi:app
├─15484 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /home/ubuntu/jenkins_task/flask-backend/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 wsgi:app
└─15485 /home/ubuntu/jenkins_task/flask-backend/venv/bin/python3 /home/ubuntu/jenkins_task/flask-backend/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 wsgi:app
Oct 30 05:43:30 ip-10-0-232-127 systemd[1]: Started flask.service - Gunicorn Flask Service.
Oct 30 05:43:30 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [15436] [INFO] Starting gunicorn 23.0.0
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [15436] [INFO] Listening at: http://0.0.0.0:5000 (15436)
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15436]: [2025-10-30 05:43:30 +0000] [15436] [INFO] Using worker: sync
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15482]: [2025-10-30 05:43:30 +0000] [15482] [INFO] Booting worker with pid: 15482
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15484]: [2025-10-30 05:43:31 +0000] [15484] [INFO] Booting worker with pid: 15484
Oct 30 05:43:31 ip-10-0-232-127 gunicorn[15485]: [2025-10-30 05:43:31 +0000] [15485] [INFO] Booting worker with pid: 15485
25°C
Sunny
Search
ENG
IN
11:14
30-10-2025


Placement Prep Course - MicroDegree
bhatadarsh/jenkins_task
Instances | EC2 | ap-south-1
15.206.94.215:5000
Not secure
15.206.94.215:5000
Pretty print
{
"message": "Hello from Flask backend!"
}
25°C
Sunny
Search
ENG
IN
11:16
30-10-2025

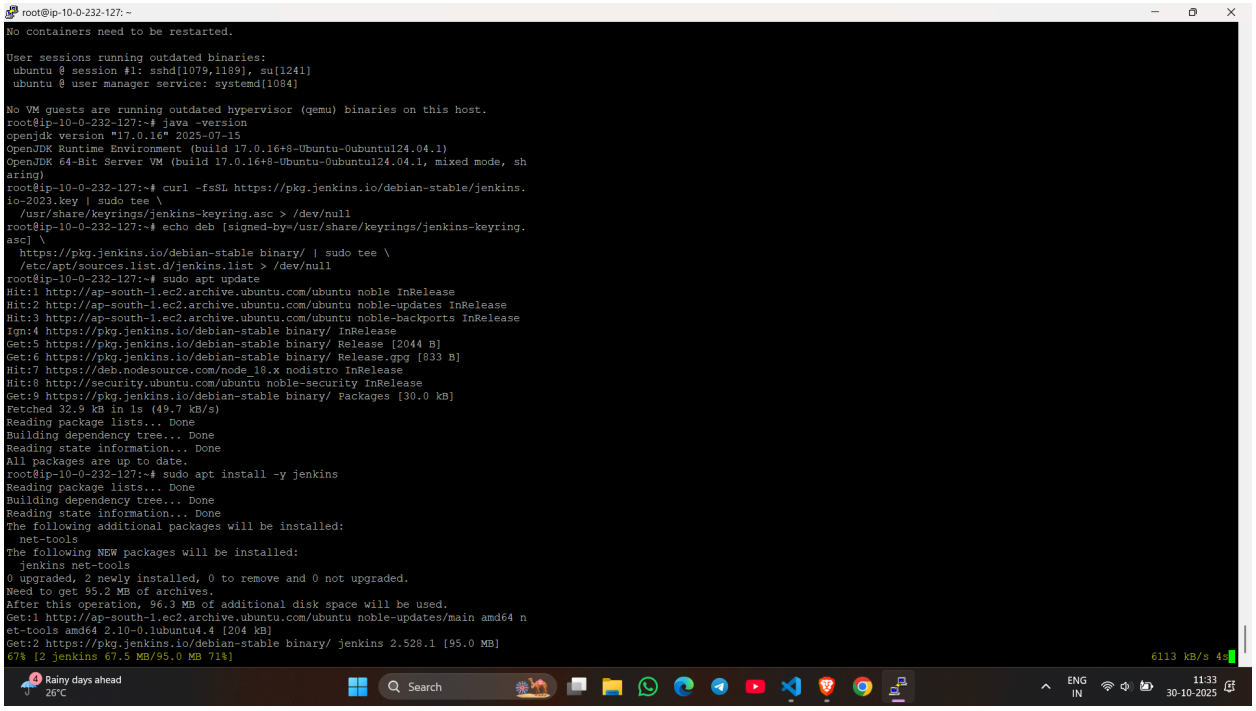

Placement Prep Course - MicroDegree
bhatadarsh/jenkins_task
Instances | EC2 | ap-south-1
15.206.94.215:5000
github.com/bhatadarsh/jenkins_task
bhatadarsh / jenkins_task
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
jenkins_task
Public
Pin
Watch 0
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
bhatadarsh adding express and flask
1a8d19d · 26 minutes ago
1 Commit
express-frontend
adding express and flask
26 minutes ago
flask-backend
adding express and flask
26 minutes ago
README
Add a README
Help people interested in this repository understand your project by adding a README.
Add a README
About
No description, website, or topics provided.
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
JavaScript 61.9%
Python 38.1%
25°C
Sunny
Search
ENG
IN
11:16
30-10-2025

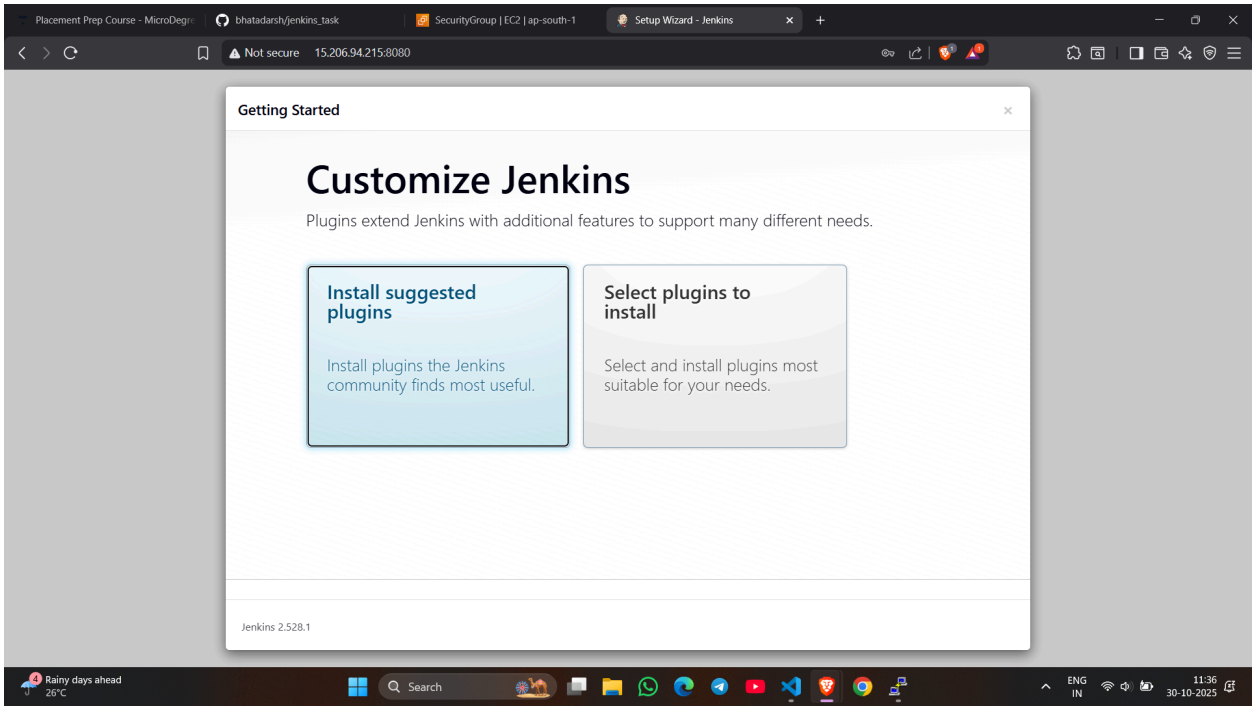


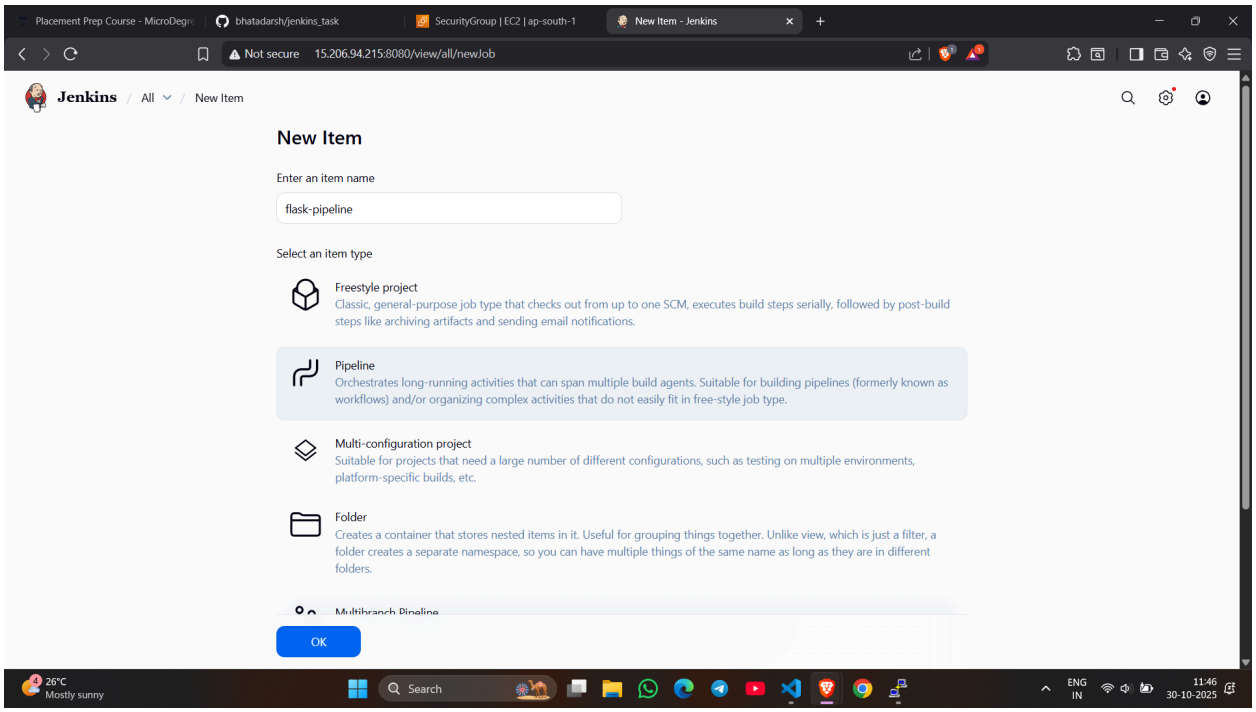
In this task, both the Flask backend and Express frontend applications were deployed on a single Ubuntu EC2 instance in the AWS ap-south-1 (Mumbai) region using a t3.micro instance type. The instance was launched with the existing key pair key-06-10-2025, and the required inbound ports (22 for SSH, 5000 for Flask, and 3000 for Express) were opened in the security group. After connecting to the instance via SSH, essential packages like Git, Python3, pip, Node.js, npm, Gunicorn, and PM2 were installed. The project repository containing both

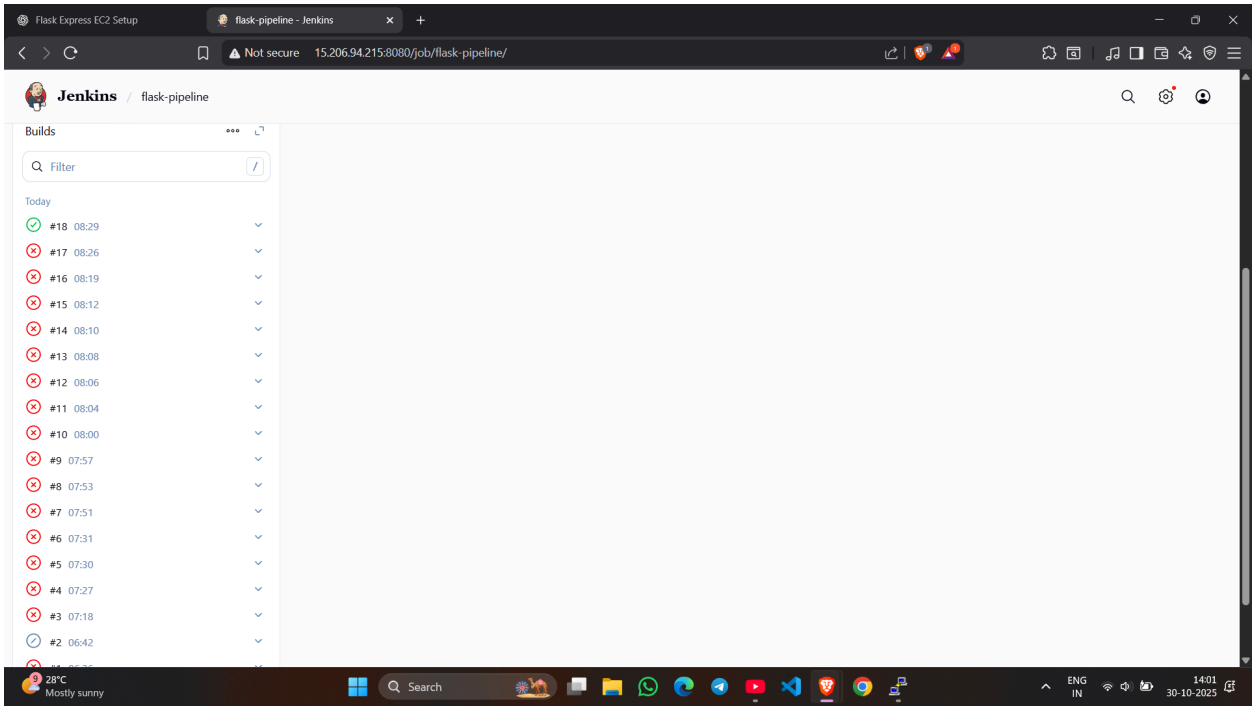
applications (jenkins_task) was cloned from GitHub. The Flask backend was set up inside a Python virtual environment and configured to run persistently using Gunicorn and a systemd service on port 5000. The Express frontend was installed with npm and managed using PM2 on port 3000. Both applications were verified to be running and accessible through the public IP address of the EC2 instance — Flask on http://15.206.94.215:5000 and Express on http://15.206.94.215:3000. To save cost under the AWS Free Tier, an Elastic IP was not used. This setup demonstrates a full-stack deployment on a single EC2 instance, ensuring both services start automatically on system reboot and remain continuously available.

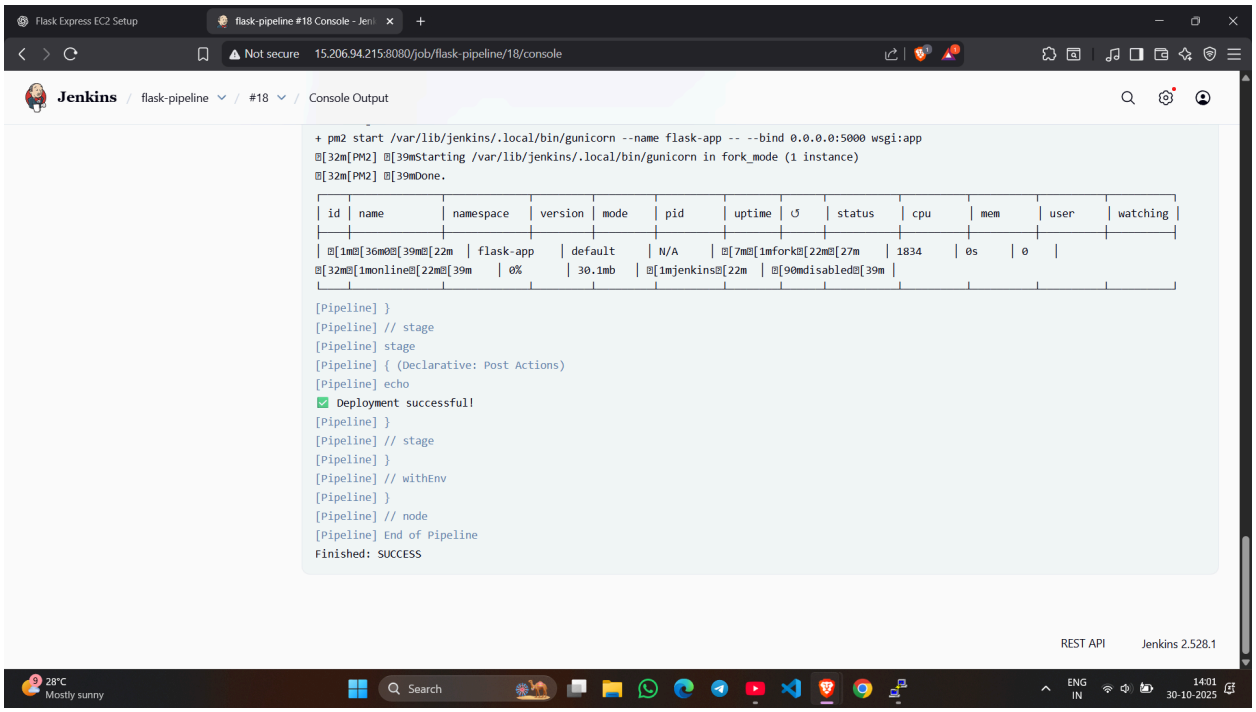
Task2:CI|CD

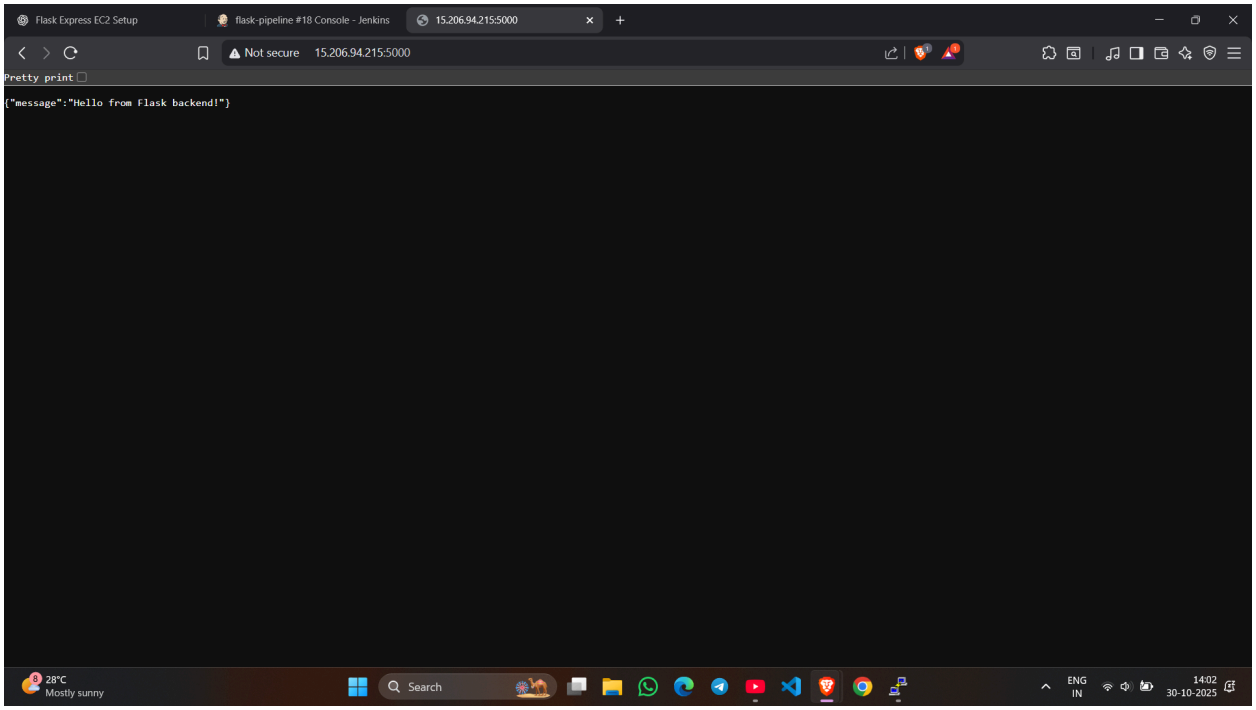


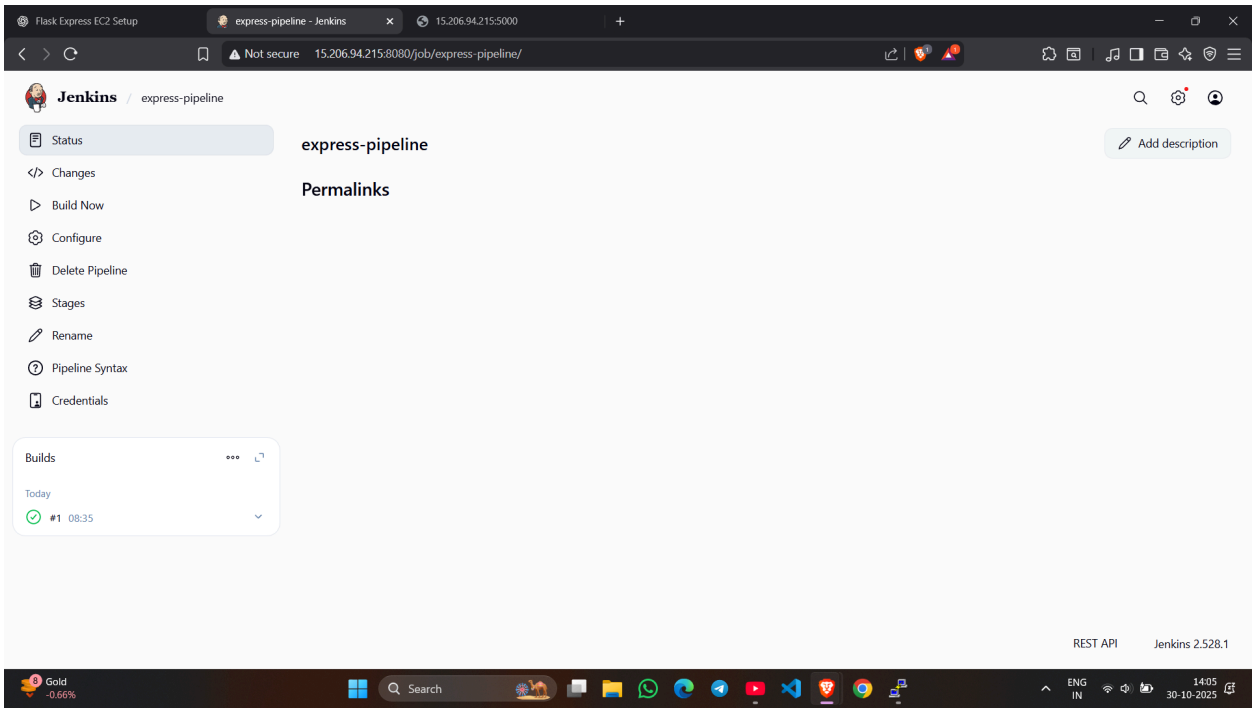

Placement Prep Course - MicroDegre
bhatadarsh/jenkins_task
SecurityGroup | EC2 | ap-south-1
Setup Wizard - Jenkins
Not secure 15.206.94.215:8080
Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.528.1
Rainy days ahead
26°C
Search
ENG
IN
11:36
30-10-2025

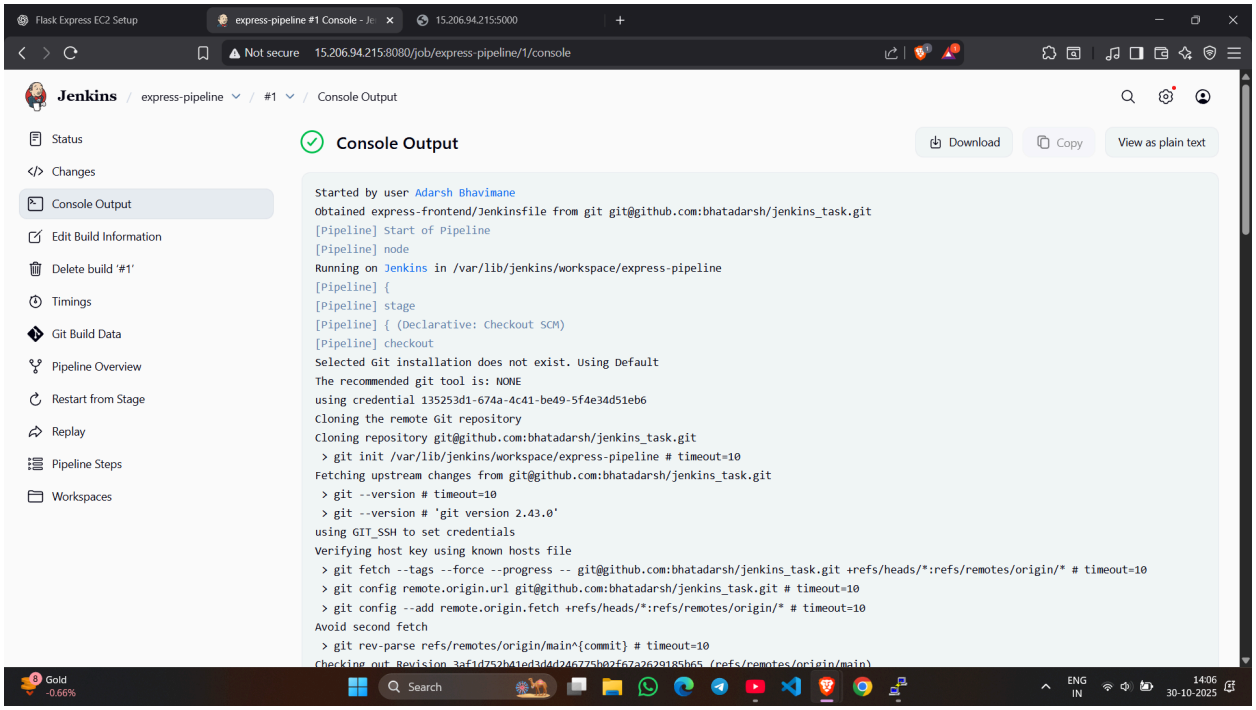

Placement Prep Course - MicroDegre
bhatadarsh/jenkins_task
SecurityGroup | EC2 | ap-south-1
New Item - Jenkins
Not secure 15.206.94.215:8080/view/all/newJob
Jenkins
All
New Item
New Item
Enter an item name
flask-pipeline
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
OK
26°C
Mostly sunny
Search
ENG
IN
11:46
30-10-2025

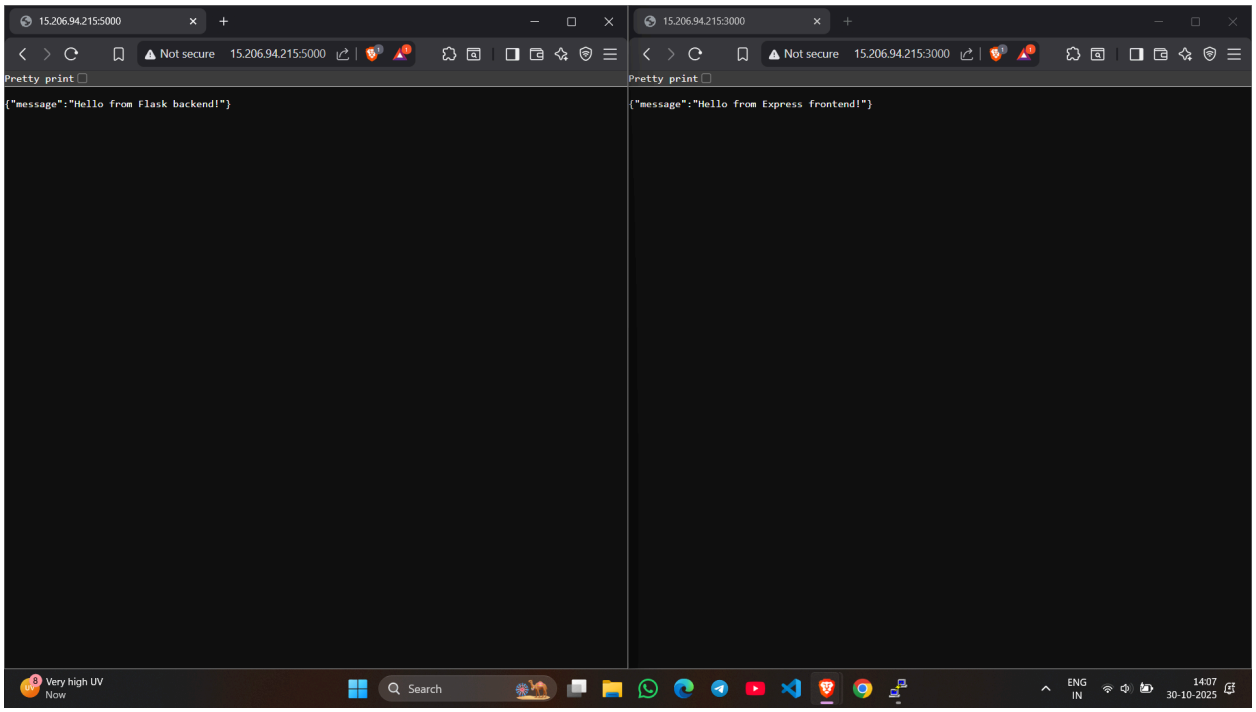

Flask Express EC2 Setup
flask-pipeline - Jenkins
Not secure 15.206.94.215:8080/job/flask-pipeline/
Jenkins / flask-pipeline
Builds
Filter
Today
#18 08:29
#17 08:26
#16 08:19
#15 08:12
#14 08:10
#13 08:08
#12 08:06
#11 08:04
#10 08:00
#9 07:57
#8 07:53
#7 07:51
#6 07:31
#5 07:30
#4 07:27
#3 07:18
#2 06:42
28°C
Mostly sunny
Search
ENG
IN
14:01
30-10-2025

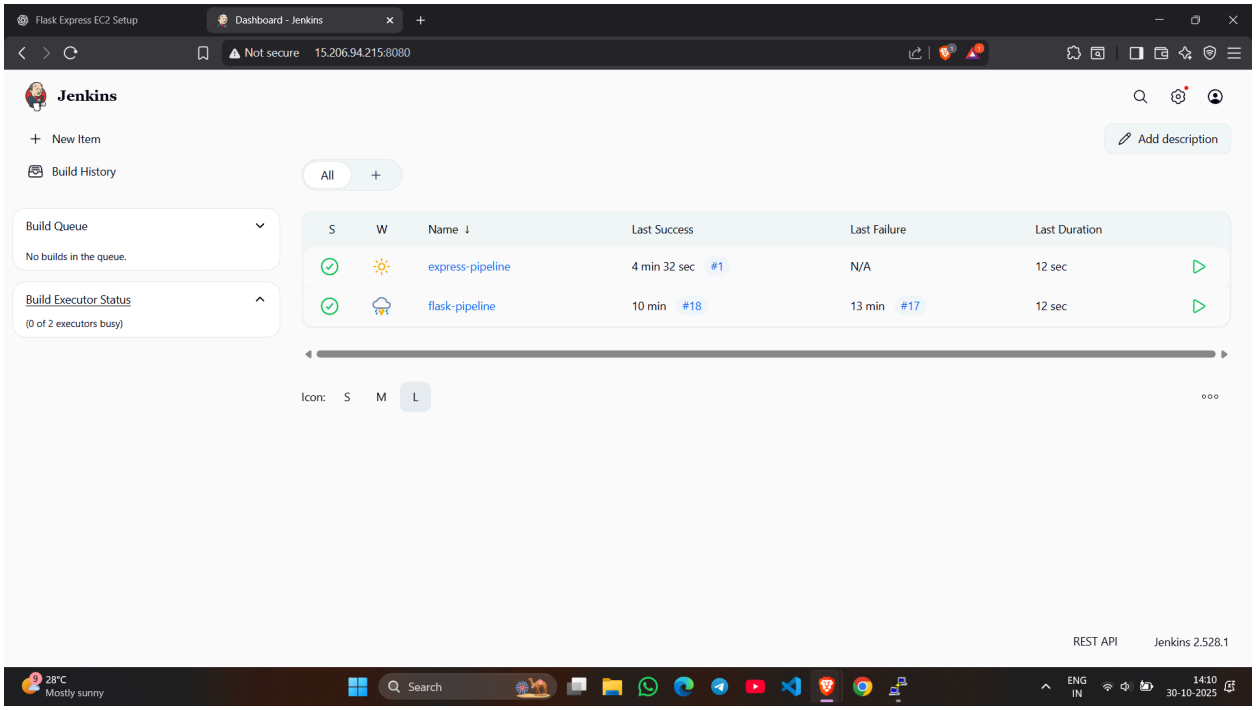

Flask Express EC2 Setup
flask-pipeline #18 Console - Jenk
Not secure 15.206.94.215:8080/job/flask-pipeline/18/console
Jenkins / flask-pipeline / #18 / Console Output
+ pm2 start /var/lib/jenkins/.local/bin/gunicorn --name flask-app -- --bind 0.0.0.0:5000 wsgi:app
[32m[PM2] [39mStarting /var/lib/jenkins/.local/bin/gunicorn in fork_mode (1 instance)
[32m[PM2] [39mDone.
id | name | namespace | version | mode | pid | uptime | ↺ | status | cpu | mem | user | watching
[1m[36m0[39m[22m | flask-app | default | N/A | [7m[1mfork[22m[27m | 1834 | 0s | 0 |
[32m[1monline[22m[39m | 0% | 30.1mb | [1mjenkins[22m | [90mdisabled[39m |
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deployment successful!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
REST API
Jenkins 2.528.1
28°C
Mostly sunny
Search
ENG
IN
14:01
30-10-2025

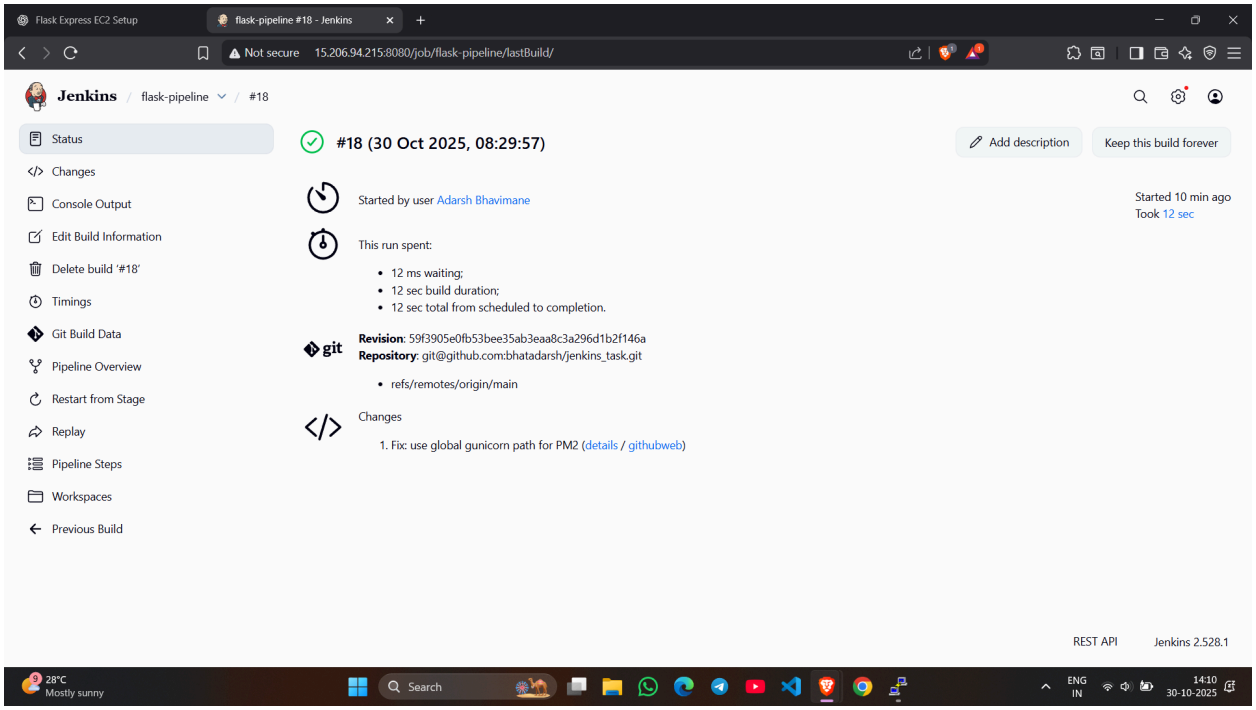

Flask Express EC2 Setup
flask-pipeline #18 Console - Jenkins
15.206.94.215:5000
Not secure 15.206.94.215:5000
Pretty print
{"message":"Hello from Flask backend!"}
28°C
Mostly sunny
Search
ENG
IN
14:02
30-10-2025

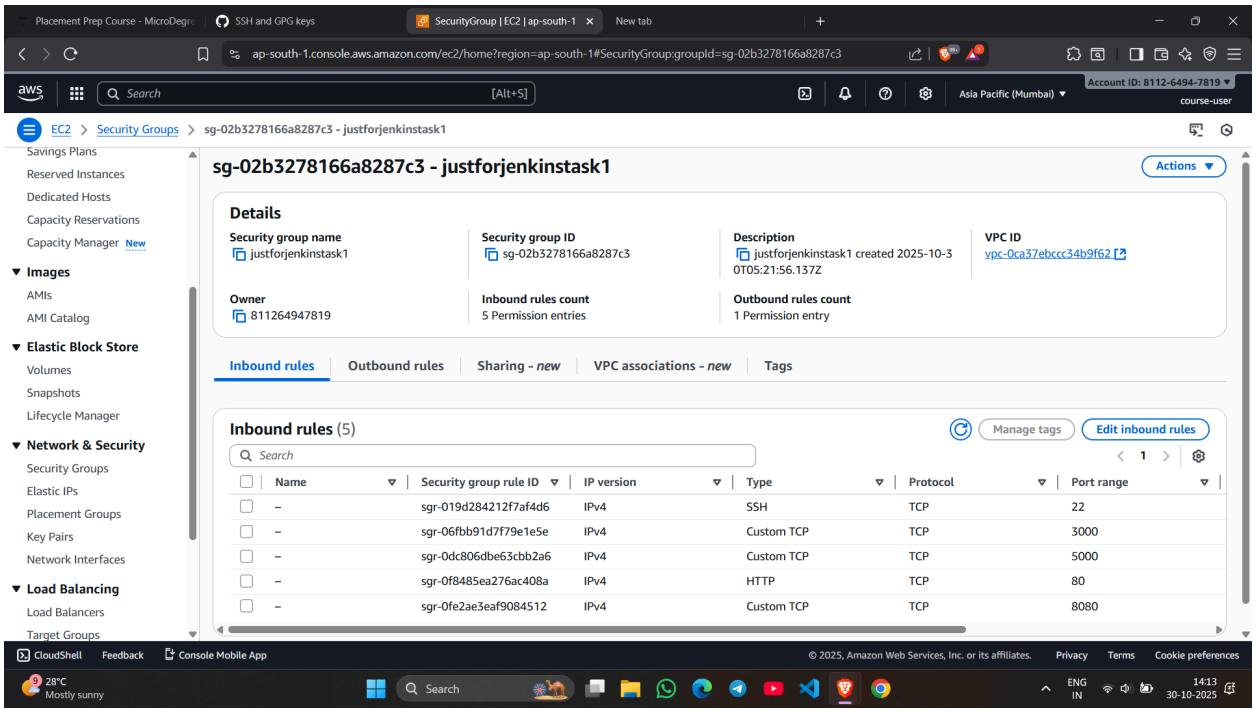

Flask Express EC2 Setup
express-pipeline - Jenkins
15.206.94.215:5000
Not secure 15.206.94.215:8080/job/express-pipeline/
Jenkins
express-pipeline
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Credentials
Builds
Today
#1 08:35
express-pipeline
Add description
Permalinks
REST API
Jenkins 2.528.1
Gold
-0.66%
Search
ENG
IN
14:05
30-10-2025


Flask Express EC2 Setup
express-pipeline #1 Console - Je
15.206.94.215:5000
Not secure 15.206.94.215:8080/job/express-pipeline/1/console
Jenkins / express-pipeline / #1 / Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
Git Build Data
Pipeline Overview
Restart from Stage
Replay
Pipeline Steps
Workspaces
Console Output
Download
Copy
View as plain text
Started by user Adarsh Bhavimane
Obtained express-frontend/Jenkinsfile from git git@github.com:bhatadarsh/jenkins_task.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/express-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential 135253d1-674a-4c41-be49-5f4e34d51eb6
Cloning the remote Git repository
Cloning repository git@github.com:bhatadarsh/jenkins_task.git
> git init /var/lib/jenkins/workspace/express-pipeline # timeout=10
Fetching upstream changes from git@github.com:bhatadarsh/jenkins_task.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT_SSH to set credentials
Verifying host key using known hosts file
> git fetch --tags --force --progress -- git@github.com:bhatadarsh/jenkins_task.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url git@github.com:bhatadarsh/jenkins_task.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Gold -0.66%
Search
ENG IN
14:06 30-10-2025


15.206.94.215:5000
Not secure 15.206.94.215:5000
Pretty print
{"message":"Hello from Flask backend!"}
15.206.94.215:3000
Not secure 15.206.94.215:3000
Pretty print
{"message":"Hello from Express frontend!"}
Very high UV Now
Search
ENG IN
14:07 30-10-2025


Flask Express EC2 Setup
Dashboard - Jenkins
Not secure 15.206.94.215:8080
Jenkins
New Item
Build History
Add description
All
Build Queue
No builds in the queue.
Build Executor Status
(0 of 2 executors busy)
S W Name ↓ Last Success Last Failure Last Duration
express-pipeline 4 min 32 sec #1 N/A 12 sec
flask-pipeline 10 min #18 13 min #17 12 sec
Icon: S M L
REST API Jenkins 2.528.1
28°C Mostly sunny
Search
ENG IN
14:10 30-10-2025





In this task, I implemented a CI/CD pipeline using Jenkins to automate the deployment of both Flask (backend) and Express (frontend) applications. I installed and configured Jenkins on my EC2 instance, added essential plugins such as Git, NodeJS, and Python, and created two separate Jenkins pipelines — one for each application. Each pipeline was designed to automatically pull the latest code from GitHub, install dependencies (pip install -r requirements.txt for Flask and npm install for Express), and restart the respective services using

PM2. This setup ensures that any new code pushed to GitHub can be automatically deployed without manual intervention, achieving a fully functional CI/CD workflow for both applications.