DevOps Pipeline Implementation using GitHub, Jenkins, and Docker

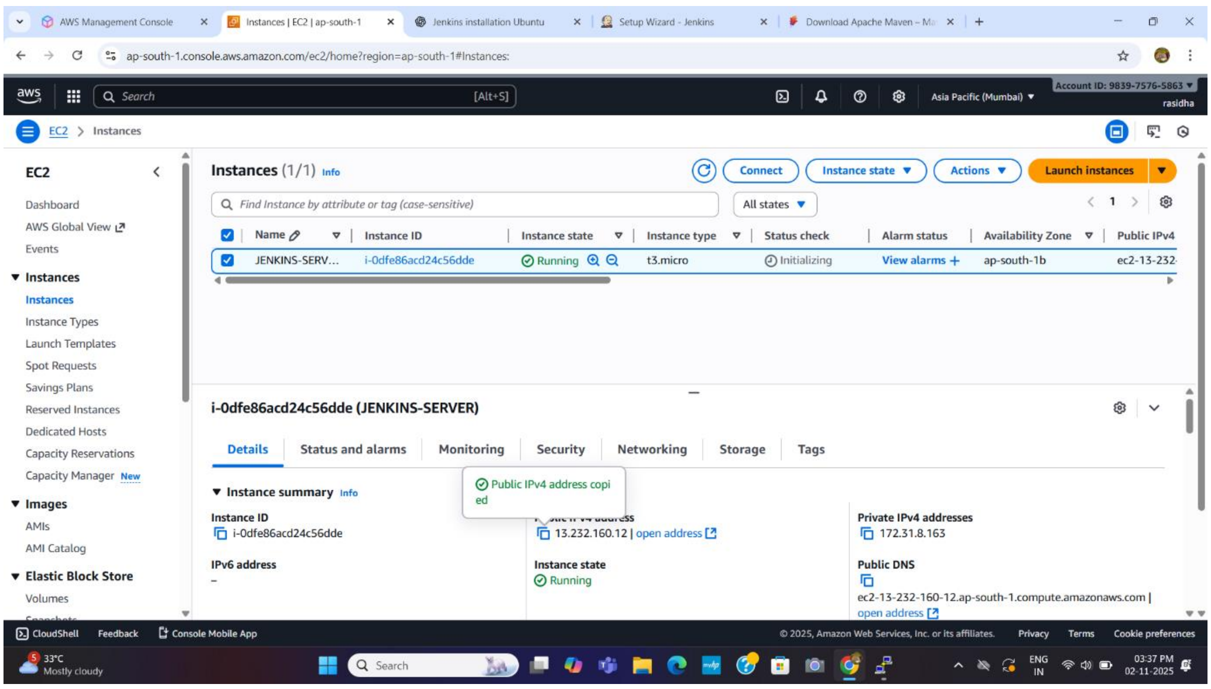
This project demonstrates a simple DevOps pipeline that automates the process of building and deploying an application using Jenkins and Docker. The system pulls the source code directly from a GitHub repository, builds a Docker image, and deploys it as a running container. Jenkins acts as the automation server, managing continuous integration and continuous deployment (CI/CD) efficiently. Docker ensures the application runs in a consistent environment regardless of system differences. The pipeline reduces manual effort, speeds up deployment, and improves reliability. It provides developers with faster feedback and easier version control integration. The entire setup promotes agile development practices and helps maintain stable releases. This project serves as a foundational example for beginners in DevOps automation.

Tools Used in the Project

1. **GitHub** – Used for storing and managing the project source code and Jenkinsfile.
2. **Jenkins** – Automates the build, test, and deployment process through CI/CD pipelines.
3. **Docker** – Packages the application and its dependencies into containers for consistent deployment.
4. **EC2 (Amazon Web Services)** – Hosts Jenkins and Docker to run the automation pipeline in the cloud.
5. **Nginx** – Acts as a lightweight web server to host and serve the deployed application.

Steps Involved:

Step 1: AWS EC2 Instance



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
JENKINS-SERV...	i-0dfe86acd24c56dde	Running	t3.micro	Initializing	View alarms +	ap-south-1b	ec2-13-232-

i-0dfe86acd24c56dde (JENKINS-SERVER)

Public IPv4 address copied

Instance summary

Instance ID: i-0dfe86acd24c56dde

Public IPv4 address: 13.232.160.12 | open address

Private IPv4 addresses: 172.31.8.163

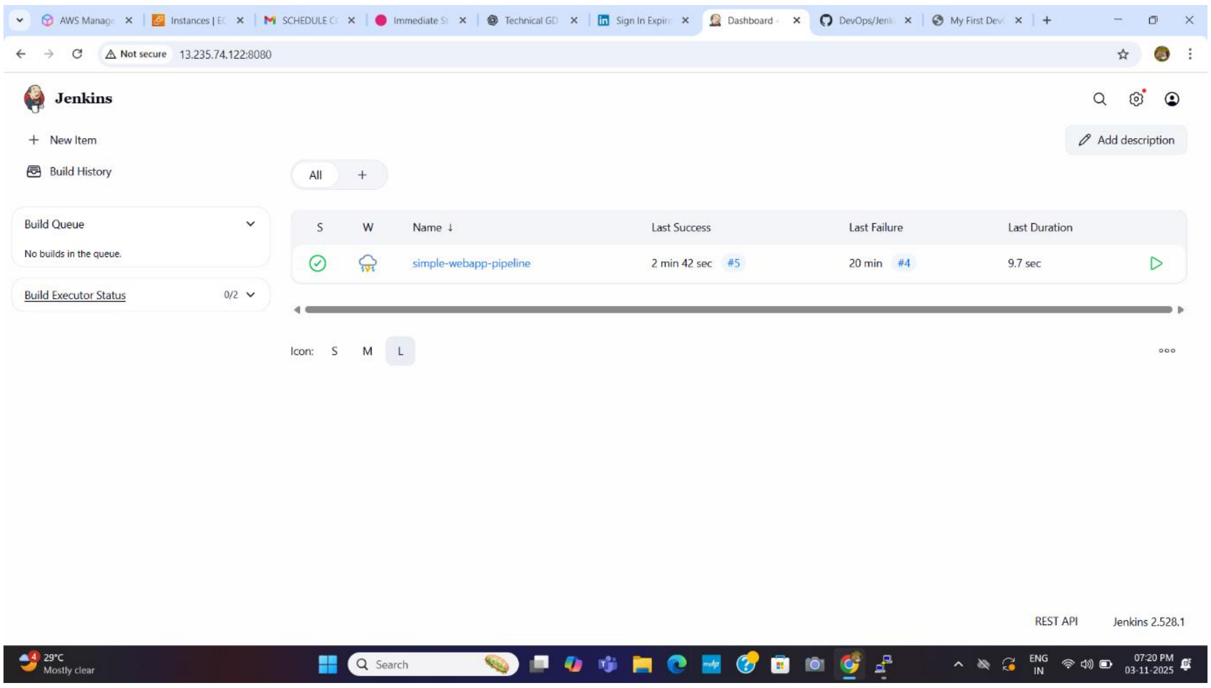
IPv6 address: –

Instance state: Running

Public DNS: ec2-13-232-160-12.ap-south-1.compute.amazonaws.com | open address

This figure shows the AWS EC2 instance used to host Jenkins and Docker. The instance acts as the main server where all DevOps tools are installed and integrated. It provides a virtual environment to execute the CI/CD pipeline, automate builds, and deploy the web application. The instance runs continuously, allowing seamless automation between code updates and deployment.

Step 2 : Jenkins Pipeline Creation

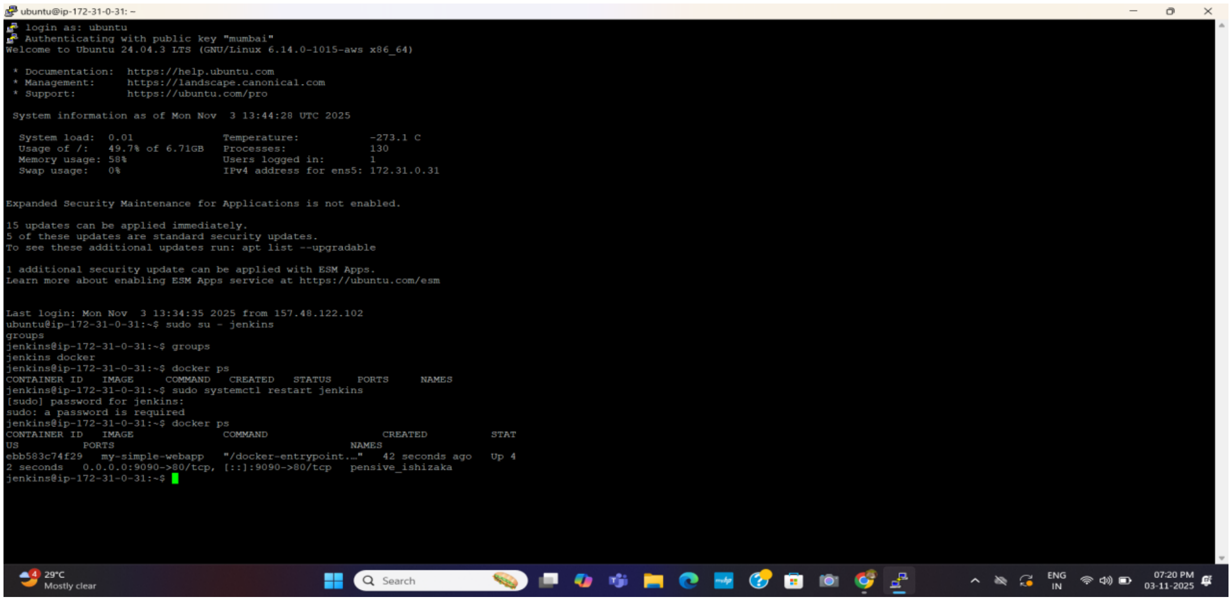


Jenkins — 13.235.74.122:8080

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		simple-webapp-pipeline	2 min 42 sec #5	20 min #4	9.7 sec

This figure represents the Jenkins dashboard, where the project pipeline named *"simple-webapp-pipeline"* is configured and executed. Jenkins automates the process of pulling code from GitHub, building a Docker image, and deploying the application. The green build status indicates that all stages of the pipeline have completed successfully.

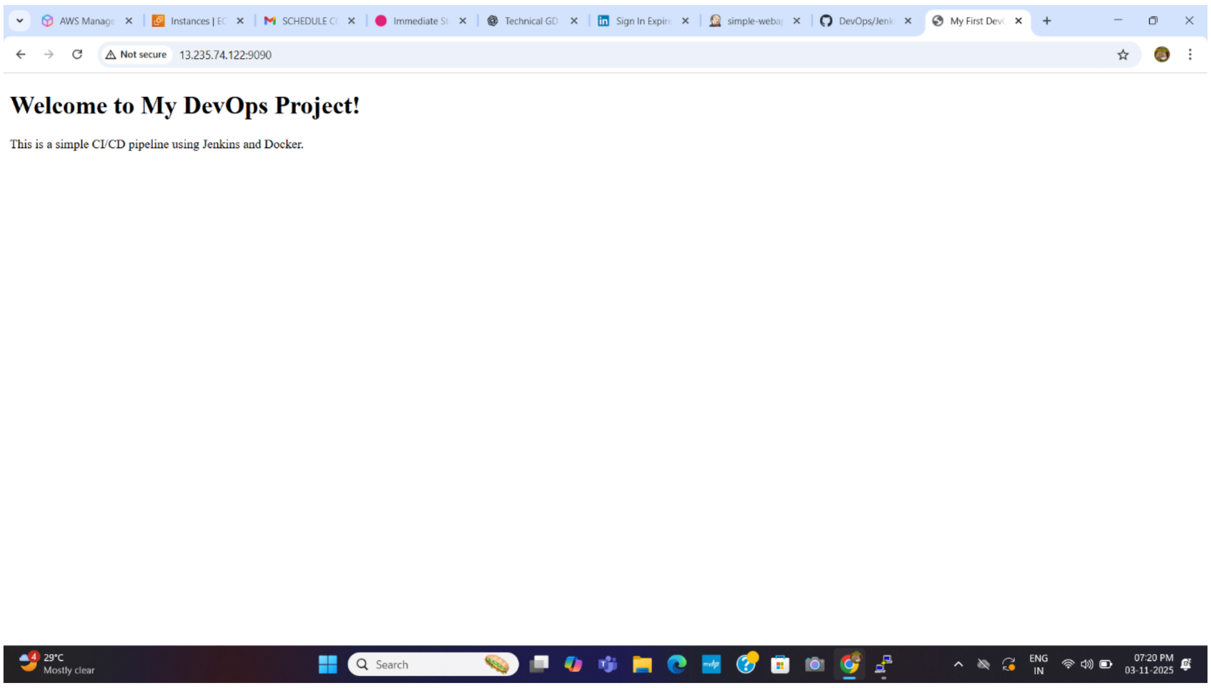
Step 3 : Docker Container Running



```
ubuntu@ip-172-31-0-31: ~
login as: ubuntu
Authenticating with public key "mumbai"
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Nov  3 13:44:28 UTC 2025

  System load:  0.01               Temperature:           -273.1 C
  Usage of /:   49.7% of 6.71GB    Processes:             130
  Memory usage: 58%                Users logged in:       1
  Swap usage:   0%                 IPv4 address for ens5: 172.31.0.31


Expanded Security Maintenance for Applications is not enabled.

15 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Mon Nov  3 13:34:35 2025 from 157.48.122.102
ubuntu@ip-172-31-0-31:~$ sudo su - jenkins
groups
jenkins@ip-172-31-0-31:~$ groups
jenkins docker
jenkins@ip-172-31-0-31:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
jenkins@ip-172-31-0-31:~$ sudo systemctl restart jenkins
[sudo] password for jenkins:
sudo: a password is required
jenkins@ip-172-31-0-31:~$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED          STAT
US          PORTS                                     NAMES
ebb583c74f29   my-simple-webapp   "/docker-entrypoint.…"   42 seconds ago   Up 4
2 seconds   0.0.0.0:9090->80/tcp, [::]:9090->80/tcp   pensive_ishizaka
jenkins@ip-172-31-0-31:~$
```

This figure shows the command-line interface (CLI) where Docker commands are executed on the EC2 instance. It confirms that the Docker image has been successfully built and the container is running. The command output displays the container ID and status, verifying that the web application is deployed and active inside the container.

Step 4 : Deployed Web Application



Welcome to My DevOps Project!

This is a simple CI/CD pipeline using Jenkins and Docker.

This figure shows the **final deployed web application** accessed through the EC2 public IP and port 9090. The webpage displays the message *"Welcome to My DevOps Project!"*, proving that the CI/CD pipeline successfully built, deployed, and hosted the application using Jenkins and Docker.