Jenkins CI/CD Pipeline for Flask Application

1. Project Overview

This project demonstrates the implementation of a Jenkins CI/CD pipeline for a Python Flask web application. The objective was to automate the process of building, testing, and deploying the application using Jenkins, GitHub, and Python tools.

2. Tools and Technologies Used

- Jenkins (CI/CD Automation)
- Git & GitHub (Source Code Management)
- Python 3 & Flask (Application)
- Pytest (Testing Framework)
- Linux (Ubuntu)

3. Project Structure

The repository is structured as follows:

```
jenkins_CI_CD_pipeline_for_flask_application/
├── Jenkinsfile
├── app.py
├── requirements.txt
├── test_app.py
├── README.md
└── documentation/
    ├── Jenkins_CICD_Overview.md
    ├── Pipeline_Stages.md
    ├── Setup_Prerequisites.md
    └── Screenshots.md
```

4. Jenkins Pipeline Explanation

The Jenkins pipeline is defined using a Jenkinsfile and consists of the following stages:

Checkout:
Jenkins pulls the latest source code from the GitHub repository.

Build:
A Python virtual environment is created and application dependencies are installed using pip.

Test:

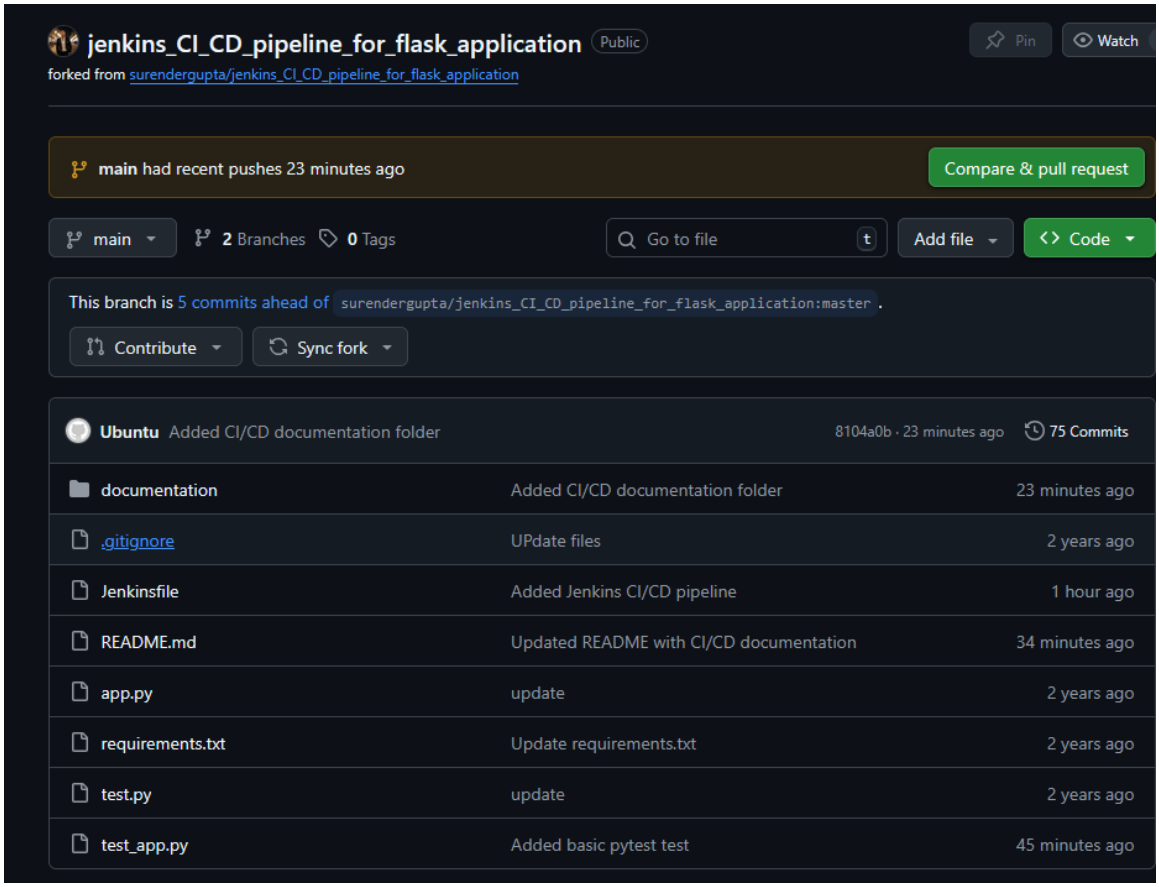
Unit tests are executed using pytest. The pipeline proceeds only if tests pass successfully.

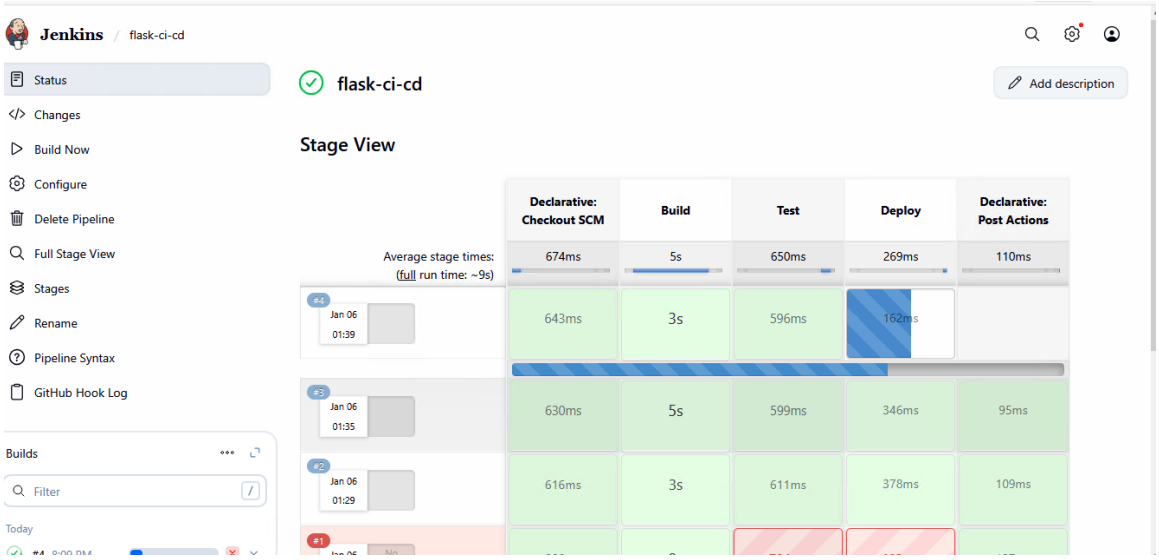
Deploy:
The Flask application is deployed to a staging environment after successful build and testing.

5. GitHub Webhook and Automation

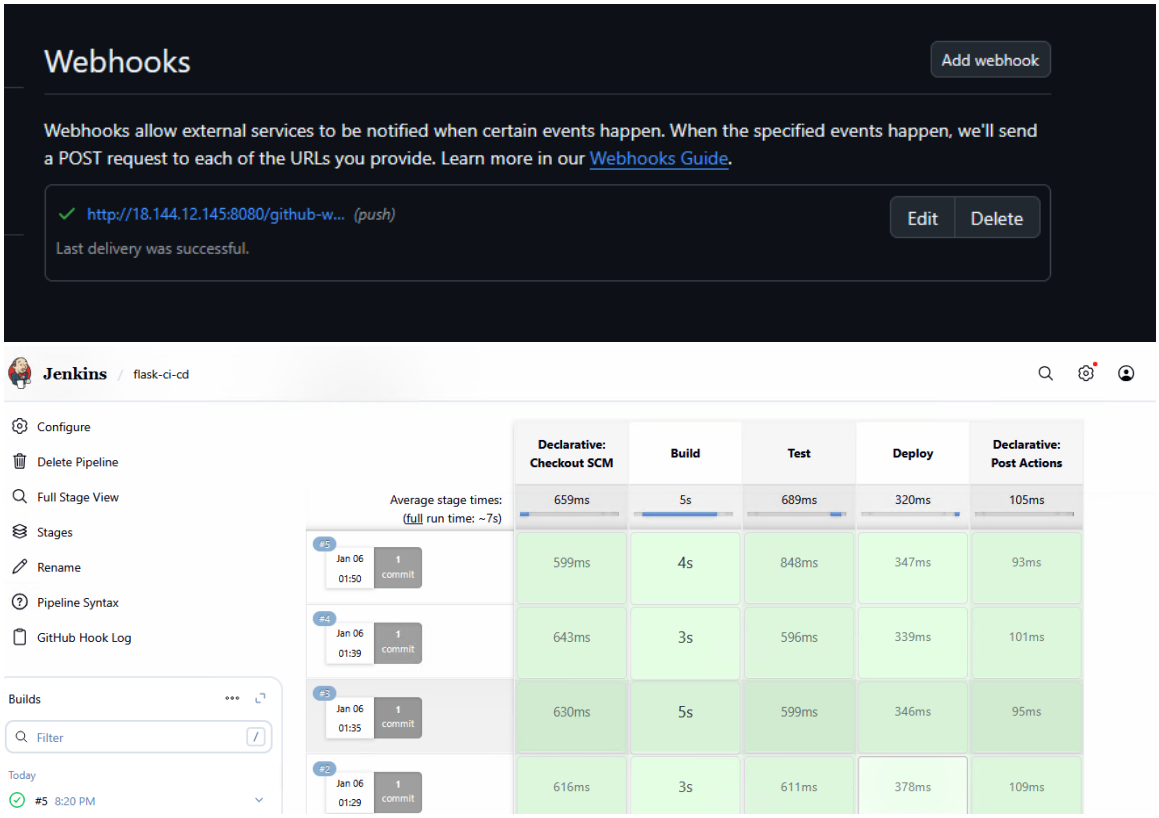
A GitHub webhook is configured to automatically trigger the Jenkins pipeline whenever changes are pushed to the main branch. This ensures continuous integration without manual intervention.

6. Screenshots




Jenkins / flask-ci-cd
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Stages
Rename
Pipeline Syntax
GitHub Hook Log
Builds
Filter
Today
flask-ci-cd
Add description
Stage View
Declarative: Checkout SCM
Build
Test
Deploy
Declarative: Post Actions
Average stage times:
(full run time: ~9s)
674ms
5s
650ms
269ms
110ms
#4
Jan 06
01:39
643ms
3s
596ms
162ms
#3
Jan 06
01:35
630ms
5s
599ms
346ms
95ms
#2
Jan 06
01:29
616ms
3s
611ms
378ms
109ms
#1


Jenkins / flask-ci-cd / #5 / Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#5'
Polling Log
Timings
Git Build Data
Pipeline Overview
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Download
Copy
View as plain text
Started by GitHub push by Rushiargade
Obtained Jenkinsfile from git https://github.com/Rushiargade/jenkins_CI_CD_pipeline_for_flask_application.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/flask-ci-cd
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/flask-ci-cd/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Rushiargade/jenkins_CI_CD_pipeline_for_flask_application.git #
timeout=10
Fetching upstream changes from https://github.com/Rushiargade/jenkins_CI_CD_pipeline_for_flask_application.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/Rushiargade/



7. Conclusion

This project successfully demonstrates a complete CI/CD workflow using Jenkins for a Flask application. Automation of build, test, and deployment improves reliability, reduces manual effort, and ensures faster delivery of applications following DevOps best practices.