Exercise 5: Pipeline Job - Deploy a Flask Application with Gunicorn

1. Objective: Automate the deployment of a Flask application with Gunicorn on Windows.

2. Steps:

o Create a Pipeline Job.

o Write a Jenkinsfile to:

▪ Clone a Flask application from GitHub.

▪ Set up a Python virtual environment.

▪ Install required packages using pip install -r requirements.txt.

▪ Configure and start the Gunicorn server:

gunicorn -b 127.0.0.1:8000 app:app

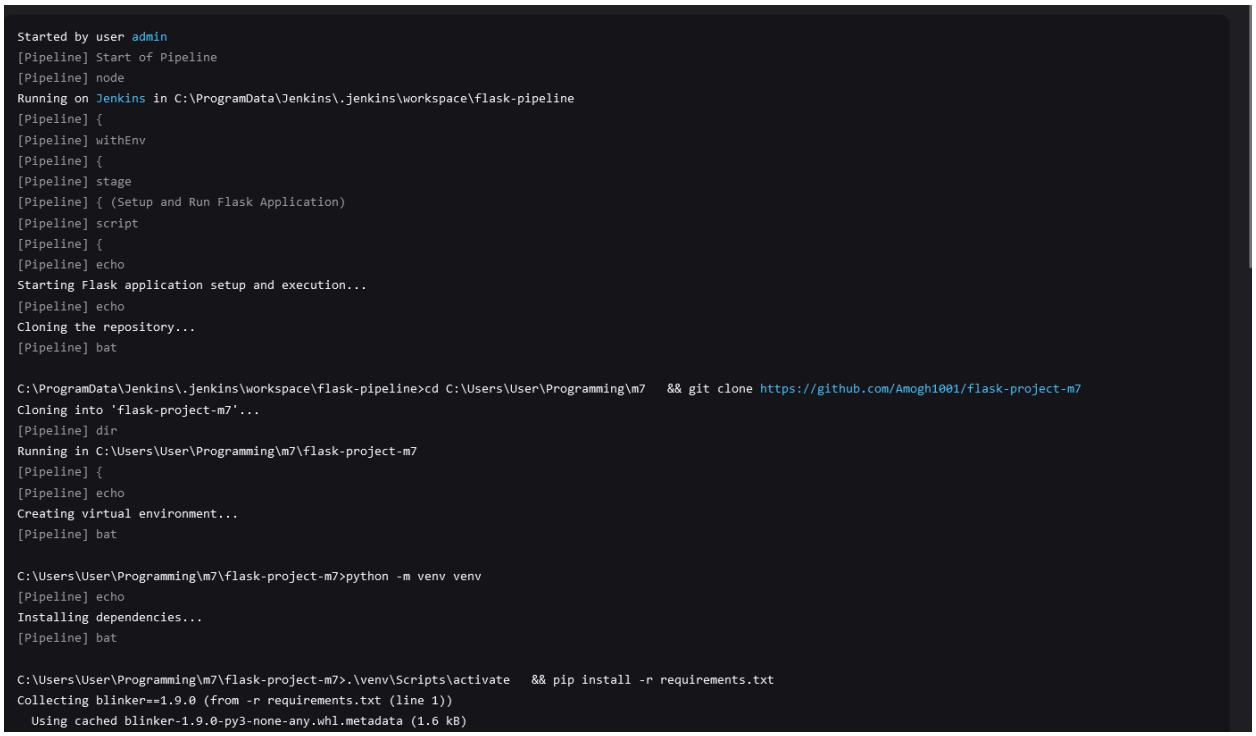
▪ Verify the deployment using a curl command in the pipeline.

o Add stages for:

▪ Unit tests using pytest.

▪ Post-deployment endpoint checks.

3. Task: Trigger the pipeline and ensure the Flask app is accessible on localhost



```
  Using cached flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting itsdangerous==2.2.0 (from -r requirements.txt (line 5))
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting Jinja2==3.1.5 (from -r requirements.txt (line 6))
  Using cached jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
Collecting MarkupSafe==3.0.2 (from -r requirements.txt (line 7))
  Using cached MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
Collecting packaging==24.2 (from -r requirements.txt (line 8))
  Downloading packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
Collecting waitress==3.0.2 (from -r requirements.txt (line 9))
  Downloading waitress-3.0.2-py3-none-any.whl.metadata (5.8 kB)
Collecting Werkzeug==3.1.3 (from -r requirements.txt (line 10))
  Using cached werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
Using cached click-8.1.8-py3-none-any.whl (98 kB)
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Using cached flask-3.1.0-py3-none-any.whl (102 kB)
Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Using cached jinja2-3.1.5-py3-none-any.whl (134 kB)
Using cached MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl (15 kB)
Downloading packaging-24.2-py3-none-any.whl (65 kB)
Downloading waitress-3.0.2-py3-none-any.whl (56 kB)
Using cached werkzeug-3.1.3-py3-none-any.whl (224 kB)
Installing collected packages: waitress, packaging, MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, Flask
Successfully installed Flask-3.1.0 Jinja2-3.1.5 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 colorama-0.4.6 itsdangerous-2.2.0 packaging-24.2
waitress-3.0.2
[Pipeline] echo
Starting the Flask application using waitress-serve...
[Pipeline] bat

C:\Users\User\Programming\m7\flask-project-m7>.\venv\Scripts\activate   && waitress-serve --listen=127.0.0.1:5000 app:app
INFO:waitress:Serving on http://127.0.0.1:5000
```

REST API Jenkins 2.479.3

```
pipeline {
    agent any
    environment {
        WORKSPACE = "C:\\Users\\User\\Programming\\m7"
        REPO_URL = "https://github.com/Amogh1001/flask-project-m7"
        CLONE_DIR = "${WORKSPACE}\\flask-project-m7"
        VENV_DIR = "${CLONE_DIR}\\venv"
    }
    stages {
        stage('Setup and Run Flask Application') {
            steps {
                script {
                    echo "Starting Flask application setup and execution..."

                    echo "Cloning the repository..."
                    bat "cd ${WORKSPACE} && git clone ${REPO_URL}"

                    dir(CLONE_DIR) {
                    }
                }
            }
        }
    }
    post {
        always {
            echo 'Pipeline execution completed.'
        }
        success {
            echo 'Flask application is running successfully!'
        }
        failure {
            echo 'Pipeline execution failed. Please check the logs for details.'
        }
    }
}
```