



This Jenkins pipeline defines a multi-stage CI/CD process for a Python project, consisting of stages for setup, linting, testing, coverage analysis, security scanning, and deployment.

In the setup stage, a Python virtual environment (`venv`) is created if it doesn’t already exist, and dependencies specified in `requirements.txt` are installed along with the `coverage` and `bandit` tools, which are used for coverage reporting and security scanning, respectively.

The lint stage enforces code standards by running `flake8` on the main script, `app.py`.

In the test stage, `pytest` is executed within the virtual environment to run the unit tests.

For coverage analysis, the pipeline uses `coverage.py` to generate a test coverage report, including an HTML version for detailed insights.

The security scan stage analyzes the codebase with `bandit`, excluding the virtual environment directory to ensure only project files are scanned.

The deployment stage copies project files to a specified deployment directory, starts the application in the background using `nohup`, and provides feedback on the deployment status.

Finally, a post-stage action cleans the workspace to remove any files generated during the pipeline's execution, ensuring a clean environment for subsequent runs.

This pipeline automates a comprehensive testing, security analysis, and deployment workflow tailored for Python projects in Jenkins.