

Python Code Quality & Security Implementation Guide

Overview

This document provides a comprehensive plan for implementing code quality, typing, and security checks for Python projects using GitHub Actions, from local development through CI/CD deployment.

Local Development (Pre-commit)

Pre-commit Hook Configuration

- Precommit hooks built in
 - id: trailing-whitespace
 - id: end-of-file-fixer
 - id: check-yaml
 - id: check-added-large-files
 - id: check-merge-conflict
 - id: debug-statements
- id: pyright Pyright is a powerful, fast, and feature-packed static type checker explicitly designed for Python. It helps to ensure code quality, catch errors early, and boost productivity through static type checking.
- id: ruff-check linter args: [-fix]
- id: ruff-format
- id: conventional-pre-commit ensures commit message is standardised
 - stages: [commit-msg]
- id: python-script LEAK DETECTION CUSTOM
 - **enables easy dev change if needed.**
 - name: Python Script
 - entry: python custom.py
 - language: python
 - stages: [pre-commit]
 - *We can add other checks here if needed*
- id: bandit

Conventional Commit Messages

- This provides structure around what has been committed and can be used for versioning.

CI/CD Pipeline (GitHub Actions)

The screenshot shows a GitHub Actions workflow run titled "UV Lint, Test, Matrix, Security #12". The workflow is triggered manually and has a status of "Success". It took 1m 4s to complete and produced 6 artifacts. The workflow is defined in the file `uv_run_lint_tests_calc_matrix_security.yaml` on the `workflow_dispatch` event.

The workflow consists of several jobs:

- Lint & Test**: 14s
- Run Tests and Coverage (3.13)**: 19s
- Run Tests and Coverage (3.12)**: 50s
- Run Tests and Coverage (3.11)**: 6s
- Run Tests and Coverage (3.10)**: 6s
- Security Scans With Bandit**: 19s
- CodeQL Analysis**: 50s
- Check GitLeaks for Secrets**: 6s

The workflow also includes a matrix for testing across different Python versions (3.9, 3.10, 3.11, 3.12) and Ubuntu versions (20.04, 22.04).

The "Check GitLeaks for Secrets" job summary shows "No leaks detected".

The workflow produced the following artifacts:

Name	Size	Digest
<code>gitleaks-results.sarif</code>	6.61 KB	<code>sha256:35576c815c859e01465a3b90f9fdea58fab5c96f21e7f717ab7f8ae66620e...</code>
<code>html-test-report-3.10</code>	8.8 KB	<code>sha256:a13269474681d088a05680e0e43f1cc786822ab49ecc272ab2c3f329a674...</code>
<code>html-test-report-3.11</code>	8.81 KB	<code>sha256:470692ab0a46d1d33e930390957de774877297c86120ff351f8631150b0de...</code>
<code>html-test-report-3.12</code>	8.81 KB	<code>sha256:18cc1f83090ea28a547a3bbc348d32b706121f314d94492b2b0d048f83f1d...</code>
<code>html-test-report-3.13</code>	8.81 KB	<code>sha256:8cc929a73fcb3d19420bd17e26ca49079423f953e5763bd2051aea26cfa3dc...</code>
<code>security-reports</code>	1.65 KB	<code>sha256:0fab6f324e66c998b745b612281e0b413e925f832551c8ae5baefecff376b...</code>

Build Matrix Strategy

- Set up matrix testing across Python versions (3.9, 3.10, 3.11, 3.12).
- Test across Ubuntu only.
- Install dependencies with caching for faster build times.
- Download Pytest-HTML reports.

Code and Security Quality Checks

- Run Ruff, MyPy/Pyright, Unit tests/Coverage, CodeQL, Bandit, Safety, Pip-audit in CI
- Scan for secrets and credentials in commit history and codebase
- Check for dependency vulnerabilities with GitHub's dependency review TO DO

Docker

← Docker Build Scan Push Calculator Application

✓ Docker Build Scan Push Calculator Application #2 Re-run all jobs

Summary

Jobs

- ✓ Application Setup & Validation
- ✓ Validate Docker Compose
- ✓ Build Docker Image
- ✓ Docker Image Security Scan
- ✓ Deploy to Docker Hub

Run details

- Usage
- Workflow file

Manually triggered 4 minutes ago Python-Test-Engineer 33224ff rule Success Total duration: 2m 8s Artifacts: 1

docker_validate_build_scan_deployyaml
on: workflow_dispatch

Application Setup & Val... 25s → Validate Docker Compose 13s → Build Docker Image 11s → Docker Image Security ... 41s → Deploy to Docker Hub 13s

Artifacts

Produced during runtime

Name	Size	Digest
security-scan-results	3.56 KB	sha256:a98342ac366cd586caabff7734287e4c89d79980bc354c84ffbfac3de602f20 📄

We use Trivy and Docker Scout, inbuilt in GitHub Actions, to scan Docker images for vulnerabilities and secrets, with a downloadable report.

We can run matrix Python version for the Dockerfile using arguments: `-build-arg PYTHON_VERSION=3.9` etc `02_CI/DockerfileMultiple` and `02/cicd_pipeline.yaml`.