Python Code Quality & Security Implementation Guide

Overview

The requirements were:

Create CI/CD workflow that included Python Version testing, Dependency Management, code quality and security as well as full deployment.

The following is an MVP that covers pre-commit to final deployment of a Docker image to my Docker Hub account.

Use of GitHub Advanced Security, Azure security and the use of environments and branch protection rules along with best practices will add/delete to this as appropriate.

My MVP contains the following:

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
- 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. MyPy can also be used.
- **Ruff** linting, formatting and fixing.
- **conventional pre-commit** ensures commit messages to provide helpful annotations as well as being used in automatic version numbering.
- **Custom python-script** to provide LEAK DETECTION CUSTOM. Home made enabling developers to adjust as needed to prevent friction. There are 3rd party services we can use.
- **Bandit** provides security checking.

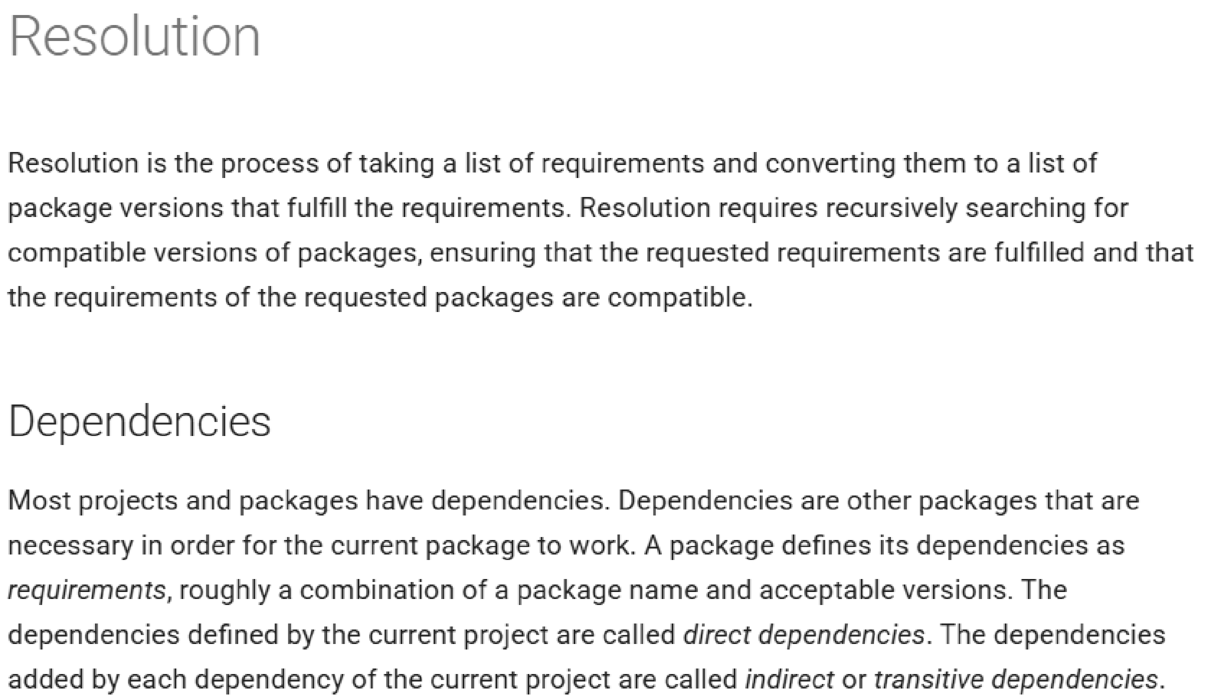
This is amendable.

Conventional Commit Messages

This provides structure around what has been committed and can be used for versioning. These commit messages can also be used to add information to the CHANGELOG.md file as we can search the Git database for filtered commit messages.

Dependency Management

For a given commit, UV enables a uv.lock file to be saved that pins all dependencies.

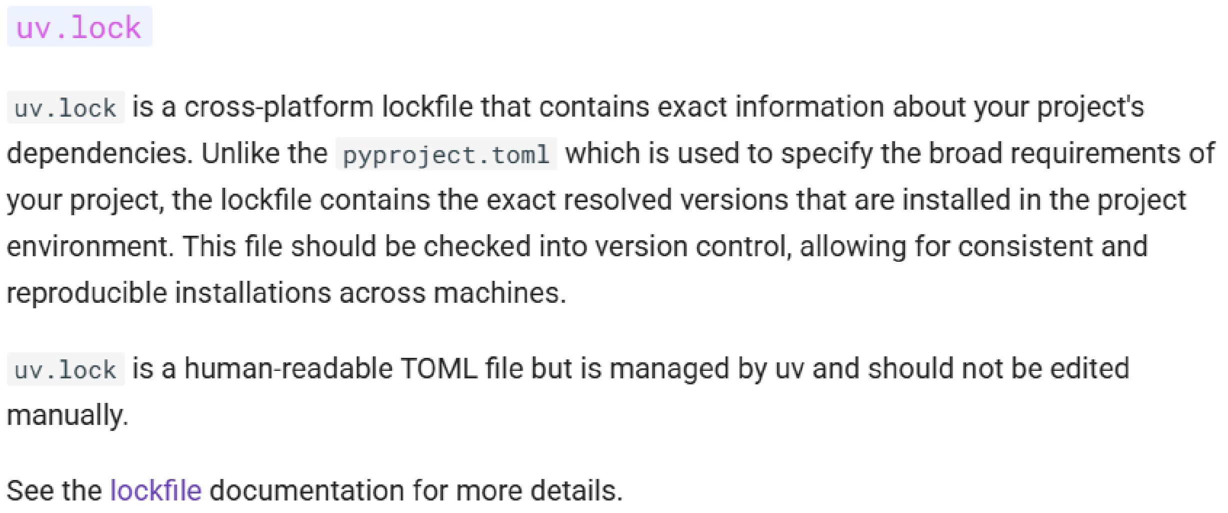


Resolution

Resolution is the process of taking a list of requirements and converting them to a list of package versions that fulfill the requirements. Resolution requires recursively searching for compatible versions of packages, ensuring that the requested requirements are fulfilled and that the requirements of the requested packages are compatible.

Dependencies

Most projects and packages have dependencies. Dependencies are other packages that are necessary in order for the current package to work. A package defines its dependencies as *requirements*, roughly a combination of a package name and acceptable versions. The dependencies defined by the current project are called *direct dependencies*. The dependencies added by each dependency of the current project are called *indirect* or *transitive dependencies*.

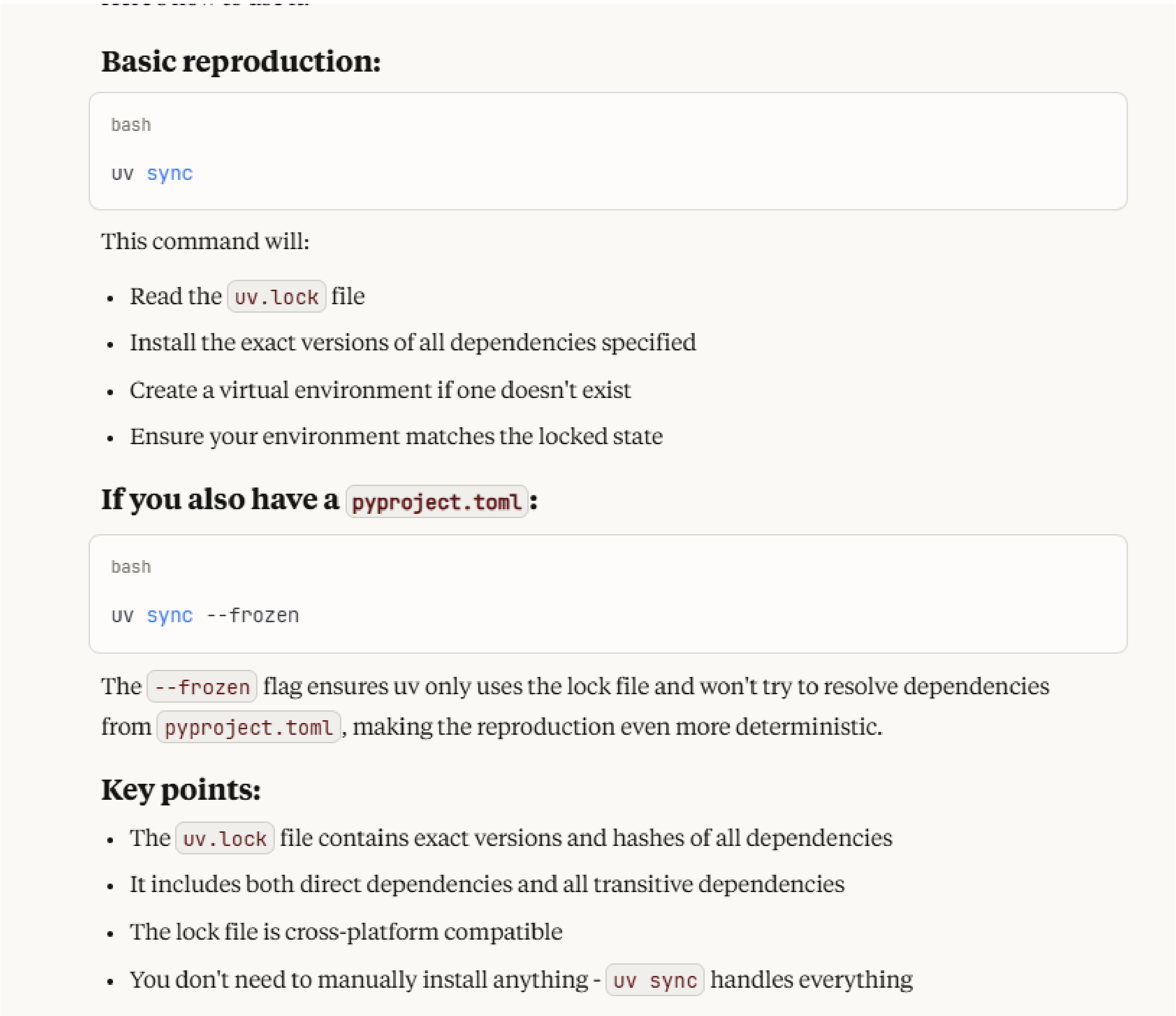


`uv.lock`

`uv.lock` is a cross-platform lockfile that contains exact information about your project's dependencies. Unlike the `pyproject.toml` which is used to specify the broad requirements of your project, the lockfile contains the exact resolved versions that are installed in the project environment. This file should be checked into version control, allowing for consistent and reproducible installations across machines.

`uv.lock` is a human-readable TOML file but is managed by uv and should not be edited manually.

See the lockfile documentation for more details.



Basic reproduction:

```bash
uv sync
```

This command will:

- Read the `uv.lock` file
- Install the exact versions of all dependencies specified
- Create a virtual environment if one doesn't exist
- Ensure your environment matches the locked state

If you also have a `pyproject.toml`:

```bash
uv sync --frozen
```

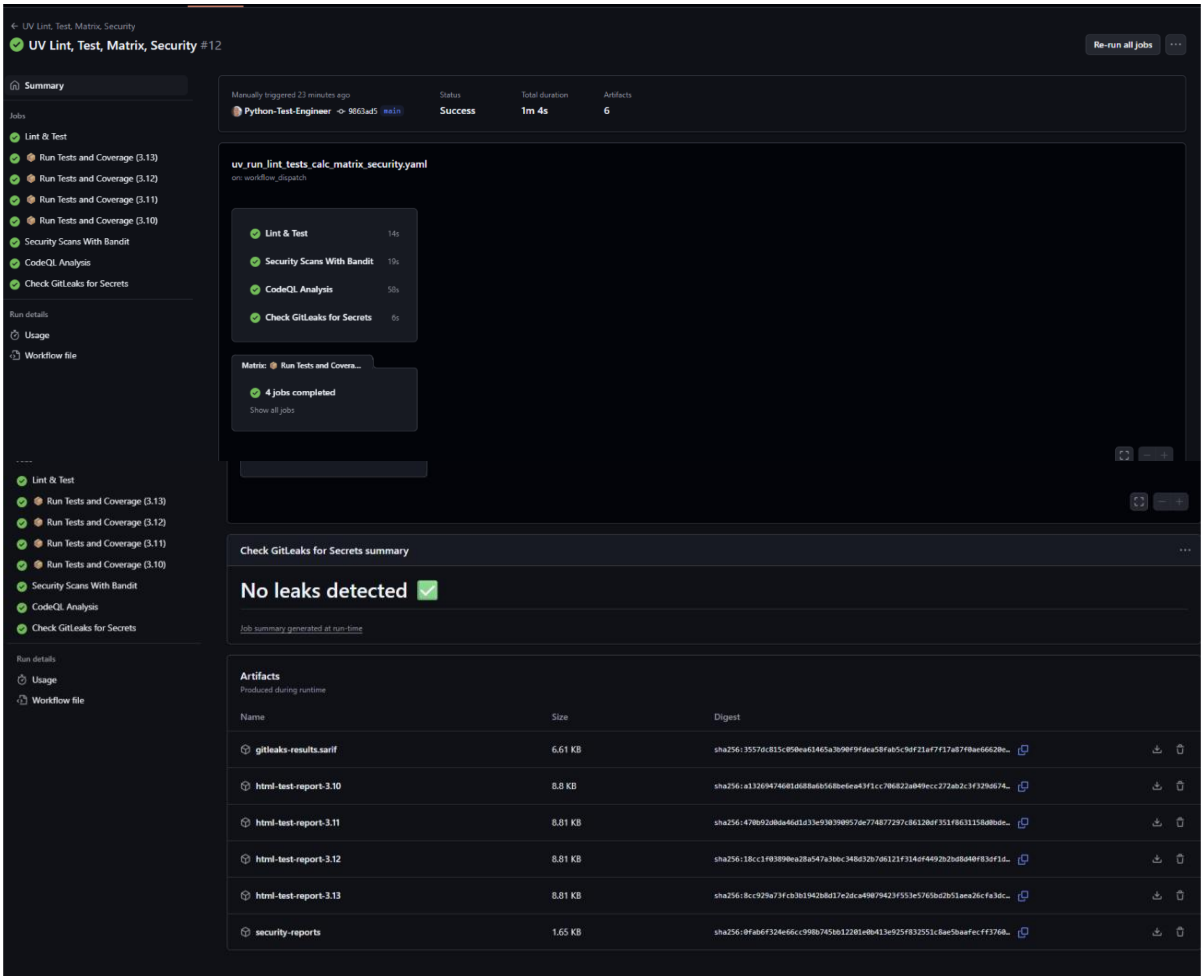
The `--frozen` flag ensures uv only uses the lock file and won't try to resolve dependencies from `pyproject.toml`, making the reproduction even more deterministic.

Key points:

- The `uv.lock` file contains exact versions and hashes of all dependencies
- It includes both direct dependencies and all transitive dependencies
- The lock file is cross-platform compatible
- You don't need to manually install anything - `uv sync` handles everything

UV is a more powerful resolver algorithm than pip and we can have a versioning system for uv.lock for each release.

CI/CD Pipeline (GitHub Actions)



← UV Lint, Test, Matrix, Security

UV Lint, Test, Matrix, Security #12

- Lint & Test
- Run Tests and Coverage (3.13)
- Run Tests and Coverage (3.12)
- Run Tests and Coverage (3.11)
- Run Tests and Coverage (3.10)
- Security Scans With Bandit
- CodeQL Analysis
- Check GitLeaks for Secrets

Manually triggered 23 minutes ago	Status	Total duration	Artifacts
Python-Test-Engineer -o- 9863ad5 main	Success	1m 4s	6

uv_run_lint_tests_calc_matrix_security.yaml
on: workflow_dispatch

Lint & Test	14s
Security Scans With Bandit	19s
CodeQL Analysis	58s
Check GitLeaks for Secrets	6s

Matrix: Run Tests and Covera...
4 jobs completed
Show all jobs

- Lint & Test
- Run Tests and Coverage (3.13)
- Run Tests and Coverage (3.12)
- Run Tests and Coverage (3.11)
- Run Tests and Coverage (3.10)
- Security Scans With Bandit
- CodeQL Analysis
- Check GitLeaks for Secrets

Run details
Usage
Workflow file

Check GitLeaks for Secrets summary

No leaks detected ✅

Job summary generated at run-time

Artifacts
Produced during runtime

Name	Size	Digest
gitleaks-results.sarif	6.61 KB	sha256:3557dc815c050ea61465a3b90f9fdea58fab5c9df21af7f17a87f0ae66620e...
html-test-report-3.10	8.8 KB	sha256:a13269474601d688a6b568be6ea43f1cc706822a049ecc272ab2c3f329d674...
html-test-report-3.11	8.81 KB	sha256:470b92d0da46d1d33e930390957de774877297c86120df351f8631158d0bde...
html-test-report-3.12	8.81 KB	sha256:18cc1f03890ea28a547a3bbc348d32b7d6121f314df4492b2bd8d40f83df1d...
html-test-report-3.13	8.81 KB	sha256:8cc929a73fcb3b1942b8d17e2dca49079423f553e5765bd2b51aea26cfa3dc...
security-reports	1.65 KB	sha256:0fab6f324e66cc998b745bb12201e0b413e925f832551c8ae5baafecff3760...

Linting, typing and formatting checks

Similar to those at the pre-commit stage.

Build Matrix Strategy

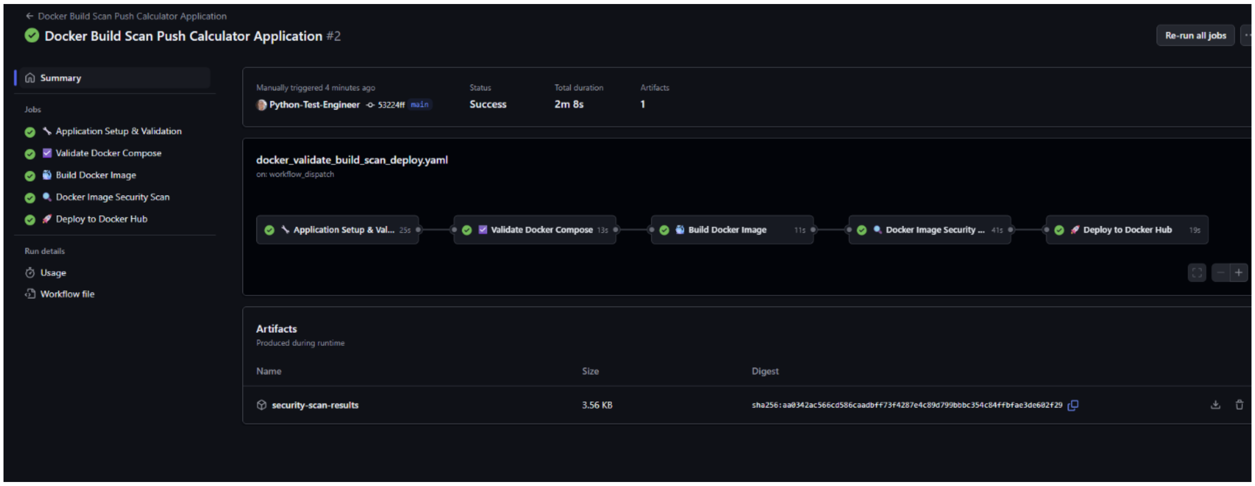
- Set up matrix testing across Python versions (3.10, 3.11, 3.12).
- Test across Ubuntu only.
- Install dependencies with caching for faster build times.
- Download Pytest-HTML reports and coverage reports.

Code and Security Quality Checks

- Unit tests/Coverage, CodeQL, Bandit, Safety, Pip-audit in CI.
- Scan for secrets and credentials in commit history and codebase.

Docker

The MVP can create Docker images across many Python versions if needed, scan for security and deploy to Azure if needed but uses my Docker Hub account in lieu of access to Azure.



← 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

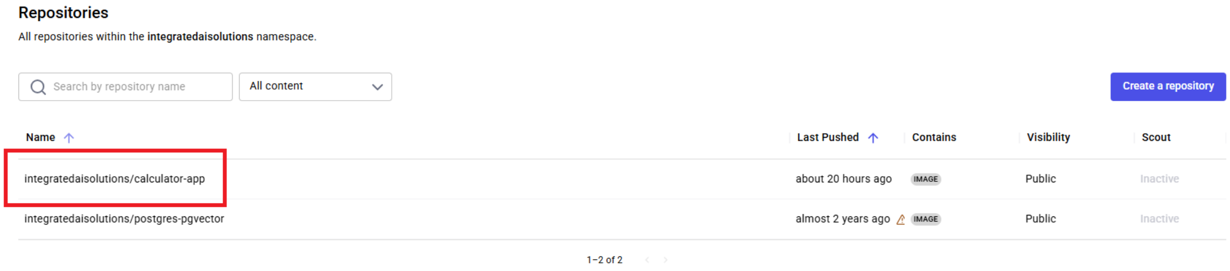
Manually triggered 4 minutes ago	Status	Total duration	Artifacts
Python-Test-Engineer -o- 53224ff main	Success	2m 8s	1

docker_validate_build_scan_deploy.yaml
on: workflow_dispatch

Application Setup & Val... 25s
Validate Docker Compose 13s
Build Docker Image 11s
Docker Image Security ... 41s
Deploy to Docker Hub 19s

Artifacts
Produced during runtime

Name	Size	Digest
security-scan-results	3.56 KB	sha256:aa0342ac566cd586caadbff73f4287e4c89d799bbbc354c84ffbfae3de602f29



Repositories

All repositories within the **integratedaisolutions** namespace.

Name ↑	Last Pushed ↑	Contains	Visibility	Scout
integratedaisolutions/calculator-app	about 20 hours ago	IMAGE	Public	Inactive
integratedaisolutions/postgres-pgvector	almost 2 years ago	IMAGE	Public	Inactive

1–2 of 2

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 matric Python version for the Dockerfile using arguments: –build-arg PYTHON_VERSION=3.10 etc. 02_CI/DockerfileMultiple and 02/cicd_pipeline.yaml.

Summary

This MVP is sufficient but not necessarily complete.

It covers the requirements stated at the beginning of the document.

Processes and protocols will be formulated later to provide the best possible security measures that will need to be monitored and adjusted as needed.

Work time

I would bill all this work as one day, subject to agreement.