**Case Study: Multi-job CI Pipeline with PR Status Checks & Artefact Upload**

**Scenario**

A development team wants a robust Continuous Integration (CI) pipeline that:

1. Runs linting, unit tests, and packaging in separate jobs.
2. Ensures that pull requests (PRs) can only be merged if all jobs pass.
3. Uploads build artefacts for later use.

**Goals**

* **Automated code quality checks** before merging.
* **Parallelized execution** of jobs for efficiency.
* **Artefact retention** for debugging and deployment.

**Workflow Design**

The workflow will have three jobs:

1. **Lint** – Checks code formatting and static analysis.
2. **Unit Test** – Runs test suite.
3. **Package** – Builds the application and uploads artefacts.

**Workflow YAML Example**

name: Multi-job CI

on:

pull\_request:

branches: [main]

jobs:

lint:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- name: Set up Python

uses: actions/setup-python@v5

with:

python-version: '3.10'

- run: pip install flake8

- run: flake8 .

unit\_tests:

runs-on: ubuntu-latest

needs: lint

steps:

- uses: actions/checkout@v4

- uses: actions/setup-python@v5

with:

python-version: '3.10'

- run: pip install -r requirements.txt

- run: pytest --junitxml=results.xml

- uses: actions/upload-artifact@v4

with:

name: test-results

path: results.xml

retention-days: 7

package:

runs-on: ubuntu-latest

needs: unit\_tests

steps:

- uses: actions/checkout@v4

- name: Build Package

run: |

mkdir dist

echo "Build Output" > dist/output.txt

- uses: actions/upload-artifact@v4

with:

name: build-package

path: dist

retention-days: 7

**PR Status Checks Configuration**

1. Go to **Settings → Branches → Add Rule**.
2. Select main branch.
3. Enable **Require status checks to pass before merging**.
4. Select lint, unit\_tests, and package jobs.

**Execution Flow**

1. **PR Created** → Workflow triggered.
2. **Lint Job** runs first.
3. **Unit Tests** run only if lint passes.
4. **Package** job runs after tests pass.
5. Build output and test results are uploaded as artefacts.

**Benefits**

* **Code Quality Enforcement**: PRs cannot be merged if lint or tests fail.
* **Efficiency**: Jobs run in sequence with clear dependencies.
* **Traceability**: Artefacts allow investigation of build/test results.

**Extensions**

* Add **matrix builds** to test across Python versions.
* Integrate **security scanning** in lint stage.
* Deploy artefacts automatically after PR merge.