**Step-by-Step Implementation**

**1. Project Structure**

Assume your project is named my\_etl.

bash

my\_etl/

├── my\_etl/ # Python package

│ ├── \_\_init\_\_.py

│ └── etl.py

├── tests/ # Unit tests

│ └── test\_etl.py

├── pyproject.toml # Build system config

├── setup.cfg # Metadata + lint/test config

├── setup.py # Setup script (if needed)

├── requirements.txt

├── .gitignore

├── .pre-commit-config.yaml

└── README.md

**2. Write Tests with pytest**

Install pytest and coverage:

bash

pip install pytest coverage

Sample test (tests/test\_etl.py):

python

from my\_etl import etl

def test\_transform\_data():

data = [{"name": "john"}, {"name": "EMMA"}]

result = etl.transform(data)

assert result == [{"name": "John"}, {"name": "Emma"}]

**3. Ensure > 80% Test Coverage**

Run:

bash

coverage run -m pytest

coverage report --fail-under=80

Optionally generate an HTML report:

bash

coverage html

**4. Lint with flake8 or ruff**

Install and configure linter:

bash

pip install flake8

In setup.cfg:

ini

[flake8]

max-line-length = 88

exclude = .git,\_\_pycache\_\_,build,dist

**5. Package as a Wheel**

Use pyproject.toml:

toml

[build-system]

requires = ["setuptools", "wheel"]

build-backend = "setuptools.build\_meta"

Use setup.cfg:

ini

[metadata]

name = my\_etl

version = 0.1.0

description = An ETL pipeline

author = Your Name

packages = find:

license = MIT

[options]

packages = find:

install\_requires =

requests

pandas

Build the wheel:

bash

python -m build

This creates .whl and .tar.gz files in the dist/ directory.

**6. Automate Lint and Tests with Pre-commit**

Install pre-commit:

bash

pip install pre-commit

Create .pre-commit-config.yaml:

yaml

repos:

- repo: https://github.com/pre-commit/pre-commit-hooks

rev: v4.5.0

hooks:

- id: trailing-whitespace

- id: end-of-file-fixer

- repo: https://github.com/psf/black

rev: 24.3.0

hooks:

- id: black

- repo: https://github.com/pycqa/flake8

rev: 7.0.0

hooks:

- id: flake8

- repo: local

hooks:

- id: pytest-coverage

name: Run Pytest with Coverage

entry: bash -c "coverage run -m pytest && coverage report --fail-under=80"

language: system

pass\_filenames: false

Install the hooks:

bash

pre-commit install

Now, every commit will run black, flake8, and pytest with coverage check.

**7. Run Everything**

Run tests and lint:

bash

coverage run -m pytest

flake8

Build the wheel:

bash

python -m build

Manually trigger pre-commit:

bash

pre-commit run --all-files

**Summary Checklist**

| **Item** | **Status** |
| --- | --- |
| Unit tests with pytest | Ready |
| > 80% coverage with coverage | Ready |
| Linting with flake8 or black | Ready |
| Build wheel using setuptools | Ready |
| Pre-commit hook automation | Ready |