

Development Workflow

Development Workflow

1. Branch off of `main`, name your branch with the following convention: `f/$FEATURE_NAME`, `b/$BUG_FIX` or `r/RELEASE_TAG`.
2. Make new changes in your branch, and add/modify relevant unit tests in `tests`. Please follow the styling guide in the next section.
3. Test your changes, once ready, open a PR for review. See below for detailed testing guide.
4. The PR will trigger pre-commit hooks and run GitHub actions that would check code style and linting, run unit tests, build documentation and Docker images. All GitHub actions need to pass (along with 2 approvals) to merge the PR.

Code Style and Linting

We use several tools to ensure code quality:

- **ruff** for linting and formatting
- **mypy** for type checking

You can run these tools with:

```
# Linting
uv run ruff check .

# Type checking
uv run mypy

# Format code
uv run ruff format .
```

!!! note "Pre-commit Hooks" The pre-commit hooks will automatically run these checks before each commit. If the hooks fail, you will need to fix the issues before you can commit.

Testing

Testing new changes involves 2 major steps: unit tests and integration tests.

Unit Tests

All new features and bug fixes should include tests. We use pytest for testing:

```
# Change to test directory
cd tests

# Run all tests
```

```
uv run pytest

# Run tests with coverage
uv run pytest --cov=vec_inf
```

Integration Tests

Integration tests are stored in a different repository: [vec-inf-maintenance](#)

1. Clone the repository and run `cd vec-inf-maintenance/deployment`
2. Follow the README to test model launch and inference

Revision #1

Created 2026-02-04 04:35:58 UTC

Updated 2026-02-04 05:07:52 UTC