

Branch Protection Configuration

This document outlines the recommended branch protection rules for the UtilityFog-Fractal-TreeOpen repository.

Main Branch Protection

The `main` branch should have the following protection rules:

Required Status Checks

- ☒ Require status checks to pass before merging
- ☒ Require branches to be up to date before merging
- Required checks:
 - `CI / verify`
 - `CodeQL Security Analysis / analyze (python)`
 - `Release Smoke Test / smoke-test`

Pull Request Requirements

- ☒ Require pull request reviews before merging
- Number of required reviewers: **1**
- ☒ Dismiss stale reviews when new commits are pushed
- ☒ Require review from code owners (if CODEOWNERS file exists)

Additional Restrictions

- ☒ Restrict pushes that create files larger than 100 MB
- ☒ Require signed commits
- ☒ Require linear history
- ☒ Include administrators in restrictions

Branch Deletion

- ☒ Allow force pushes
- ☒ Allow deletions

Release Branch Protection (release/*)


Release branches should have similar but slightly relaxed rules:

Required Status Checks



- ☒ Require status checks to pass before merging
- Required checks:
 - `CI / verify`
 - `Release Smoke Test / smoke-test`

Pull Request Requirements

- ☒ Require pull request reviews before merging
- Number of required reviewers: **1**

-  Dismiss stale reviews when new commits are pushed

Additional Restrictions

-  Restrict pushes that create files larger than 100 MB
-  Require linear history

Implementation

To apply these rules, repository administrators can:

1. Go to Settings → Branches
2. Add branch protection rule for `main`
3. Add branch protection rule for `release/*`
4. Configure the settings as outlined above

Alternatively, use the GitHub API or terraform for automated configuration.

Notes

- These rules ensure code quality and security
- They prevent accidental force pushes and deletions
- They require peer review for all changes
- They ensure CI passes before merging
- They maintain a clean, linear git history