GitHub Branch Protection Setup Guide

1. Overview

What is Branch Protection?

> Branch Protection is a GitHub feature that prevents direct changes to critical branches (like `main` or `dev`). It enforces code review workflows by requiring pull requests and approvals before any code can be merged.

Why do we need it?

Problem: Human Error Risk

- Developers can accidentally push directly to `main` (production)
- Untested code can break the live application

- No review process = potential bugs slip through

Solution: Branch Protection

- Blocks direct pushes to protected branches
- Requires Pull Request + Approval before merge
- Ensures all code is reviewed before reaching production

Benefits by Role

For PM (Anshita):

- Visibility: All changes go through PR review before production
- Control: At least 1 reviewer's approval required before any merge
- Accountability: Clear audit trail of who approved what
- Protection: Direct push to main blocked (forces PR workflow)

For Developers:

- Clear workflow: Always create PR, never push directly
- No accidental production breaks
- Code review feedback improves code quality

For the Team:

- Better collaboration: PR comments enable discussion
- Peace of mind: Production is protected from human error
- Consistent workflow: System automatically guides everyone through the same PR process
- Extensible workflow: Can integrate automated tests, linting, or CI checks into the PR process

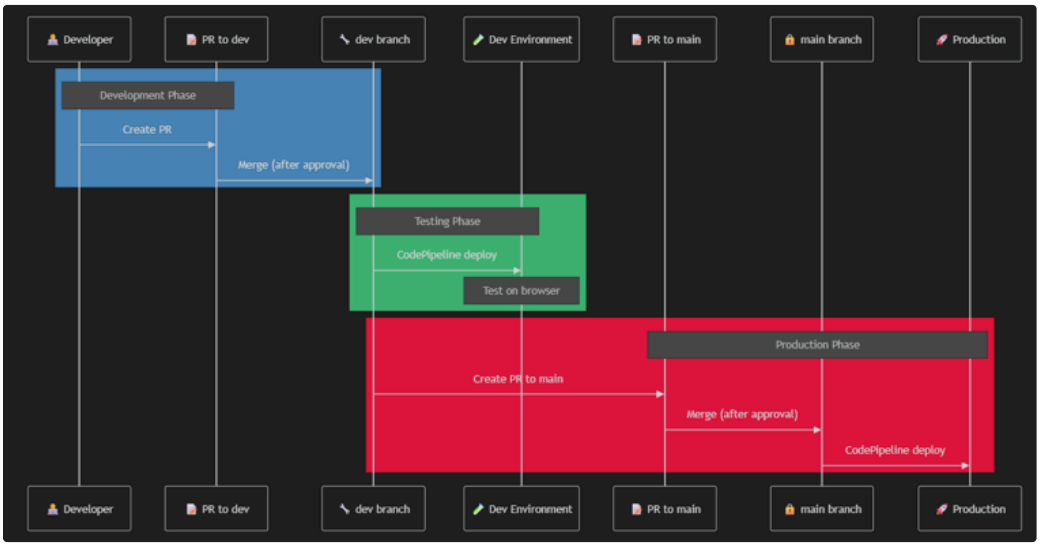
2. Prerequisites

Required

- GitHub Pro ($4/month) or GitHub Team plan
- Free plan does not support Branch Protection on private repositories

3. Workflow with Main Branch Protected

- Before: Anyone can push to main
- After: PR + Approval required

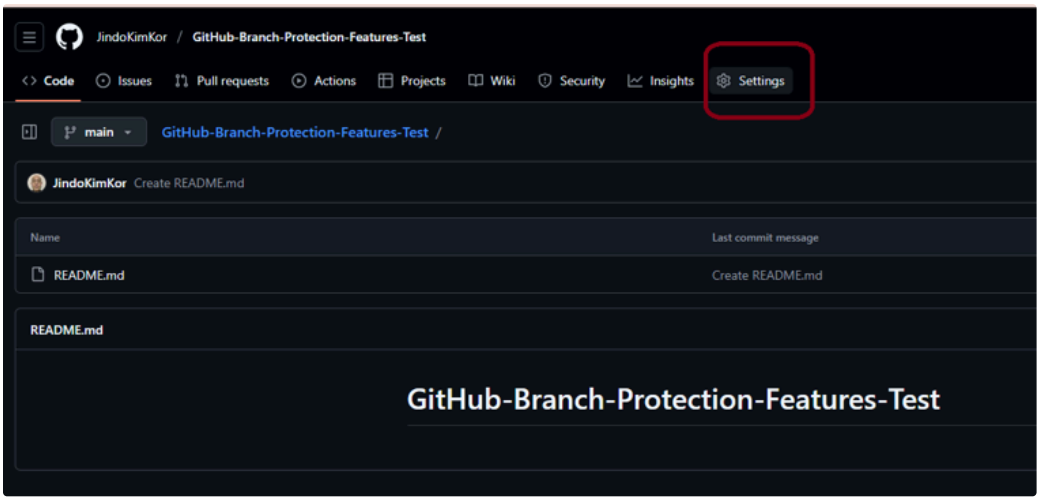


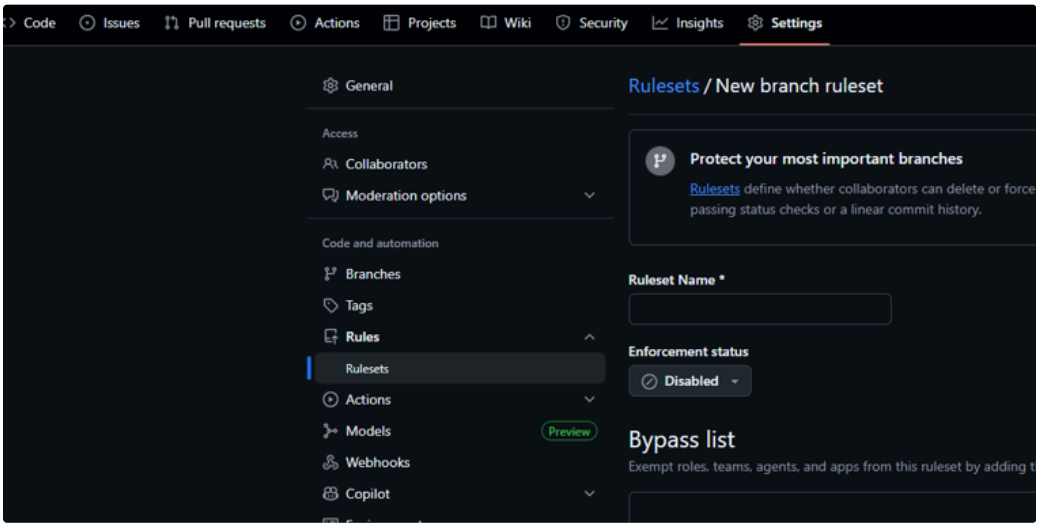
> ⚠️ **Note:** This document covers **main branch protection only** (Production Phase).
>
> Protecting dev branch is optional and can be configured later based on team needs. Adding approval requirements to dev may slow down the development flow, so it's recommended to start with main protection first and evaluate dev protection after observing the workflow.

4. Setup Guide (with Screenshots)

4.1 Access Repository Settings

1. Go to your repository on GitHub
2. Click Settings tab
3. Navigate to Rules → Rulesets
4. Click New ruleset → New branch ruleset





4.2 Create main Branch Ruleset

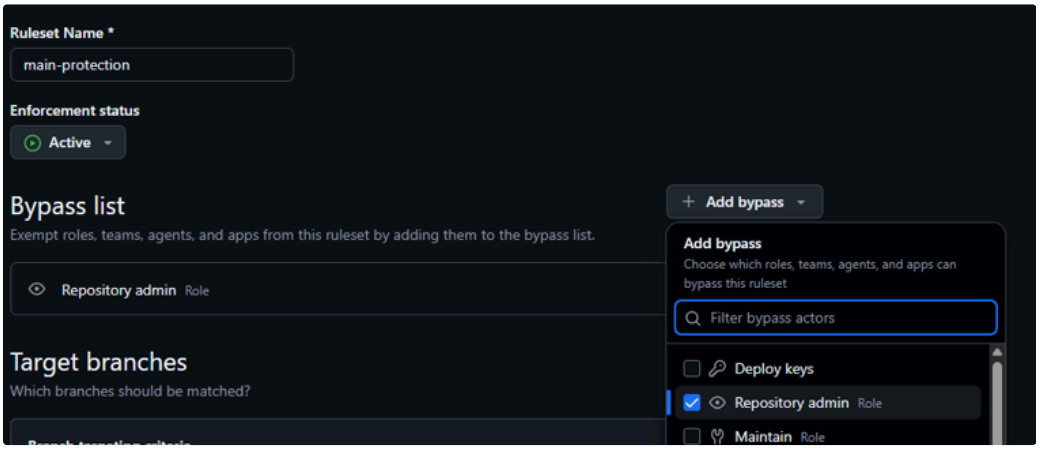
Basic Settings

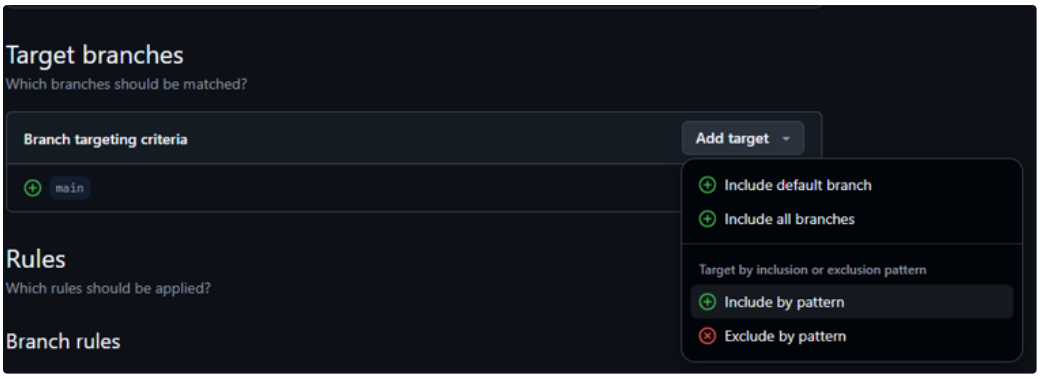
- Ruleset Name: main-protection
- Bypass list: Repository admin → Anshita
- Enforcement Status: Active
- Target Branches: Add Target → Include by pattern → main

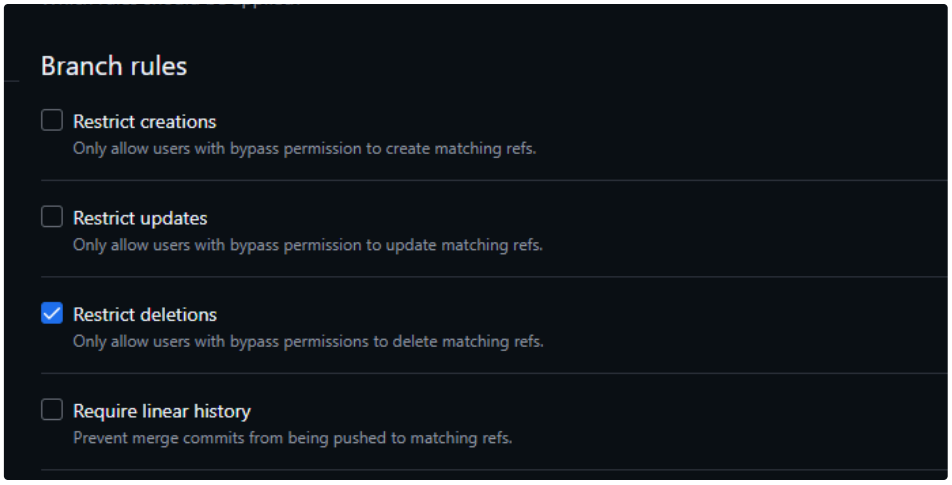
Rules to enable

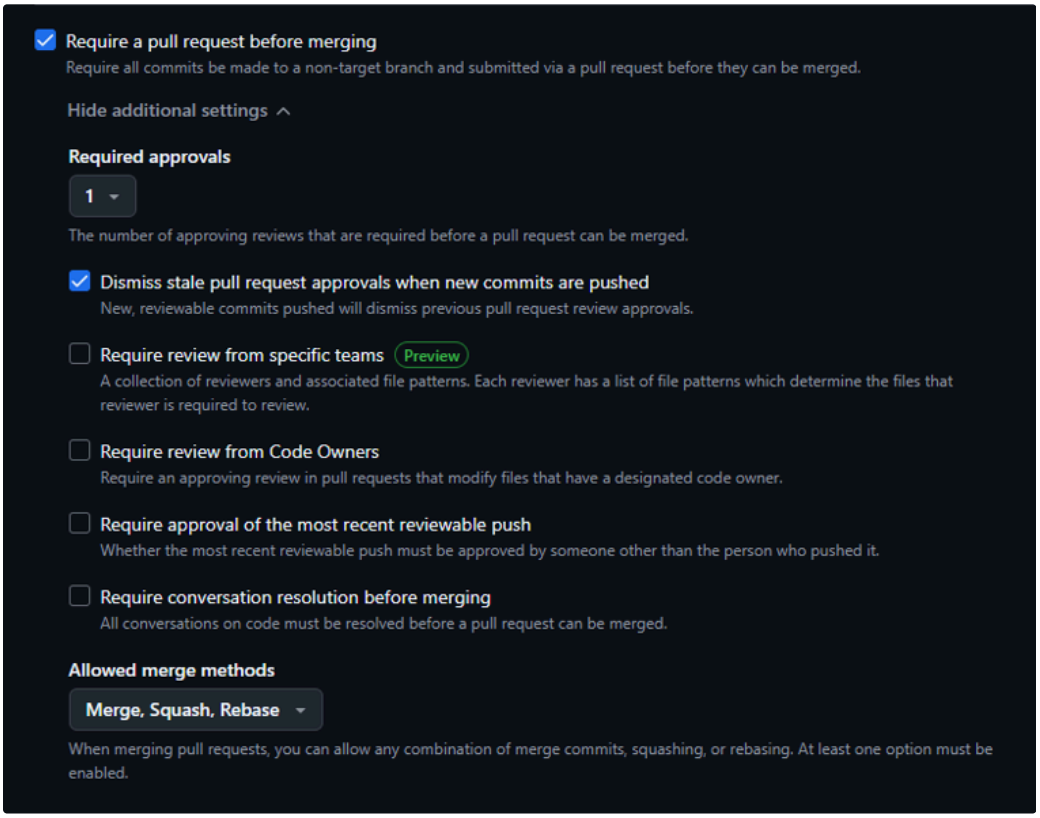
- Restrict deletion:
  - Prevent branch from being deleted

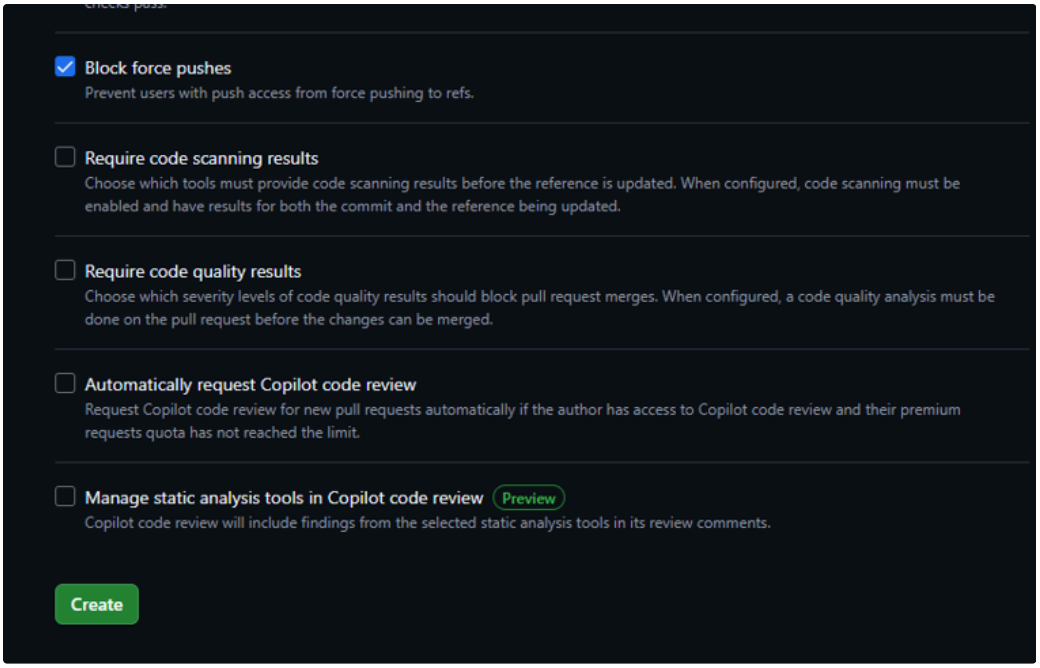
- Require a pull request before merging
  - No direct push allowed to main, even if anyone make a change on GitHub directly, except Anshita
  - Required approvals: 1
    - At least 1 reviewers must approve
      - → Developers can not merge without an approval
    - Dismiss stale approvals
      - → If changes are made to an approved PR, the approval is dismissed. This ensures no unreviewed code gets merged.
- Block force pushes
  - Prevent history rewriting






Branch rules
Restrict creations
Only allow users with bypass permission to create matching refs.
Restrict updates
Only allow users with bypass permission to update matching refs.
Restrict deletions
Only allow users with bypass permissions to delete matching refs.
Require linear history
Prevent merge commits from being pushed to matching refs.


Require a pull request before merging
Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.
Hide additional settings
Required approvals
1
The number of approving reviews that are required before a pull request can be merged.
Dismiss stale pull request approvals when new commits are pushed
New, reviewable commits pushed will dismiss previous pull request review approvals.
Require review from specific teams
Preview
A collection of reviewers and associated file patterns. Each reviewer has a list of file patterns which determine the files that reviewer is required to review.
Require review from Code Owners
Require an approving review in pull requests that modify files that have a designated code owner.
Require approval of the most recent reviewable push
Whether the most recent reviewable push must be approved by someone other than the person who pushed it.
Require conversation resolution before merging
All conversations on code must be resolved before a pull request can be merged.
Allowed merge methods
Merge, Squash, Rebase
When merging pull requests, you can allow any combination of merge commits, squashing, or rebasing. At least one option must be enabled.



4.3 (Optional) Create dev Branch Ruleset

> Protecting dev branch is optional. Since dev environment mistakes don't affect production, you may want to keep dev branch unprotected for faster iteration. Decide based on team workflow needs.

If you choose to protect dev branch, you have options:

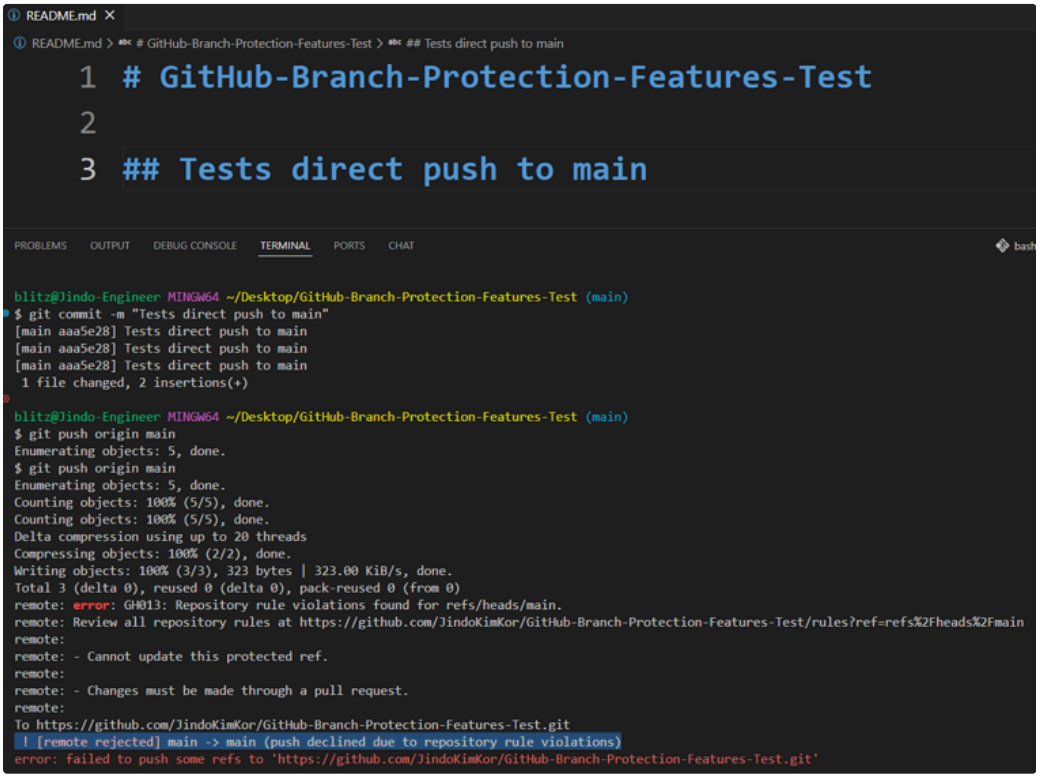
- Option A: Same as main (strict)
  - Same rules as main branch
  - All PRs require approval
- Option B: PR required, no approval
  - Ruleset name: `dev-protection`
  - Target branches: `dev`
  - Rules:
    - Require a pull request before merging
    - Required approvals: 0
    - Block force pushes

> This could ensure that code changes are tracked via PRs without slowing down development.

5. Validate main-protection Ruleset

> **After creating the ruleset, verify it works correctly:**

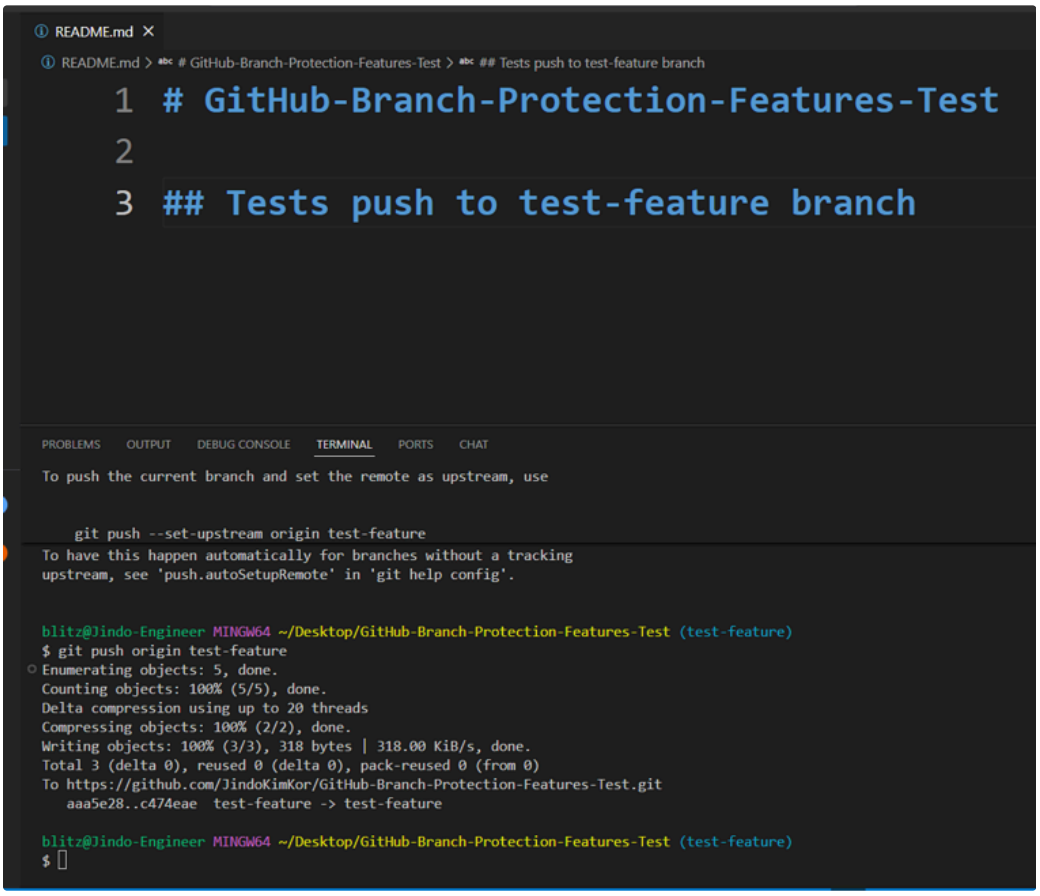
Test 1: Direct push blocked

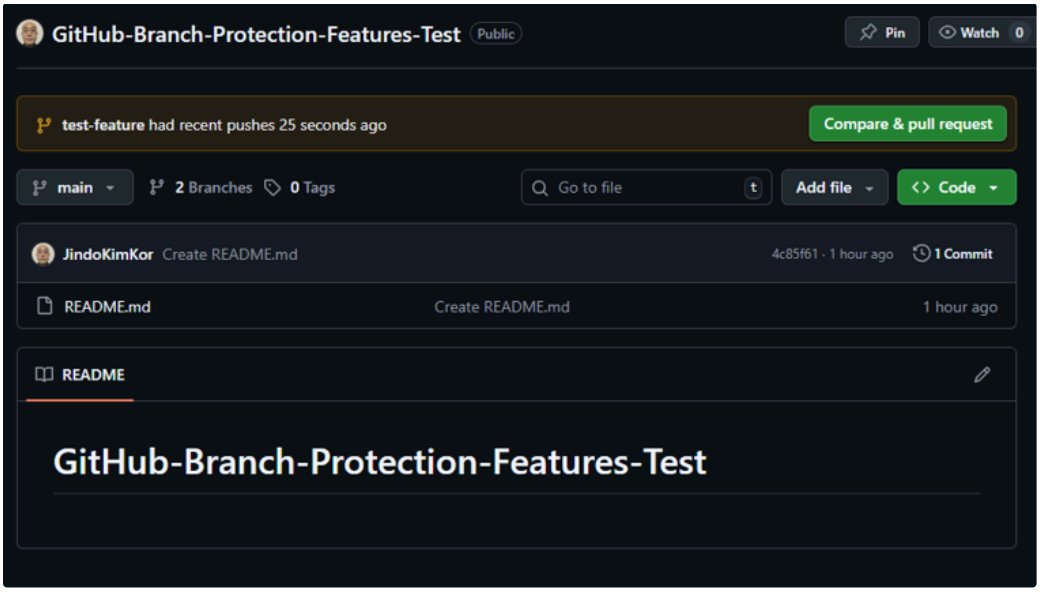


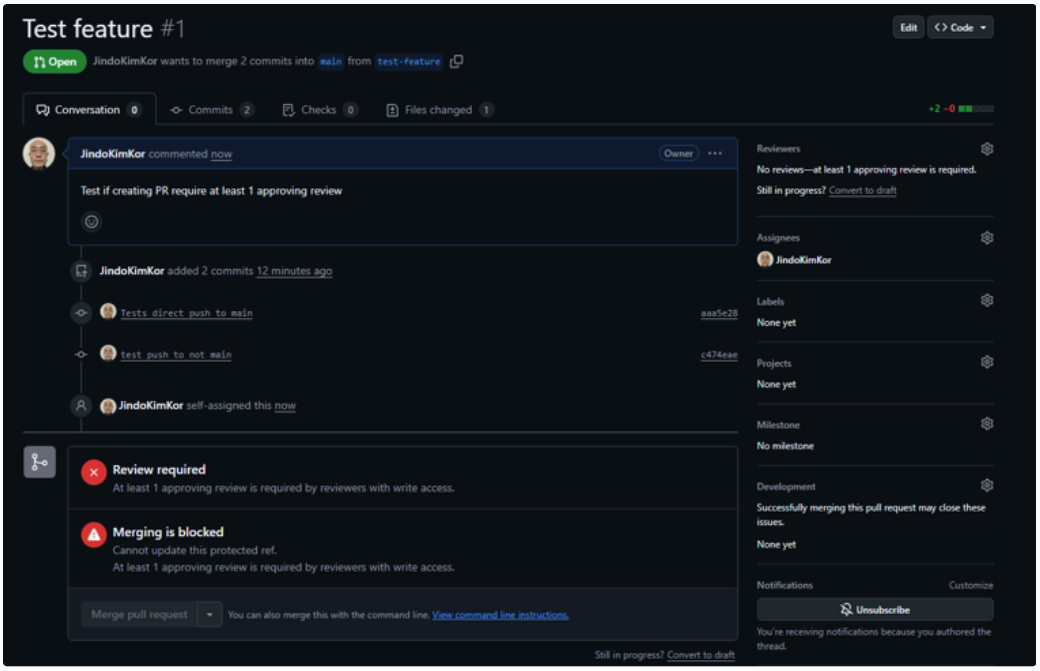
1. **Make changes locally and try to push directly to main**
2. **Expected: Push rejected with error message**

> **It works as expected**

Test 2: PR without approval blocked



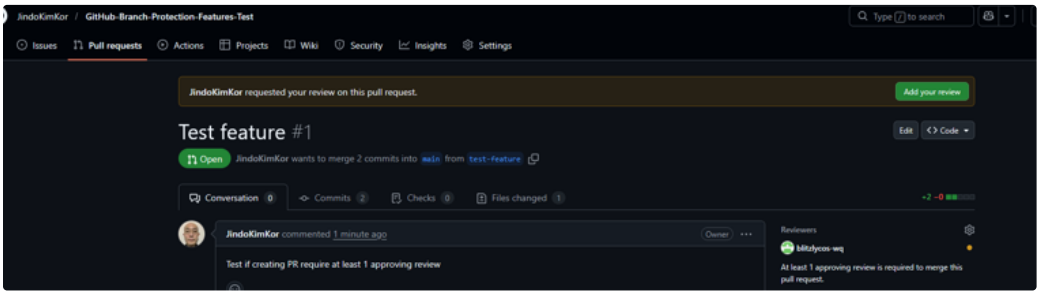


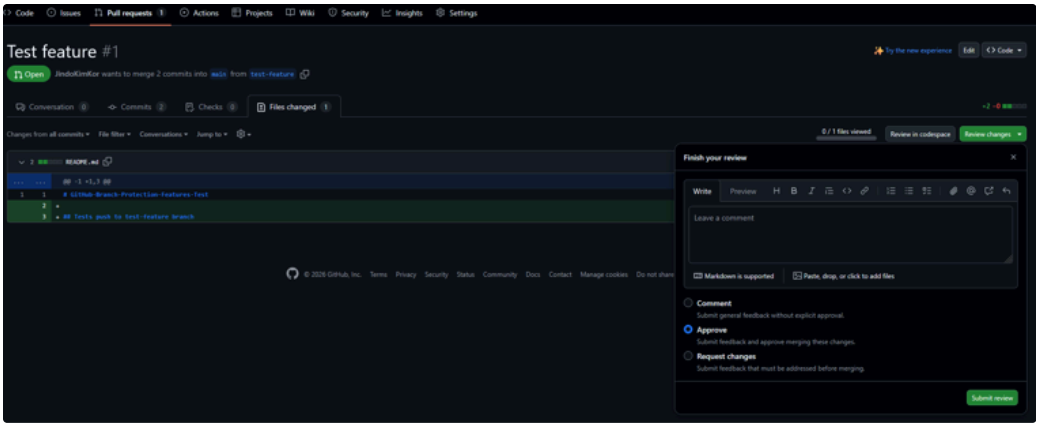


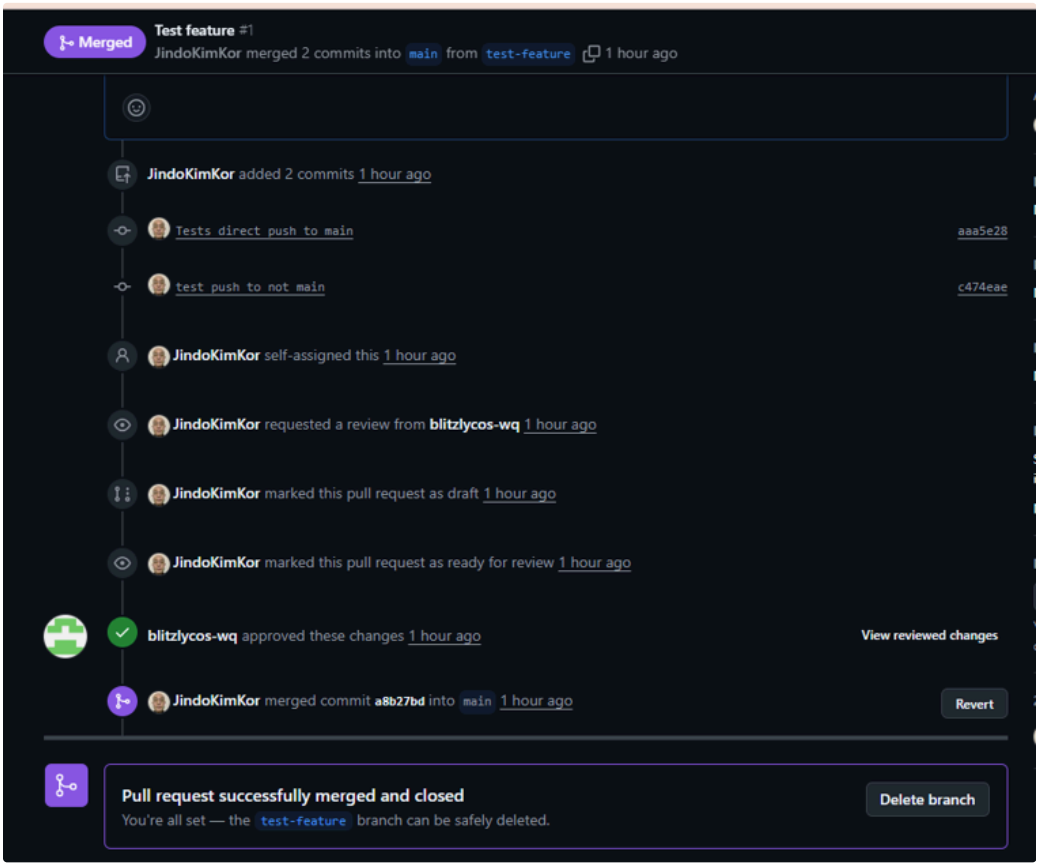
1. **Create a PR to main**
2. **Try to merge without approval**
3. **Expected: Merge button is disabled with message "Review required"**

> **It works as expected**

Test 3: PR with approval succeeds



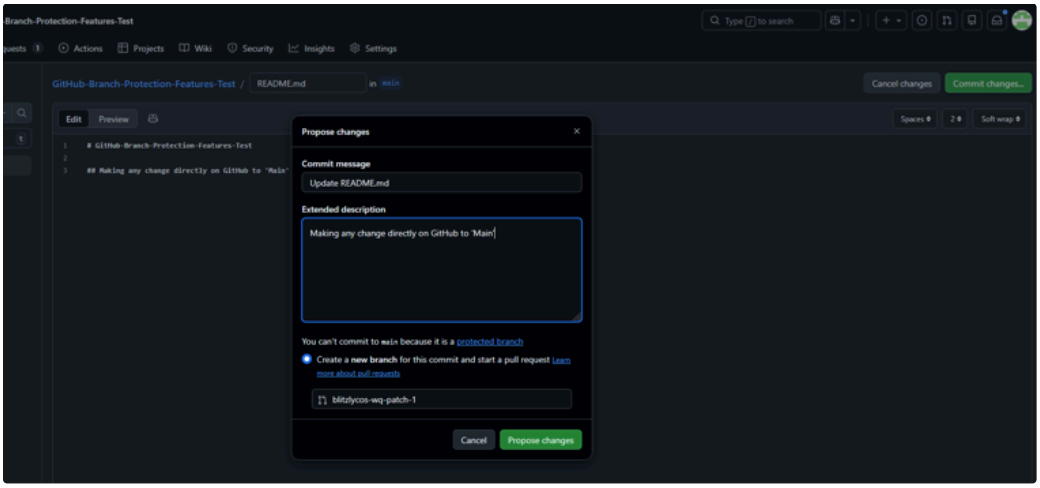


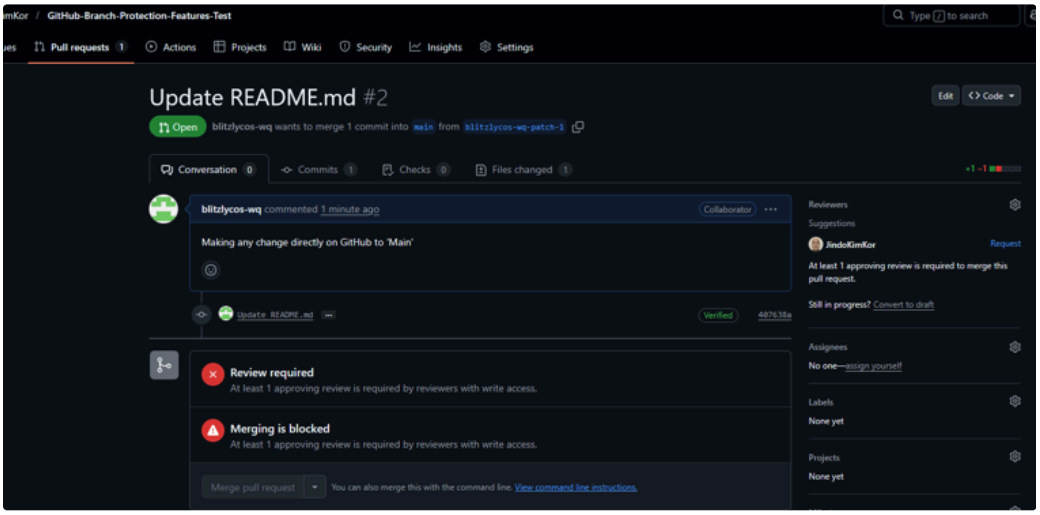


1. **Have another team member approve the PR**
2. **Expected: Merge button becomes enabled**

> **It works as expected**

Test (Edge Case) 4: Direct edit on main branch blocked (via GitHub Web)





1. **Try to edit a file directly on main branch via GitHub web**
2. **Expected: GitHub creates a new branch and prompts you to create a PR**

> **It works as expected**