

Part 1

Branch Protection rules

The screenshot shows the GitHub interface for configuring branch protection rules. The left sidebar contains navigation links: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Settings' tab is active, and the 'Rulesets' section is selected in the left sidebar. The main content area is titled 'Rulesets / the rules' and shows the configuration for a rule named 'the rule'. The rule is currently 'Active'. Below the rule name, there is a 'Bypass list' section which is currently empty. The 'Target branches' section shows that the rule applies to 'All branches'. The 'Rules' section lists several protection rules, with 'Require linear history' highlighted by a red rectangle. Other rules include 'Restrict creations', 'Restrict updates', 'Restrict deletions', 'Require deployments to succeed', and 'Require signed commits'.

dipto718 / Software-Supply Chain-HW-1

Rulesets / the rules (Active)

Rule Name *
the rule

Enforcement status
Active

Bypass list
Exempt roles, teams, and apps from this ruleset by adding them to the bypass list.
+ Add bypass

Bypass list is empty

Target branches
Which branches should be matched?
Branch targeting criteria
All branches
Applies to 1 target

Rules
Which rules should be applied?

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 deployments to succeed
Choose which environments must be successfully deployed to before refs can be pushed into a ref that matches this rule.
- ☐ Require signed commits
Commits pushed to matching refs must have verified signatures.

Rules

Which rules should be applied?

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 deployments to succeed**
Choose which environments must be successfully deployed to before refs can be pushed into a ref that matches this rule.
- ☐ **Require signed commits**
Commits pushed to matching refs must have verified signatures.
- ☒ **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.
[Show additional settings](#) ▾
- ☐ **Require status checks to pass**
Choose which status checks must pass before the ref is updated. When enabled, commits must first be pushed to another ref where the checks pass.
- ☒ **Block force pushes**
Prevent users with push access from force pushing to refs.
- ☐ **Require code scanning results**
Choose which tools must provide code scanning results before the reference is updated. When configured, code scanning must be enabled and have results for both the commit and the reference being updated.
- ☐ **Require code quality results**
Choose which severity levels of code quality results should block pull request merges. When configured, a code quality analysis must be done on the pull request before the changes can be merged.
- ☐ **Automatically request Copilot code review**
Request Copilot code review for new pull requests automatically if the author has access to Copilot code review.

Save changes

Revert changes

Part 3

personal.txt in the repo

This screenshot shows the GitHub interface for the repository 'Software-Supply-Chain-HW-1' by user 'dipto718'. The 'Files' tab is active, displaying the file 'personal.txt' which was added 6 minutes ago. The file's content is visible in the main area, showing a single line: 'sample data'. The left sidebar shows a file tree with various files and folders, including 'github', 'githgnore', 'pre-commit-config.yaml', 'CONTRIBUTING.md', 'LICENSE', 'README.md', 'SECURITY.md', 'artifact.bundle', 'artifact.md', 'bardi-final.txt', 'bardi-initial.txt', 'github_link.txt', 'main.py', 'merkle_proof.py', 'mygy-final.txt', 'mygy-initial.txt', 'personal.txt' (selected), 'pyint-final.txt', 'pyint-initial.txt', 'required_functionality_demostra...', 'ruff-final.txt', 'ruff-initial.txt', and 'util.py'.

This screenshot shows the GitHub commit view for commit '3d4c1f4' by user 'dipto718', committed 6 minutes ago. The commit message is 'added personal.txt'. The 'Files changed' tab is active, showing a diff for 'personal.txt' with one line added: 'sample data'. The 'Comments' section is empty, and there is a 'Subscribe' button at the bottom.

Part 5

Code Coverage is $\geq 75\%$

```
===== test session starts =====
platform linux -- Python 3.12.3, pytest-9.0.0, pluggy-1.6.0
rootdir: /home/dipto/Software-Supply-Chain-HW-1
configfile: pyproject.toml
plugins: cov-7.0.0
collected 11 items

tests/test_cases.py ..... [100%]

===== tests coverage =====
_____ coverage: platform linux, python 3.12.3-final-0 _____

Name                Stmts   Miss  Cover
-----
main.py              118     39    67%
merkle_proof.py      118     25    79%
tests/test_cases.py   91     11    88%
util.py              27      3    89%
-----
TOTAL                354     78    78%

===== 11 passed in 30.89s =====
(myenv) dipto@dipto-VirtualBox:~/Software-Supply-Chain-HW-1$
```