
Pull requests

Key takeaways

1. A pull request is a branch proposal plus the evidence needed to decide it.
2. One focused claim per PR: why, what changed, how tested, and remaining risk.
3. Review the claim before the lines; decline what evidence cannot support.

Common misunderstandings of pull requests

Which claim would you defend right now? Choose before the evidence.

“The code is the proposal; the pull-request description is optional context.”

“A thorough review starts by commenting on changed lines one by one.”

“Green automated checks are enough to approve a performance claim.”

Keep your choice. Look for the mechanism or counterexample that would make it fail.

Every pull request does three jobs

Make a change easy to understand, test, discuss, and decide

Propose

Show one focused change from a branch without putting it on the main branch yet.

Review

Inspect the exact differences, test the claim, and discuss risks on the relevant lines.

Decide

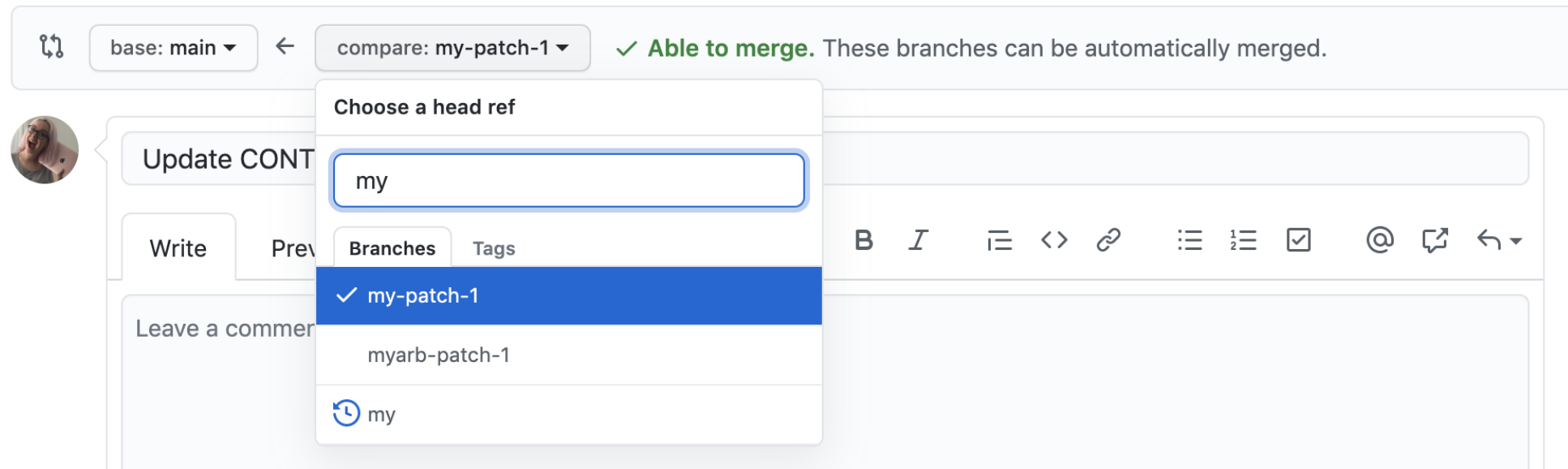
Merge when the evidence is convincing, or decline with a reason when it is not.

A pull request proposes one branch

GitHub compares a source branch with the branch that would receive it

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



BASE is where the change would land. COMPARE is the branch that contains the proposed change.

One GitHub page holds the whole decision

The page updates when the contributor pushes another commit

Adding sparkles #45834

Open

octocat wants to merge 6 commits into `main` from `octocats-branch`

Conversation34

Commits6

Checks57

Files changed2

+32-4

Changes from all commits

File filter

Conversations

0 / 2 files viewed

Review in codespace

Review changes

Area	Question it answers
Conversation	Why is this proposed, and what did people decide?
Commits	How did the proposal develop?
Checks	Did the automated tests finish successfully?
Files changed	What exact lines differ from the base branch?

source: GitHub Docs, Reviewing proposed changes in a pull request — docs.github.com (accessed 2026-07-21)

Small pull requests get better reviews

One purpose gives the reviewer one claim to verify

Hard to review	Easy to review
Improve performance everywhere	Reduce wasted simulation on machines with growing tape coverage
Refactor, optimize, and rename	Change the allocation policy; leave unrelated cleanup out
Trust me — it is faster	Show the base, the proposed version, the same racks, and the result
Twenty noisy commits	Keep useful experimental commits; remove accidental noise

A reviewer should be able to finish the sentence: “This PR claims that ...”

source: GitHub Docs, Helping others review your changes — docs.github.com (accessed 2026-07-21)

The description is the map for the review

Explain the purpose, the change, the evidence, and the remaining risk

A reviewable tm-compute description

Target and hypothesis

- Which machine family is mis-ranked?
- Why does the frozen bot make that mistake?

Evidence

- Base result, exact command, racks, and environment
- Proposed result under the same conditions

Final proposal

- What changed and why it fits this bot
- Known risks or cases not addressed

Write for a reviewer who did not watch your agent session.

Self-review before opening the PR

Check the destination, the diff, the evidence, and the reviewer

 octo-repo had recent pushes less than a minute ago

Compare & pull request

One safe path from a reviewed branch to GitHub

```
git status
git diff BASE...HEAD
git push -u origin my-direction
gh pr create --base main --head my-direction --web
```

- Use a draft only while the work is not ready for a decision.
- Request the assigned reviewer when the description and evidence are complete.

tm-compute offers two contributor paths

Both paths require visible, reproducible work in the one assigned PR

The final version improves
The course measures the frozen base and final PR version. The final version is safe and performs better.

Two substantial directions do not improve
Keep code, commits, rationale, and fair measurements for two technically distinct attempts.

Substantial direction	Not a separate direction
A different estimation method	A nearby constant for the same method
A different compute-budget allocation	Another run of the same implementation
New handling for a meaningful machine family	Comments, cleanup, or an unimplemented idea

Review the claim before reviewing the code

Purpose first keeps line comments connected to the proposed outcome

Pass	What to do	What you are deciding
1. Understand	Read the description and identify the claimed outcome	Is the goal specific and relevant?
2. Inspect	Read each changed file and mark it viewed	Does the code implement the stated idea?
3. Verify	Run the checks and reproduce the important measurement	Does the evidence support the claim?
4. Respond	Leave focused line comments, then a summary	Approve, request changes, or comment?

Do not spend ten minutes on naming while a ranking error or unfair comparison goes unchecked.

Review the details in Files changed

Move file by file; GitHub tracks which ones you have viewed

 Conversation 0

 Commits 1

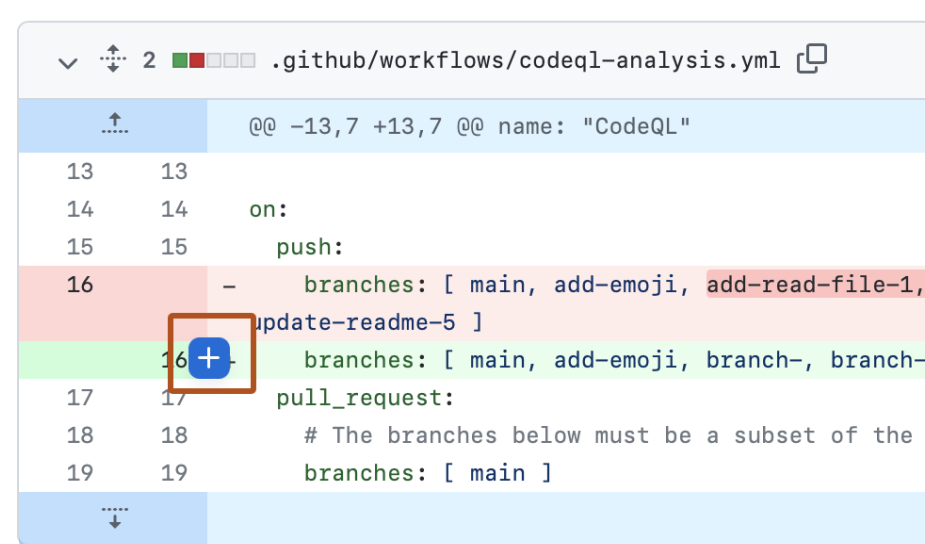
 Checks 1

 Files changed 24

- Use the file tree and filters when the change spans several files.
- Hide whitespace only when whitespace is not part of the behavior.
- Mark each finished file Viewed; a later edit makes it unviewed again.
- Check the final combined diff, not only one commit at a time.

Anchor feedback to the risky line

Select one line or a range, then start a review to batch your comments



Weak	Useful line comment
Looks wrong	This doubles every probe budget, so the total can exceed the limit. Please cap the share or show a bound.
Why?	This removes the signal for long runners. Which rack shows that the lost signal does not invert the top pair?
Rewrite	This can be expressed directly; here is a suggested change that preserves behavior and exposes the boundary.

Name the observation, impact, and action

Ask for evidence when evidence can settle the disagreement

Part	Example
Observation	The repeat check now runs only every 128 steps.
Impact	A short cycle can consume most of the budget before it is recognized.
Action	Please measure this on a rack with short cycles, or restore the frequent check.

Blocking
Correctness, safety, unfair measurement, or a claim that lacks required evidence.

Non-blocking
A clear improvement that is useful but not required for this proposal to be accepted.

Question
A real request for context whose answer may change your judgment.

Finish with one coherent review

Batch line comments, then summarize the decision

Adding sparkles #45834

Edit < > Code ▾

 Open octocat wants to merge 6 commits into `main` from `octocats-branch` 

 Conversation 34

 Commits 6

 Checks 57

 Files changed 2

+32 -4 

 Changes from all commits ▾ File filter ▾ Conversations ▾ 

0 / 2 files viewed

Review in codespace

Review changes ▾

Choice	Use it when
Comment	You have feedback but are not making the final accept-or-change decision
Approve	You tested enough to accept the change as it stands
Request changes	A specific issue must be addressed before acceptance

Respond to every substantive review point

A new commit updates the same PR; the conversation records why

Reviewer says	Contributor response
The code has a defect	Fix it in the PR branch, push, and reply with the commit and test
The evidence is too weak	Repeat the fair comparison or explain why another check is more decisive
You disagree	State the technical reason and attach evidence; do not dismiss the comment
The idea is outside this PR	Say so and preserve it as follow-up work rather than silently expanding scope

Resolve a conversation after the point is answered, then re-request review after substantial changes.

The final verdict follows the evidence

A reasoned decline is better maintenance than a polite unsafe merge

Evidence at the end	Maintainer verdict
The proposal is safe and the claim survives review	MERGE — name the evidence that makes the decision
The final version is safe, but improvement is not established	DECLINE — explain the remaining uncertainty or lack of benefit
The final version regresses, breaks the protocol, or exceeds limits	DECLINE — point to the failed check or measurement
Two substantial attempts failed fairly	The contributor may still earn full outcome credit; the author may still decline

You vs. the Agent: pull-request exchange

Your Responsibility
Choose the claim and the acceptable evidence
Read the final diff and decide what is safe
Own line comments, approval, and the verdict
Disagree with reasons rather than politeness

Agent's Responsibility
Summarize the diff and changed control flow
Run tests and repeat benchmark commands
Draft possible risks and review comments
Prepare a revision after you choose the response

Shared: the agent can help inspect; the people remain accountable for the proposal and decision

Key Takeaways

1. A pull request is a proposal from one branch into another, plus the evidence and discussion needed to decide it.
2. Create one focused claim; explain why, what changed, how you tested it, and what risk remains.
3. Review purpose first, then files and lines; finish with a supported approve, change request, merge, or decline.

triage a review

10 min

- A tm-compute PR changes bot.py only. Its claim: growing-tape machines are under-probed; double their probe budget and check for repeats every 128 steps (was 32). A review agent returns four findings:
- A) The doubled budget is never capped, so a rack full of growing-tape machines can exceed the step limit.
- B) The description promises median-of-three timings, but the evidence table shows one run per rack.
- C) n_prb is unreadable; rename it to probe_count.
- D) simulate.py drops its halt_guard() call, so a runaway machine never stops.
- Alone, choose the strongest merge blocker and the finding that should make you audit the reviewer. In pairs, agree on both.
- For the blocker, write the smallest test and its predicted result. For one rejected finding, write a one-sentence reason.
- Join another pair and resolve one disagreement. Deliverable: blocker + test, audited finding + clue, and rejected finding + reason.