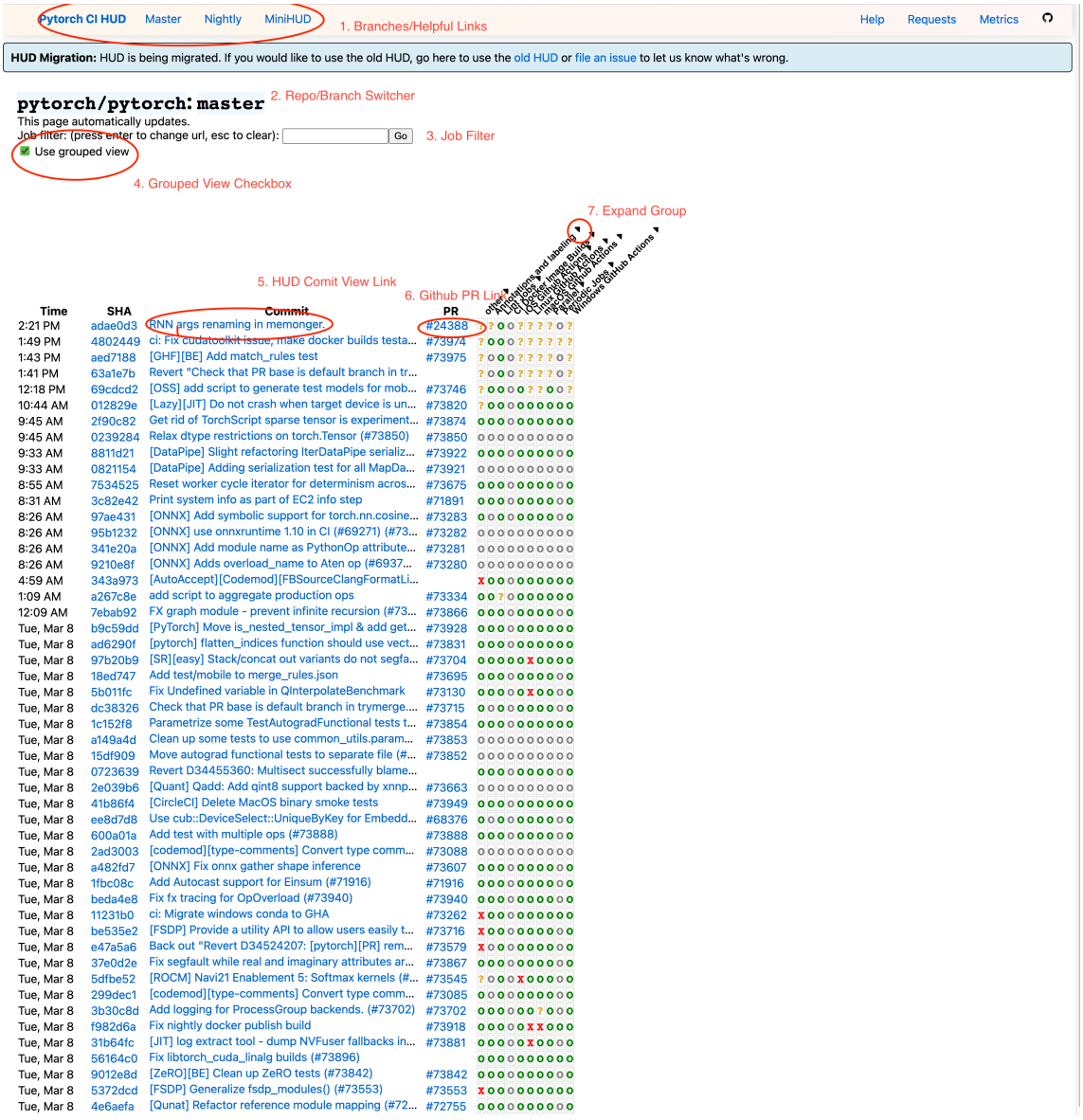
TLDR: Visit hud.pytorch.org for a quick glance into PyTorch's CI. Go to `hud.pytorch.org/pr/<pr number>` (example) or `hud.pytorch.org/commit/<long hash>` (example) for a detailed view of GitHub Actions jobs.

Please report any HUD bugs you find in our issue tracker!

Jobs

PyTorch's CI currently runs on 3 platforms, GitHub Actions, Jenkins, and CircleCI. It can be hard to tell at a glance how the various jobs across these services are doing on recent commits to PyTorch to determine if a failure on your pull request is a real failure vs. something that is broken on `master`. hud.pytorch.org aims to fill this gap by providing a quick view over all the jobs on these commits.



1. Quickly switch between different branches. Also has a link to MiniHUD, a more granular, commit based view. 2. Click on these to change the branch or repo. Only works on repos within Pytorch 3. Search for jobs in here and press "Go" to get the permalink 4. Group the workflows by logical organization 5. Click on these commits to take you to

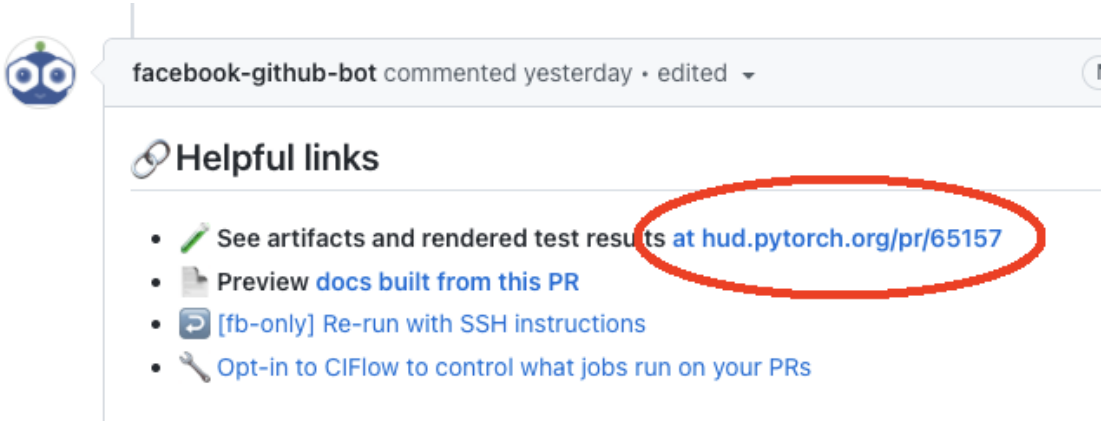
the HUD view of the jobs that ran 6. Click on these to take you to the PR on Github 7. Click on this to expand the group

Individual Pull Requests and Commits

GitHub provides a view for GitHub Action job logs for commits and PRs, but we received many reports that this was lacking in functionality and usability, so we created our own view which is also hosted on hud.pytorch.org. The page displays statuses for a commit, so if you are viewing a PR you will see statuses for the *latest* commit to that PR.

Finding the Page

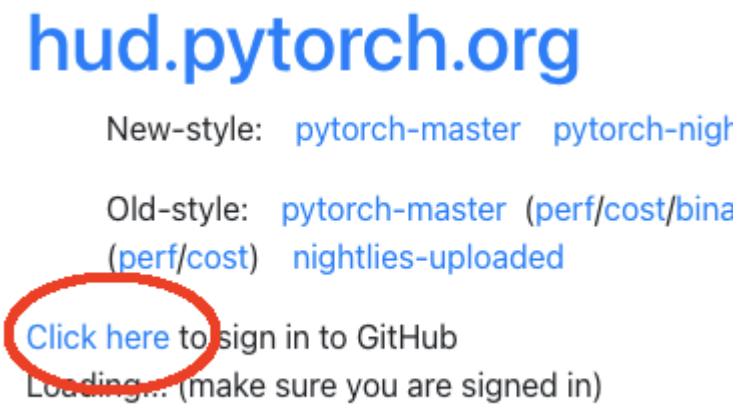
For PRs, you can navigate to the page directly by going to `hud.pytorch.org/pr/<pr number>`, or by finding the link in the automated Dr. CI facebook-github-bot comment on your PR.



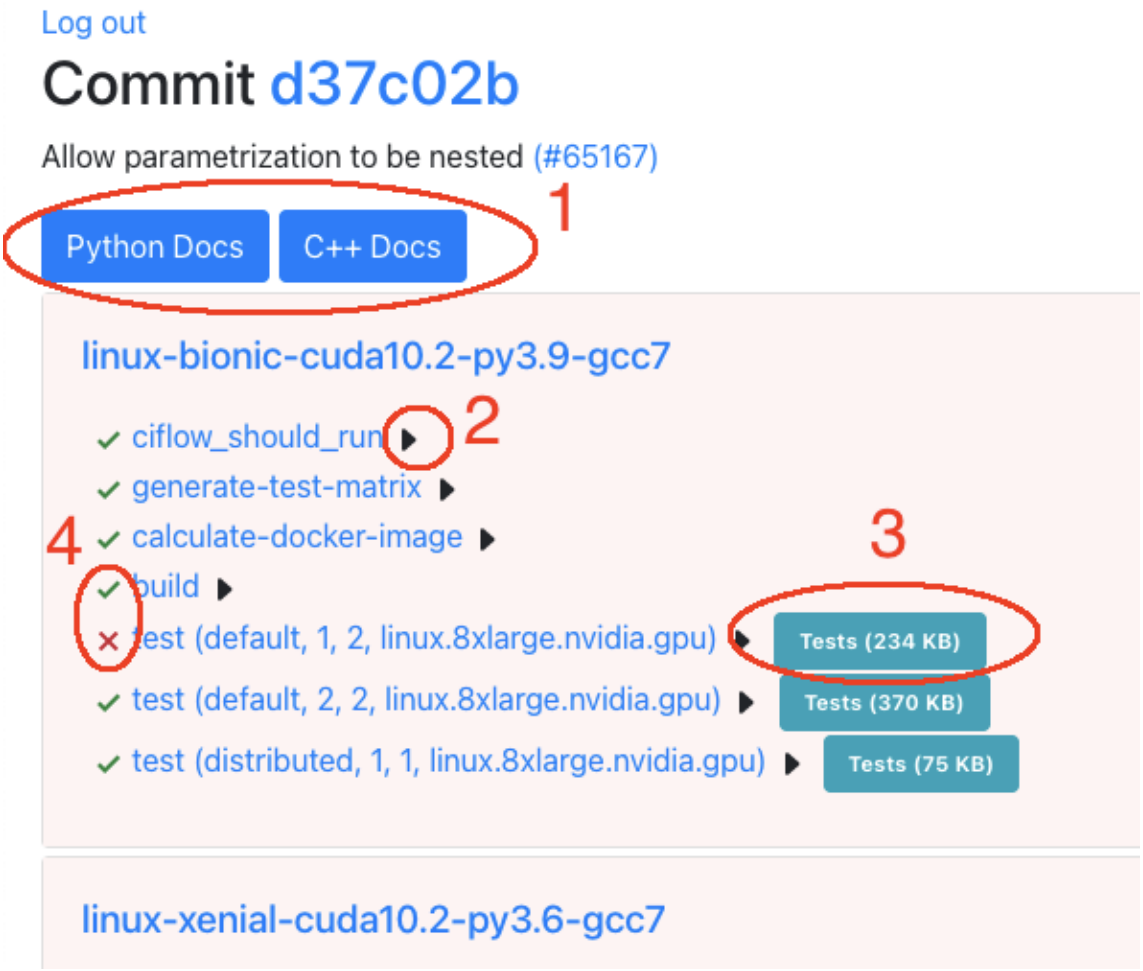
For commits, you can go to `hud.pytorch.org/commit/<long commit hash>` or by clicking the link from the main HUD page (see "Jobs" above).

Usage

You will need to sign in with GitHub on your first visit to the page. This is necessary so the HUD can make calls to the GitHub API.



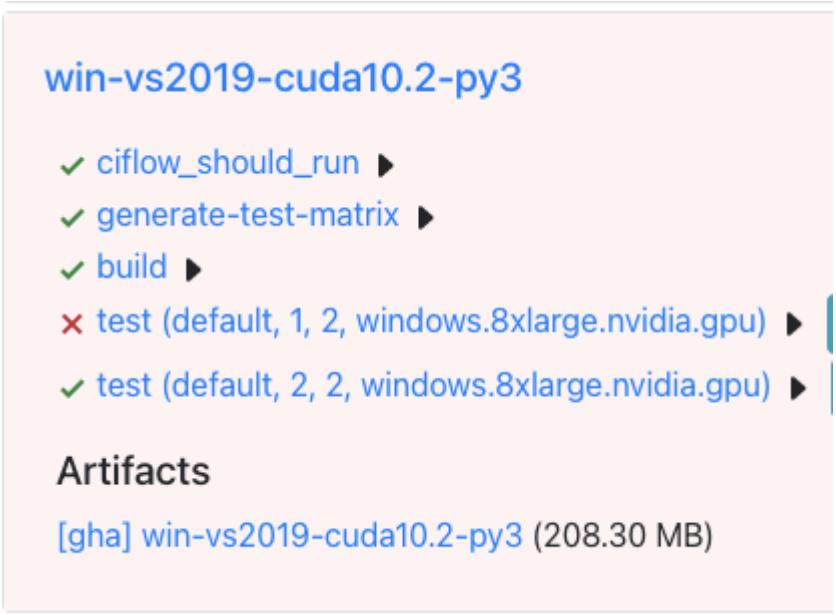
Once done, you should be able to see the GitHub Actions jobs for that commit or PR. For PRs, the jobs shown are for the latest commit pushed to that PR. The jobs are sorted so failing jobs are at the top. Some jobs that are not very helpful are grouped together at the bottom (such as the "Triage" jobs).



1. If the doc builds for this commit or PR have finished, the link to the C++ / Python previews will be shown at the top.
2. You can view logs for this job by clicking the rightward arrow.
3. Some jobs report test results. The HUD can download and render these, similar to the test view on CircleCI. Click the button to expand and see failed tests. You can also see details of which tests ran in the "Summary" section.
4. Each job reports a status, one of:
    1. Success
    2. Failed
    3. Cancelled
    4. Skipped

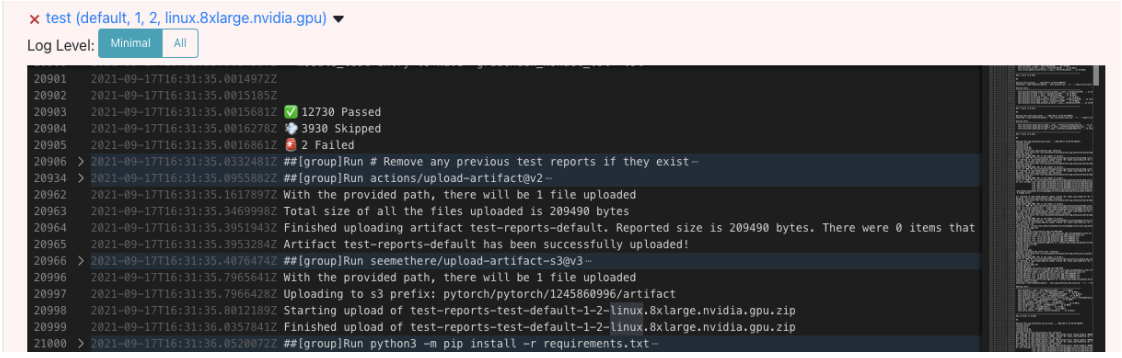
Artifacts

Some jobs upload artifacts. Test report artifacts are hidden from view since their data is exposed via the test report renderer you see when you click the blue "Tests" button next to a test job. Other artifacts, some of which are stored in GitHub's artifact store and some in AWS S3, are shown below the job if there are any.



Logs

Logs are shown using VScode, so you can look through them with the same tools you would if you had downloaded them locally. The text is editable if you wish to operate on the logs, but these edits are not saved anywhere. You can also bring up the VSCode command palatte with F1 and run most commands from there. The "Log Level" selector enables or disables line filtering of [known-noisy lines]. Usually you will not have to move it off of "Minimal".



GitHub's log viewer supports sigils to mark the start and end of groups of lines, `##[group]` and `##[endgroup]` respectively. The log viewer here detects and automatically folds these. You can expand them by clicking the arrow on the left or F1 -> "Unfold All".