

Merging Github Pull Requests

This section documents the process for reviewing and merging code changes contributed via [Github Pull Requests](#). There are 2 approaches for merging:

1. Via the GitHub merge button. This is simplest and GitHub will recognise the pull request as merge. However, it is up to the committer to format the commit message appropriately, close the relevant JIRA and cherry-pick the squashed commit to the relevant release branches (if necessary).
2. Via the `kafka-merge-pr.py` script. The script automates a number of steps as described in the Merge Script section. The main downside is that GitHub will not recognise the pull request as merged (it will simply be closed).

A pull request is ready to be merged once:

1. It has been reviewed and feedback has been addressed.
2. The branch merges cleanly to trunk. If the branch merges cleanly to a stable branch, the code hasn't diverged and the committer deems it safe, it's OK to cherry-pick to a stable branch (manually with git or via the merge script). Otherwise, a separate PR should be submitted for the backport so that it can be reviewed and tested.
3. Unit and integration tests have passed via the automated PR builds.
4. System tests have passed, if applicable. For non trivial changes, we recommend running the system tests via <https://jenkins.confluent.io/job/system-test-kafka-branch-builder/> before merging the PR. This is a manual process at the moment and it would be good to automate it in the future.

Merging via the GitHub Merge Button

This process requires the committer to have connected their ASF and github accounts via [gitbox](#). Once connected, it can take several minutes for the write access to synchronize and the workflow below to become available.

Once the pull request is ready to be merged (as per the description above):

1. Click on the "Squash and merge" button.
2. Edit the commit title and description so that it follows the convention described in [Contributing Code Changes](#). It should be a matter of copying and pasting from the pull request title and description, but they can often be improved. Unfortunately, GitHub's default strategy for the commit message is not great (it copies the message from each commit in the pull request).
3. Mention the reviewers in a separate line. For example: "Reviewers: Joe Smith <joesmith@gmail.com>, Mary Jane <maryjane@yahoo.co.uk>".
4. Click on the confirm button.

Merging via the Merge Script

`kafka-merge-pr.py` is a script that automates the process of accepting a code change into the project. It creates a temporary branch from `apache/trunk`, squashes the commits in the pull request, rewrites the commit message in the squashed commit to follow a standard format including information about each original commit, merges the squashed commit into the temporary branch, pushes the code to `apache/trunk` and closes the JIRA ticket. The push will then be mirrored to `apache-github/trunk`, which will cause the PR to be closed due to the pattern in the commit message. Note that the script will ask the user before executing remote updates (ie git push and closing JIRA ticket), so it can still be used even if the user wants to skip those steps.

Setting Up

1. Clone Kafka's Git repository (if you haven't already).
2. Add aliases for the remotes expected by the merge script (if you haven't already):

```
git remote add apache https://gitbox.apache.org/repos/asf/kafka.git
git remote add apache-github https://github.com/apache/kafka.git
```

Note that the scripts we use default to pushing changes directly to github. If you did not connect your Apache and Github accounts via [gitbox](#), you'll need to push to gitbox instead by:

```
export PUSH_REMOTE_NAME=apache
```

3. Install jira-python:

```
sudo easy_install jira
```

Merging

Once the pull request is ready to be merged (as per the description above):

1. Set the `JIRA_USERNAME` and `JIRA_PASSWORD` environment variables with the appropriate credentials if you intend to ask the script to close the issue associated with the pull request.
2. Run the merge script:

```
./kafka-merge-pr.py
```

3. Answer the questions prompted by the script.