STEPS TO CLONE CODE AND RAISE PR

Contents

[When cloning code using **https** URL: 1](#_Toc84331509)

[When cloning code using **SSH** URL 3](#_Toc84331510)

[Configure your Git username/email 5](#_Toc84331511)

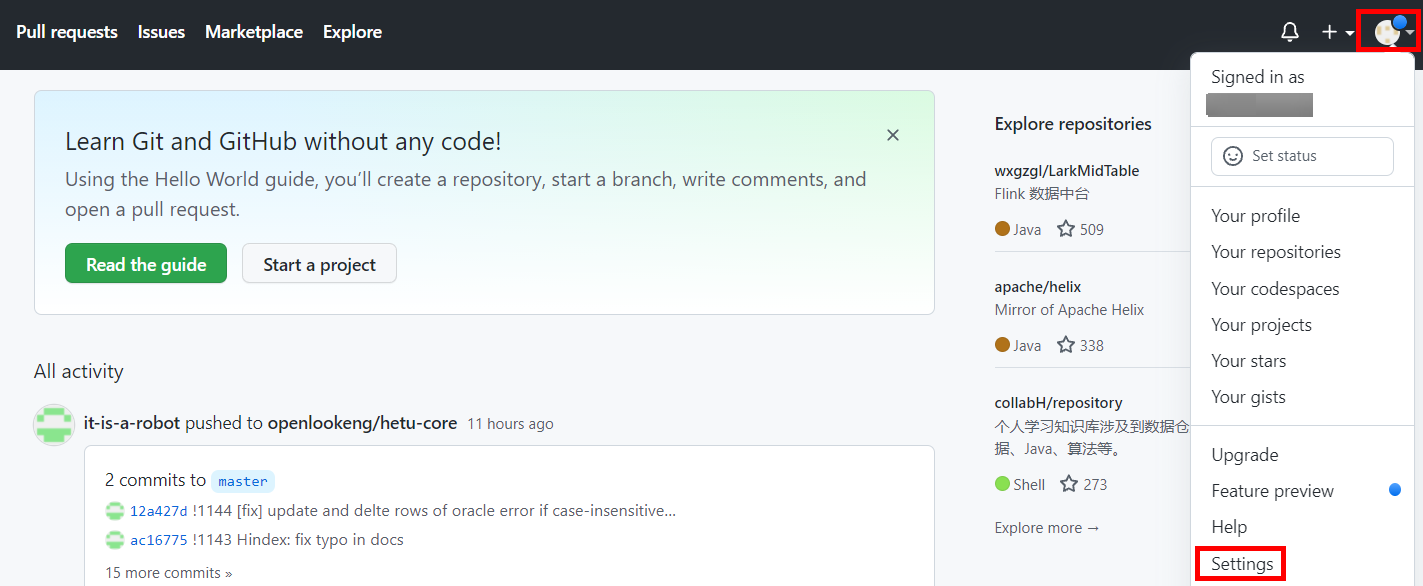
[Steps to Clone code and Commit changes 6](#_Toc84331512)

[Steps to create Issue and raise PR 7](#_Toc84331513)

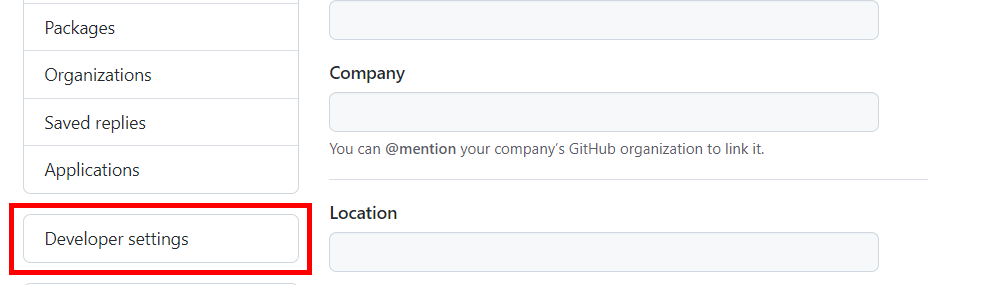
# When cloning code using **https** URL:

**Steps to Generate Private Access Token in GitHub**

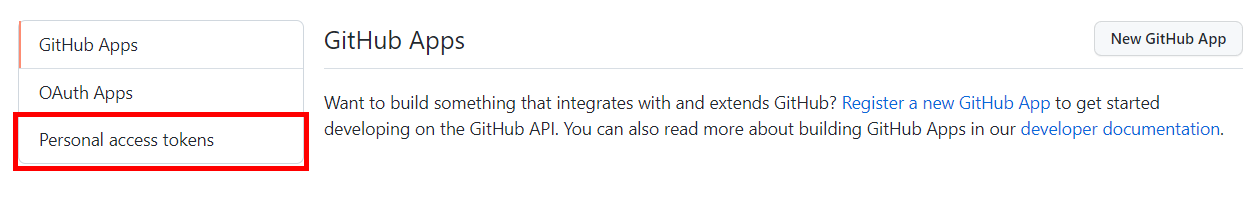
Step 1: Go to Profile, click on drop-down and click on Settings



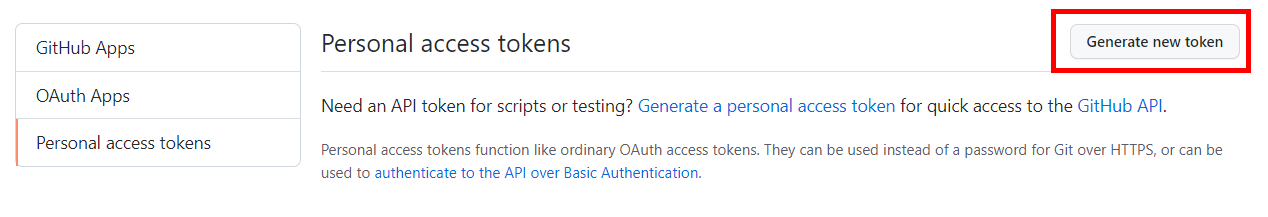
Step 2: Go to Developer settings tab



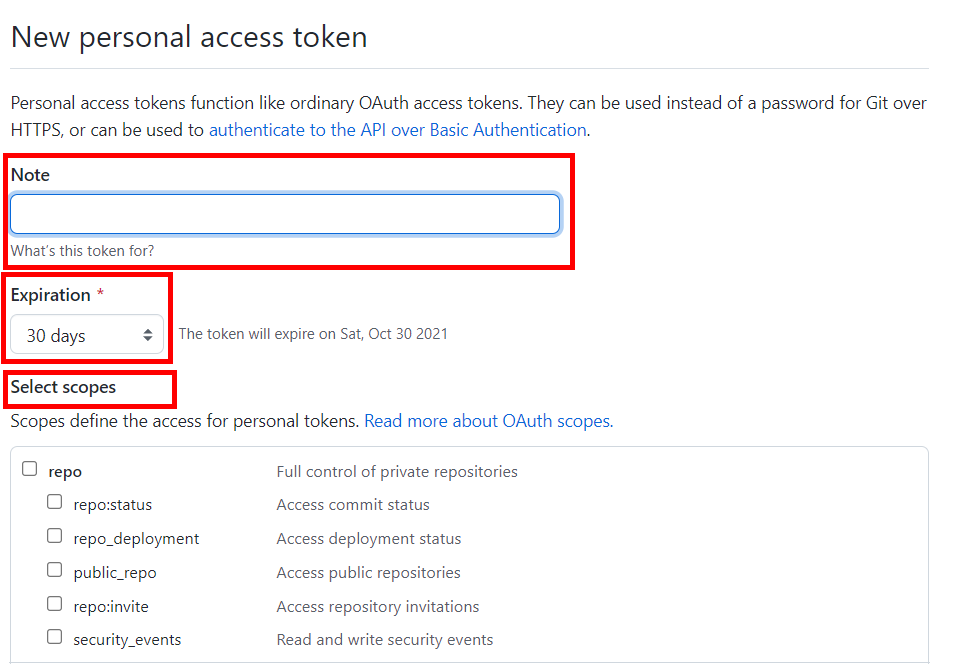
Step 3: Go to Personal access tokens tab



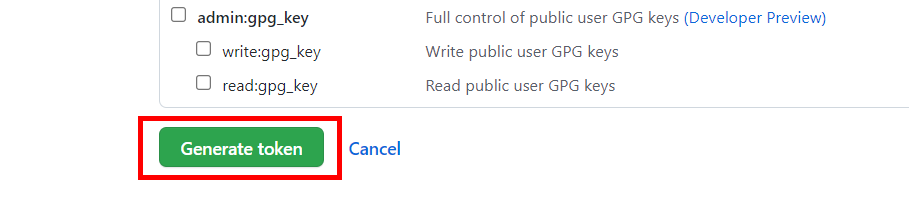
Step 4: Click on Generate New Token



Step 5: Add Note, select Expiration and Scopes.



Step 6: Click on Generate Token



Note: Copy the generated token and store it safe. This generated token is to be given as password when pushing the code.

# When cloning code using **SSH** URL

Generating SSH key

**Step 1:** Open Git Bash

**Step 2:** Paste the text below, substituting in your GitHub email address.

$ ssh-keygen -t ed25519 -C "your\_email@example.com"

**Step 3:** When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

> Enter a file in which to save the key (/c/Users/you/.ssh/id\_ed25519):[Press enter]

**Step 4:** At the prompt, type a secure passphrase.

> Enter passphrase (empty for no passphrase): [Type a passphrase]

> Enter same passphrase again: [Type passphrase again]

Adding your SSH key to the ssh-agent

**Step 5:** Ensure the ssh-agent is running. You can use the "Auto-launching the ssh-agent" instructions in "[Working with SSH key passphrases](https://docs.github.com/en/articles/working-with-ssh-key-passphrases)", or start it manually:

# start the ssh-agent in the background

$ eval "$(ssh-agent -s)"

> Agent pid 59566

**Step 6:** Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace id\_ed25519 in the command with the name of your private key file.

$ ssh-add ~/.ssh/id\_ed25519

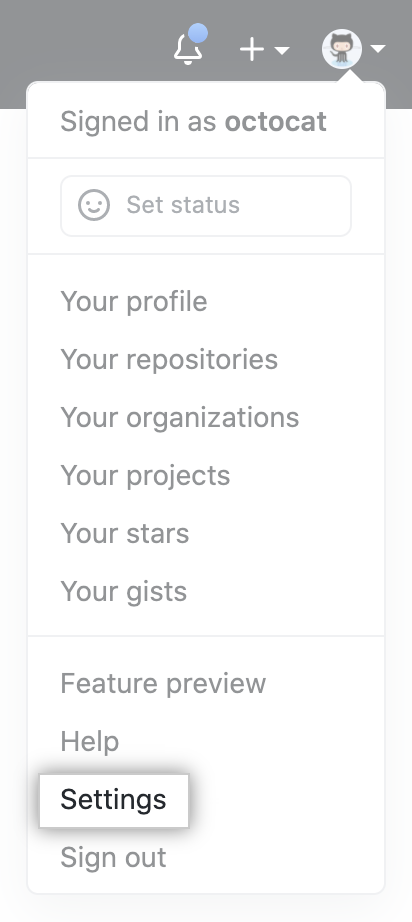
Adding a new SSH key to your GitHub account

**Step 7:** Copy the SSH public key to your clipboard. If your SSH public key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

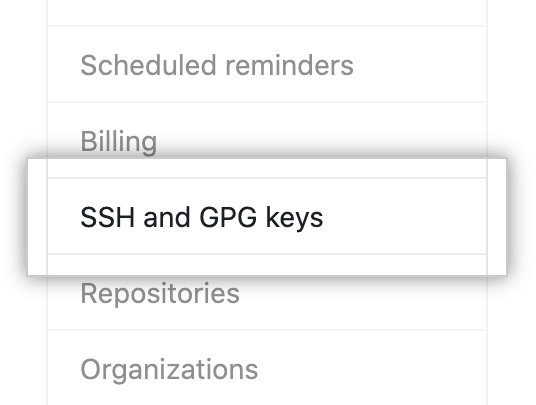
$ clip < ~/.ssh/id\_ed25519.pub

# Copies the contents of the id\_ed25519.pub file to your clipboard

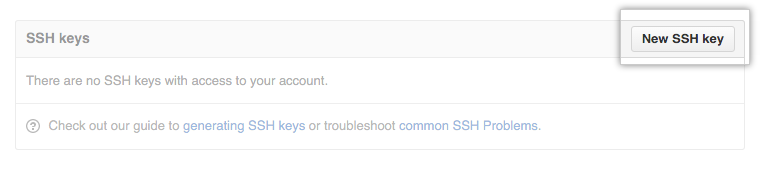
**Step 8:** In the upper-right corner of any page, click your profile photo, then click **Settings**.



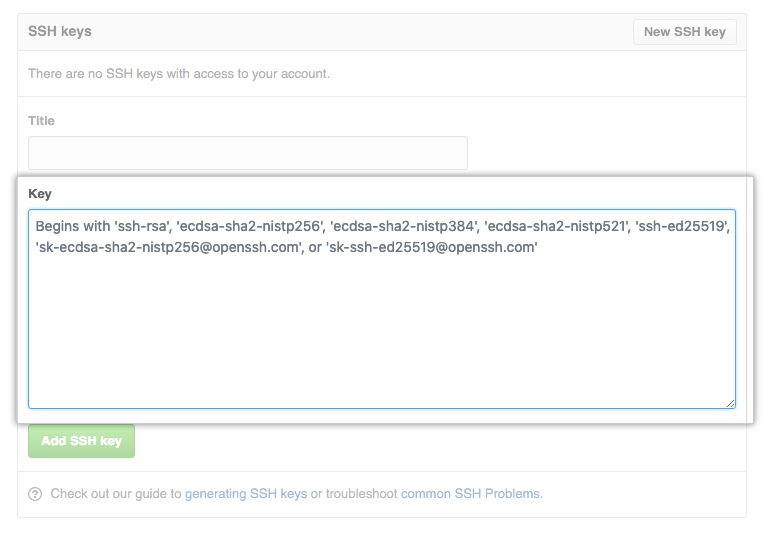
**Step 9:** In the user settings sidebar, click **SSH and GPG keys**.



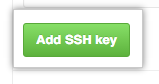
**Step 10:** Click **New SSH key** or **Add SSH key**.



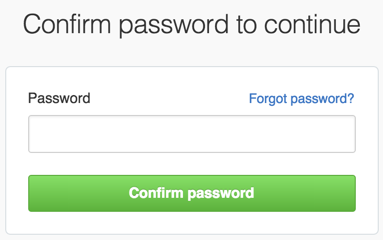
**Step 11:** In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air". Paste your key into the "Key" field.



**Step 12:** Click **Add SSH key**.



**Step 13:** If prompted, confirm your GitHub password.



**Step 14:** Clone code using SSH option

**References**

1. https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
2. <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

Note: This document includes process in Windows. Process in Linux and Mac is similar and can be referenced with the links mentioned above.

# Configure your Git username/email

Open git bash

Step1: Set your username  
git config --global user.name "FIRST\_NAME LAST\_NAME"  
  
Step2: Set your email address  
git config --global user.email "[MY\_NAME@example.com](mailto:MY_NAME@example.com)"

# Steps to Clone code and Commit changes

Step 1: Cloning forked repo. This repo will be origin by default.

$ git clone https://github.com/<name-of-forked-branch>/hetu-core.git

ex: git clone https://github.com/prasad/hetu-core.git

Go To hetu-core directory.

Step 2: Adding openLooKeng repo as main repository.

$ git remote add main https://github.com/openlookeng/hetu-core.git

Step 3: Fetch from all repositories.

$ git fetch --all

Step 4: Checkout to a new branch. Specify branch name in place of <branch-name>

$ git checkout -b <branch-name>

ex: git checkout -b spell\_fix

Step 5: Reset current branch with main repository’s master

$ git reset --hard main/master

After making the required changes in code,

Step 6: Run the following command to display the state of repository and staging area and to check the tracked, untracked files and changes.

$ git status

Step 7: Run the following commands to add files to the staging area.

To add single file

$ git add <filepath>

Ex: git add hetu-docs/en/migration/from-hive.md

To add all files

$ git add .

Step 8: Run the following command to make a commit with given commit message,

$ git commit –m “commit message”

ex: git commit -m "spelling corrections in from-hive.md"

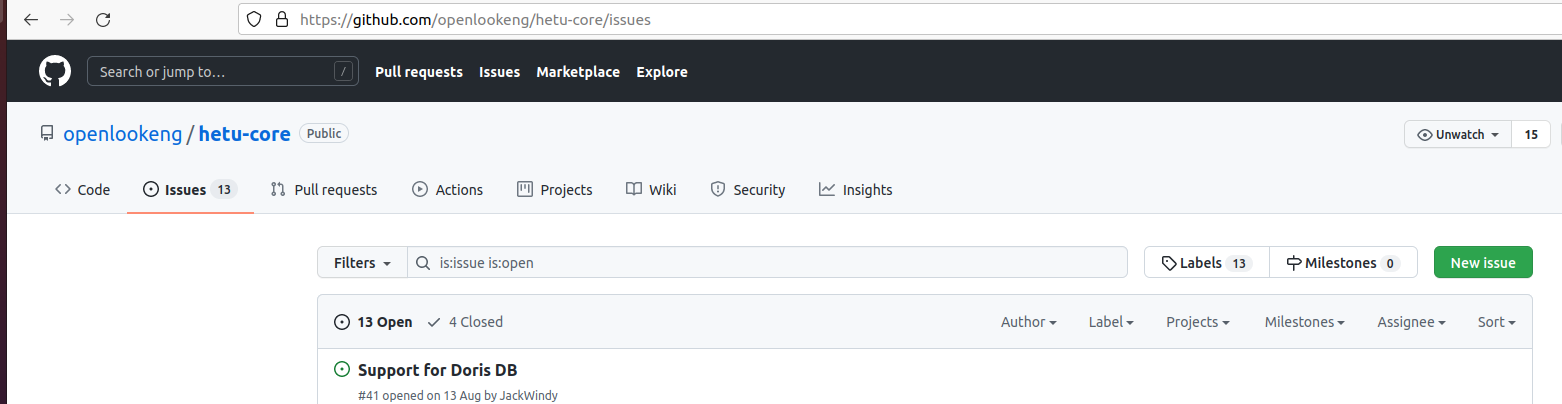
Step 9: Push updates the remote refs with local refs

$ git push origin <branch-name>

ex: git push origin spell\_fix

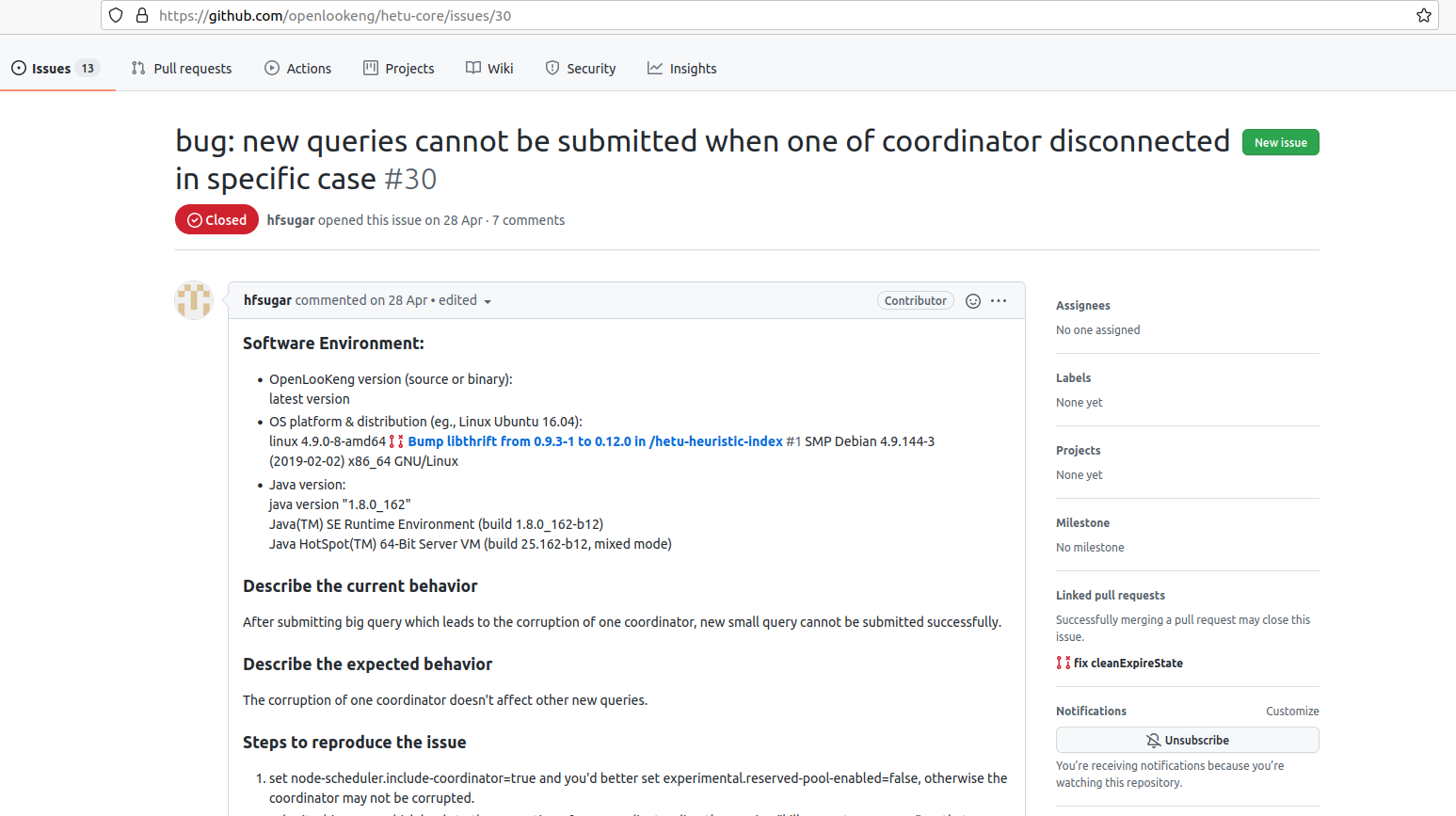
# Steps to create Issue and raise PR

**Issue:**

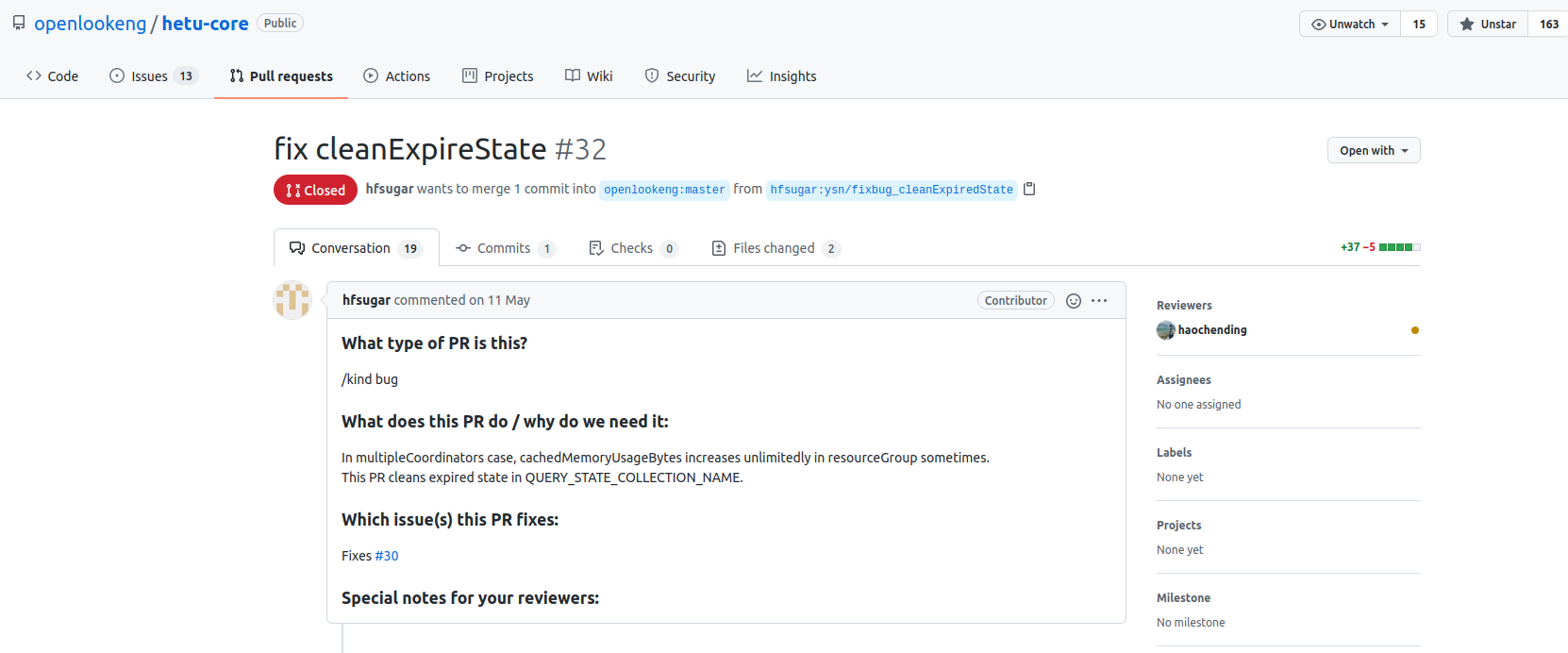


Create issue, if the issue doesn’t exist.

To create new issue, **select New issue** in github.



**PR:**



To raise PR, **select new pull** request from forked repo in github.

Select source branch and target branch. Source branch is our local branch and target branch is openlookeng/hetu-core master

Give Title as [<issue-id>] <title-name>

Provide fixes #<issue-id> and fill other required information related to PR in order to the issues fix to PR.

**Refer**: https://www.javatpoint.com/git-cheat-sheet