GitHub Workflow for akcse Contributors

Introduction

To commit code to the akcse repository, you will need to follow these steps and practices. The instructions show you how to perform these operations using VS Code. You can also implement this workflow from the command line if you have Git installed.

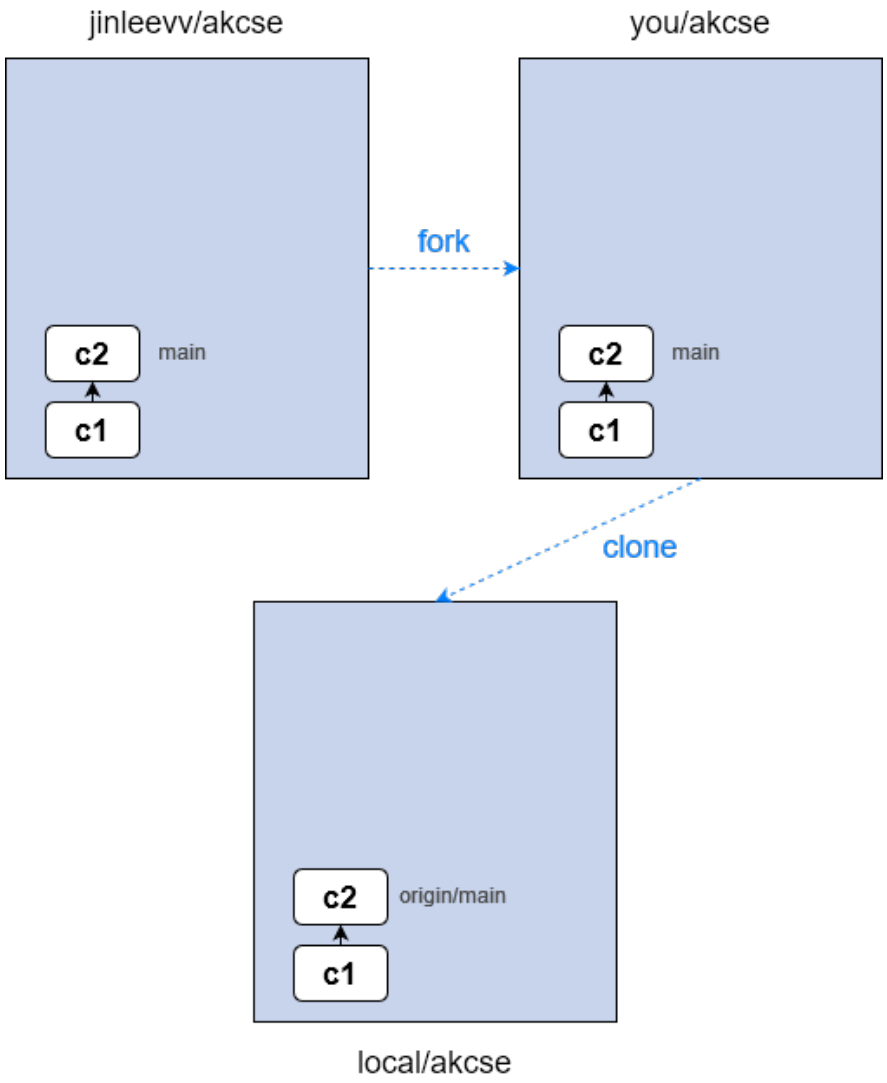
1 Fork and clone the akcse Repository

This will create your own copy of the akcse Git repository on GitHub and then make a clone of your fork on your computer. Having your own copy of the repository means you can make changes to the project without any risk of impacting the original code base. It is also necessary because you cannot commit your changes directly to the akcse repository.

1. Click on the Fork button on the top right of the akcse GitHub page.
2. On the GitHub page for your fork, click on the Clone button, then clone the repository to your computer. Alternatively, open a new VS Code window, click on Source Control and then Clone Repository.
3. In the popup, select Clone from GitHub, and search for your fork, yourUsername/akcse and select the matching result. Clone all branches.

See also: GitHub Guides: Forking Projects

At the end of the fork and clone operations, you will have a copy of the akcse project on GitHub (denoted you/akcse , where you refers to your GitHub username), and a clone of this fork on your computer (denoted local/akcse). In this diagram, each white box represents a commit, and the names in dark blue represent a branch pointer. In the local clone, the name of the branch prefixed with origin/ refers to the remote branch on the origin repository



2 Add A Reference to the Original akcse Repository

Your akcse fork is a separate repository from the original. It is possible that other contributors will make changes to akcse after you forked it. To see these changes and be able to build on them, you need to be able to synch your repositories with the original repository.

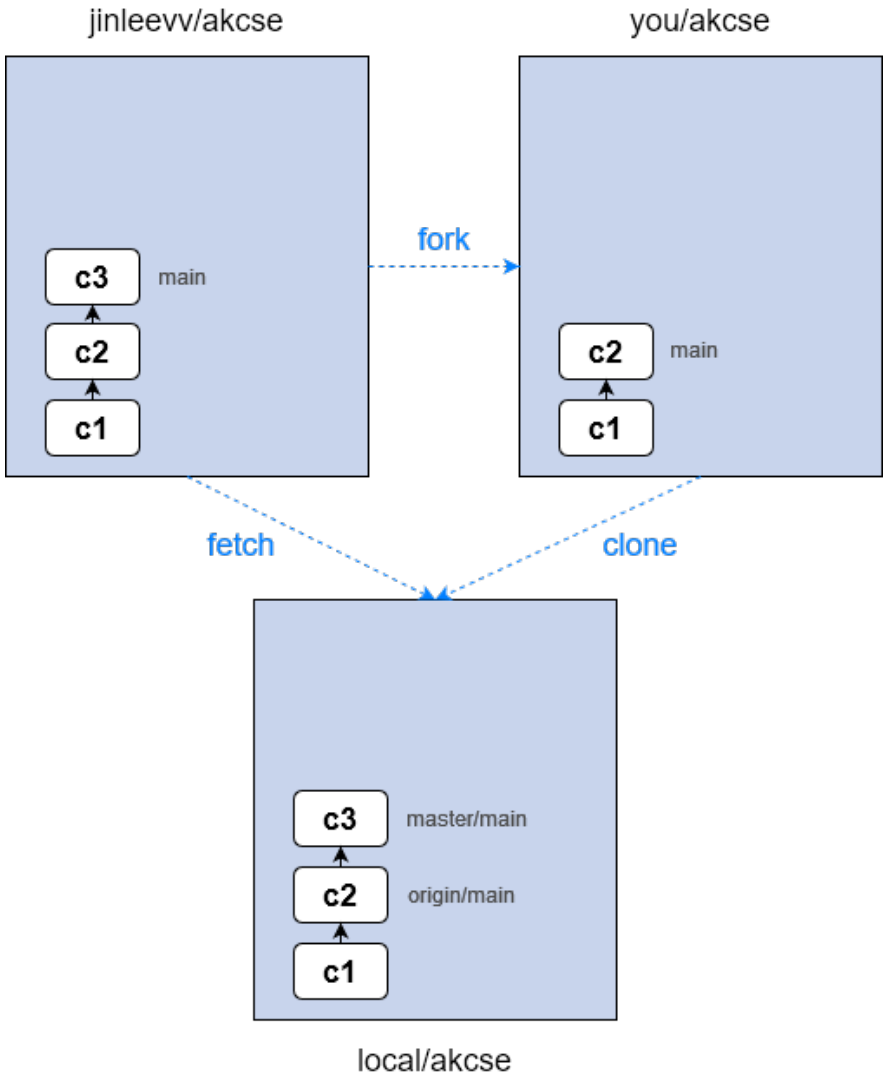
1. In the Source Control view, click on the Source Control tab.
2. Click on the ... button, select *Remote* | *Add Remote...*
3. Select Add remote from GitHub, search jinleevv/akcse, then click on the matching result.
4. Enter a name for the remote branch. You can name it master.

You can now fetch any new data from the original akcse repository at any point. In the rest of this document this will be referred to as **fetching**. To do so:

1. On the Source Control tab, Click on the ... button and select Fetch.
2. Select master, the remote branch we previously added.

All new changes from the original akcse repository will be downloaded to your local repository. In the figure below, a new commit (c3) was added to the jinleevv/akcse repo, and fetching it downloads it

to your local repository. This commit is in a branch named master/main, which is now one commit ahead of your origin/main branch.

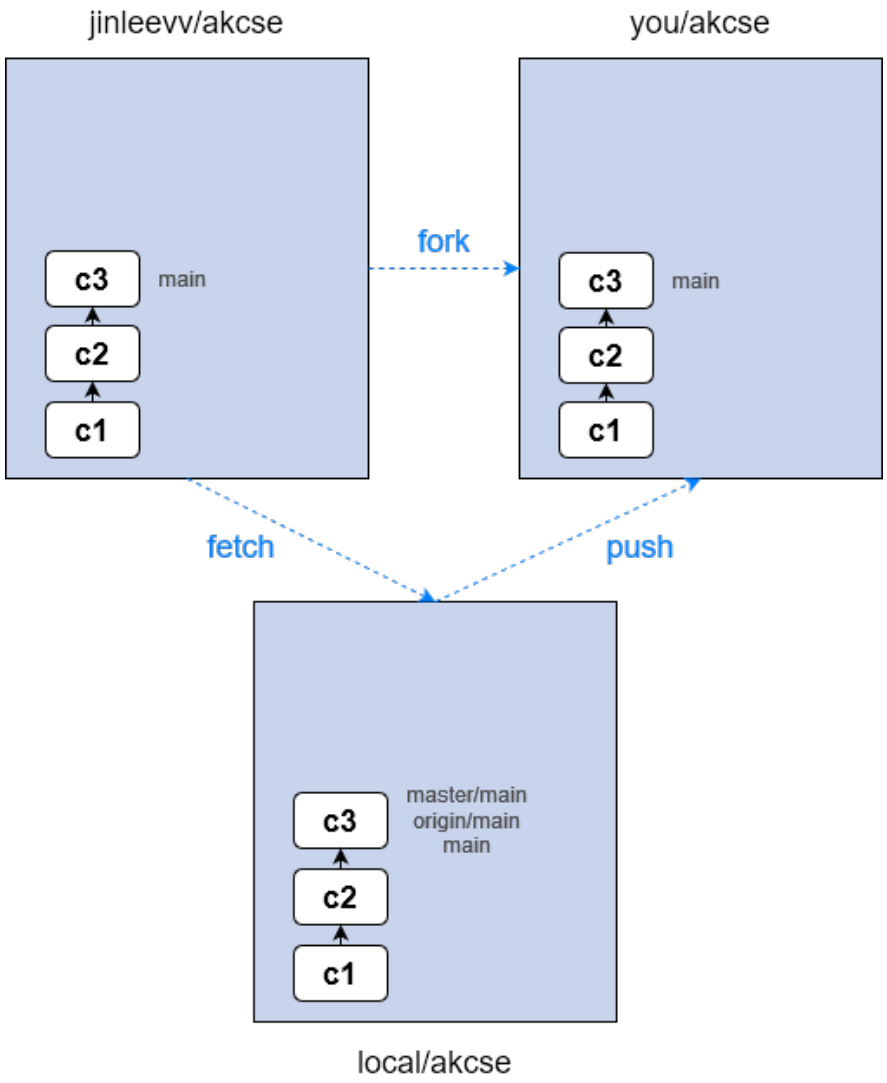


3 Integrate Changes from the Master Main Branch

New changes may be available on the original akcse repo. To obtain them:

1. Fetch
2. Make sure the local main branch is checked out
3. On the Source Control tab, Click on the ... button and select *Branch* | *Rebase Branch...*
4. Select master/main in the popup.
5. Click on the Sync Changes button in the Source Control tab.

Both your local and remote main branches will now be in sync with the akcse repository. We will refer to this step as **syncing the main branch**.



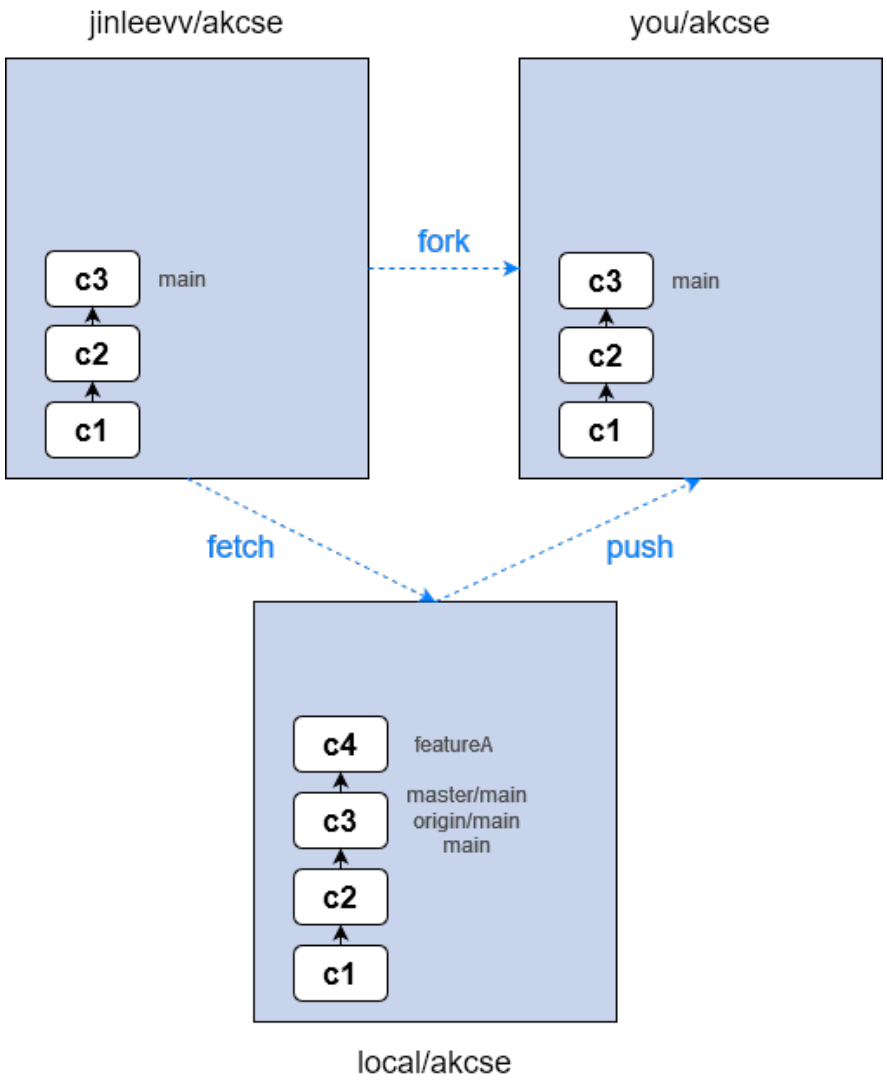
4 Create a Local Feature Branch

Commit all your changes to a ***local* feature branch**. This is a branch that only resides on your computer, so there is no consequence to anything you do on this branch, and no one outside you can see these changes unless you push the branch to your fork (see below). You can create as many local feature branches as you want. For example, you can create local feature branches to experiment with various ideas. If things lead nowhere, you can simply delete or forget about this branch and create a new one.

Create your local feature branch from the latest commit in the main branch.

1. Synch the main branch (as described in Section 3)
2. Assuming the local main branch is checked out, on the Source Control tab click ..., then select *Branch* | *Create Branch...*
3. Give your branch a descriptive name, for example featureA

You can now commit to this branch.

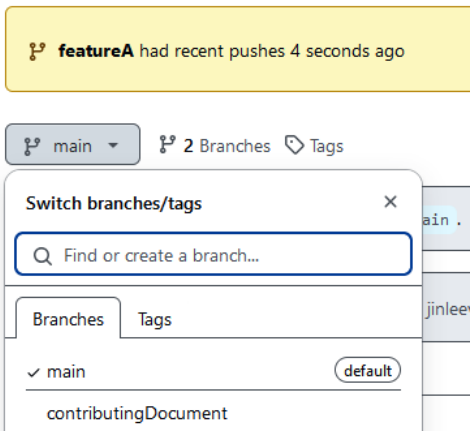


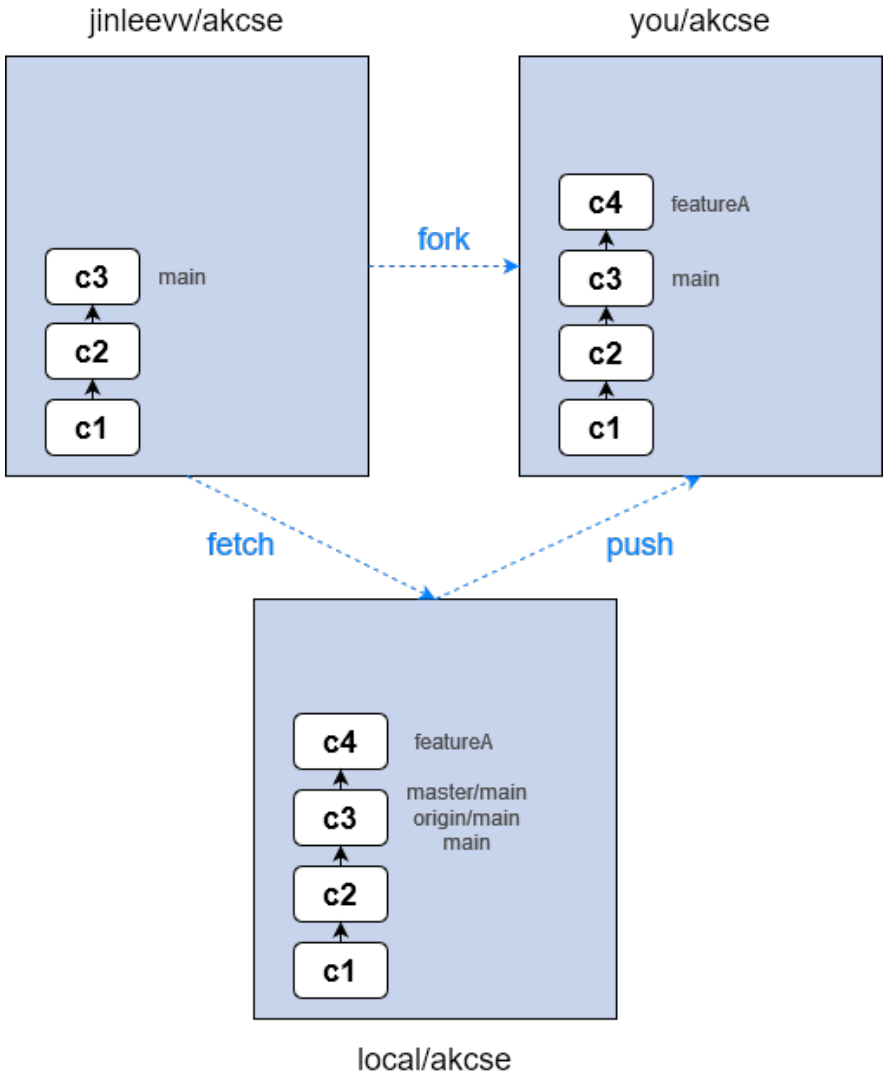
5 Share Your Work

Once you are ready to share your work publicly, you can push your branch to your fork. All the commits on this branch will now become publicly visible.

1. Make sure your feature branch is checked out.
2. In the Source Control tab, click on the Publish Branch button and select origin

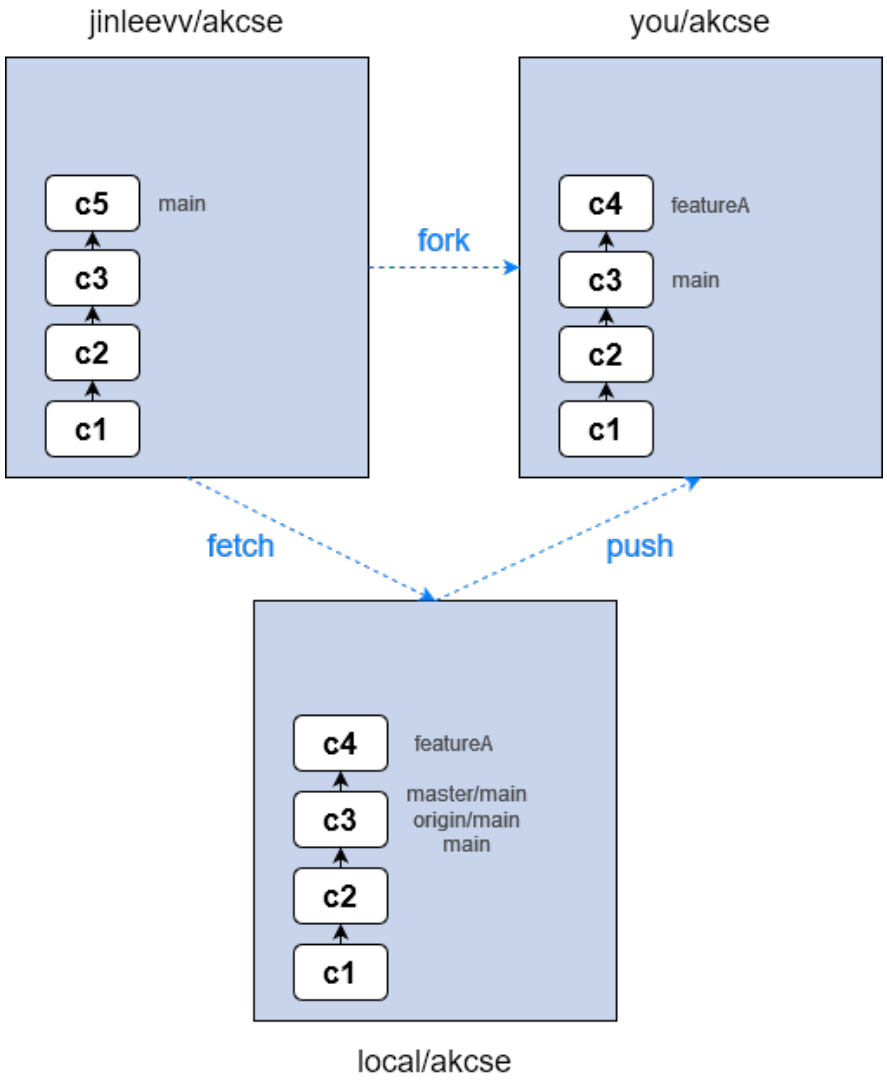
Your branch is now pushed to your fork. If you go to GitHub, you can now see and select this branch. Now when you make changes to your feature branch, in addition to committing them, you also have the option of pushing them to your fork.


featureA had recent pushes 4 seconds ago
main
2 Branches
Tags
Switch branches/tags
Find or create a branch...
Branches
Tags
main
default
contributingDocument

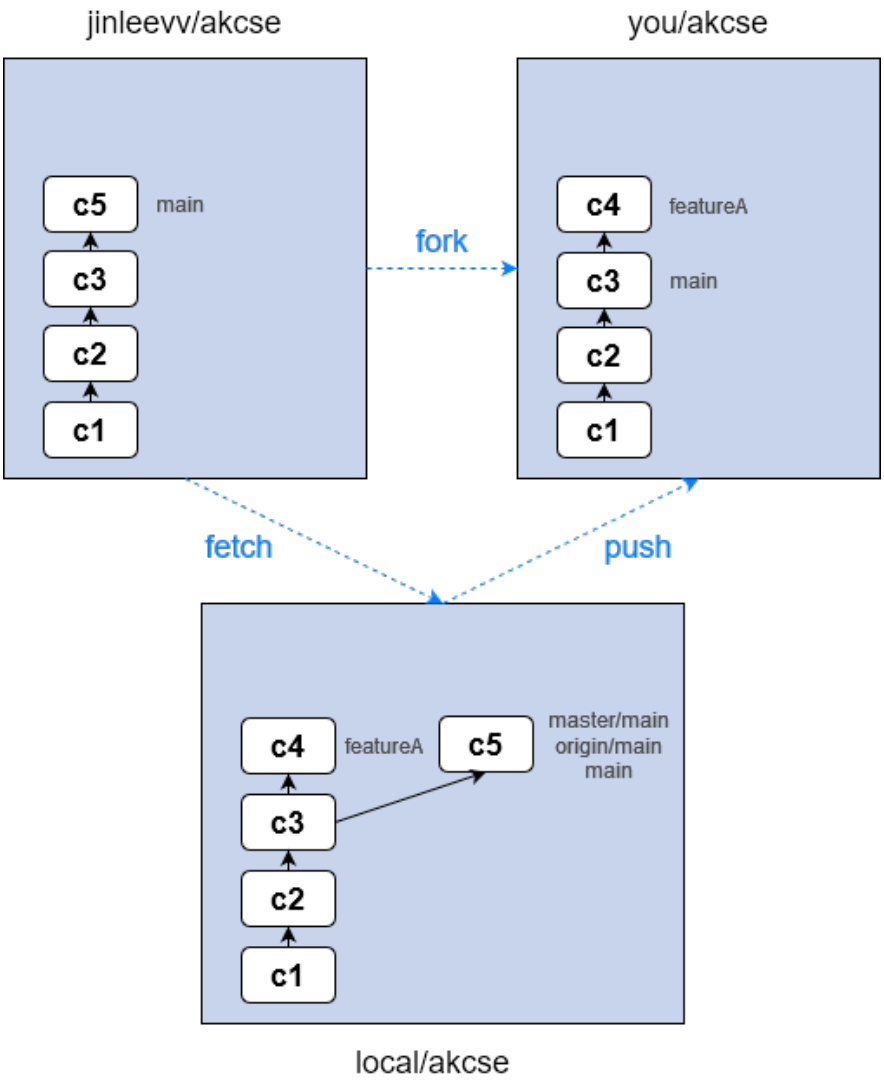

jinleevv/akcse
you/akcse
c3
main
c2
c1
fork
c4
featureA
c3
main
c2
c1
fetch
push
c4
featureA
c3
master/main
origin/main
main
c2
c1
local/akcse

6 Opening a Pull Request

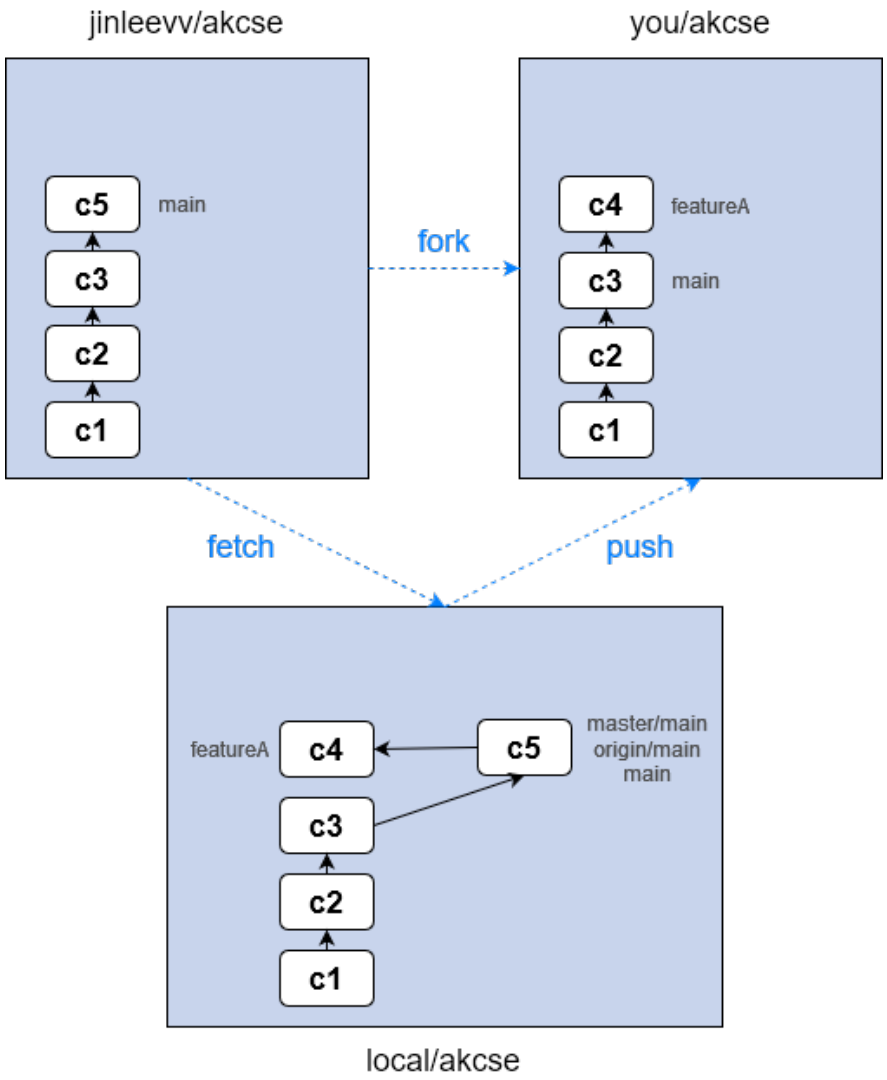
To integrate your changes into the akcse code base, you need to open a pull request. This will allow the project owner to review your changes, ask for modifications if need be and, ultimately, merge your commits into the main branch on the original akcse repo. However, before you do this you must **rebase your feature branch** because while you were working, the main branch on the main repo may have moved ahead.



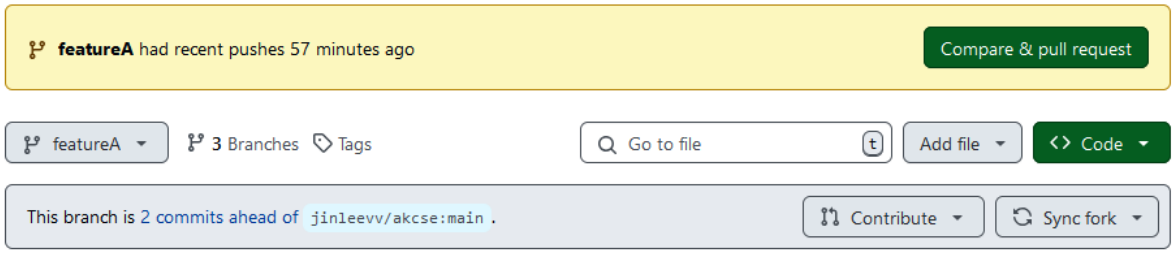
1. Sync the main branch as described above in Section 3.



2. Check out your feature branch.
3. Click on ... and select *Branch* | *Rebase Branch...* and select you main branch
4. Assuming there are no conflicts, your feature branch will now include the commits from the main branch.



5. Push your feature branch to your fork: click on Sync Changes
6. Go to your fork on GitHub
7. Select your feature branch and click on the Pull request button. If you pushed your branch recently, you can also use the big green shortcut button.



8. Make sure to select the main branch as a base. All contributions to akcse must go through the main branch.
9. Describe your pull request and click Create.