GitHub 101: Keep better track of your code

Sinziana Cristea

24th October 2019

About Me

- Sinziana Cristea

- PhD Student @Leiden University, Division of Systems Pharmacology & Biomedicine

- Use clinical data to build NLME models to optimize antibiotic treatment of newborns

Universiteit
Leiden

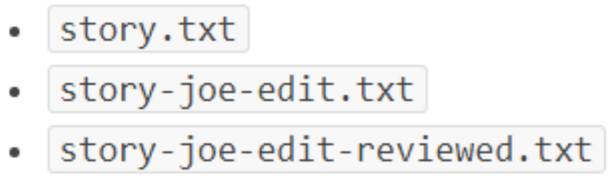
Contact: sanziana.cristea@gmail.com
Phone: 0626368948

What will you learn today?

- Basic git terminology and commands
- Working with local and remote repositories
- Working with branches
- Pull request
- Bonus: Solve conflicts when merging

Why is tracking your changes important?

- What is your current method of keeping track of your changes to your scripts?
- How do you collaborate with someone on a project?



- `story.txt`
- `story-joe-edit.txt`
- `story-joe-edit-reviewed.txt`

Solution: Git & GitHub

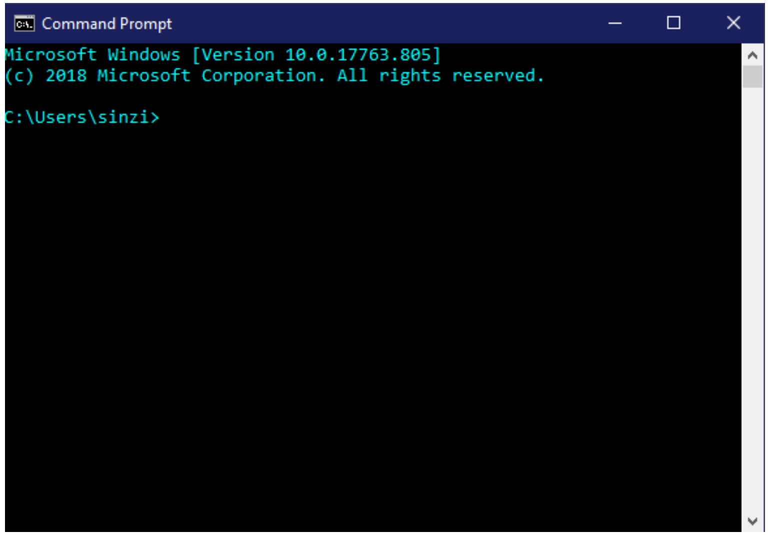
- Git: version control system
  - Records changes to files over time
  - e.g. save with different names.
- GitHub:
  - GitHub is a code hosting platform for version control and collaboration.
  - It lets you and others work together on projects from anywhere.
- Getting started:
  - With an UI: GitDesktop
  - UI has the basic functionalities
  - Does not require coding
  - (-) sometimes you can run into problems that are not solvable with UI:
    - → command line in shell/command prompt

Command prompt/Shell?

- Command window where you can type in command lines to be executed

- Useful when you want to automate tasks, execute scripts, etc..

- Git commands should start with *git* followed by what you want to do
  - This tutorial will include both UI and command lines in git, side by side.



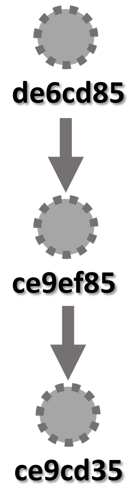
```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sinzi>
```

What are repositories (repos)?

- Similar to folder where you keep all the files necessary for a project
    - Files to keep track of
    - Files to share with others
- E.g.: images, videos, spreadsheets, data sets, scripts – anything your project needs.
- A *README*, or a file with the project description.
- Managing repositories:
    - Remote repositories (e.g.,GitHub)
        - Owner = You
        - Owner = Someone else (e.g., Open source projects)
    - Local repositories – (your computer(s))
        - Owner = You

How are your changes tracked?

- You start with an empty repo to which you add your files.
- When you save these changes you perform a *commit*.
- Git takes care of:
  - **Who** made the commit
  - **When** that commit was made
  - Assigning an unique ID (hash) to every commit
- You take care of:
  - Code changes
  - Commit message
    - Description, reason for changing code
- Commits are built on top of each other

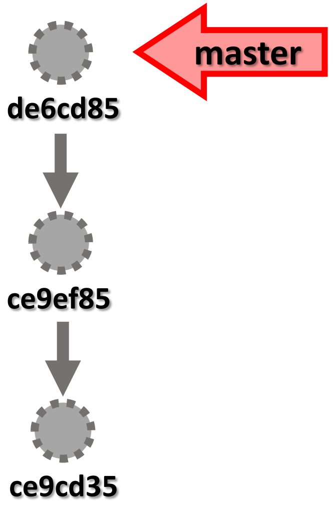


How are your changes tracked?

- *History* of your commits is tracked by a branching system.
- A *branch* is a label for a commit
  - Use: when you want to work on multiple versions of a script in parallel

  e.g.: Implement a new feature, bug fix, etc.

- The default branch is called *master*
- When switching from one branch to another we *checkout*.



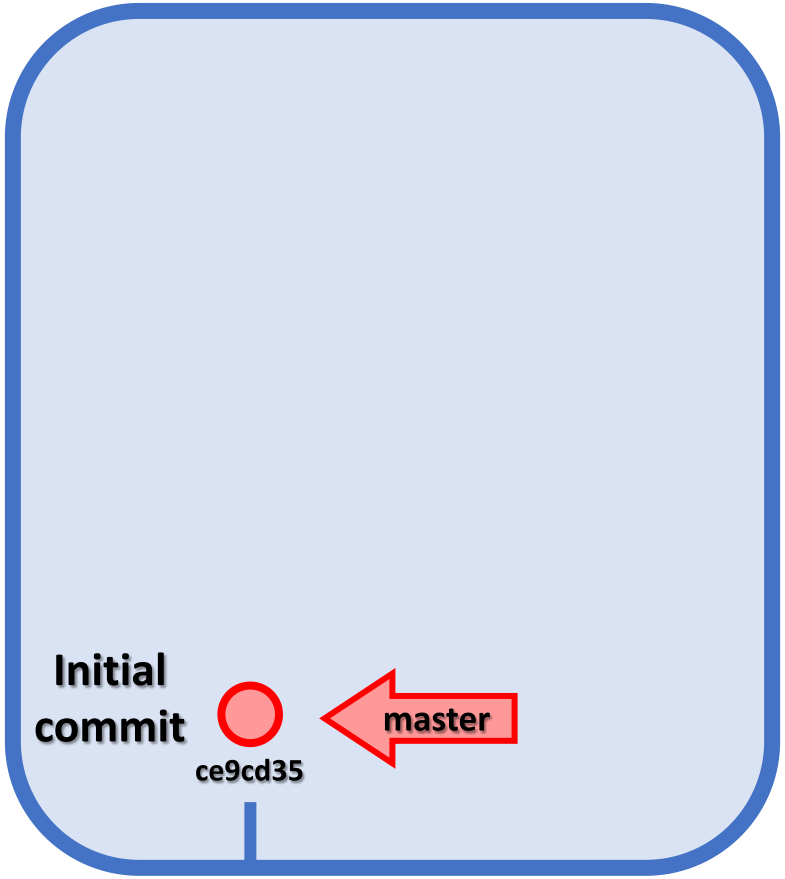
How are your changes tracked?

What happens when you create your remote repo?

- Readme file is created and added to the empty repo
- This will be the ***initial commit***
- This will be labeled as the default branch *master*

Remote Repo

GitHub



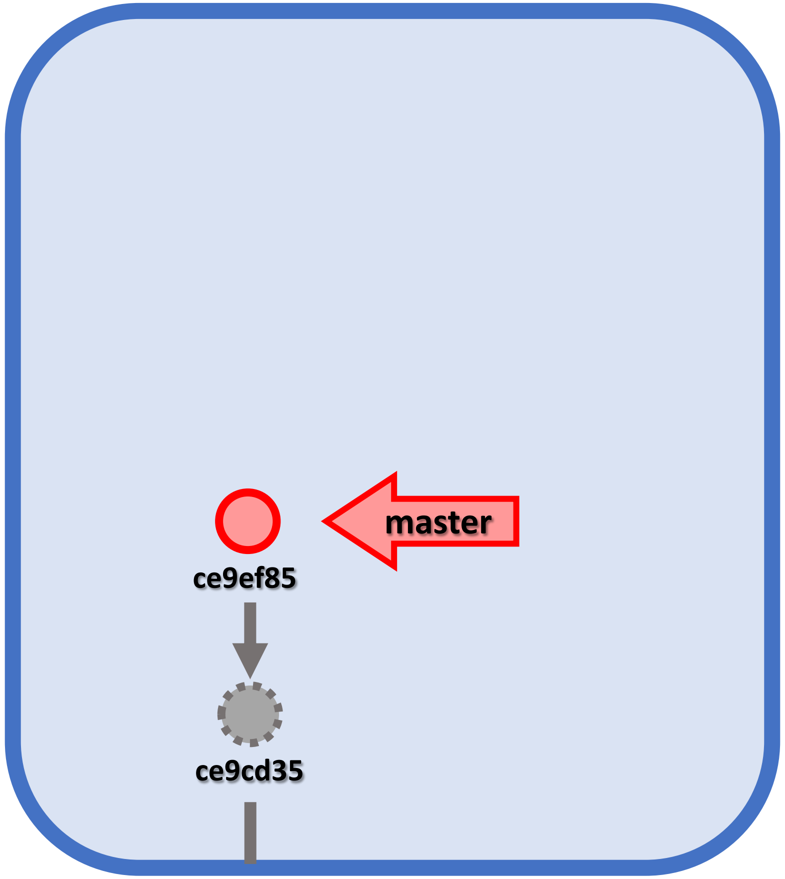
How are your changes tracked?

What happens when you make a change/commit?

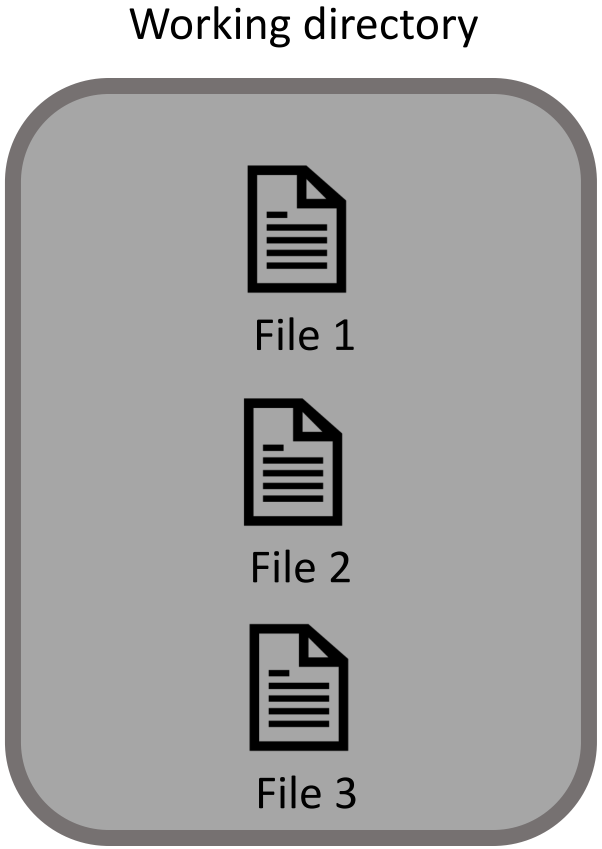
- Readme file is changed
- This will be the second commit
- Git:
  - Who, when, ID for commit
- You:
  - What, why (message)
- The branch *master* will go up one commit

Remote Repo

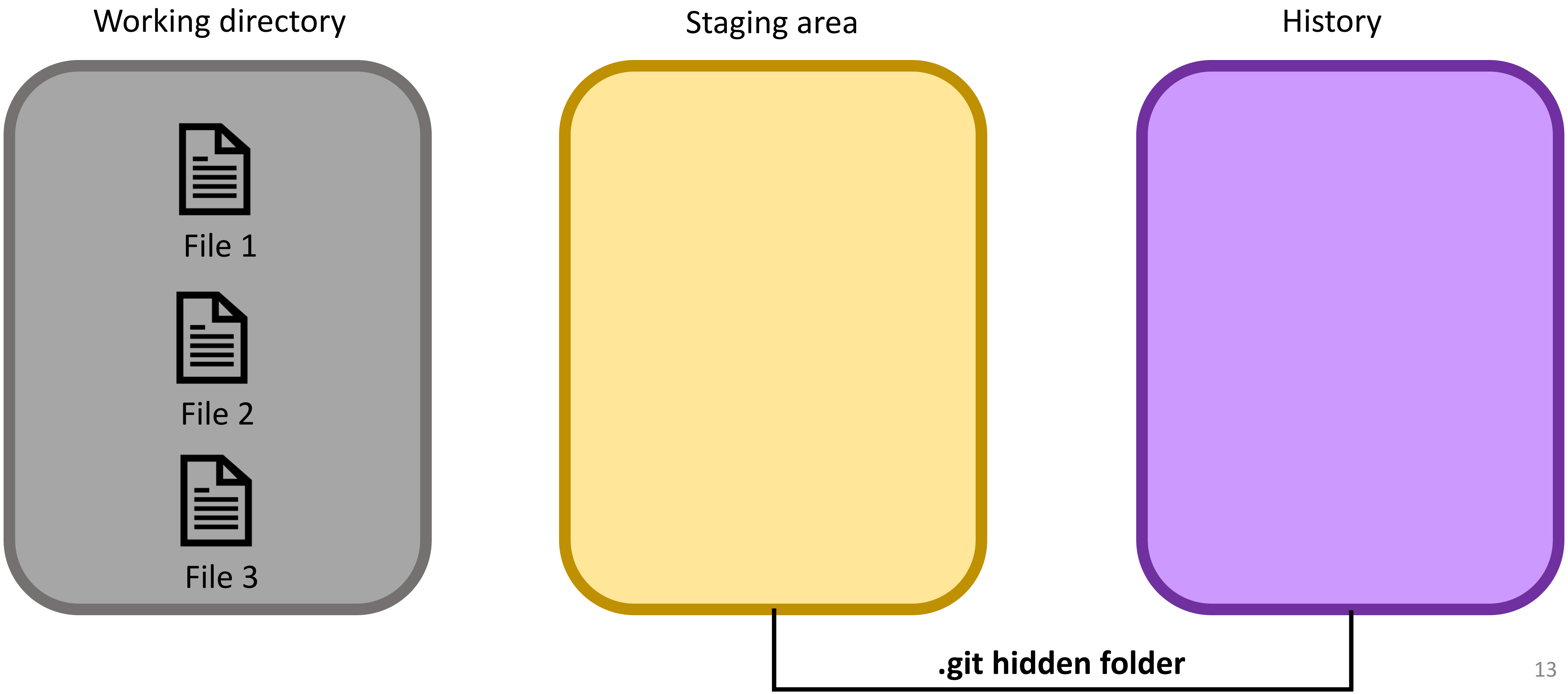
GitHub



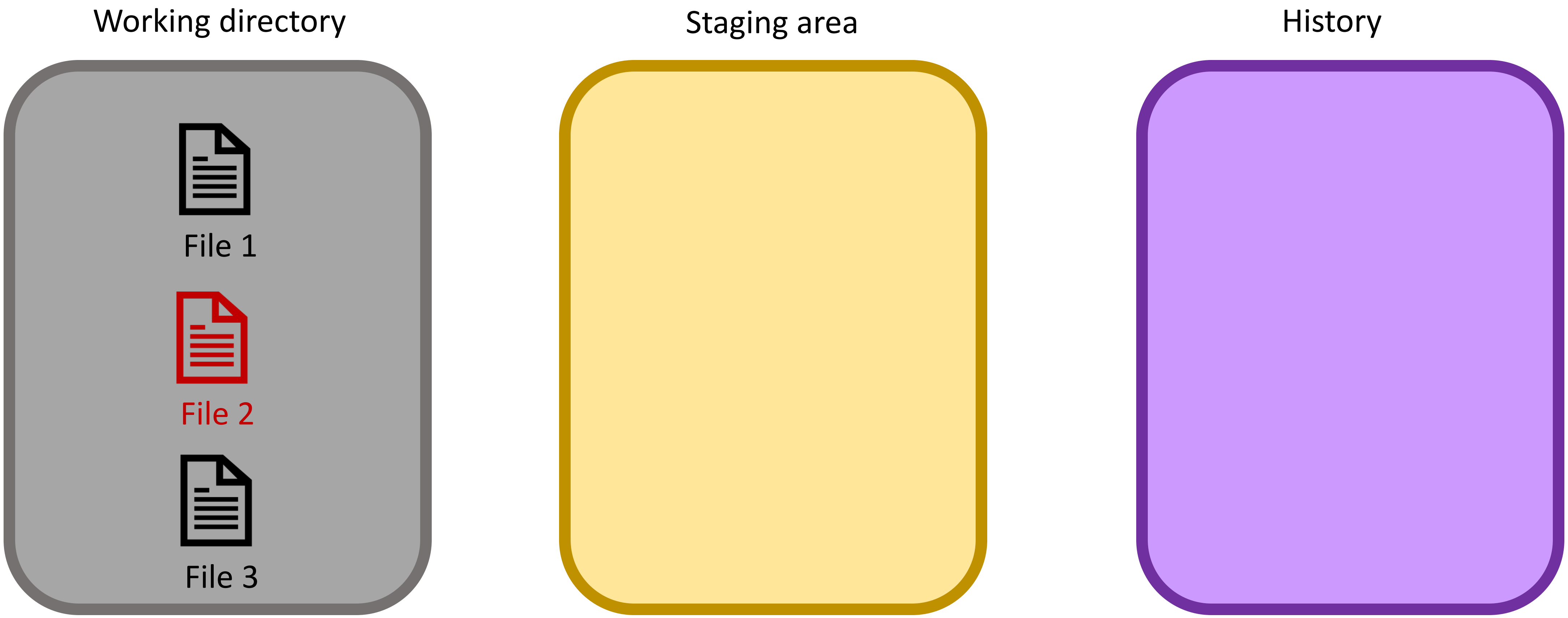
The working areas of a repo



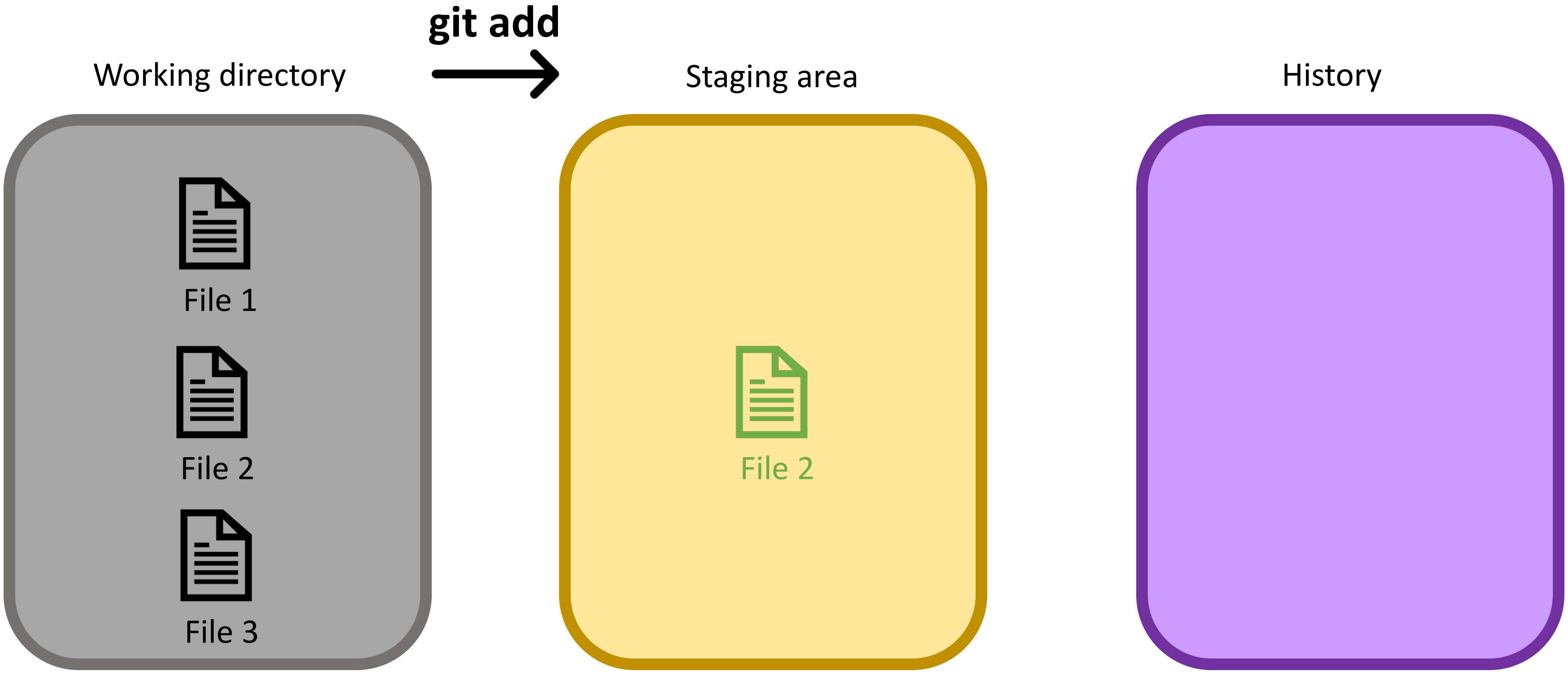
The working areas of a repo



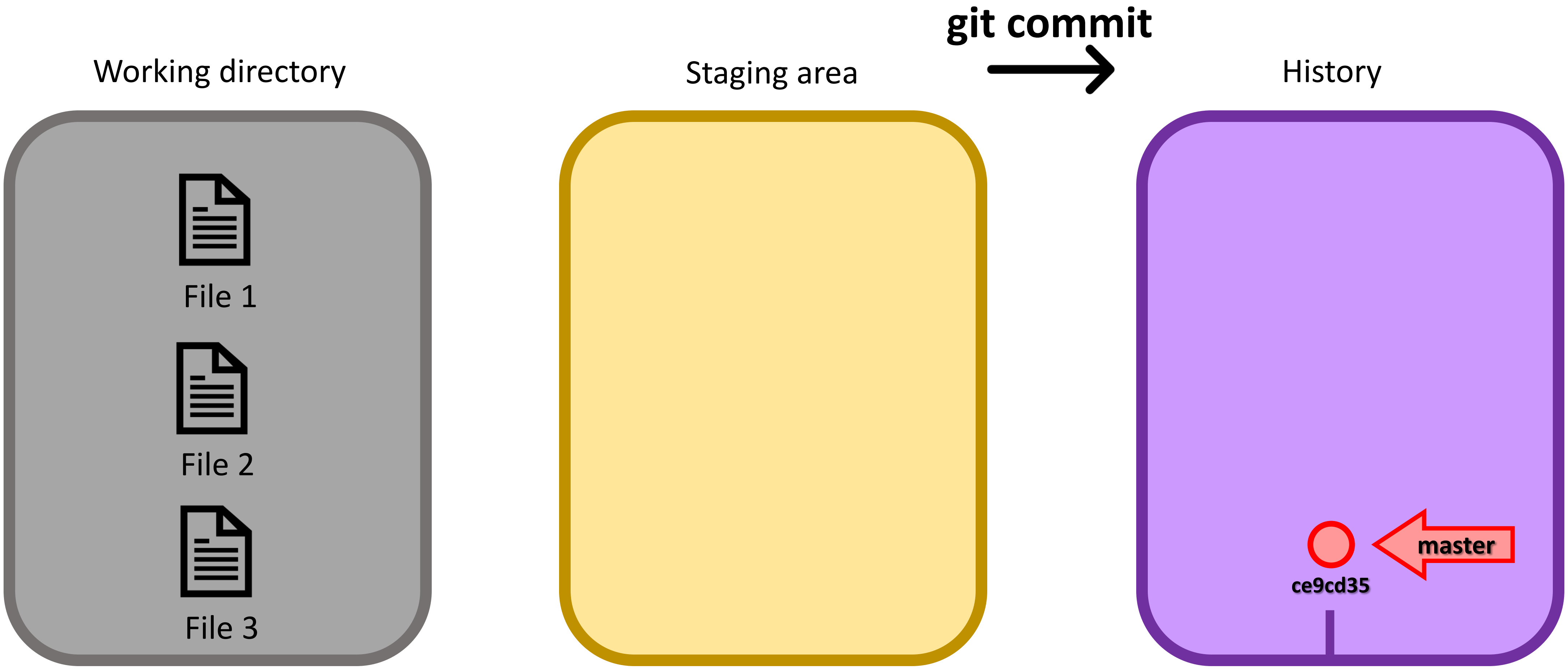
The working areas of a repo



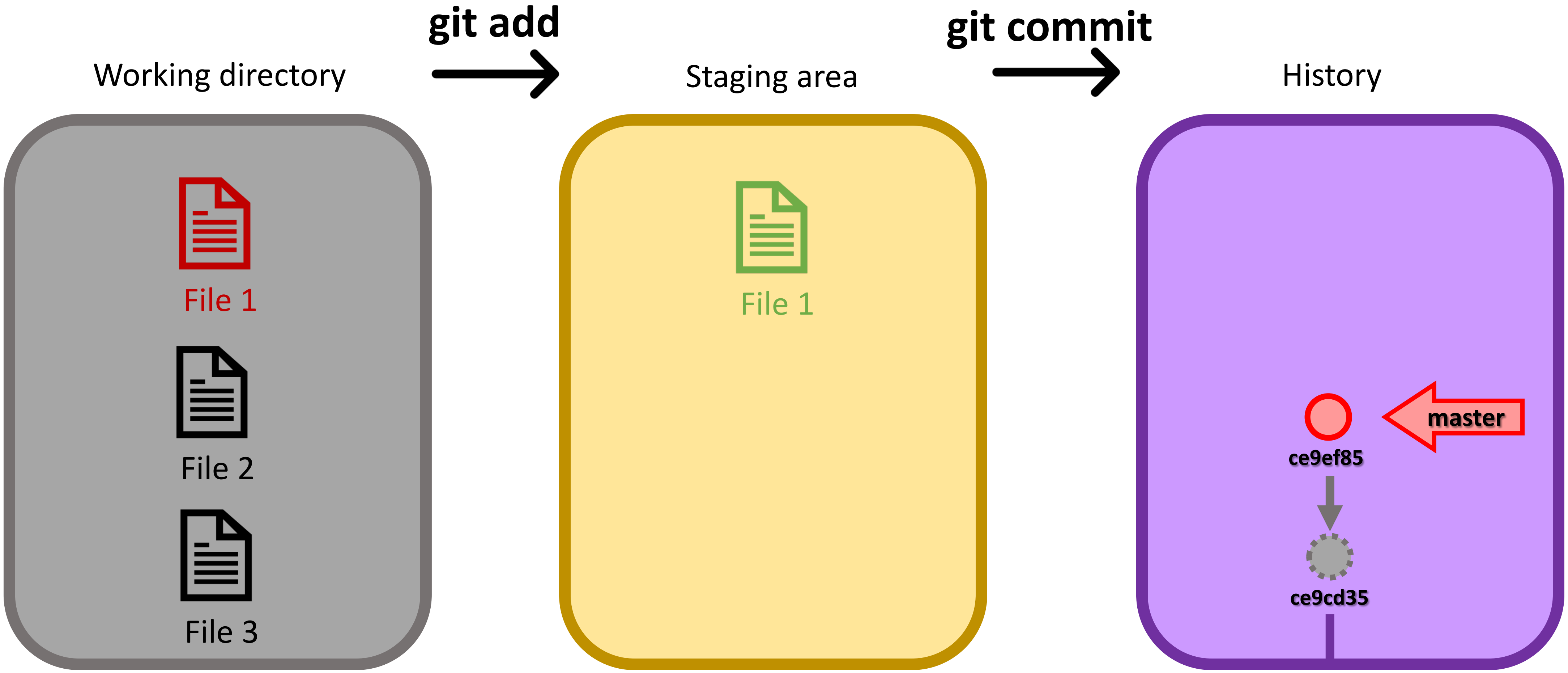
The working areas of a repo



The working areas of a repo



The working areas of a repo

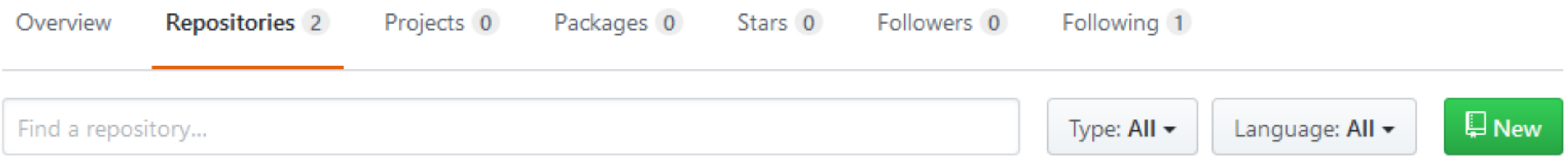


Basic git commands on GitHub

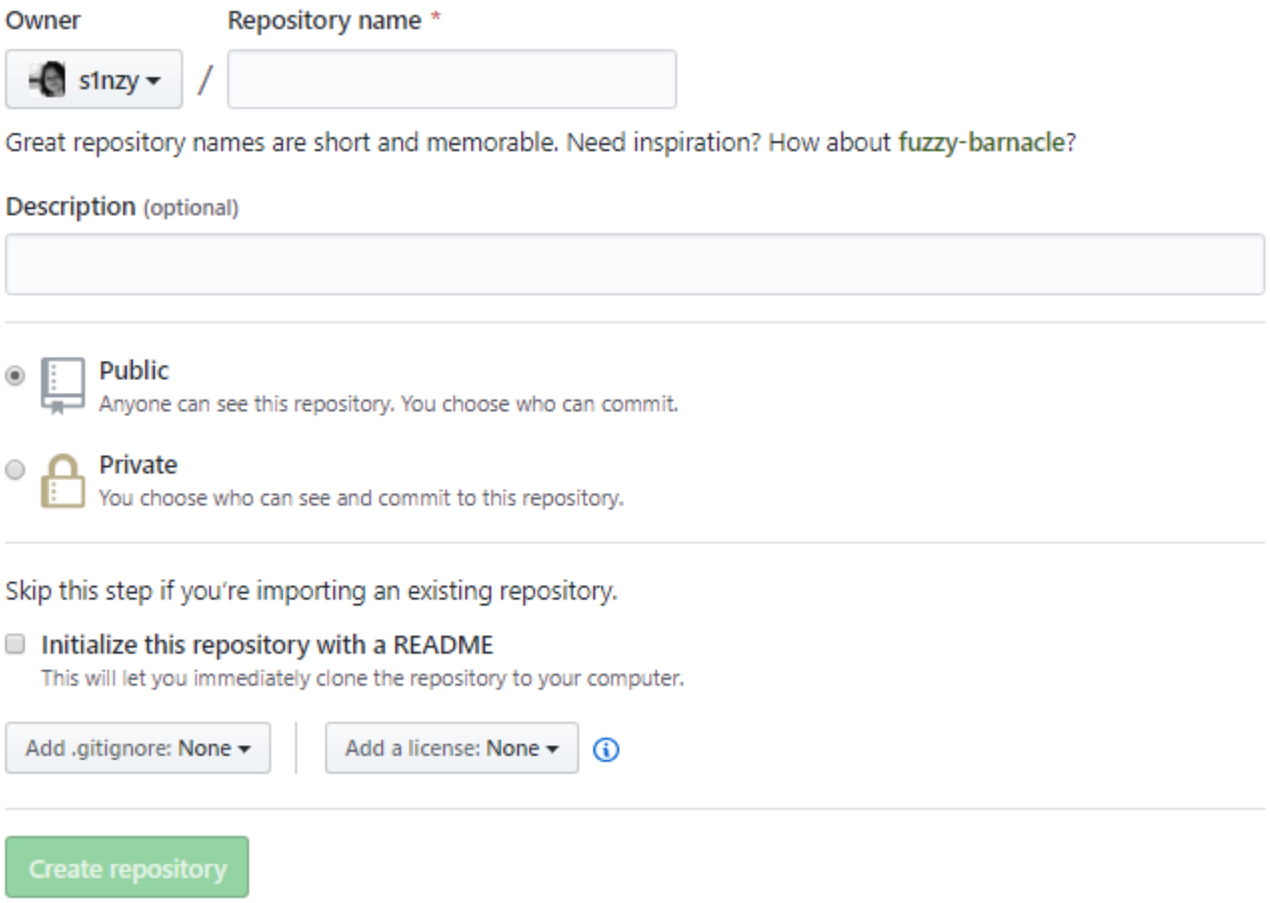
- Create remote repository on GitHub
- Make commits and look at the history
- Copy remote repository to your computer (clone)

Create a remote repo on GitHub

- Go to your profile on GitHub
- Select Repositories → click on New to create a new repo



Create a remote repo on GitHub



Owner: s1nzy ▾ / Repository name *

Great repository names are short and memorable. Need inspiration? How about fuzzy-barnacle?

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

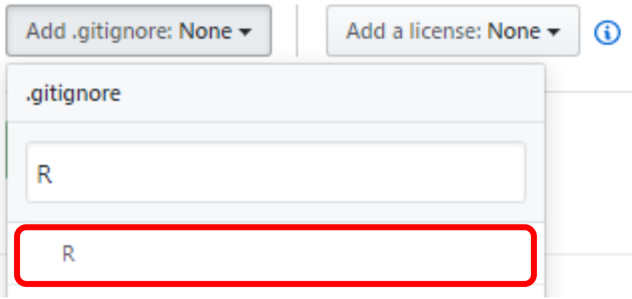
Skip this step if you're importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾

Create repository

- Repository name:
  - e.g., GitTutorial_SC_11
- Description:
  - Basics of Git and GitHub
- Public
- ✓*Initialize this repository with a README*
- Click on



Add .gitignore: None ▾ | Add a license: None ▾

.gitignore

R

R

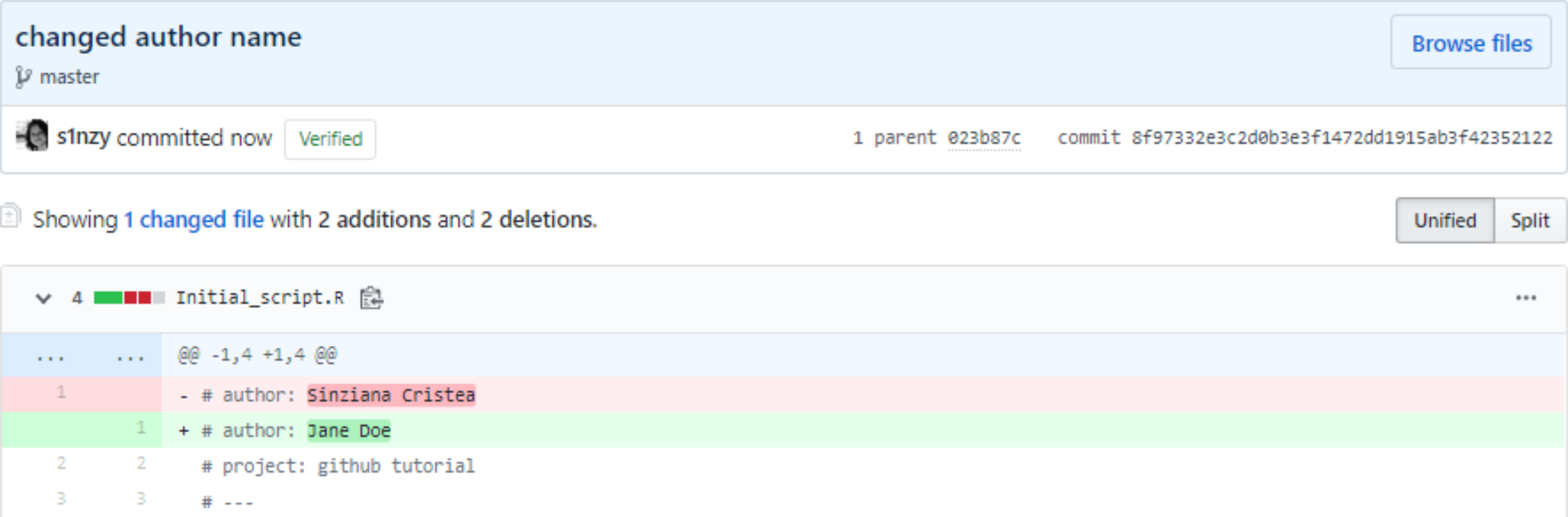
Commits in GitHub

- Change 1: Upload the R script you got in preparation for this workshop → Commit 1
- Change 2: Change the author name to your (nick)name→ Commit 2
    - Click on the R script name
    - You can see the content of the script
    - Click the *Edit* button on the right ✎
    - Change the author name

Checkout the history of our commits

- Go to your repository
- Click on the Rscript you just changed
- You should now see the content of the script
- Click the *History* button on the right
  - Who made the commits
  - When these were made
  - The unique ID numbers associated with each commit
- Click on the latest commit
  - Red background: the initial text
  - Green background: your change

Checkout the history of our commits



changed author name

master

s1nzy committed now Verified

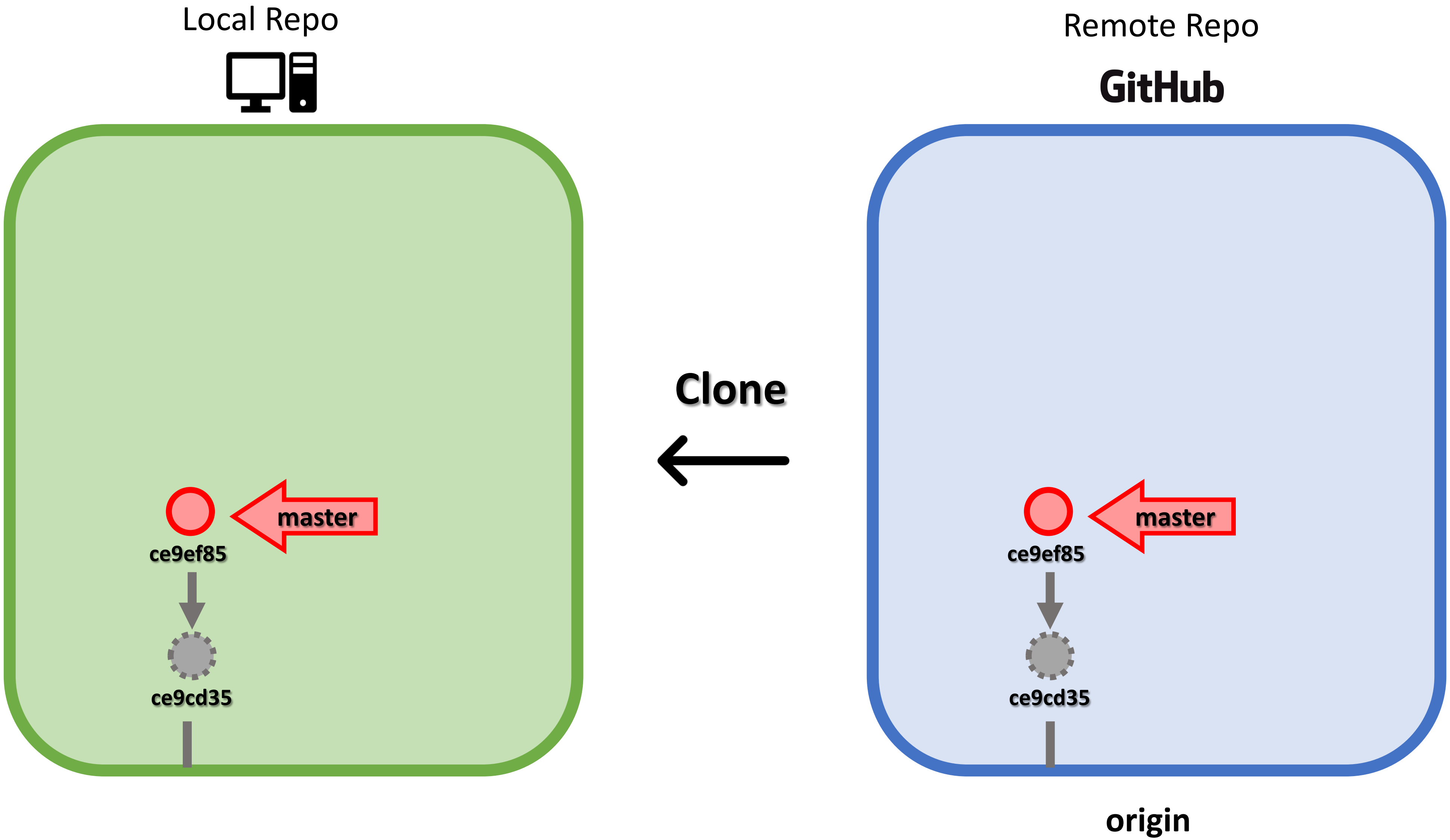
1 parent 023b87c commit 8f97332e3c2d0b3e3f1472dd1915ab3f42352122

Showing 1 changed file with 2 additions and 2 deletions.

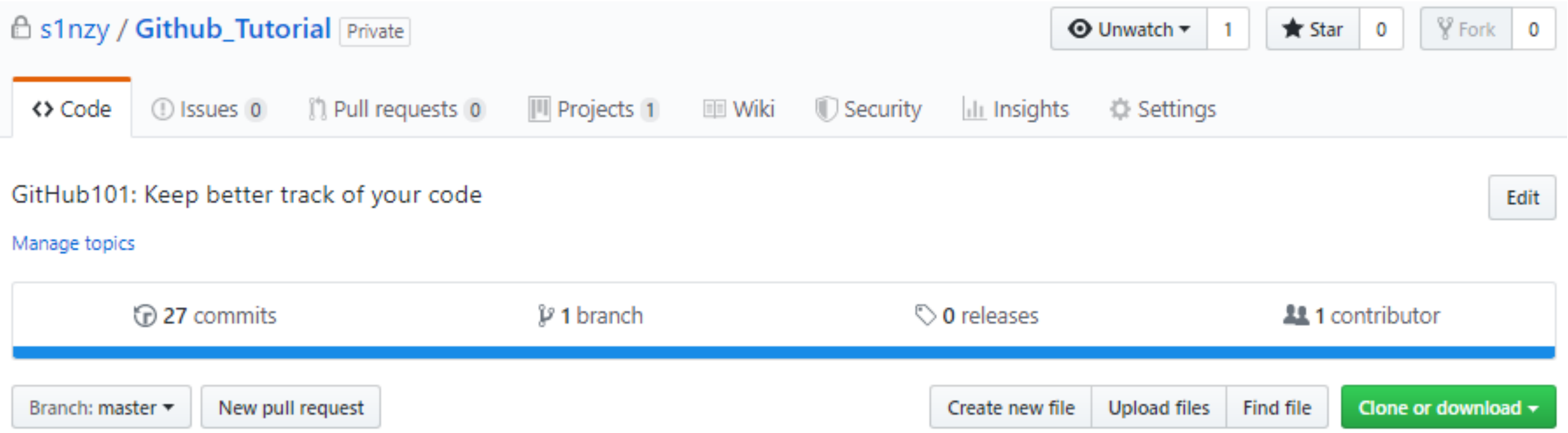
4 Initial_script.R

```
...   ...   @@ -1,4 +1,4 @@
 1          - # author: Sinziana Cristea
        1   + # author: Jane Doe
 2      2     # project: github tutorial
 3      3     # ---
```

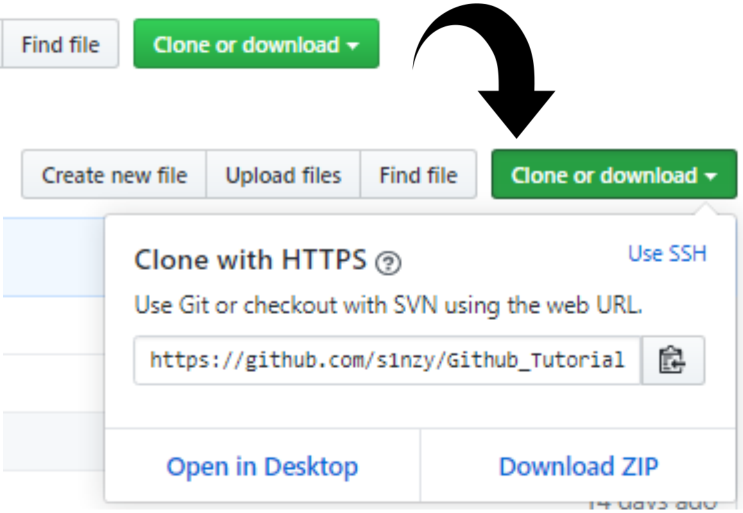
Remote repo → Local repo



How to Clone with GitDesktop (UI)?



GitHub101: Keep better track of your code



Use Git or checkout with SVN using the web URL.

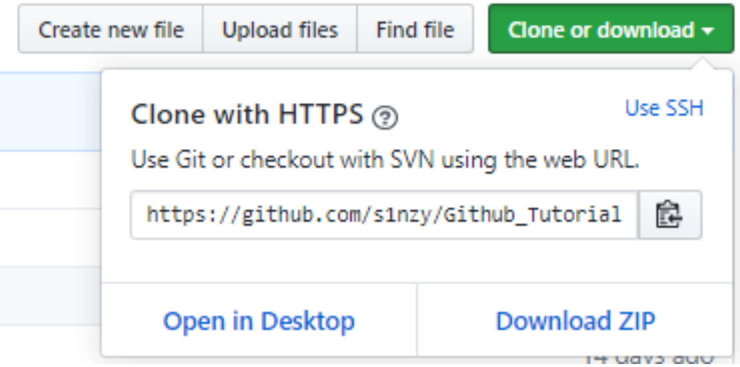
```
https://github.com/s1nzy/Github_Tutorial
```

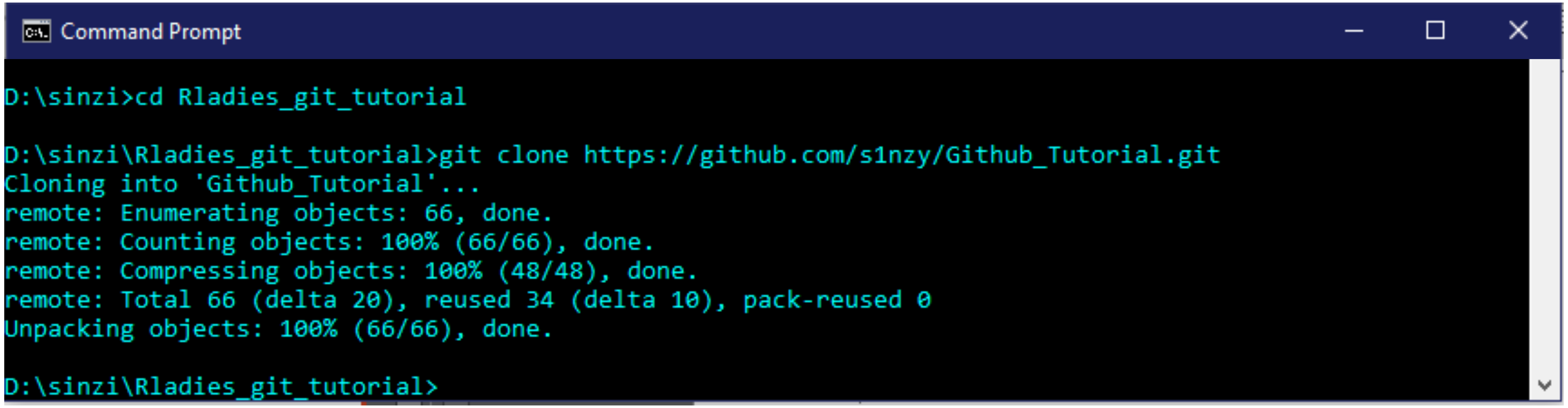
- Go to your GitHub repository
- Select the Code Tab
- Click on the *Clone or download* button
  - Open in Desktop ☜
  - Download ZIP

How to clone with the git command line?

- Go to the folder you want to clone the repo
- Command:

git clone https://github.com/s1nzy/Github_Tutorial.git



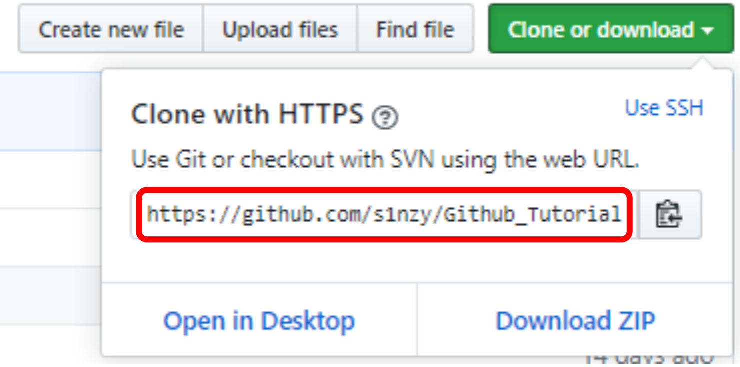


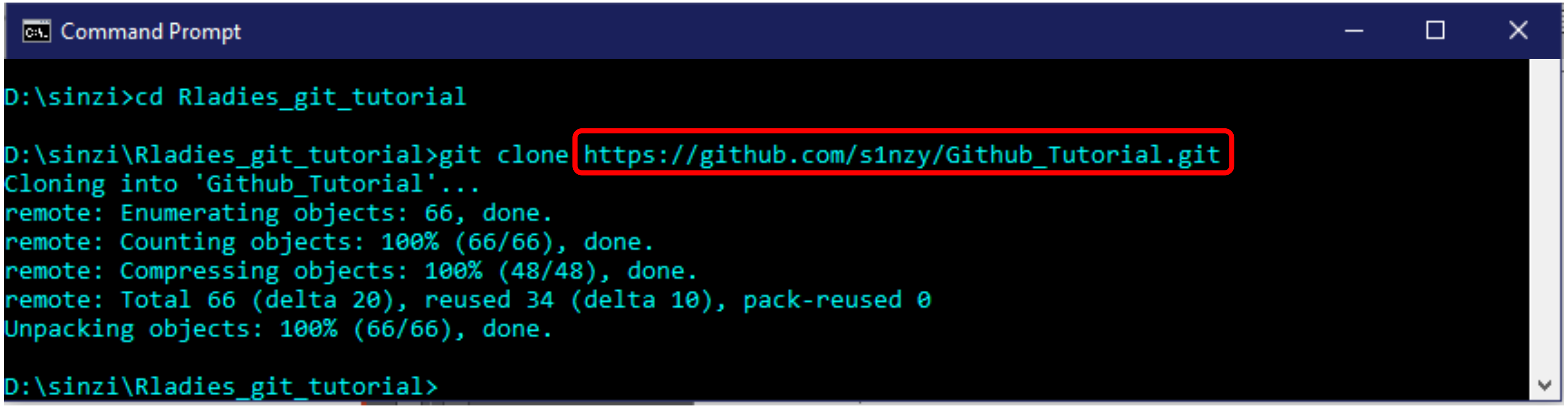
```
D:\sinzi>cd Rladies_git_tutorial

D:\sinzi\Rladies_git_tutorial>git clone https://github.com/s1nzy/Github_Tutorial.git
Cloning into 'Github_Tutorial'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 66 (delta 20), reused 34 (delta 10), pack-reused 0
Unpacking objects: 100% (66/66), done.

D:\sinzi\Rladies_git_tutorial>
```

How to clone with the git command line?

- Go to the folder you want to clone the repo
- Command:

git clone https://github.com/s1nzy/Github_Tutorial.git



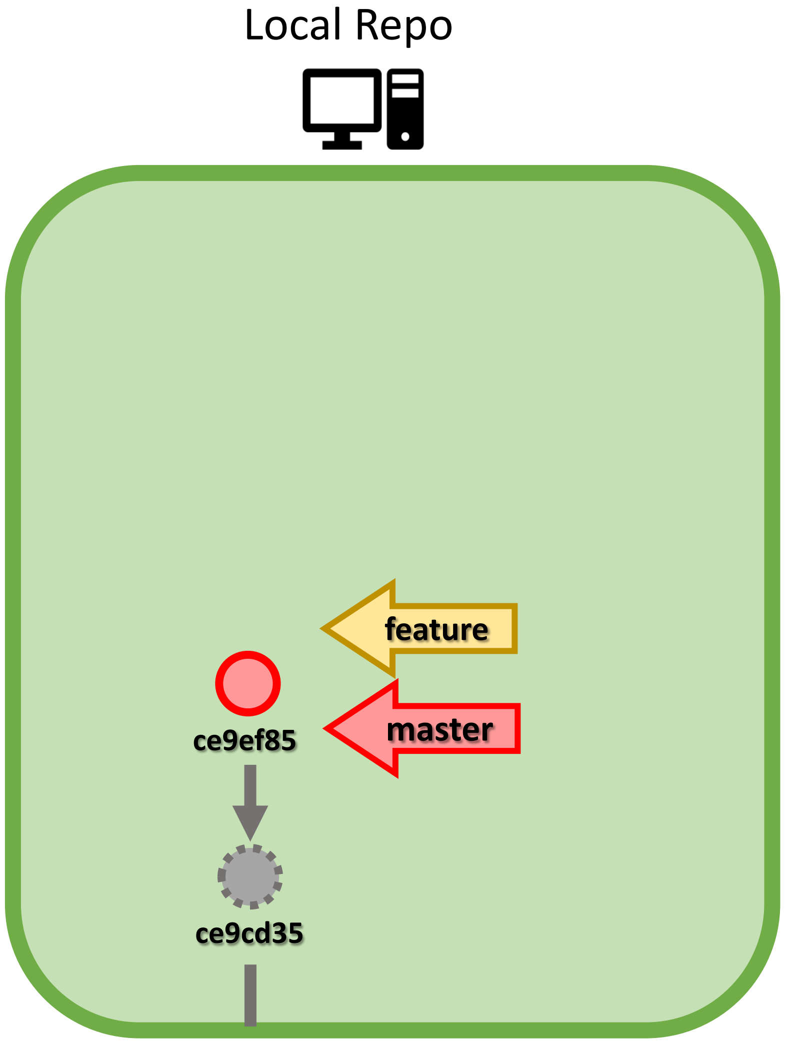


```
D:\sinzi>cd Rladies_git_tutorial

D:\sinzi\Rladies_git_tutorial>git clone https://github.com/s1nzy/Github_Tutorial.git
Cloning into 'Github_Tutorial'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 66 (delta 20), reused 34 (delta 10), pack-reused 0
Unpacking objects: 100% (66/66), done.

D:\sinzi\Rladies_git_tutorial>
```

Basic git commands on local repo

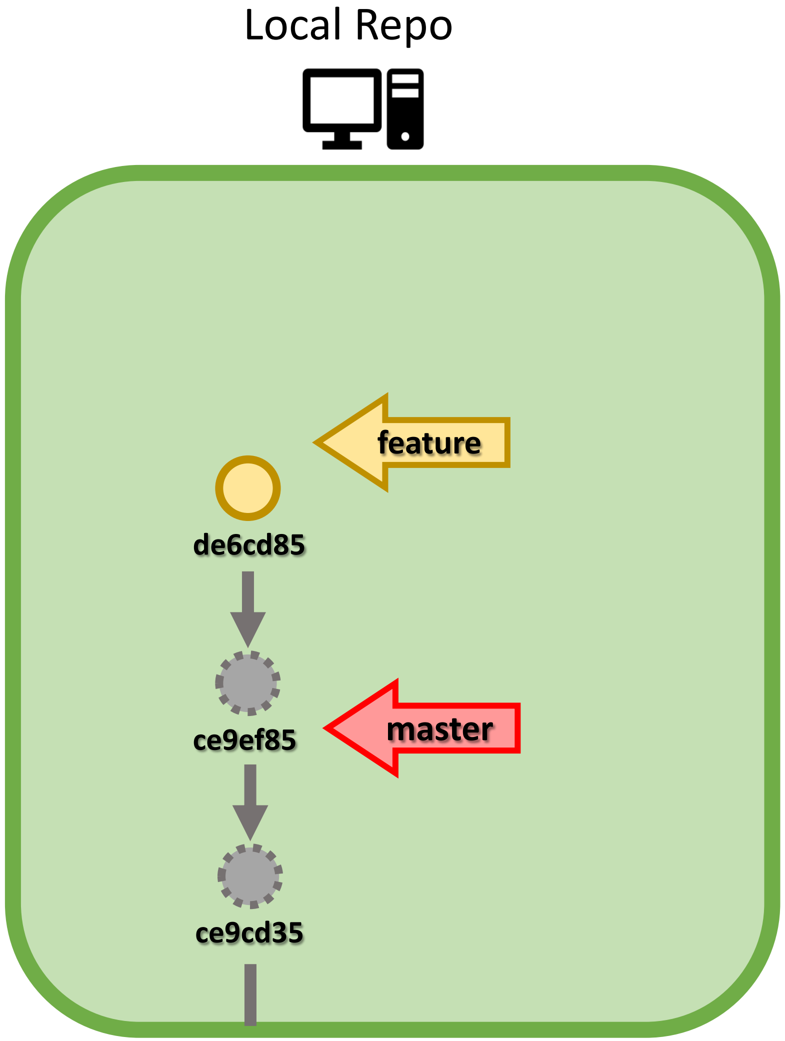
- Create a new branch
- Merge that branch into master branch
- Push local changes to remote repo

Branches & Merge



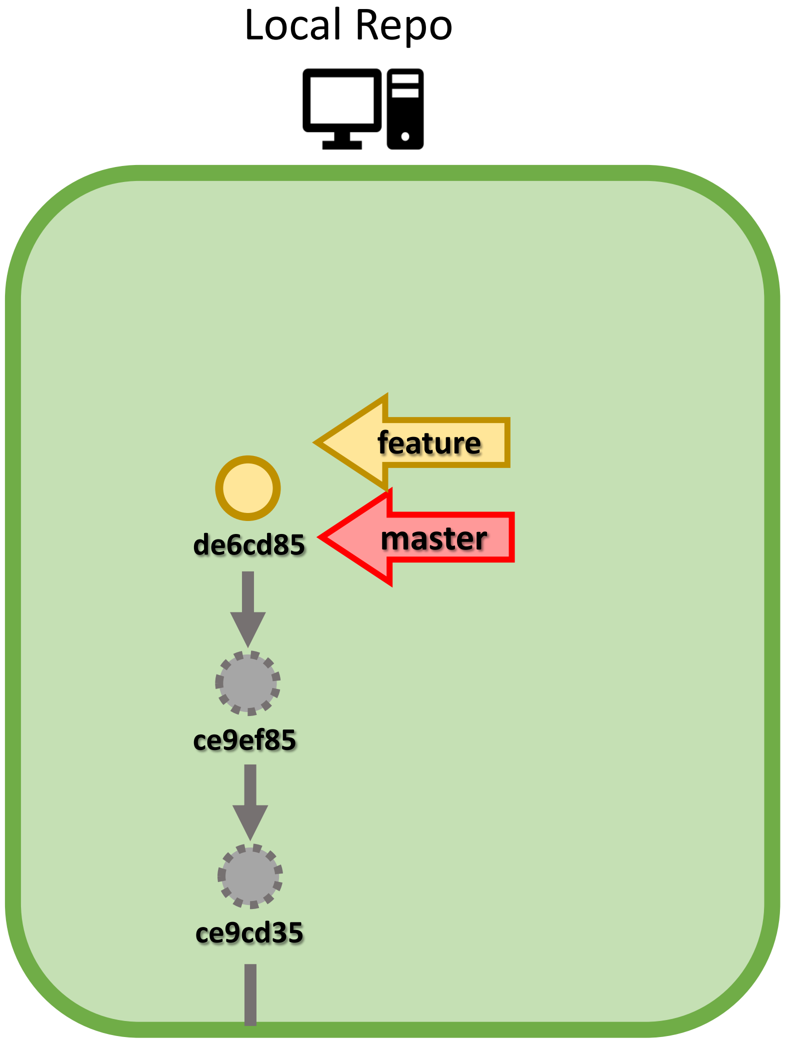
- Create a new branch called *feature*
  - master and feature will point to the same commit
- Switch to the branch *feature*
- Make a new commit

Branches & Merge



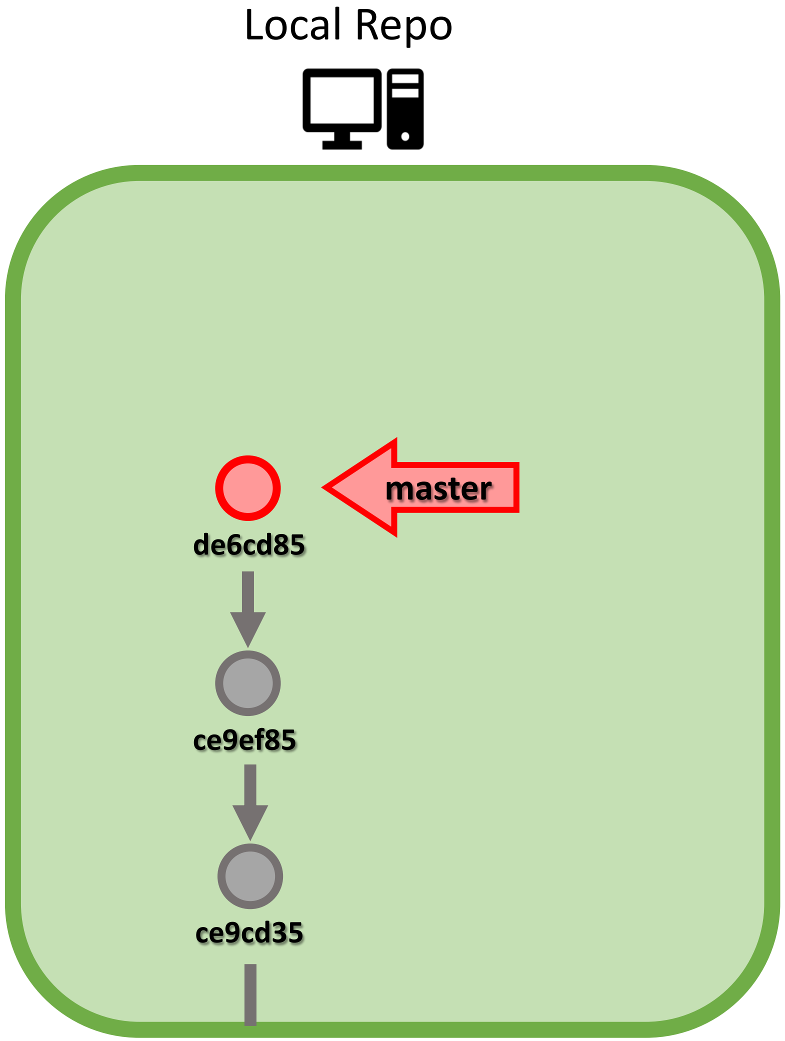
- Create a new branch called *feature*
  - master and feature will point to the same commit
- Switch to the branch *feature*
- Make a new commit
- Add changes from *feature* to *master* → Merge

Branches & Merge



- Create a new branch called *feature*
  - master and feature will point to the same commit
- Switch to the branch *feature*
- Make a new commit
- Add changes from *feature* to *master* → Merge
- Direct path between branches
  - Fast-forward merge

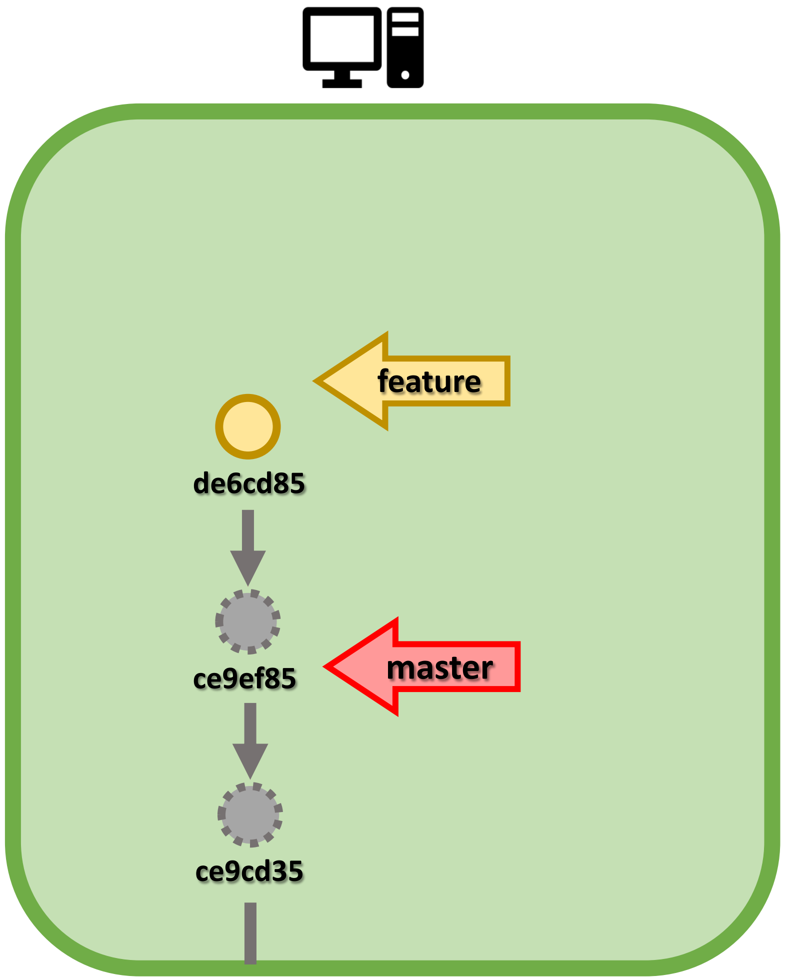
Branches & Merge



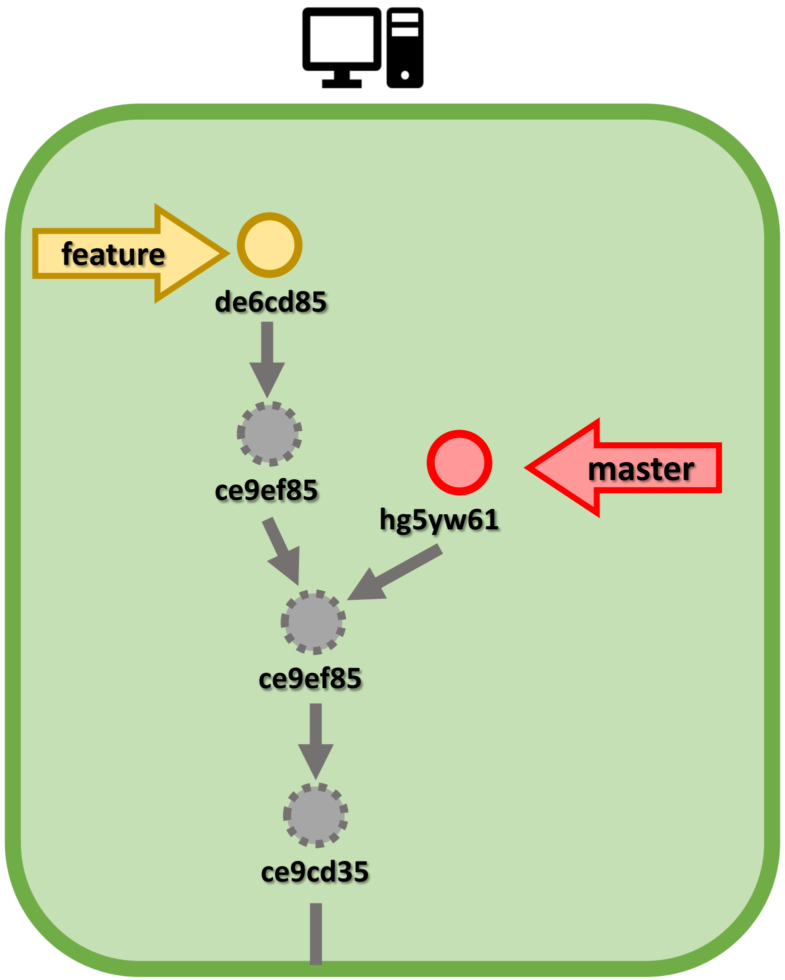
- Create a new branch called *feature*
  - master and feature will point to the same commit
- Switch to the branch *feature*
- Make a new commit
- Add changes from *feature* to *master* → Merge
- Direct path between branches
  - Fast-forward merge
- *feature* can be safely deleted

More about merge

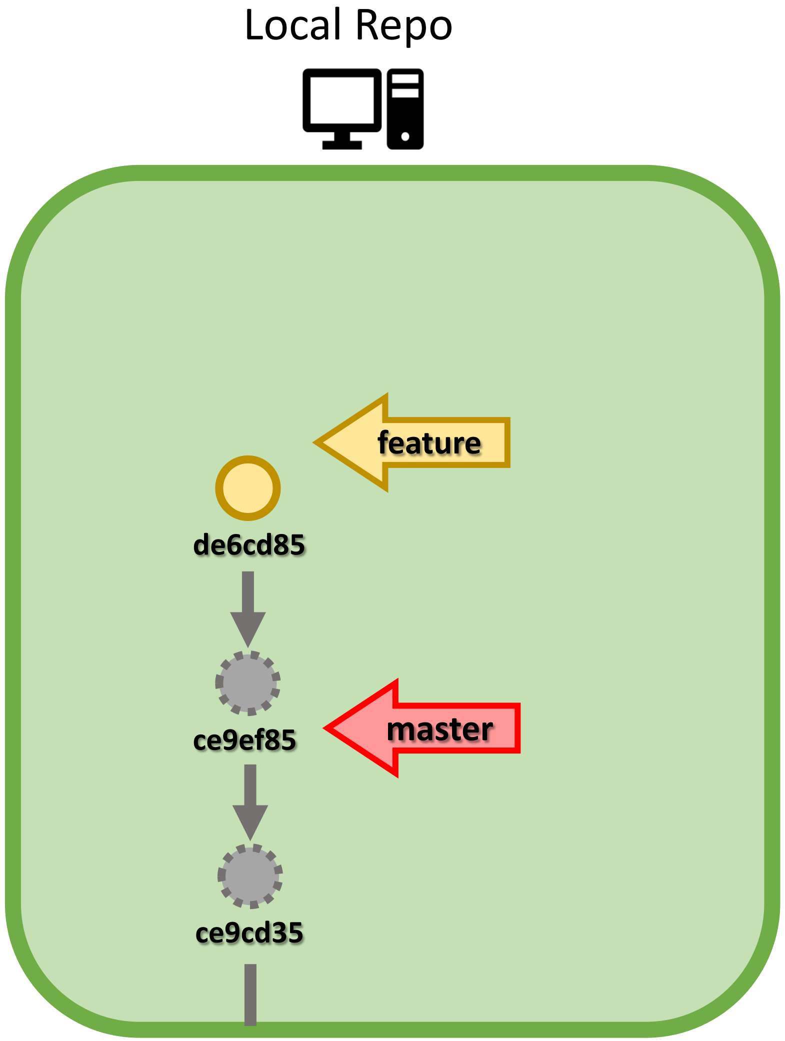
Local Repo

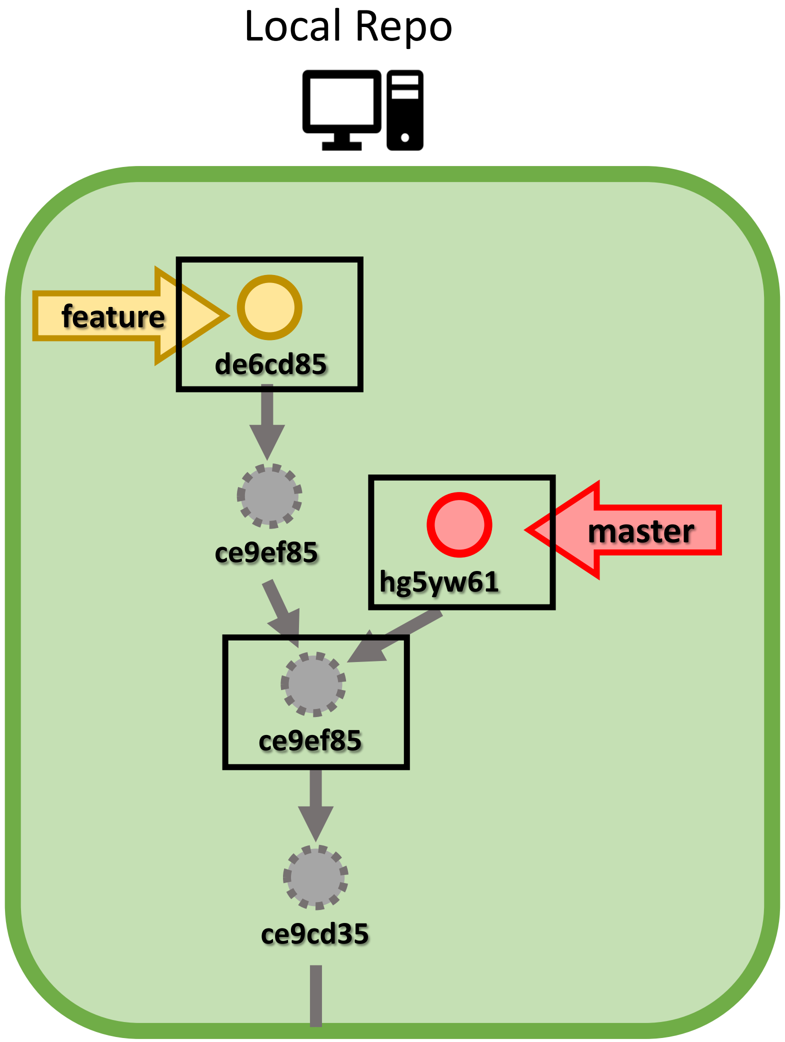


Local Repo

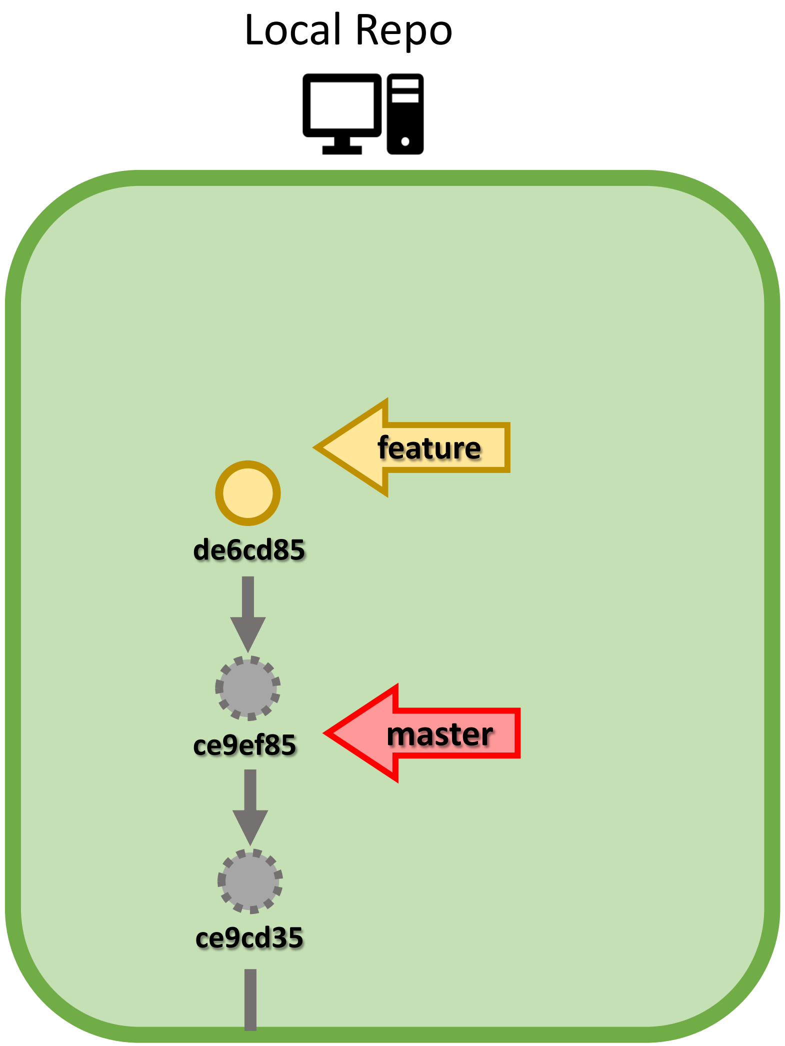


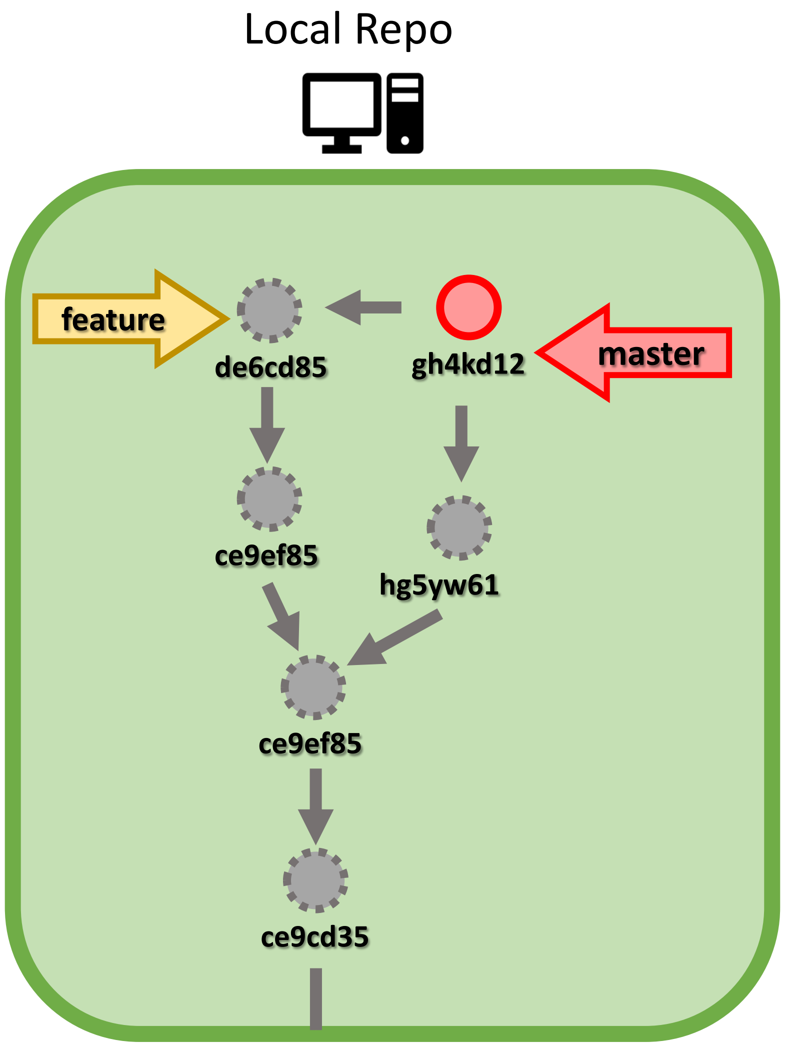
More about merge



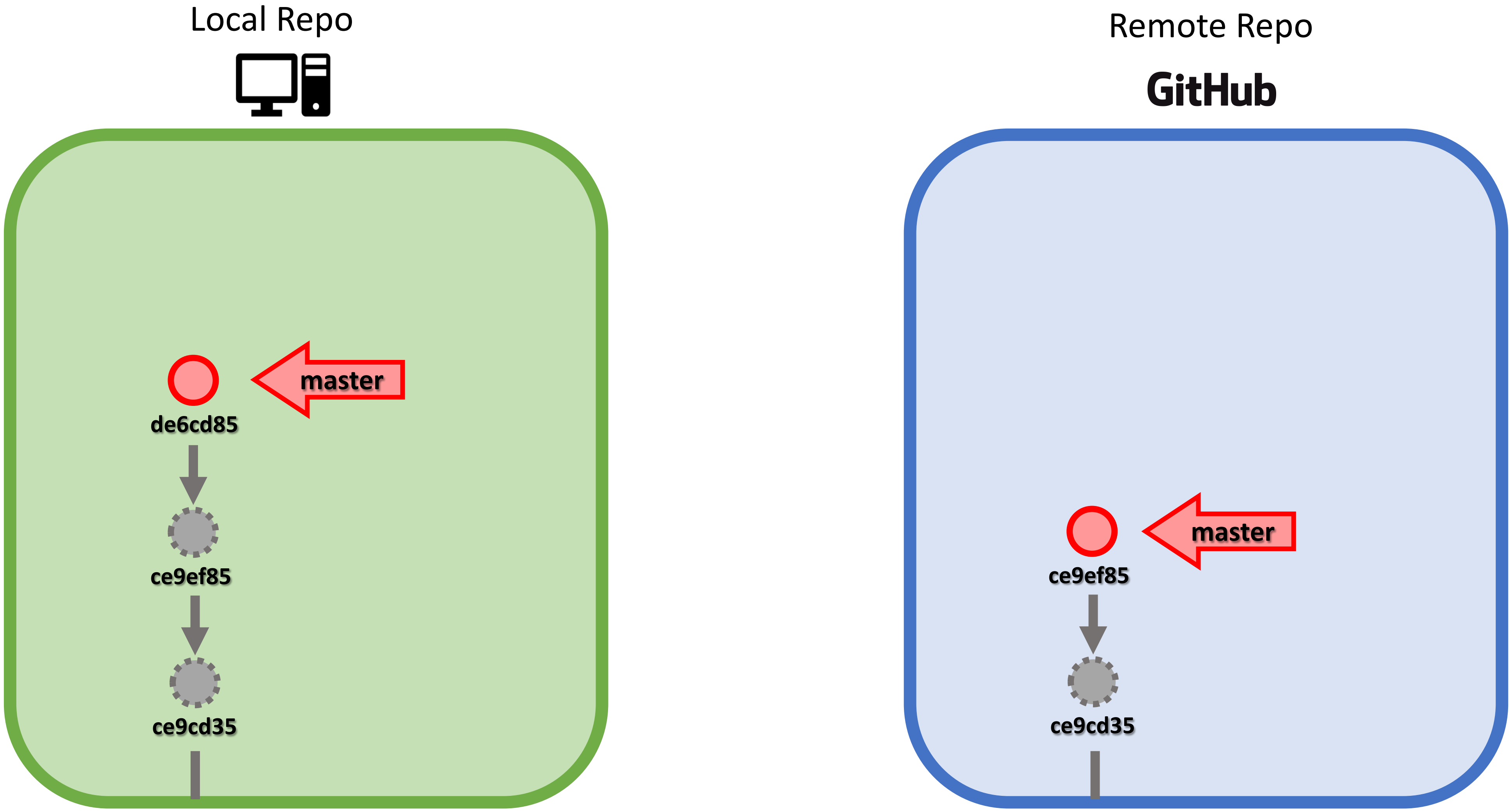


More about merge

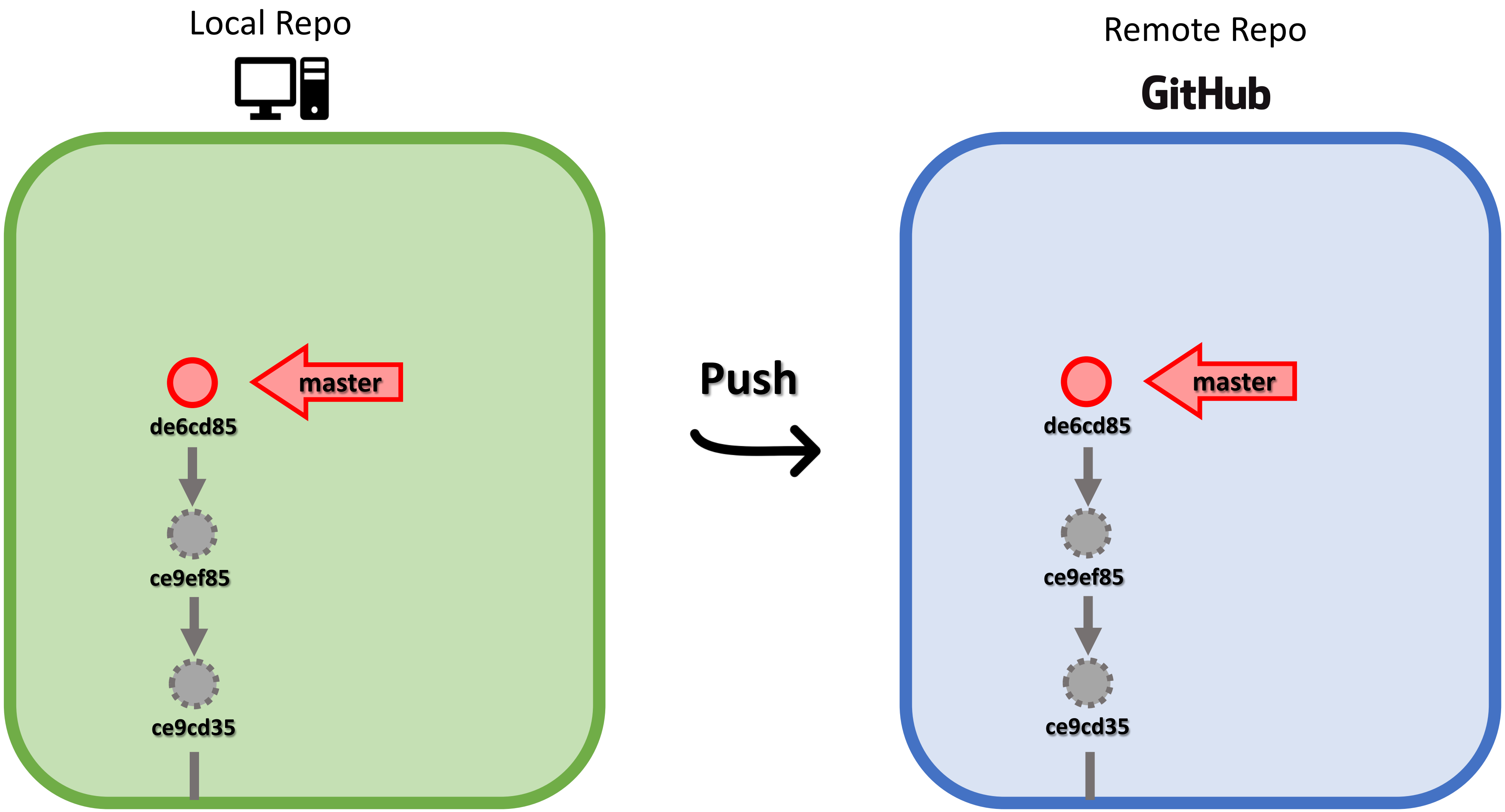




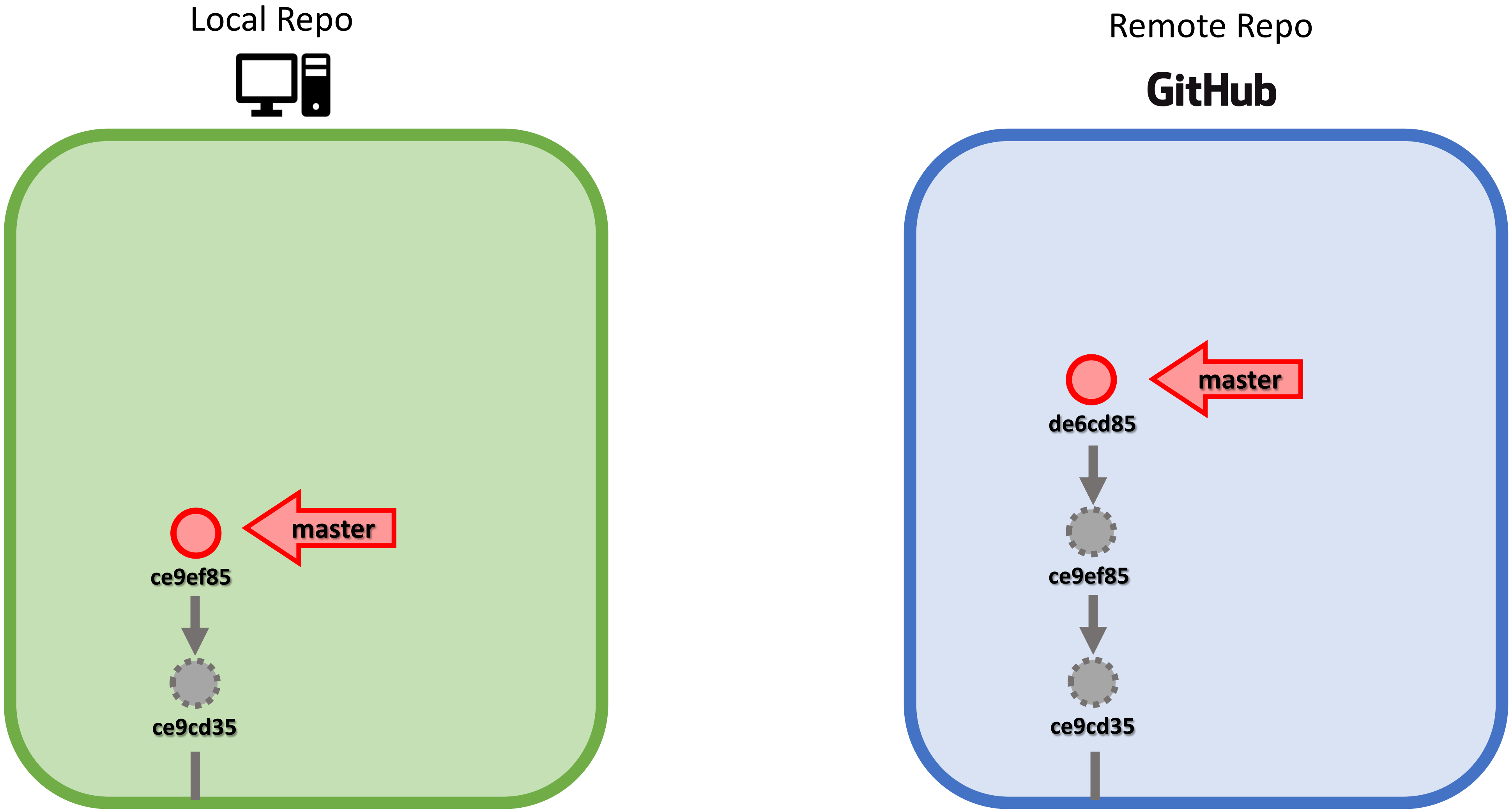
Local Repo→ Github



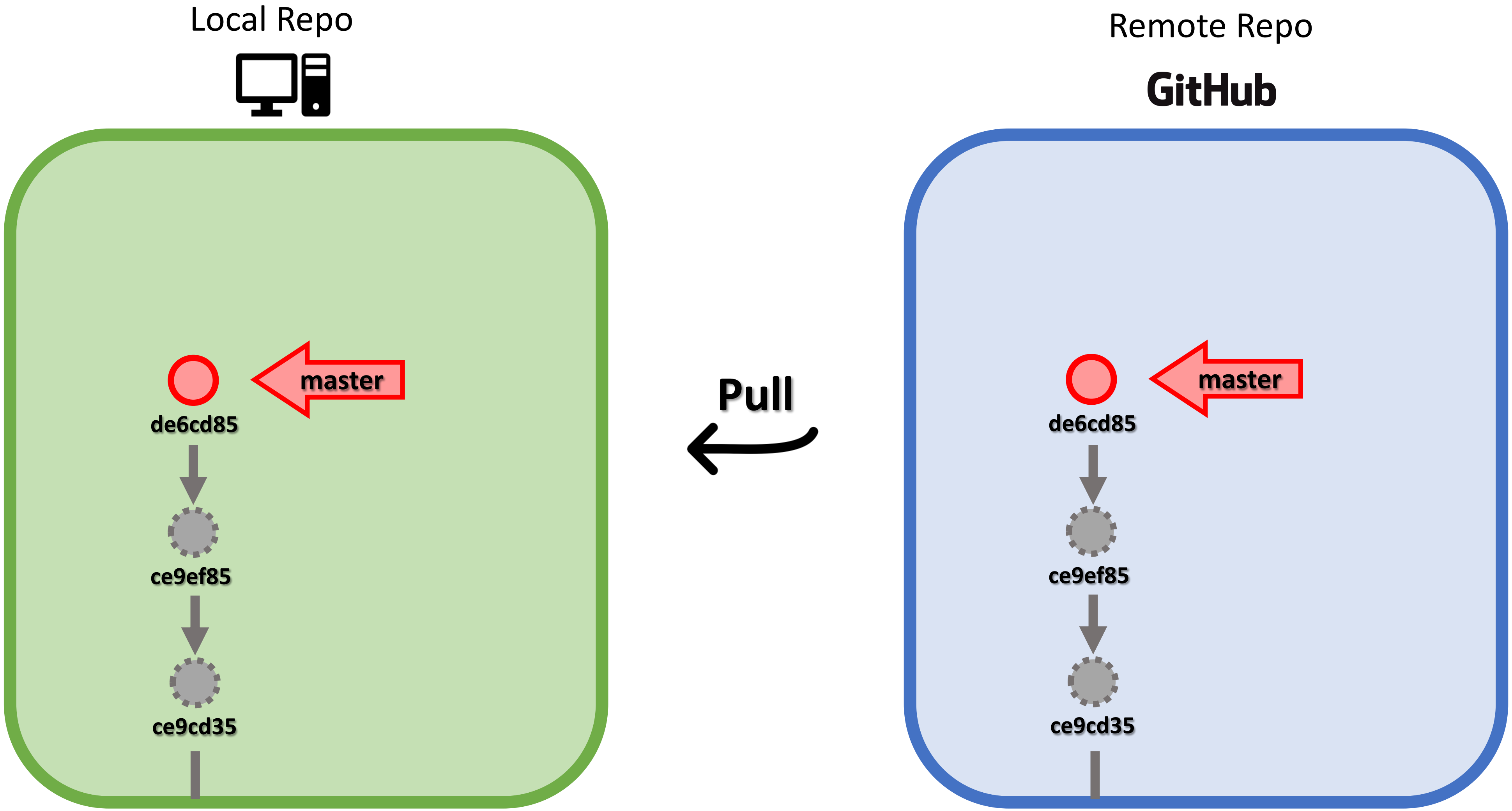
Local Repo→ Github



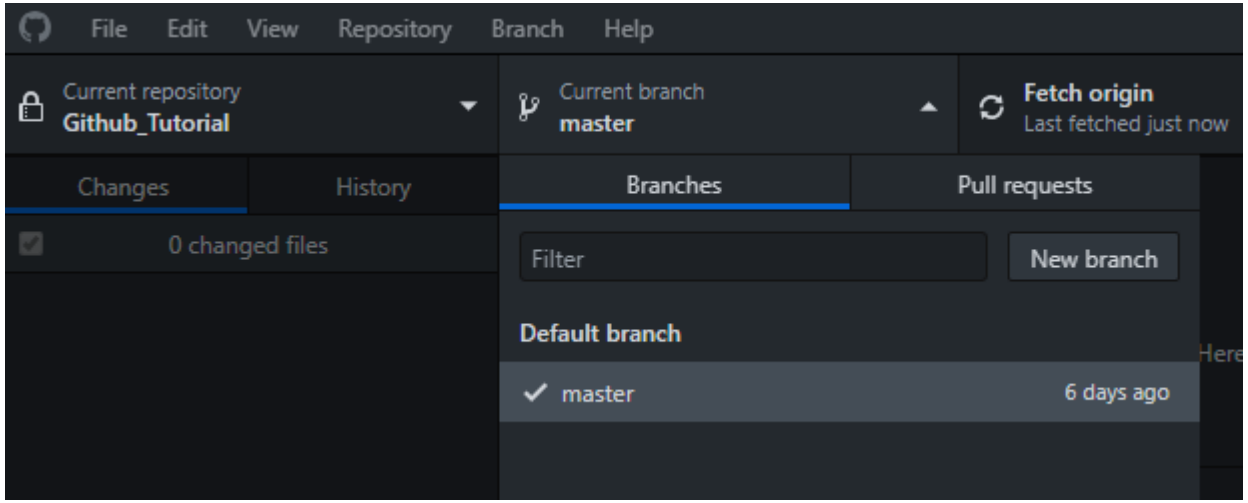
Local Repo← Github

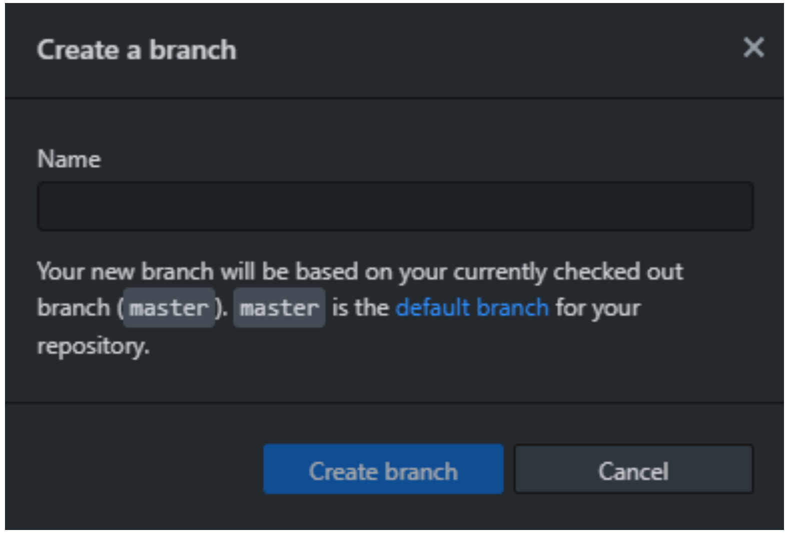


Local Repo← Github



Create branches: GitDesktop

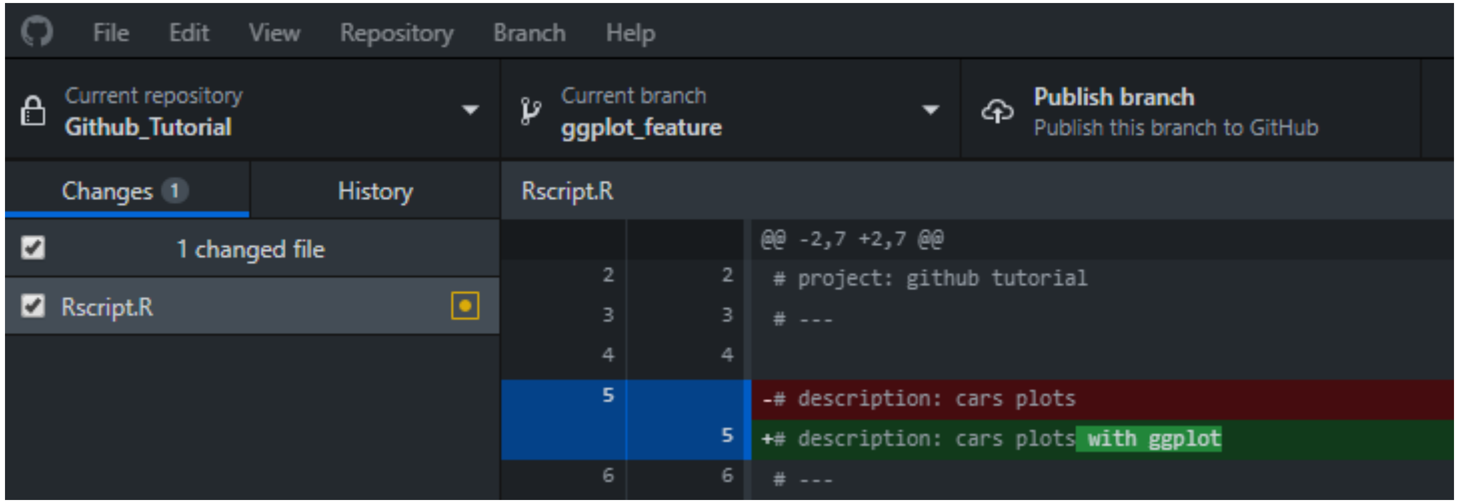




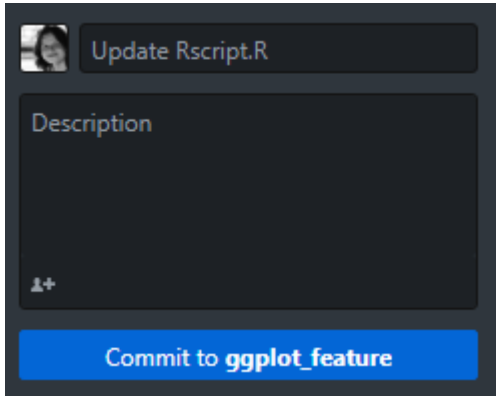
Your new branch will be based on your currently checked out branch (`master`). `master` is the default branch for your repository.

- Select the *Current branch* (as shown above)
- Click on *New branch* button
- Enter a Name in the specified field
- Click *Create branch*
- The new branch is now the current branch

Commits: GitDesktop

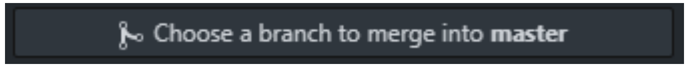


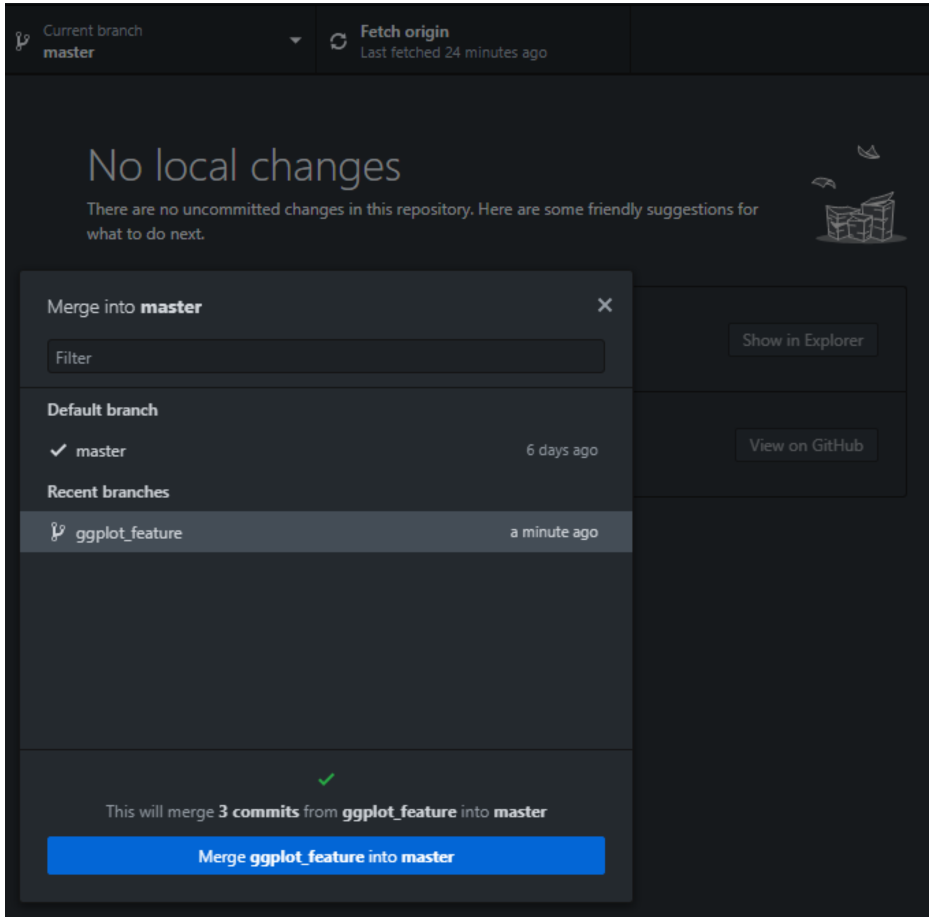
```
@@ -2,7 +2,7 @@
 # project: github tutorial
 # ---

-# description: cars plots
+# description: cars plots with ggplot
 # ---
```



- Save changes made to Rscript.R
- Rscript.R appears in the *Changes* area.
- Select the file that you want to commit (Staging)
- Add a commit message
- Press *Commit to **branch*** button (ggplot_feature)

Merge & Push: GitDesktop

- Switch from *branch* to *master*
- 
  Choose a branch to merge into **master**
- Select the *branch*
- Press the *Merge* button
- Merge is done!

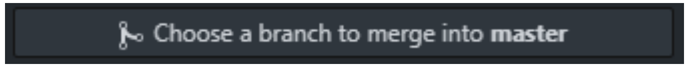


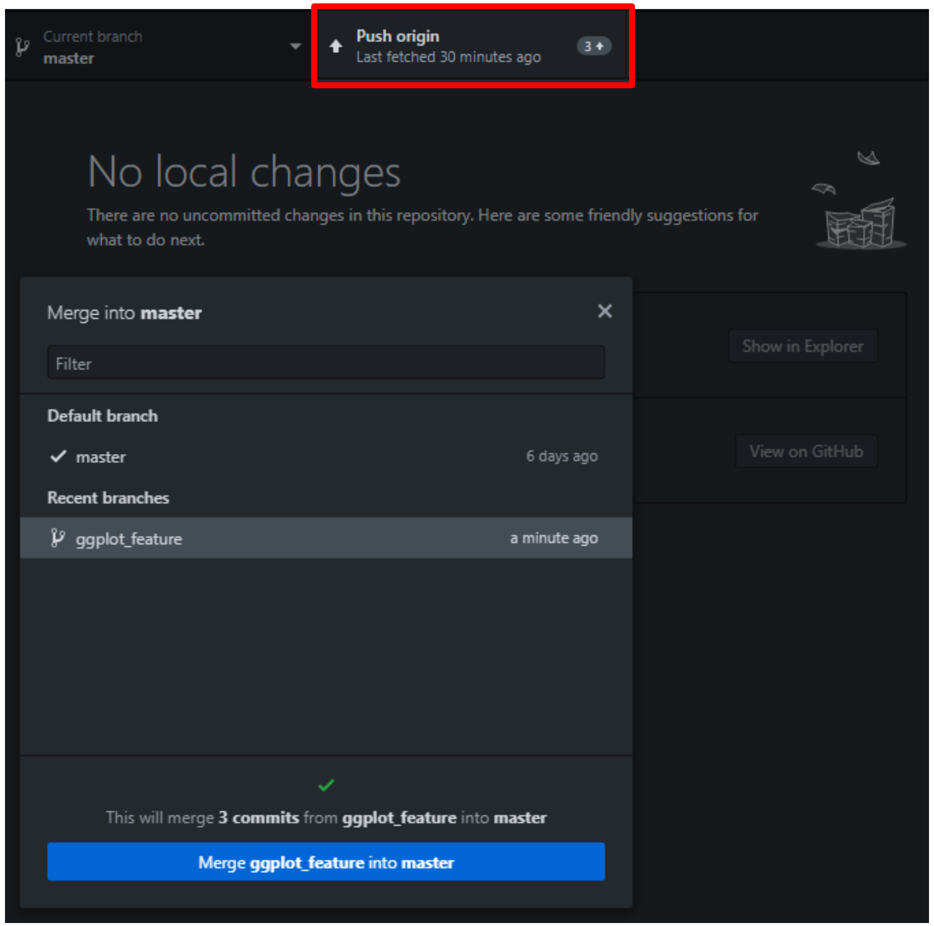
No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

This will merge **3 commits** from **ggplot_feature** into **master**

Merge & Push: GitDesktop

- Switch from *branch* to *master*
- 
  Choose a branch to merge into **master**
- Select the *branch*
- Press the *Merge* button
- Merge is done!
- Push origin becomes available
- Updates local →remote
- Got to GitHub to see your changes

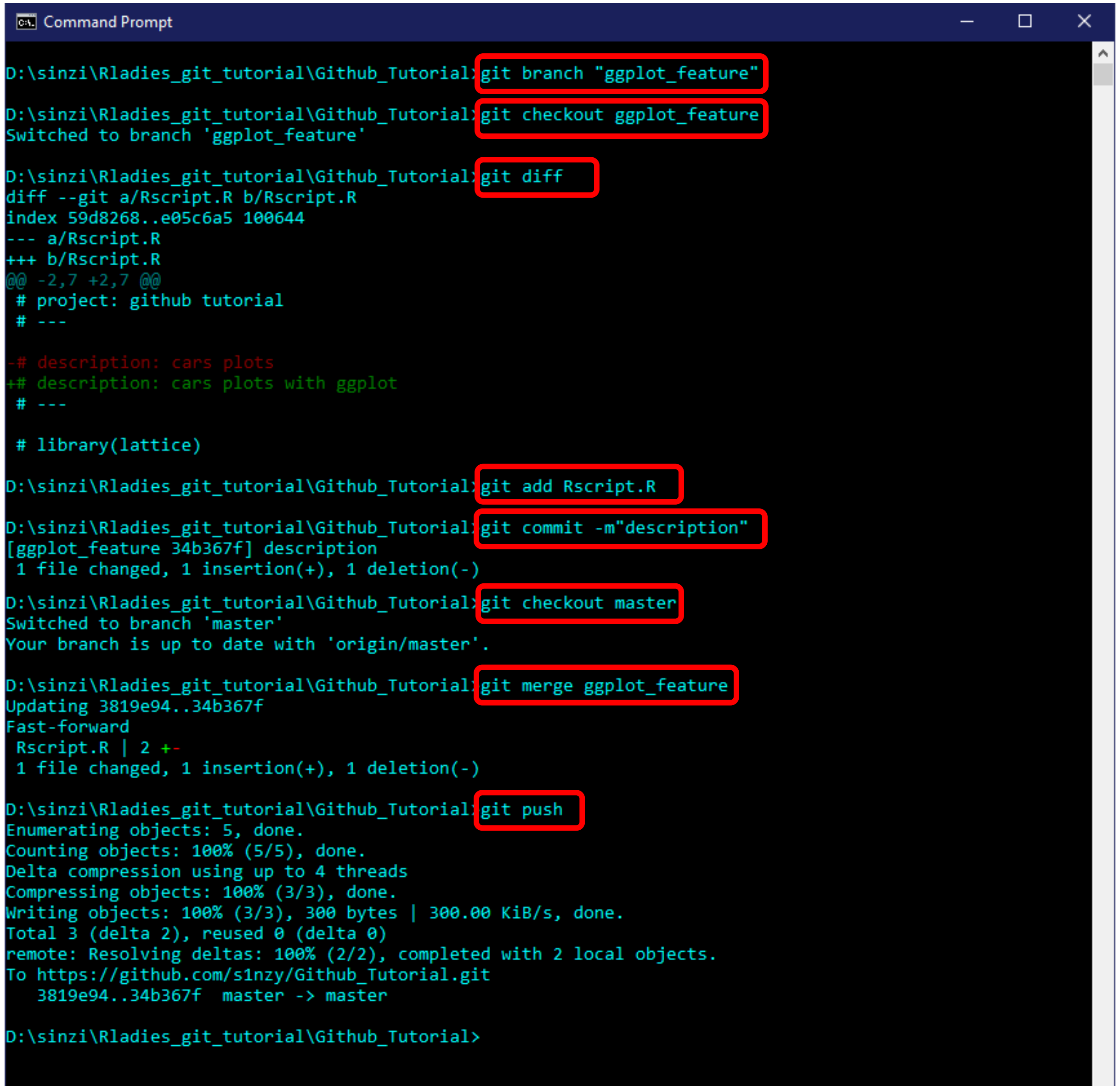


No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

This will merge **3 commits** from **ggplot_feature** into **master**

Using the command line



```
D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git branch "ggplot_feature"

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git checkout ggplot_feature
Switched to branch 'ggplot_feature'

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git diff
diff --git a/Rscript.R b/Rscript.R
index 59d8268..e05c6a5 100644
--- a/Rscript.R
+++ b/Rscript.R
@@ -2,7 +2,7 @@
 # project: github tutorial
 # ---

-# description: cars plots
+# description: cars plots with ggplot
 # ---

 # library(lattice)

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git add Rscript.R

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git commit -m"description"
[ggplot_feature 34b367f] description
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git merge ggplot_feature
Updating 3819e94..34b367f
Fast-forward
 Rscript.R | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/s1nzy/Github_Tutorial.git
   3819e94..34b367f  master -> master

D:\sinzi\Rladies_git_tutorial\Github_Tutorial>
```

`git branch “branch_name”`

creates a new branch

`git checkout branch_name`

switch to that branch

`git diff`

shows unstaged changes

`git add`

adds files to the staging area (select in GitDesktop)

`git commit –m ”message”`

makes the commit

`git checkout master`

switches back to master branch

`git merge branch_name`

updates master with changes from branch

`git push origin master`

updates remote repo on github

Rscript.R

Learn by doing! Hands-on part 1

During the break, find a partner for the 2nd hands-on!

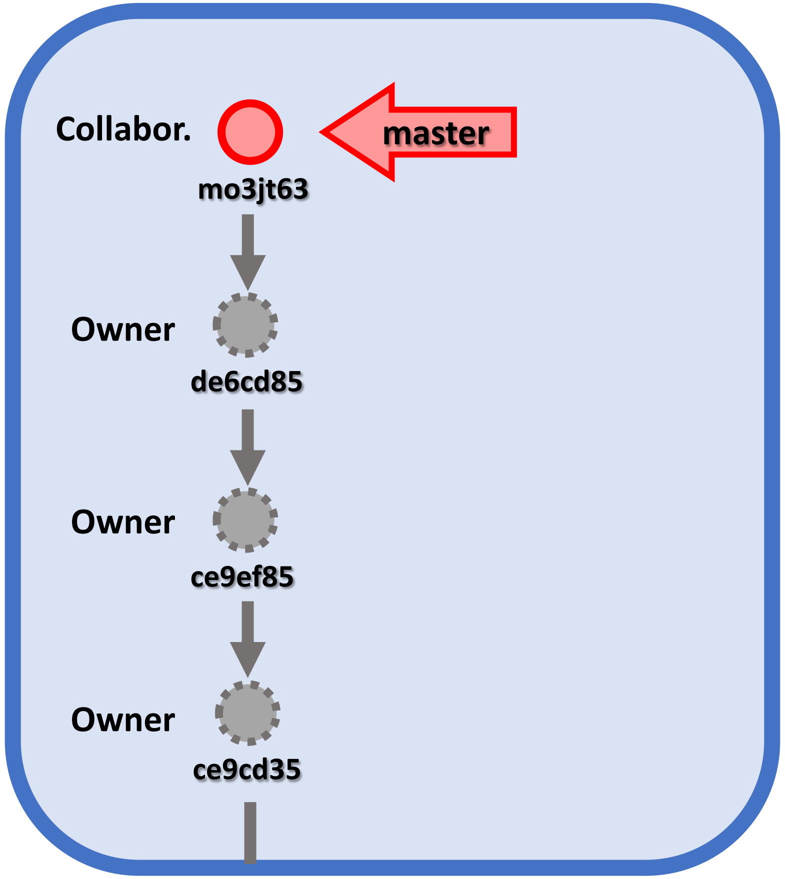
- You have downloaded a file: Rscript.R
- Basics on Github:
  - Create a new repository: GitTutorial_<initials>_<birthday> (e.g., SC_11)
  - Upload the Rscript.R to your github repo → commit
  - Edit the file with the online editor → commit
  - Check the History
- Basics on GitDesktop/CMD Prompt:
  - Clone the repository to a location of your choice
  - Create a branch (and checkout branch)
  - Click on *Show in Explorer* button (go to your local repo) and open the Rscript.R
  - Change: the color of the dots → commit
  - Change: remove the grid from the plot → commit
  - Merge these changes to your master
  - Push changes to remote repository
- While doing this pay attention to the local history and check if your changes were pushed correctly to github

Working with others on the same repo: contributor vs. collaborator

- Collaborator:
  - You can add someone to be a collaborator for your project
  - You give them rights to make changes to that repo as if they would own it
  - Repo in github → settings → collaborator → look up the username → add
  - The same working standard applies, make a working branch and do not change things directly on the master

Remote Repo

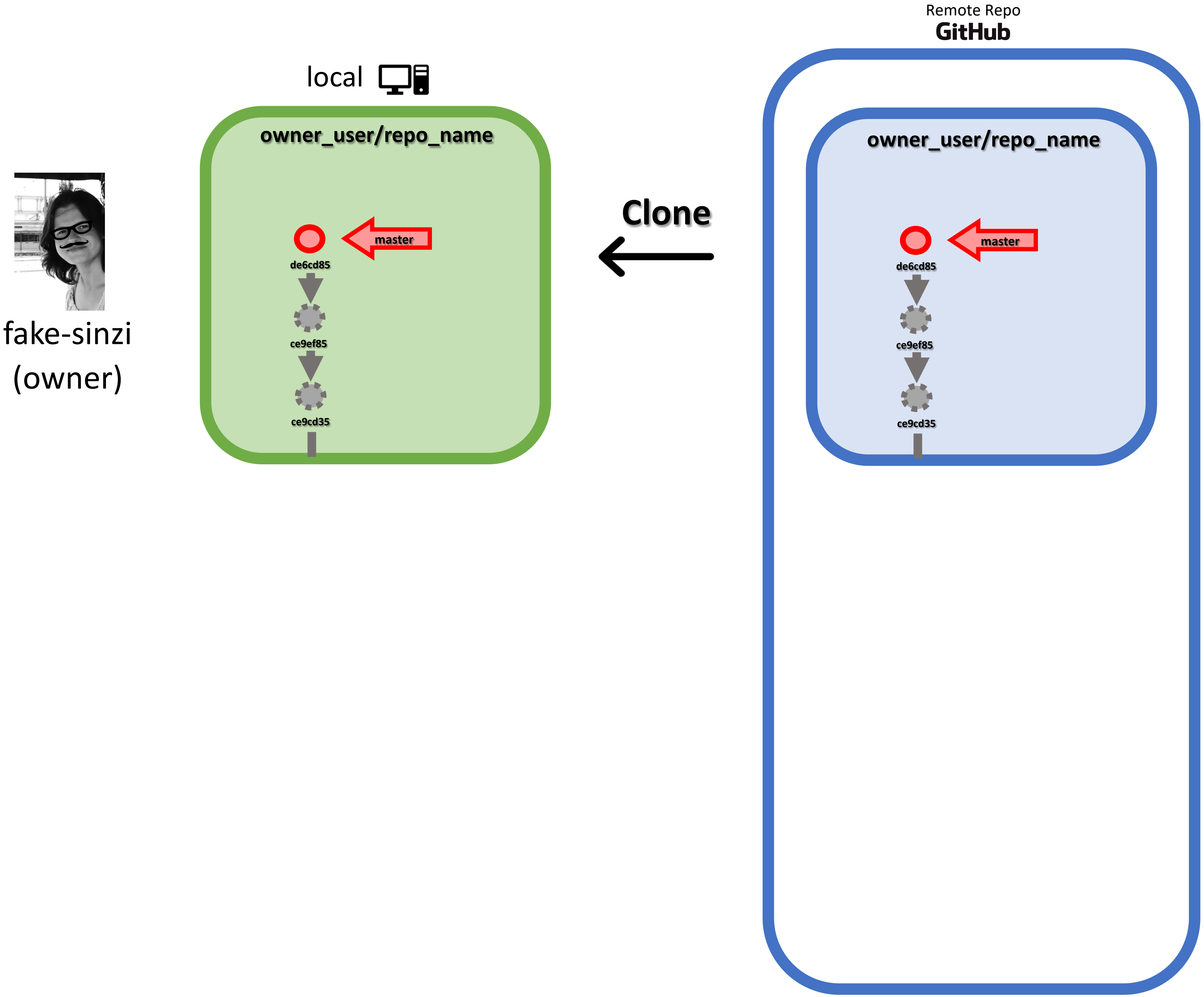
GitHub



Working with others on the same repo: contributor vs. collaborator

- Contributor:
  - Someone who wants to use/contribute your repository
  - They ask you (the owner) to revise & accept their changes before merging them to your repo

How to become a contributor?



Remote Repo

GitHub

local

fake-sinzi

(owner)

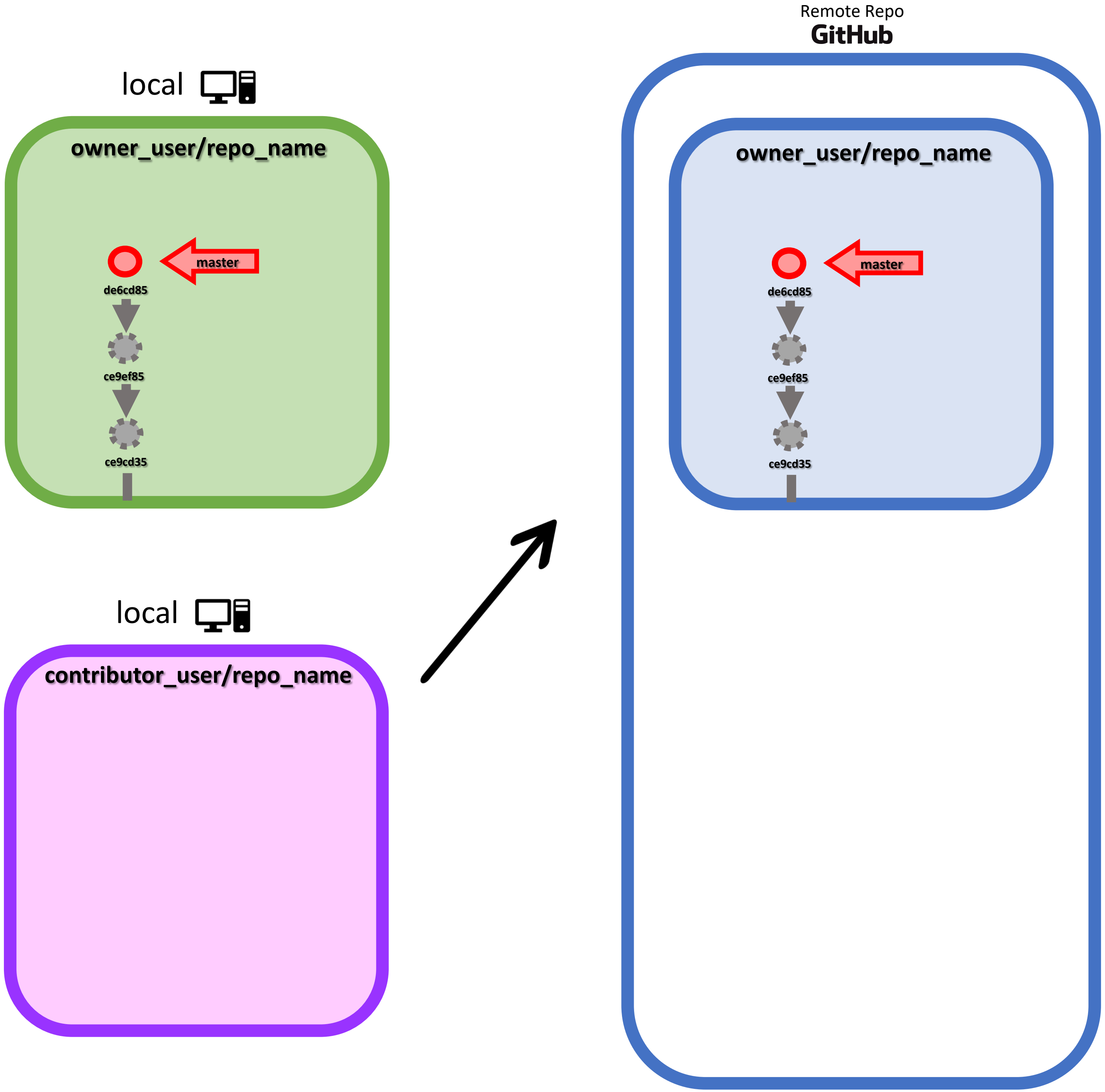
owner_user/repo_name

Clone

owner_user/repo_name

fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

local

contributor_user/repo_name

owner_user/repo_name

master

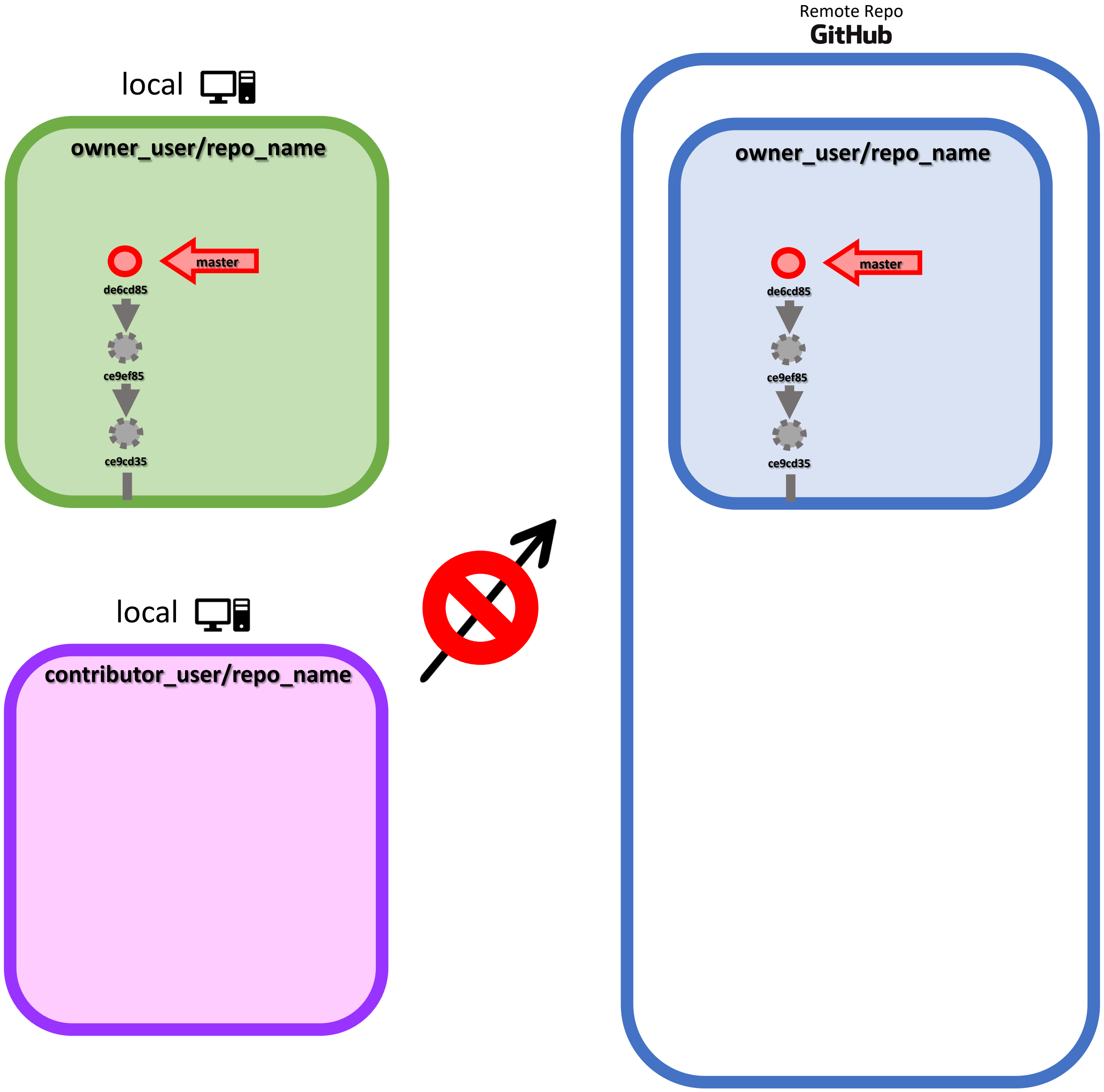
de6cd85

ce9ef85

ce9cd35

fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

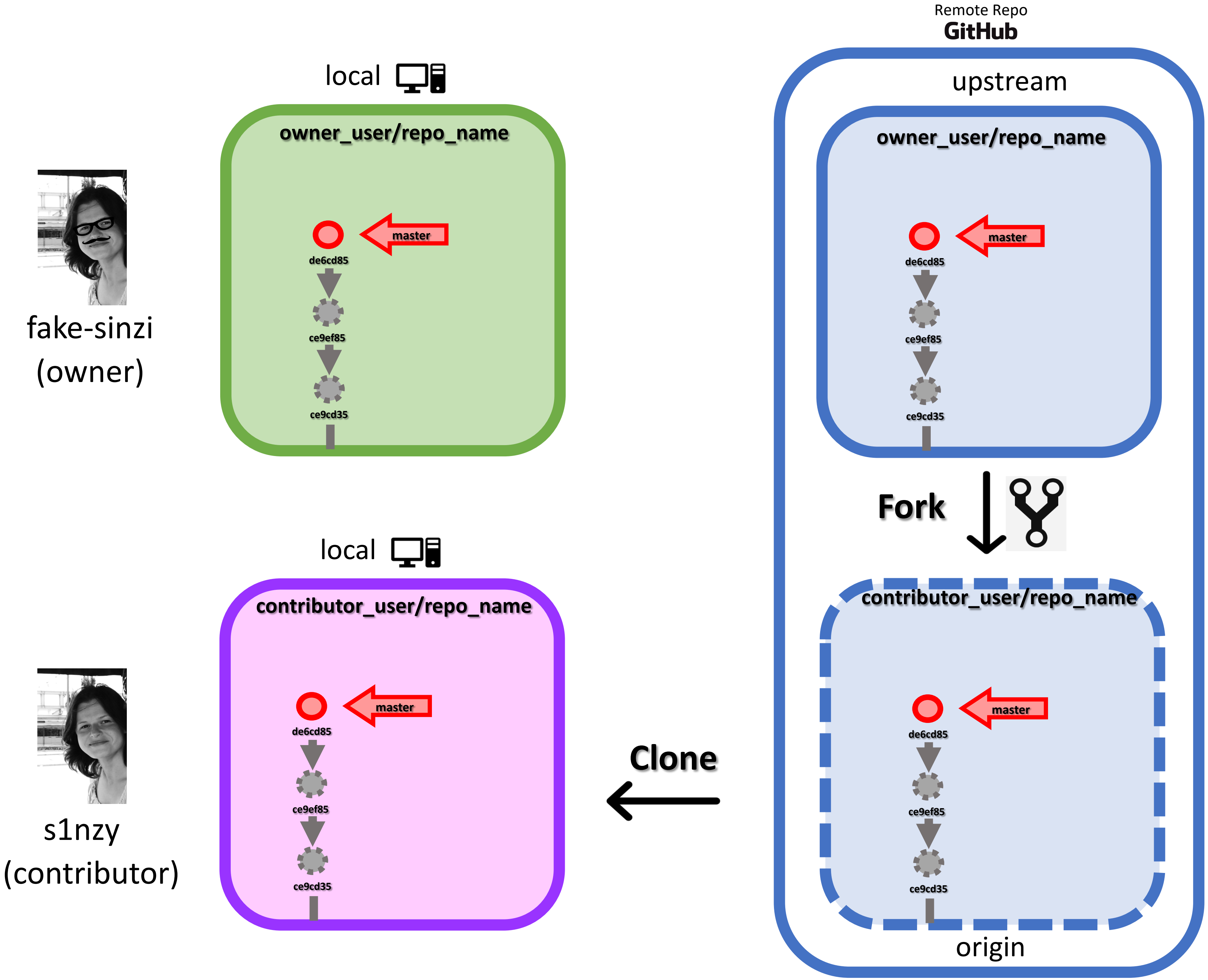
local

owner_user/repo_name

local

contributor_user/repo_name

owner_user/repo_name



Remote Repo
GitHub

upstream

owner_user/repo_name

Fork

contributor_user/repo_name

origin

local

owner_user/repo_name

fake-sinzi
(owner)

local

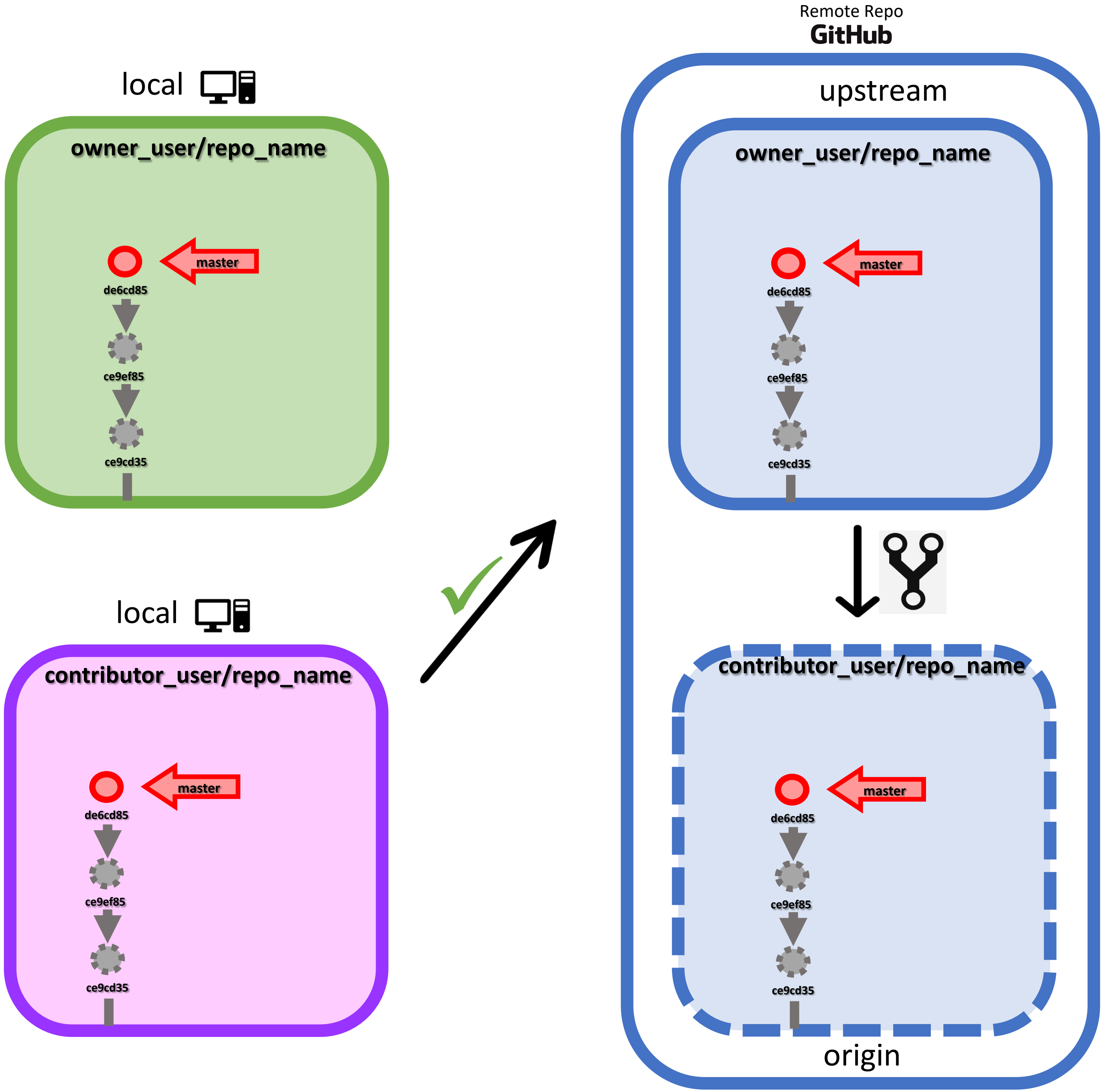
contributor_user/repo_name

s1nzy
(contributor)

Clone

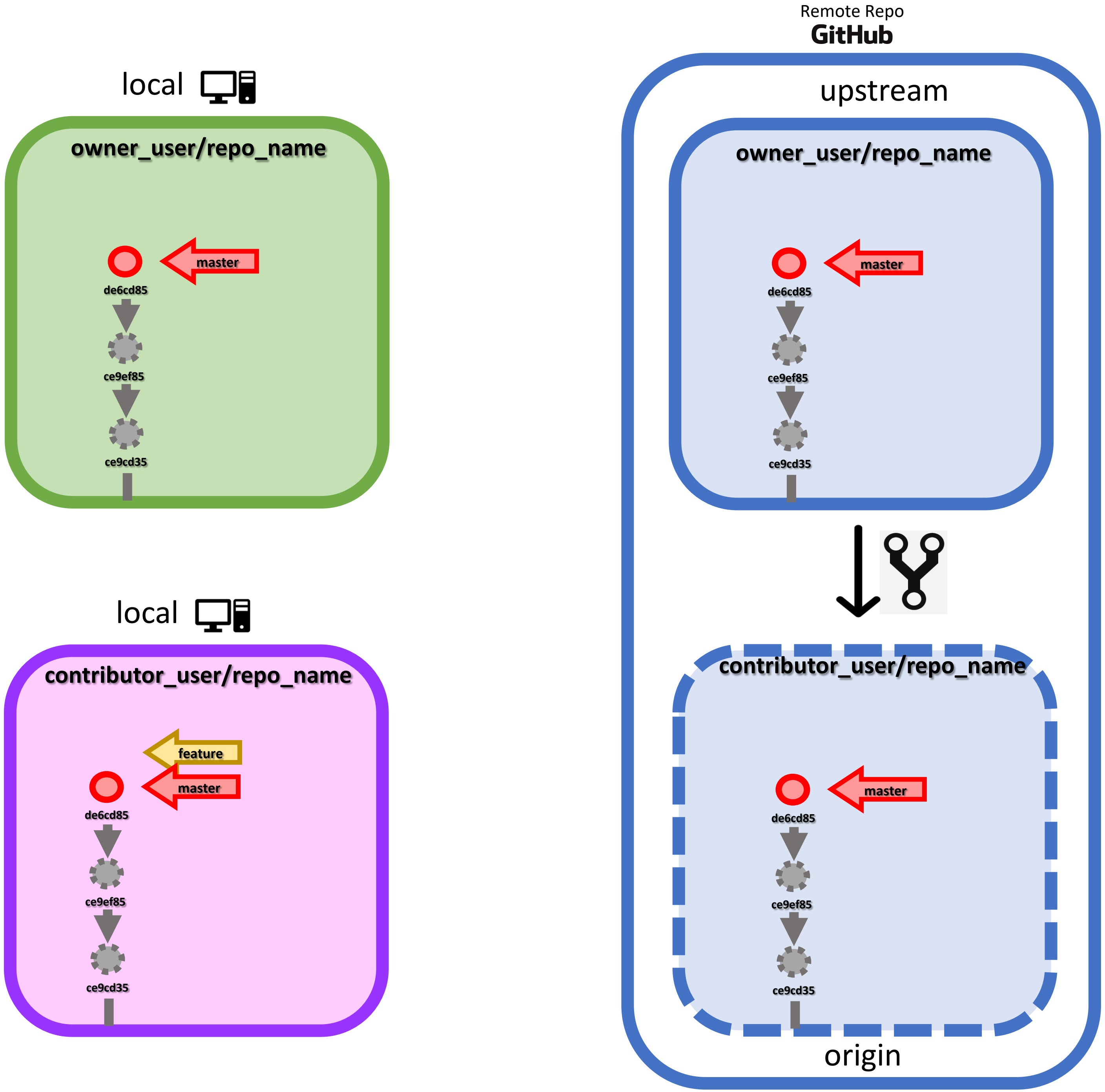
fake-sinzi
(owner)

s1nzy
(contributor)



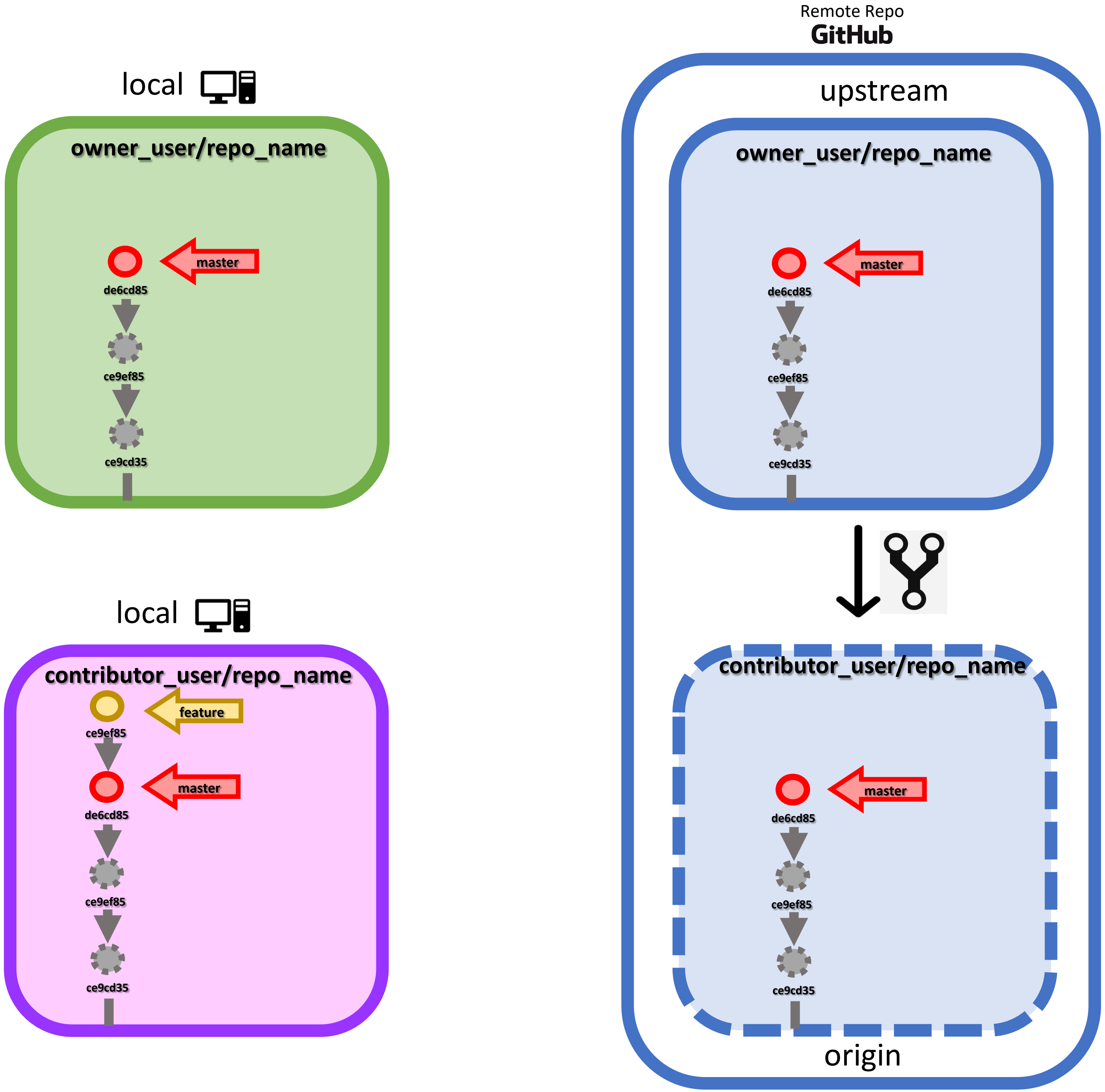
fake-sinzi
(owner)

s1nzy
(contributor)



fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

local

owner_user/repo_name

local

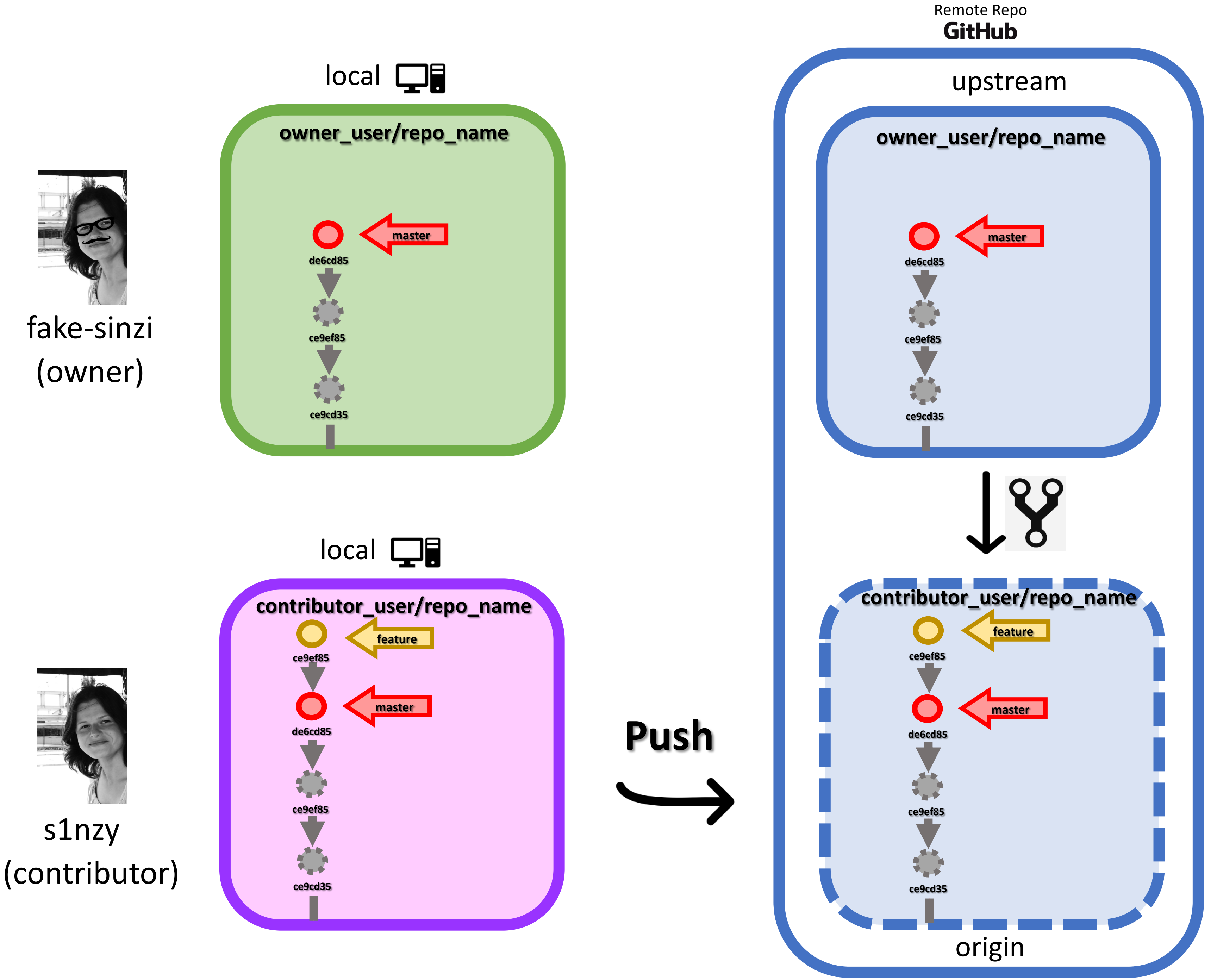
contributor_user/repo_name

upstream

owner_user/repo_name

contributor_user/repo_name

origin



Remote Repo
GitHub

fake-sinzi
(owner)

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

s1nzy
(contributor)

local

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

ce9ef85

ce9cd35

Push

upstream

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

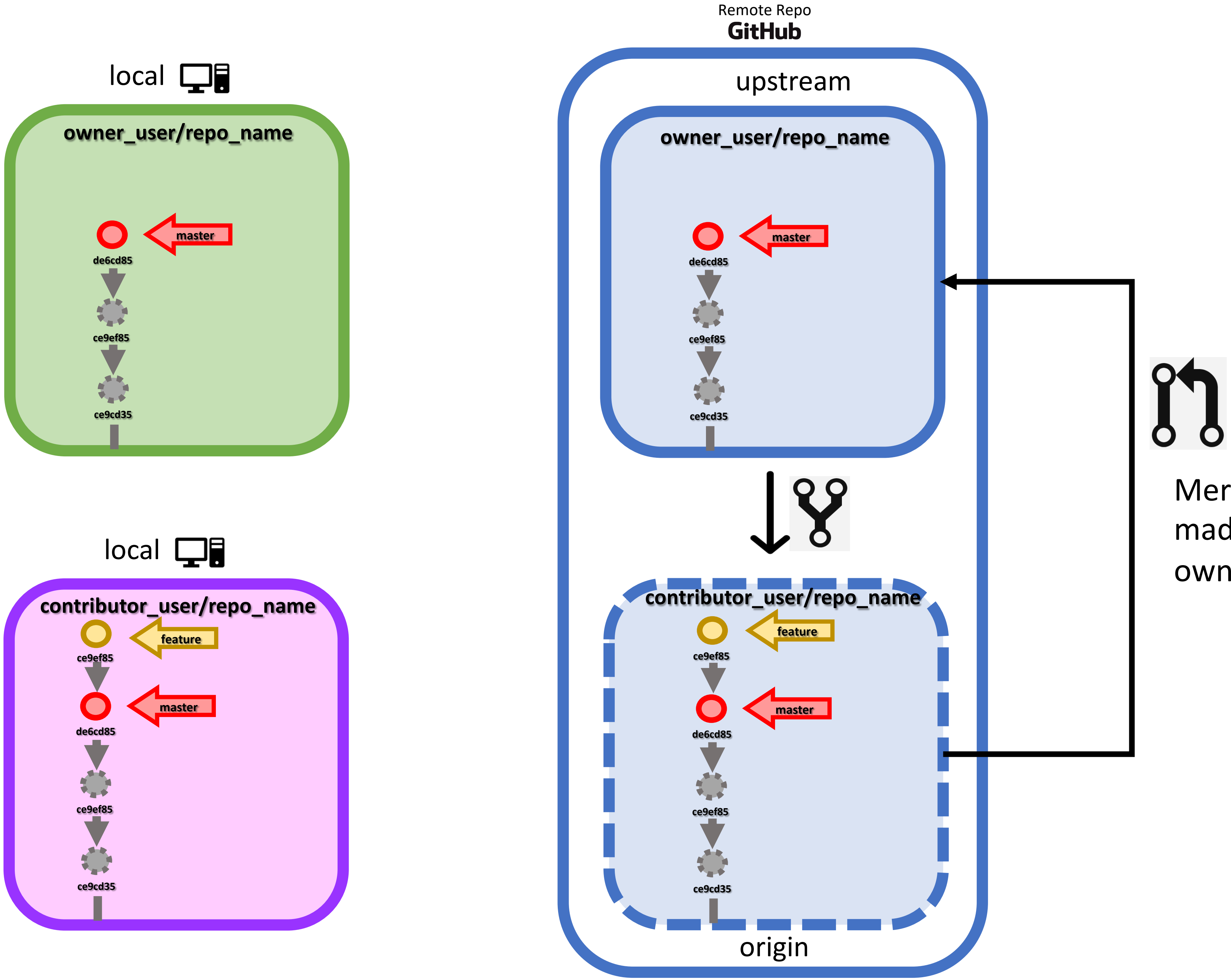
ce9ef85

ce9cd35

origin

fake-sinzi
(owner)

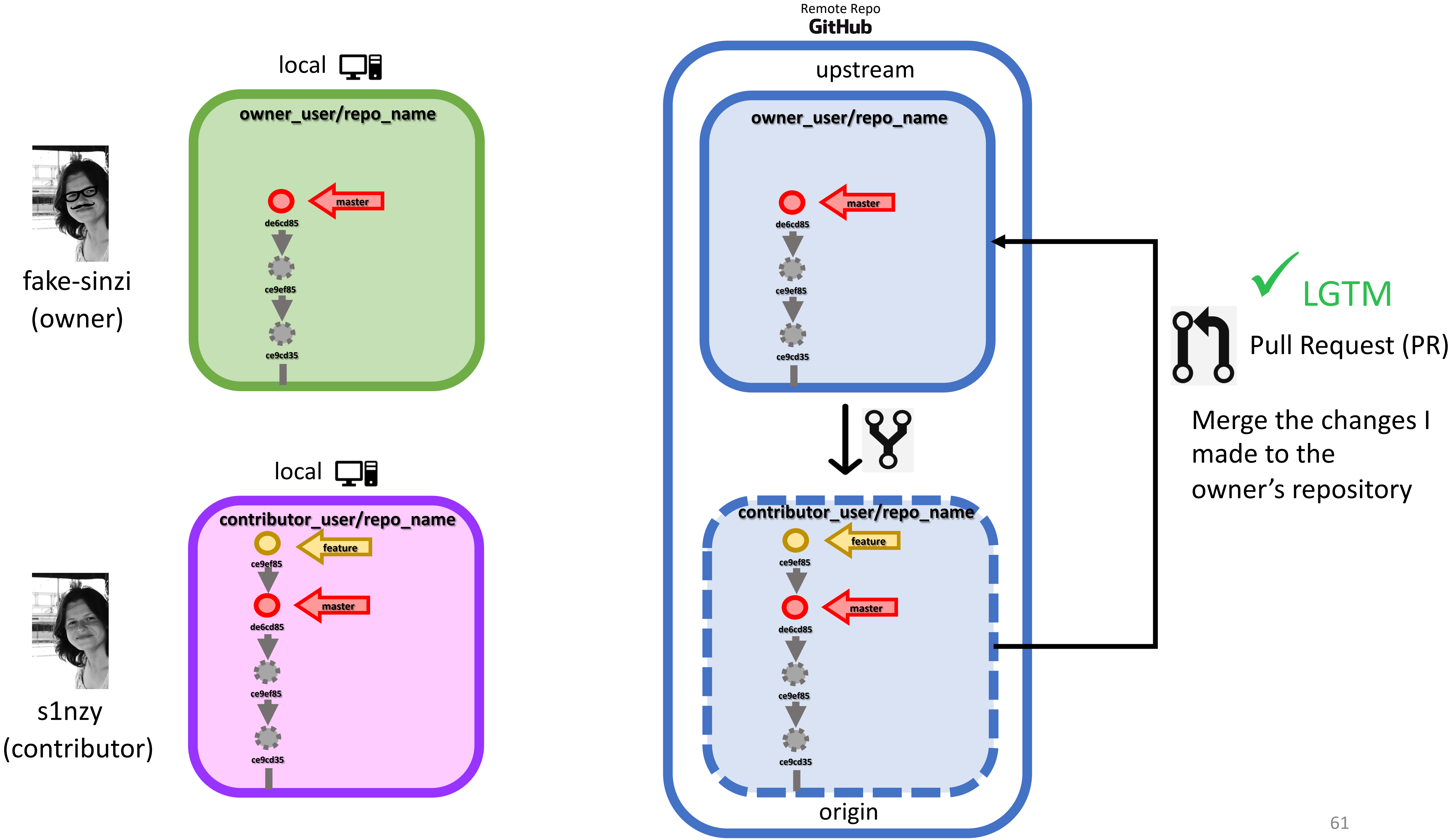
s1nzy
(contributor)



Remote Repo
GitHub

Pull Request (PR)

Merge the changes I made to the owner’s repository



Remote Repo
GitHub

upstream

origin

local

fake-sinzi
(owner)

s1nzy
(contributor)

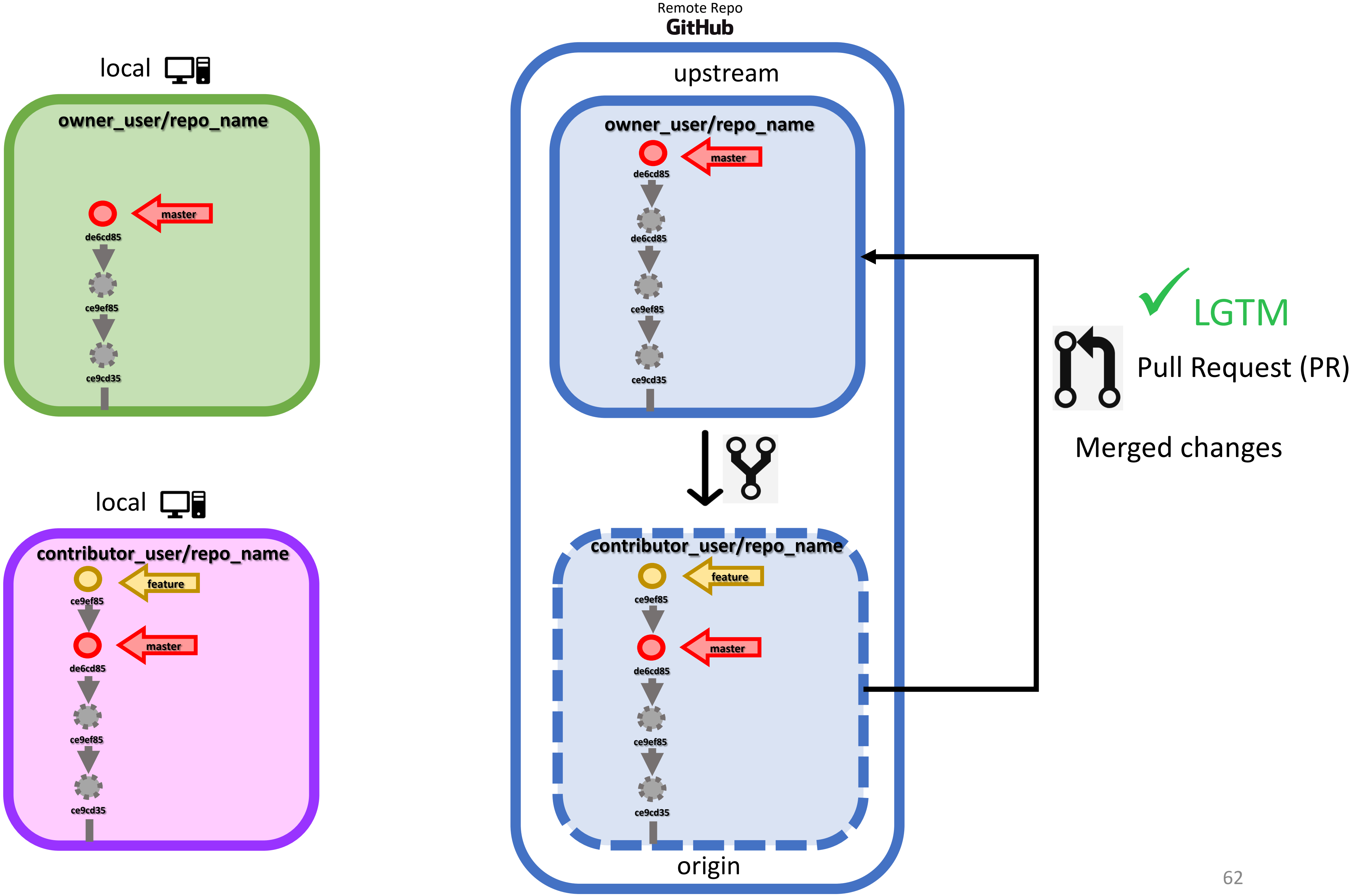
LGTM

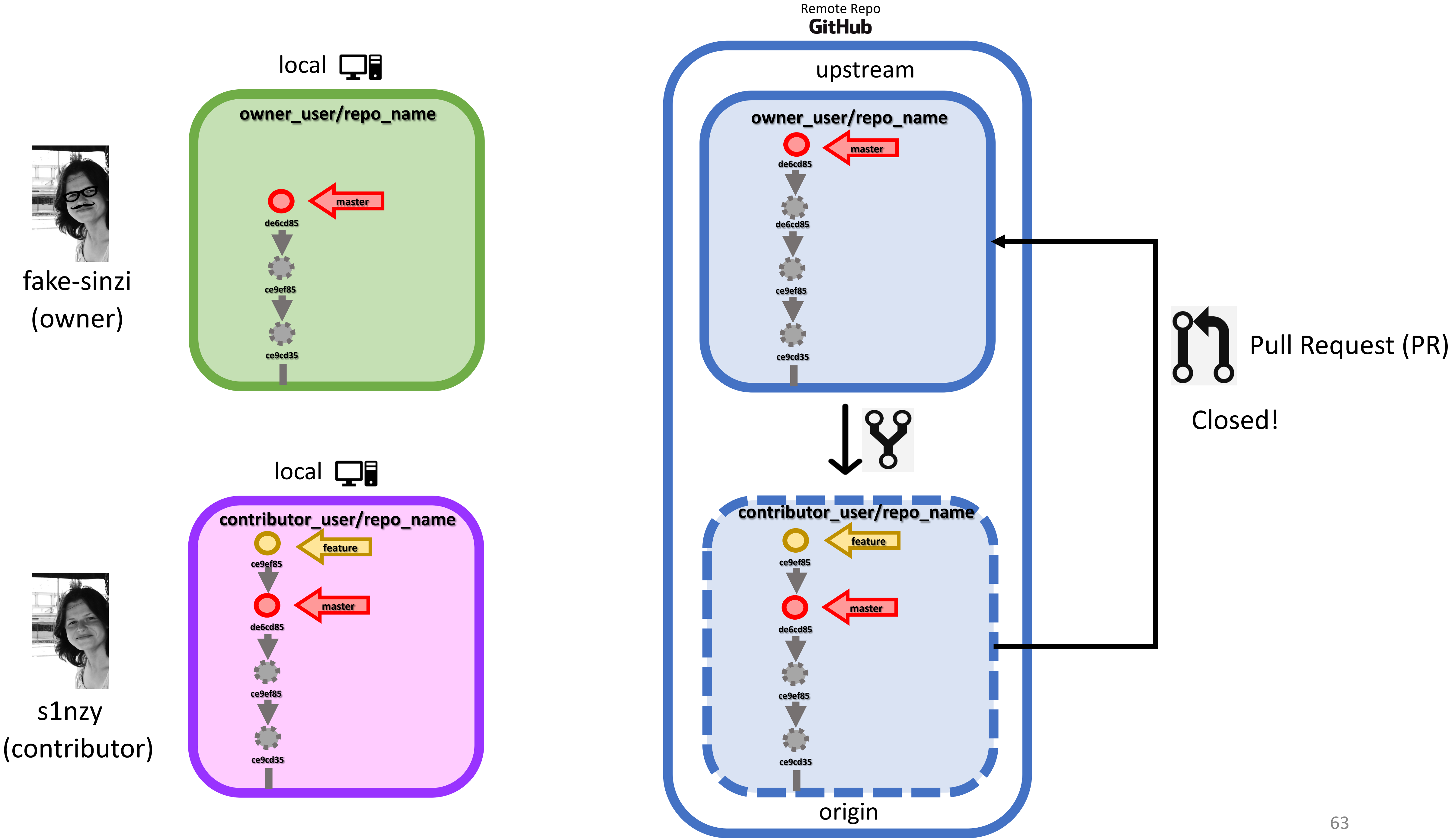
Pull Request (PR)

Merge the changes I made to the owner's repository

fake-sinzi
(owner)

s1nzy
(contributor)





Remote Repo
GitHub

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

fake-sinzi
(owner)

local

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

ce9ef85

ce9cd35

s1nzy
(contributor)

upstream

owner_user/repo_name

master

de6cd85

de6cd85

ce9ef85

ce9cd35

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

ce9ef85

ce9cd35

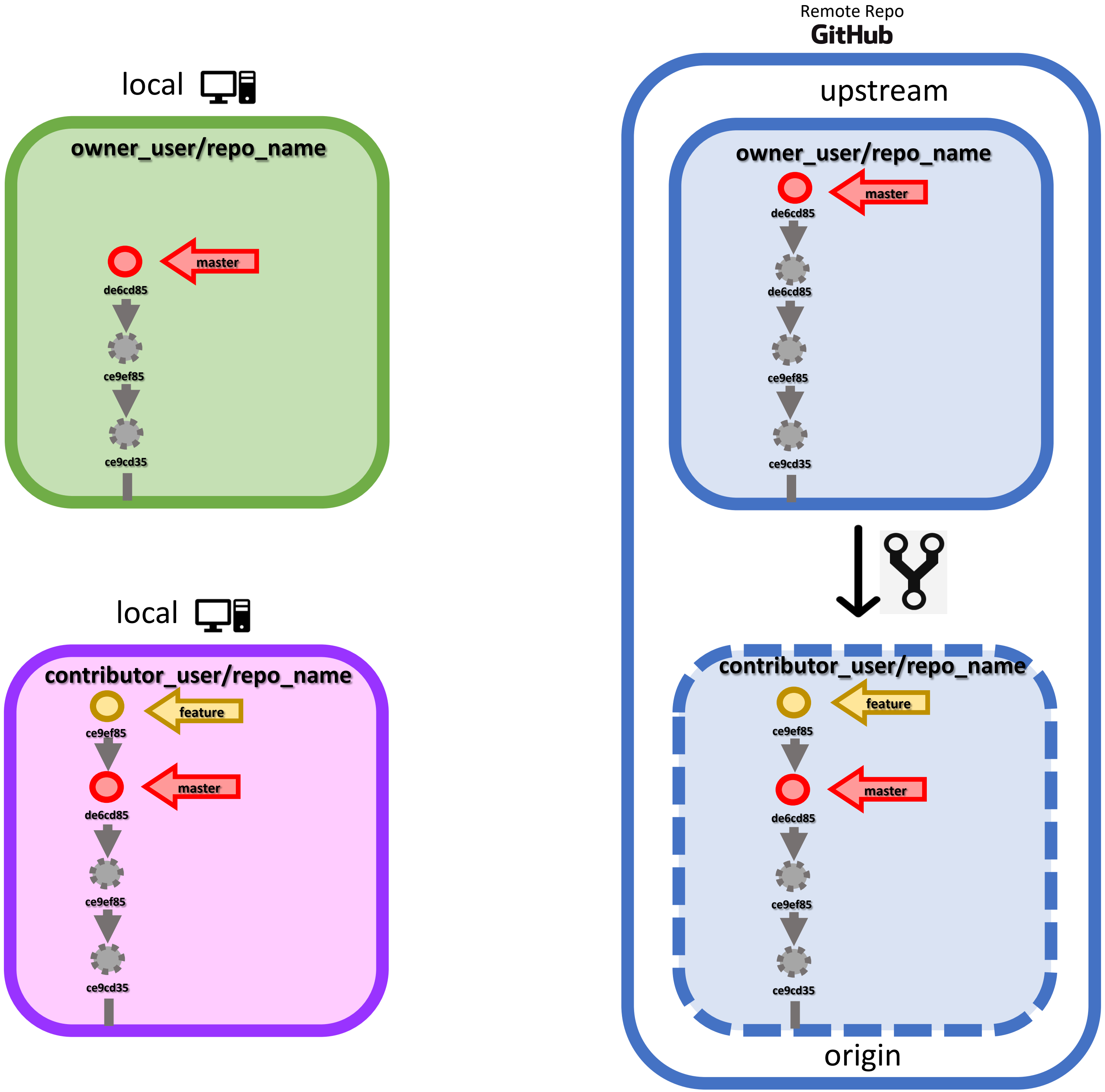
origin

Pull Request (PR)

Closed!

fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

local

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

ce9ef85

ce9cd35

upstream

owner_user/repo_name

master

de6cd85

de6cd85

ce9ef85

ce9cd35

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

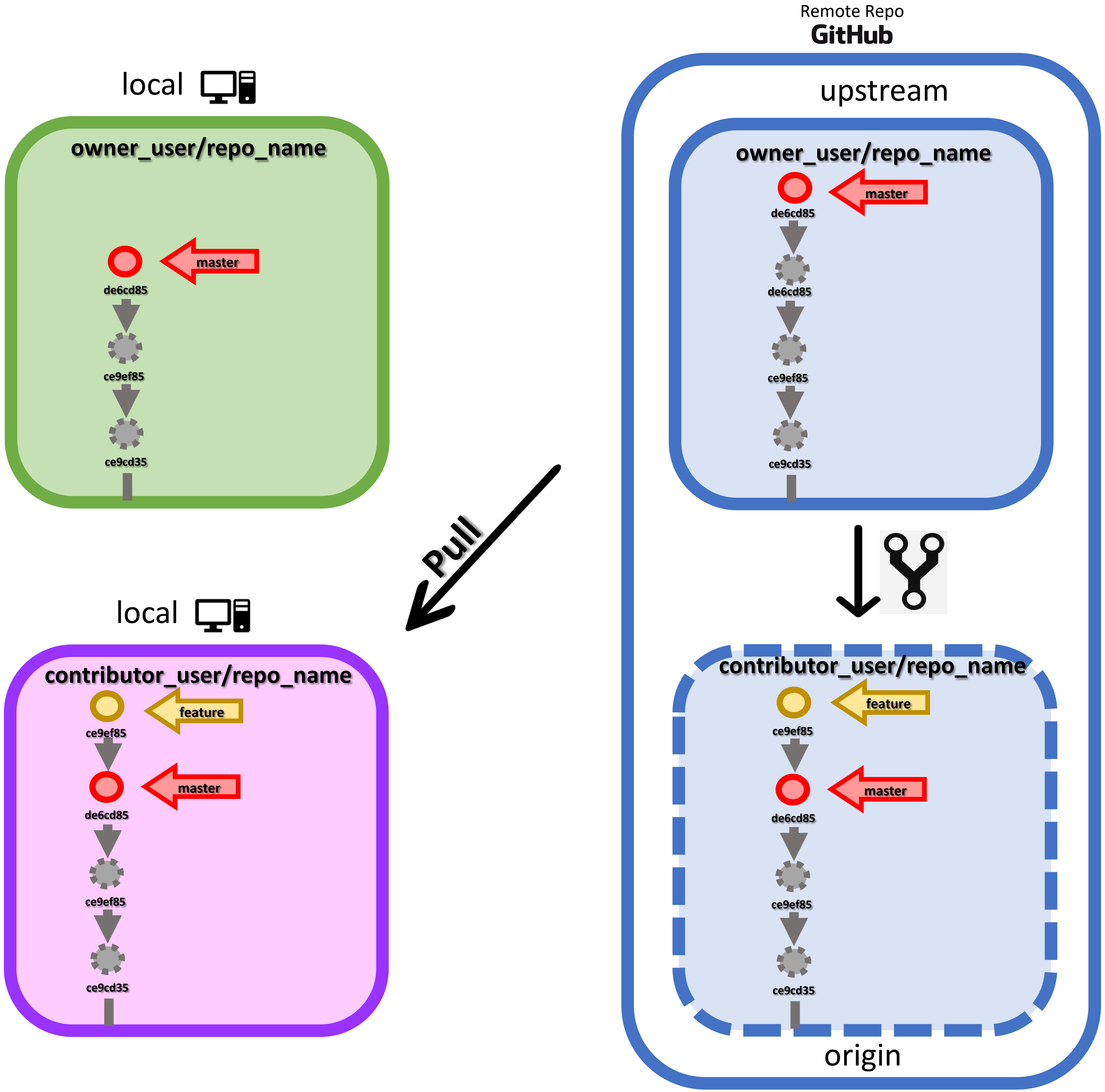
ce9ef85

ce9cd35

origin

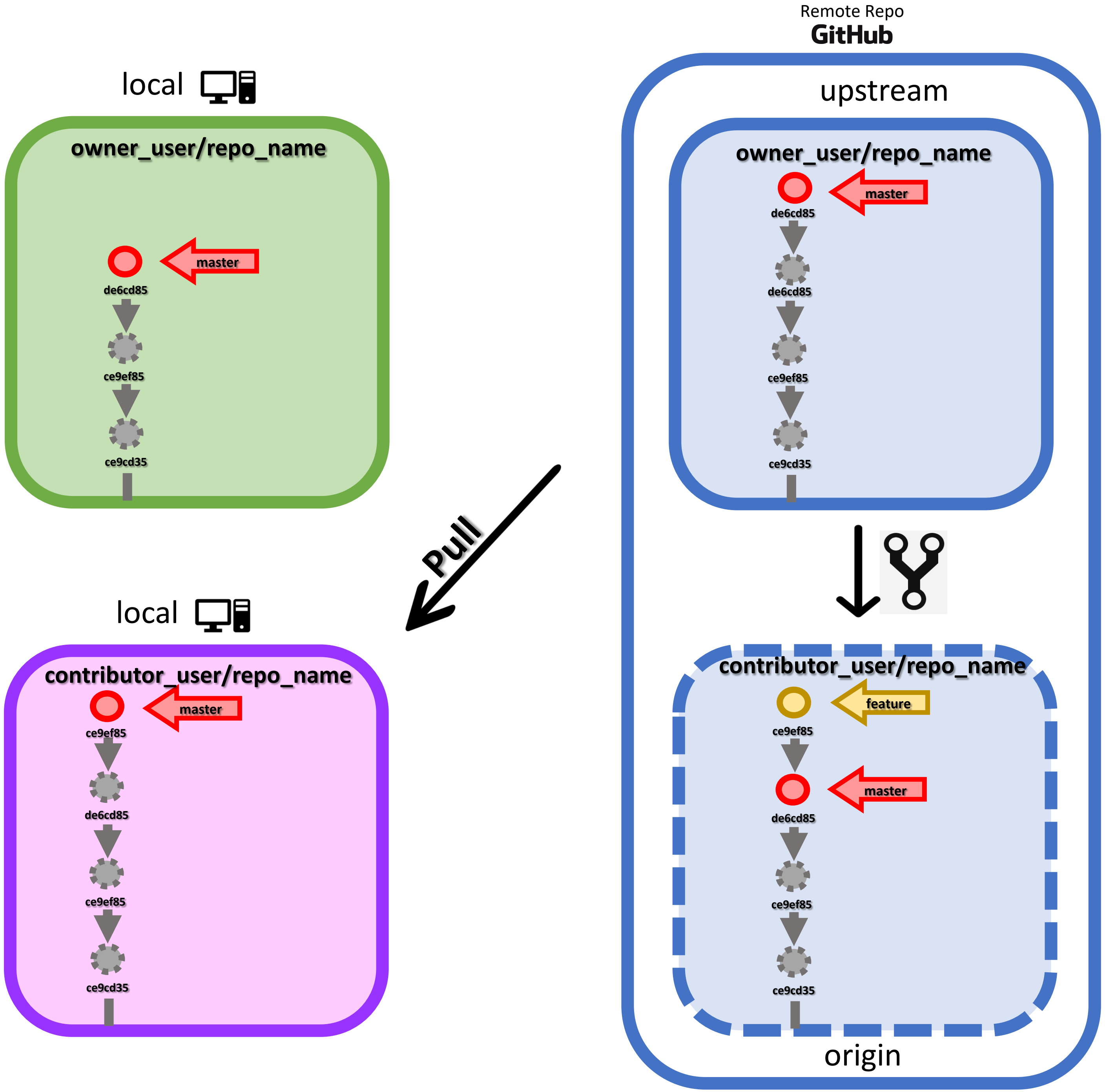
fake-sinzi
(owner)

s1nzy
(contributor)



fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

upstream

owner_user/repo_name

master

de6cd85

de6cd85

ce9ef85

ce9cd35

contributor_user/repo_name

feature

ce9ef85

master

de6cd85

ce9ef85

ce9cd35

origin

Pull

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

local

contributor_user/repo_name

master

ce9ef85

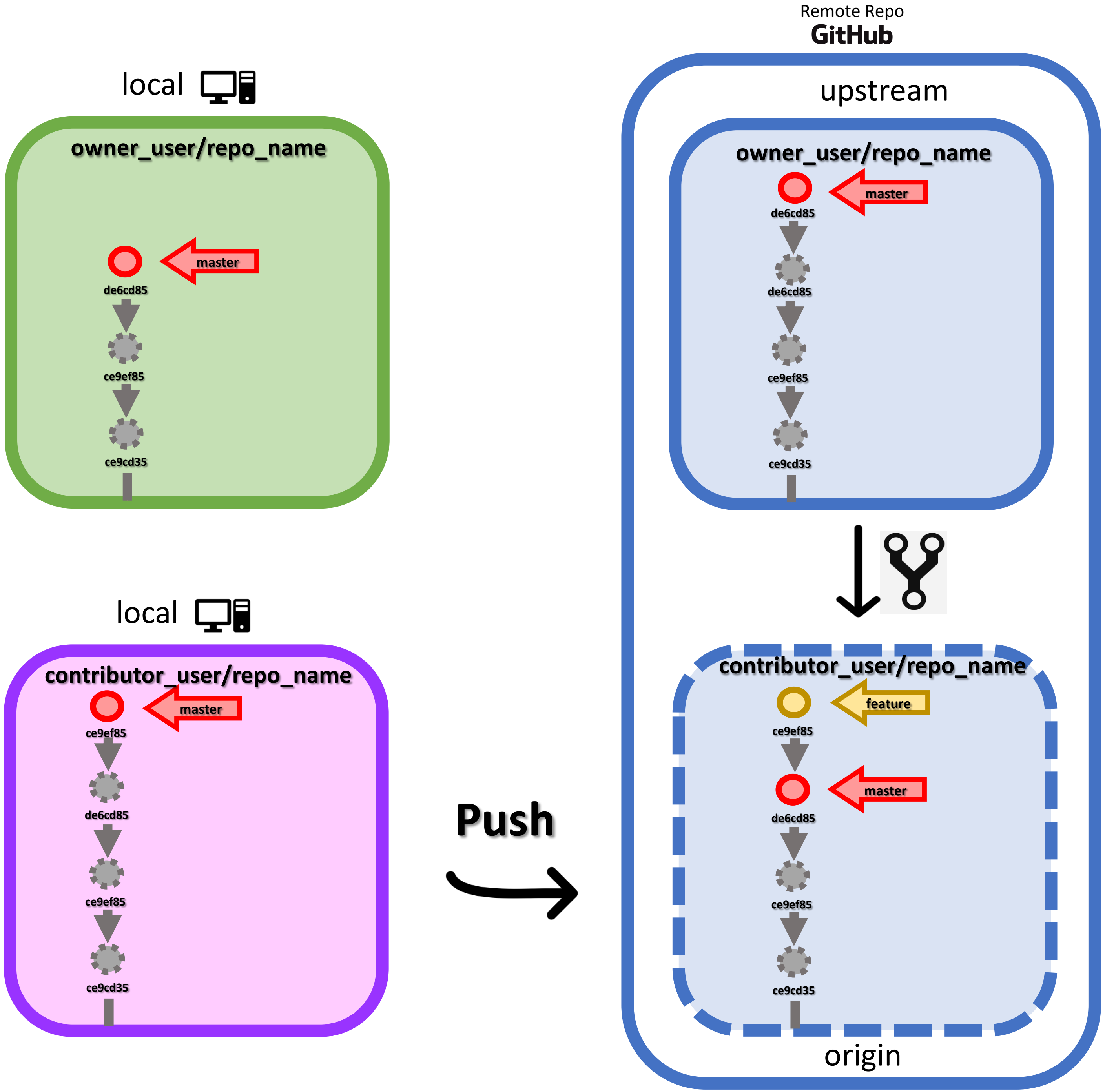
de6cd85

ce9ef85

ce9cd35

fake-sinzi
(owner)

s1nzy
(contributor)



Remote Repo
GitHub

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

local

contributor_user/repo_name

master

ce9ef85

de6cd85

ce9ef85

ce9cd35

Push

upstream

owner_user/repo_name

master

de6cd85

de6cd85

ce9ef85

ce9cd35

contributor_user/repo_name

feature

ce9ef85

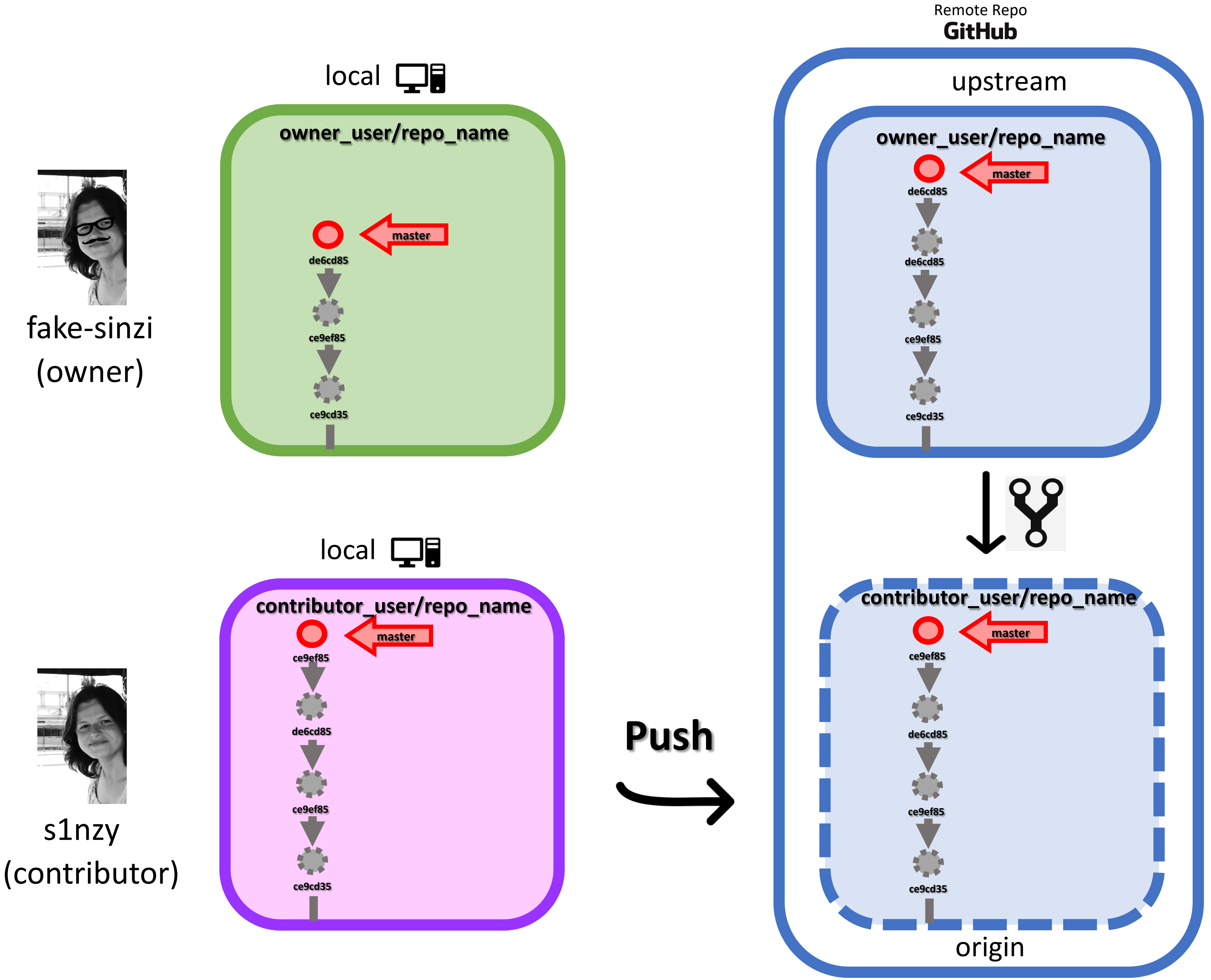
master

de6cd85

ce9ef85

ce9cd35

origin



Remote Repo
GitHub

local

owner_user/repo_name

master

de6cd85

ce9ef85

ce9cd35

fake-sinzi
(owner)

upstream

owner_user/repo_name

master

de6cd85

de6cd85

ce9ef85

ce9cd35

local

contributor_user/repo_name

master

ce9ef85

de6cd85

ce9ef85

ce9cd35

s1nzy
(contributor)

Push

contributor_user/repo_name

master

ce9ef85

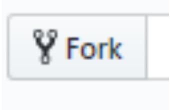
de6cd85

ce9ef85

ce9cd35

origin

How to fork a repo?

- Search the repository you are interested in
  - Owner_user/repository_name (e.g. “s1nzy/Github_Tutorial”)
- Click on the *Fork* button, next to the repo name 

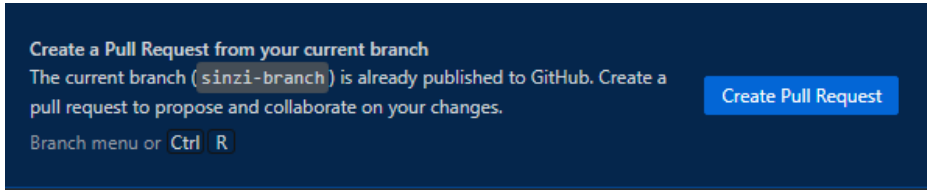
- The repo appears as one of your own repos
  - your_username/repository_name (e.g.“fake-sinzi/Github_Tutorial”)
- Clone the repository to your computer
- Strongly advised to work on a separate branch
- Once you are happy with your changes, push them to your forked remote repo (origin)
- Create a pull request to merge them with the owner’s repo (upstream)

Pull request (PR)

s1nzy

Contributor

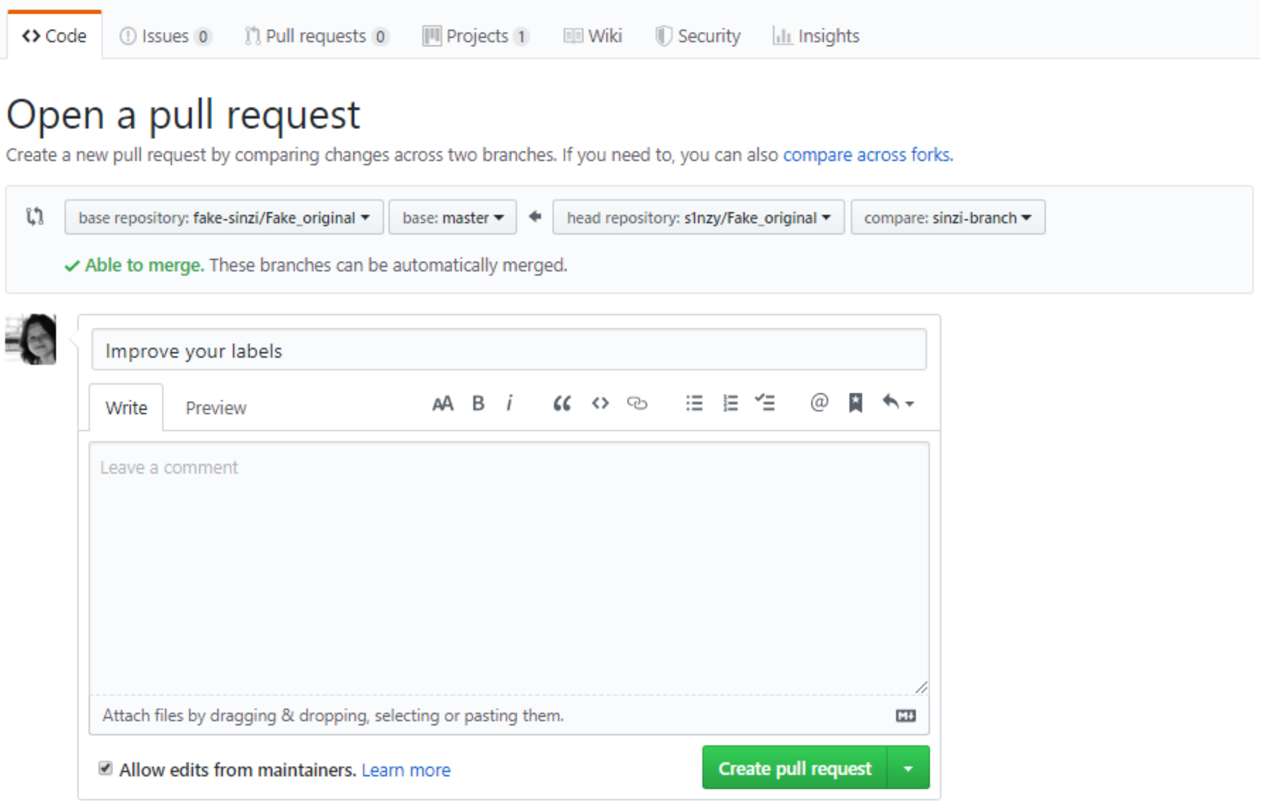
- Checkout your local *branch*
- Go to *Create Pull Request*
- Redirects you *to GitHub*



Create a Pull Request from your current branch

The current branch (`sinzi-branch`) is already published to GitHub. Create a pull request to propose and collaborate on your changes.

Branch menu or Ctrl R



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 repository: fake-sinzi/Fake_original ▾ base: master ▾ ← head repository: s1nzy/Fake_original ▾ compare: sinzi-branch ▾

✓ Able to merge. These branches can be automatically merged.

Improve your labels

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

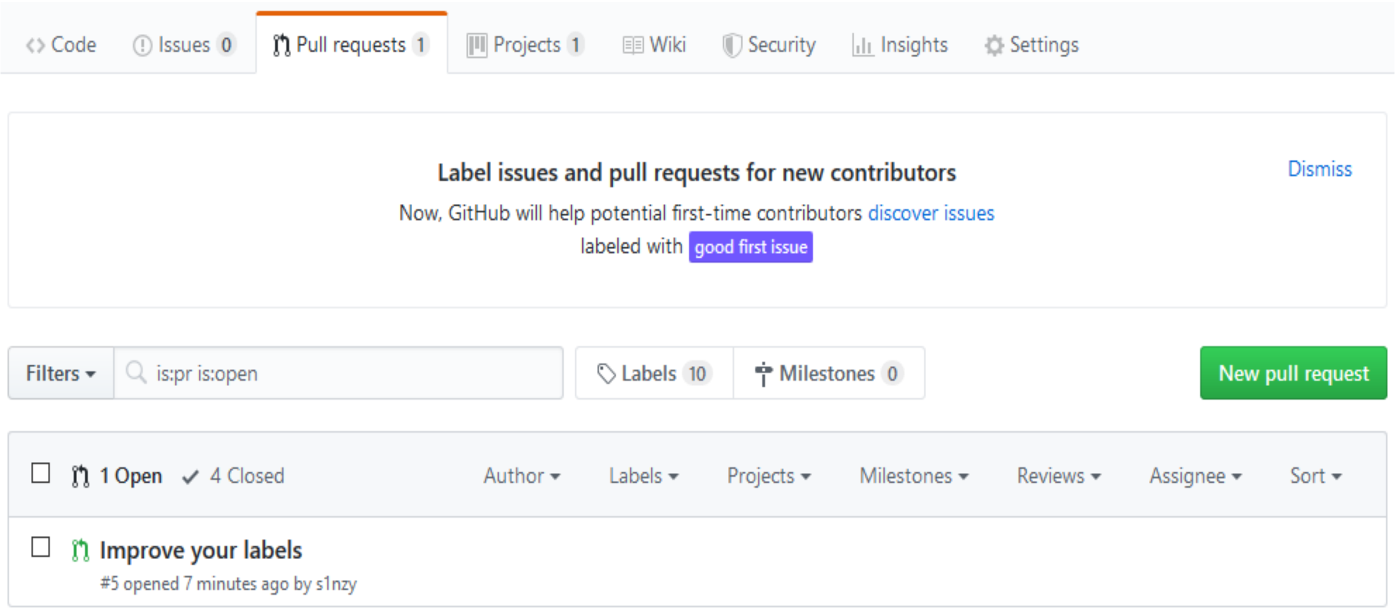
☑ Allow edits from maintainers. Learn more

Pull request (PR)

fake-sinzi

Owner

- You get an e-mail about this
- Go to your remote repo on GitHub



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

□ **Improve your labels**

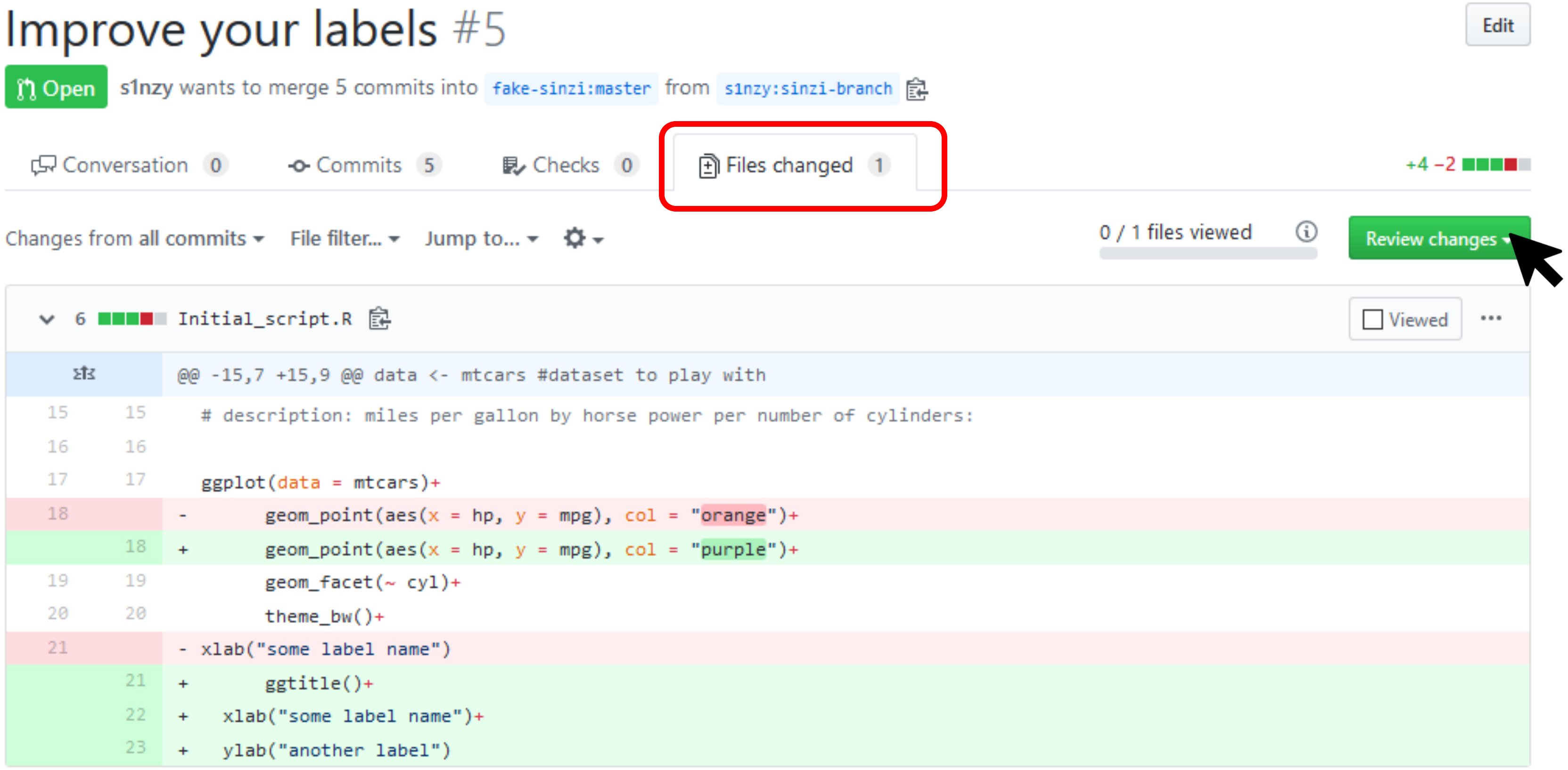
#5 opened 7 minutes ago by s1nzy

Pull request (PR)

fake-sinzi

Owner

- Time to give some feedback



Improve your labels #5

s1nzy wants to merge 5 commits into `fake-sinzi:master` from `s1nzy:sinzi-branch`

6 Initial_script.R

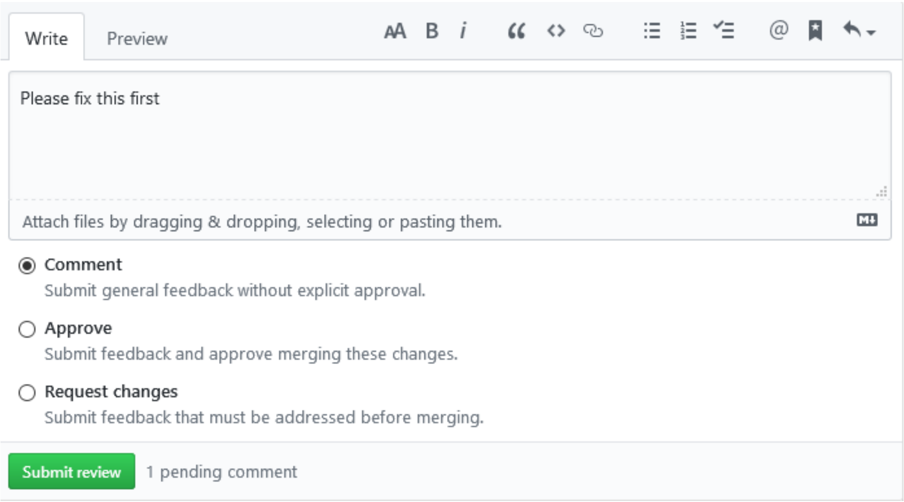
```
@@ -15,7 +15,9 @@ data <- mtcars #dataset to play with
  # description: miles per gallon by horse power per number of cylinders:

  ggplot(data = mtcars)+
-        geom_point(aes(x = hp, y = mpg), col = "orange")+
+        geom_point(aes(x = hp, y = mpg), col = "purple")+
         geom_facet(~ cyl)+
         theme_bw()+
- xlab("some label name")
+        ggtitle()+
+   xlab("some label name")+
+   ylab("another label")
```

Pull request (PR)

fake-sinzi

Owner

- Click and drag on the lines you want to comment on
- (they become highlighted)
- Click on Start a review
- When you are satisfied go to Finish your review 1 ▾



Please fix this first

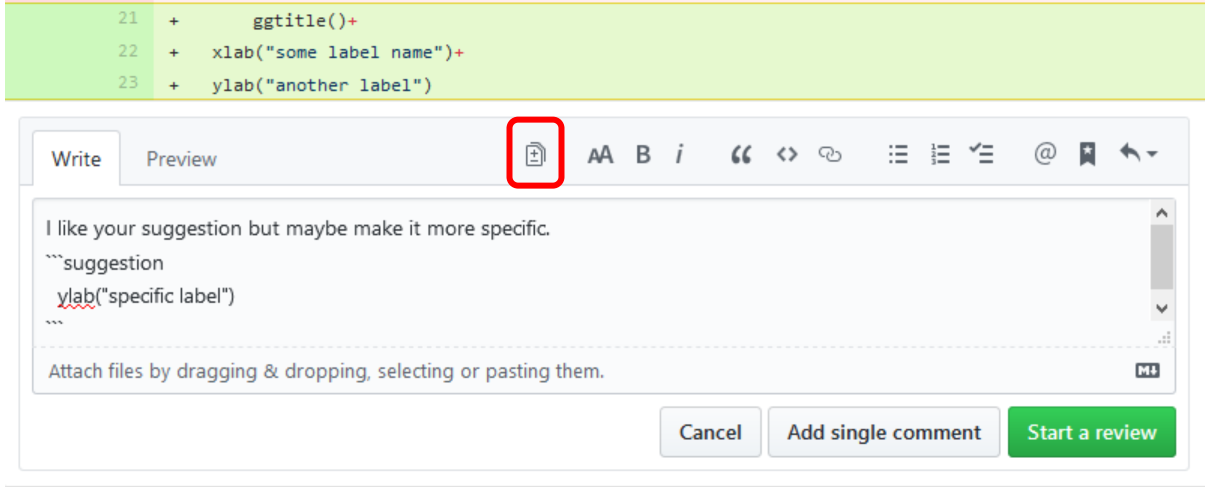
Attach files by dragging & dropping, selecting or pasting them.

● Comment
Submit general feedback without explicit approval.

○ Approve
Submit feedback and approve merging these changes.

○ Request changes
Submit feedback that must be addressed before merging.

Submit review
1 pending comment



```
+      ggtitle()+
+  xlab("some label name")+
+  ylab("another label")
```

I like your suggestion but maybe make it more specific.

~~~
```suggestion
 ylab("specific label")
```
~~~

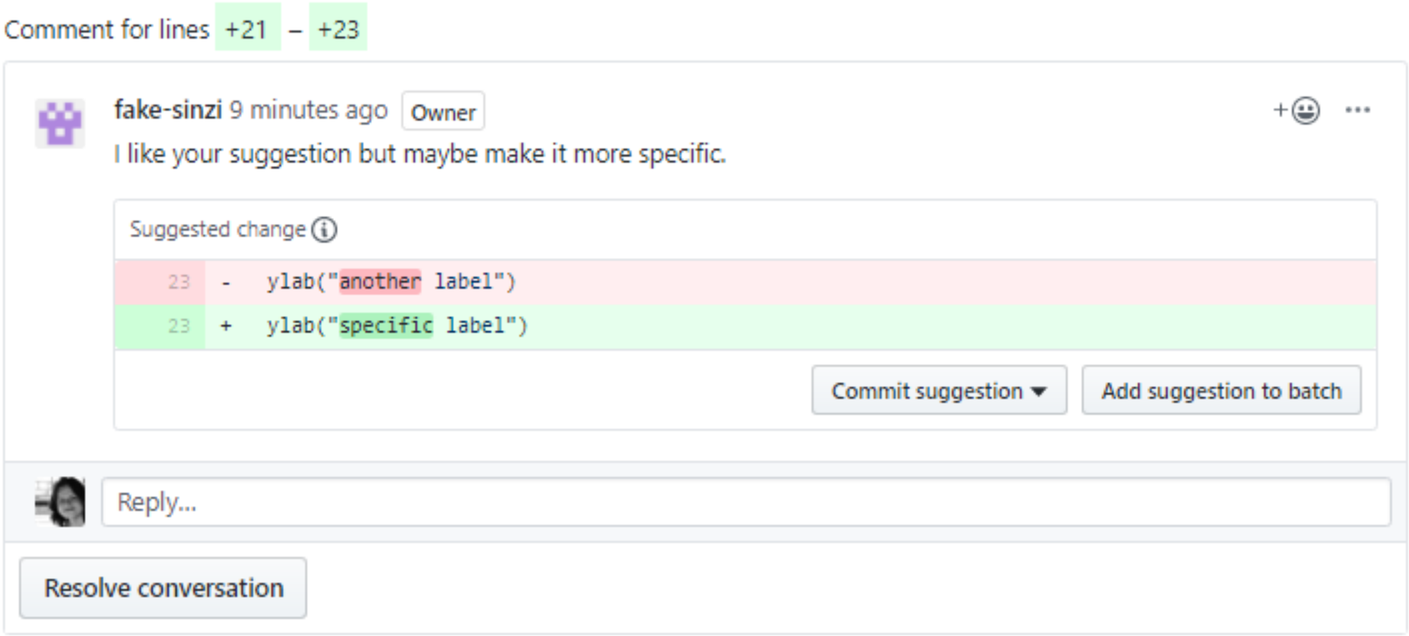
Attach files by dragging & dropping, selecting or pasting them.
Cancel
Add single comment
Start a review

Pull request (PR)

Contributor

s1nzy

- Go to Rscript.R and make change
- Commit & Push



Comment for lines +21 – +23

fake-sinzi 9 minutes ago Owner

I like your suggestion but maybe make it more specific.

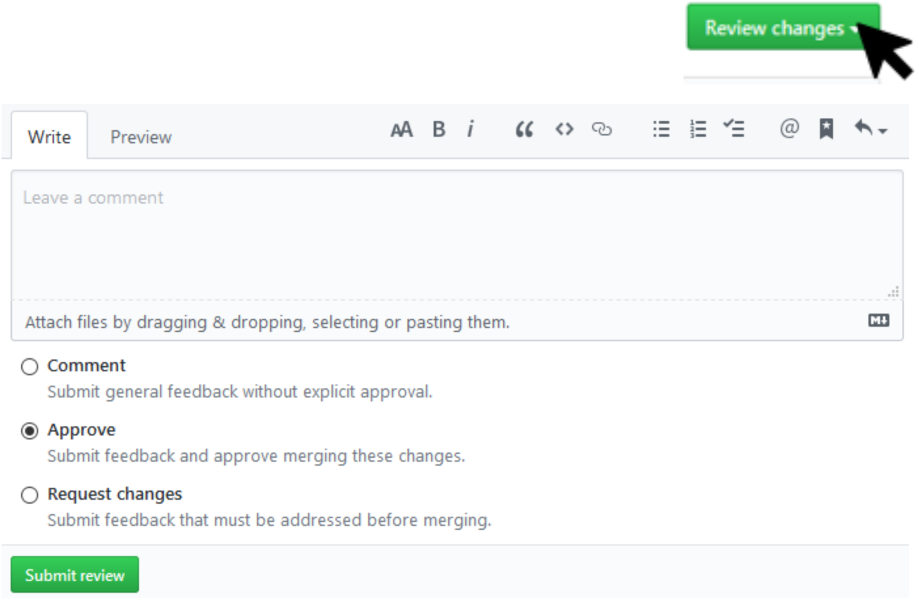
Suggested change

```
23  -   ylab("another label")
23  +   ylab("specific label")
```

Pull request (PR)

Owner

fake-sinzi



Review changes

Write Preview

Leave a comment

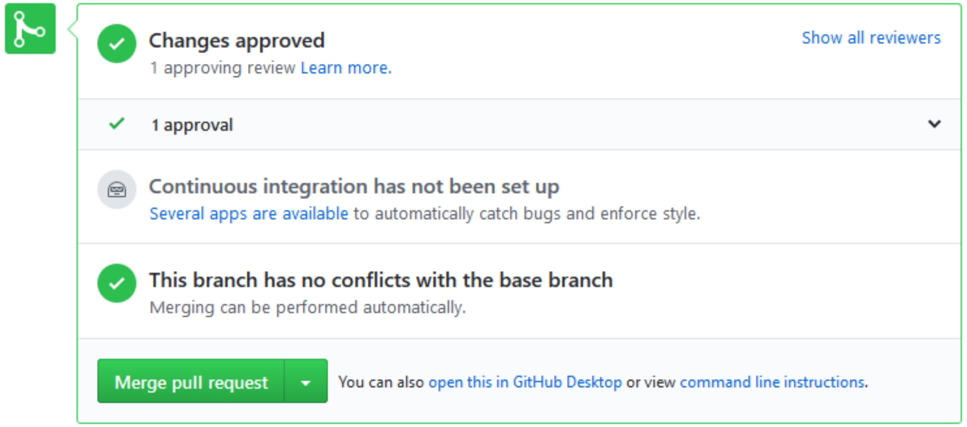
Attach files by dragging & dropping, selecting or pasting them.

○ Comment
Submit general feedback without explicit approval.

● Approve
Submit feedback and approve merging these changes.

○ Request changes
Submit feedback that must be addressed before merging.

Submit review



✓ Changes approved
1 approving review Learn more.
Show all reviewers

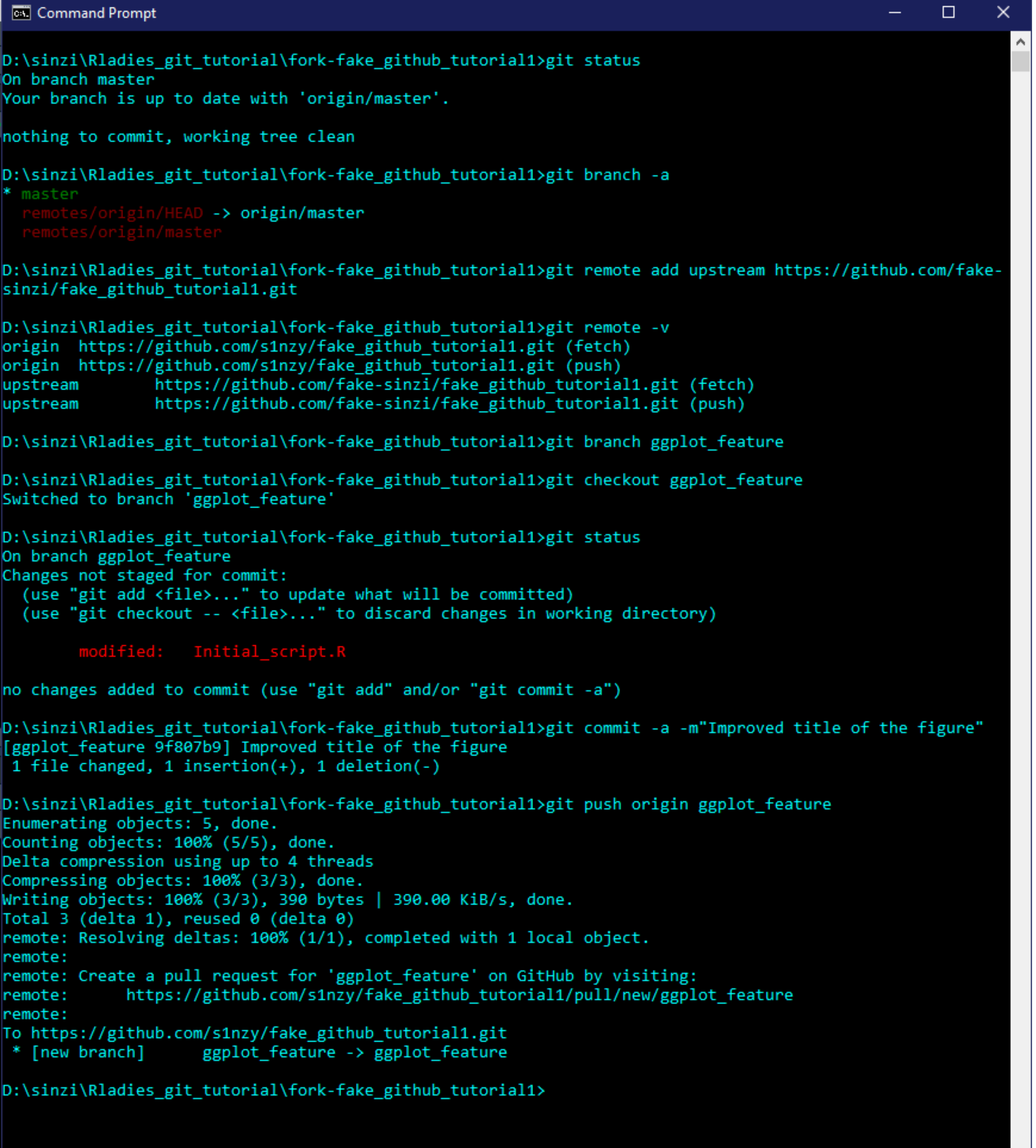
✓ 1 approval

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.

- Confirm merge
- Pull request closes automatically



```
D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git remote add upstream https://github.com/fake-
sinzi/fake_github_tutorial1.git

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git remote -v
origin  https://github.com/s1nzy/fake_github_tutorial1.git (fetch)
origin  https://github.com/s1nzy/fake_github_tutorial1.git (push)
upstream        https://github.com/fake-sinzi/fake_github_tutorial1.git (fetch)
upstream        https://github.com/fake-sinzi/fake_github_tutorial1.git (push)

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git branch ggplot_feature

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git checkout ggplot_feature
Switched to branch 'ggplot_feature'

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git status
On branch ggplot_feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Initial_script.R

no changes added to commit (use "git add" and/or "git commit -a")

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git commit -a -m"Improved title of the figure"
[ggplot_feature 9f807b9] Improved title of the figure
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>git push origin ggplot_feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 390 bytes | 390.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'ggplot_feature' on GitHub by visiting:
remote:      https://github.com/s1nzy/fake_github_tutorial1/pull/new/ggplot_feature
remote:
To https://github.com/s1nzy/fake_github_tutorial1.git
 * [new branch]      ggplot_feature -> ggplot_feature

D:\sinzi\Rladies_git_tutorial\fork-fake_github_tutorial1>
```

Using the CMD line

`git status`

which branch is checked-out and if it is up to date with the origin

`git branch -a`

shows all branches (local and remote)

`git remote add upstream <link to owner/repo>`

adds the upstream repository (same link as with cloning)

`git remote -v`

shows all remotes

`git branch branch_name`

creates new branch

`git checkout branch_name`

switches to branch

`git commit –a –m ”message”`

adds file to staging area and commits changes

`git push remote_name branch_name`

updates remote repo on github with the new branch

You now have the option to make a PR by copy-pasting the link into your browser

Pull request

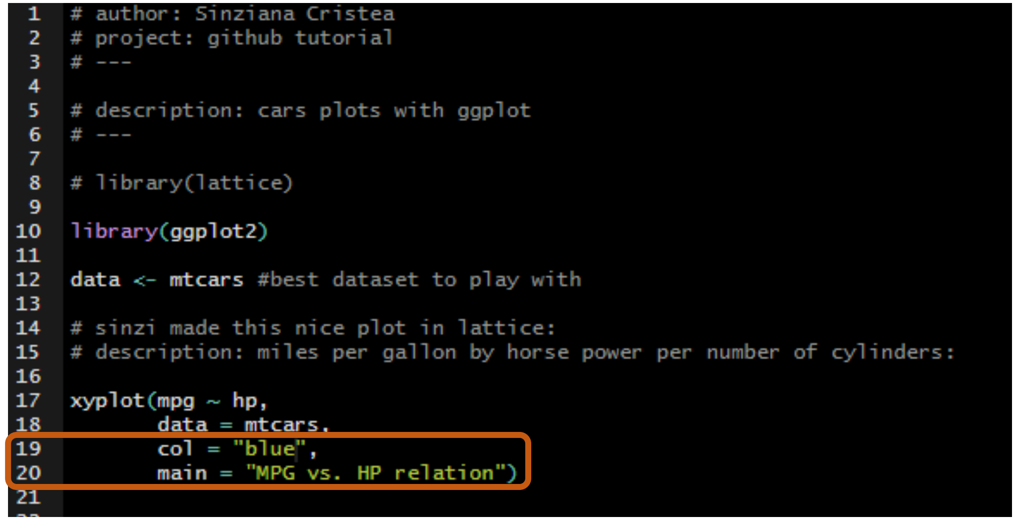
- You are directed to GitHub and you manage the PR from there as shown above.
- Your PR is approved and merged into the upstream master by the repo owner
- Congrats! You just made your first contribution!

Learn by doing! Hands-on part 2

- Work in pairs:
- Find each other’s repos and Fork your partner’s repo
- Clone the forked repo to your laptop (local forked repo)
- Create a new branch in the forked repo
  - Change the lattice plot to a ggplot
  - Push branch to origin
- Create pull request (PR) to merge your change to upstream master → become a collaborator
- Go to github to review PR for your own repo, make comments, code suggestions and merge it to your own repo.

Bonus: How to deal with *conflicts*?

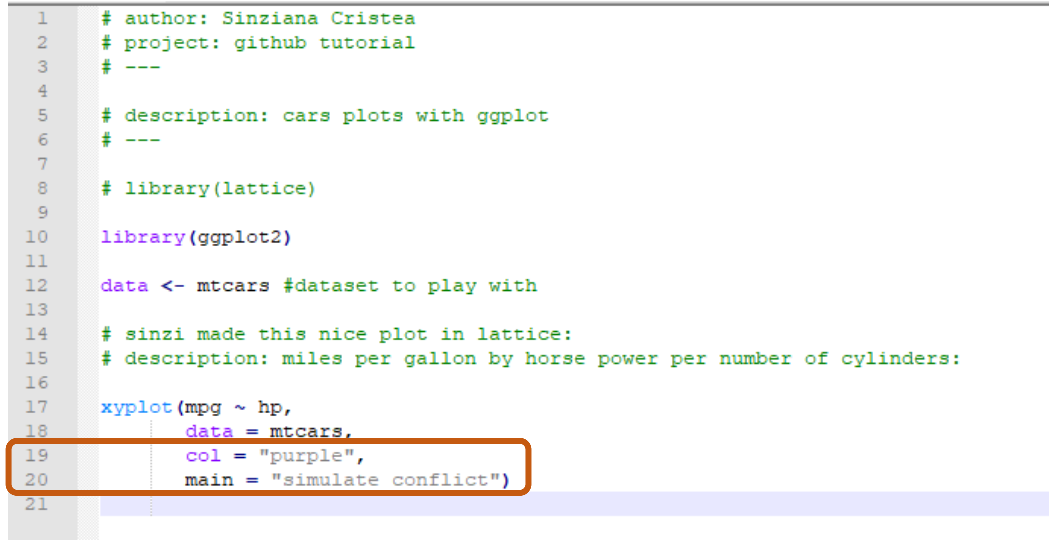
- Situation:
- Ooops! You forgot to add the title of the plot and you decided to change the colour of the points to match the Rladies theme.
- You are unaware that your collaborator made changes to the same lines
- You make the changes on your branch and submit a PR
- You see a message that the merge cannot be done automatically because there is a conflict.
- What to do?



```
# author: Sinziana Cristea
# project: github tutorial
# ---

# description: cars plots with ggplot
# ---

# library(lattice)

library(ggplot2)

data <- mtcars #best dataset to play with

# sinzi made this nice plot in lattice:
# description: miles per gallon by horse power per number of cylinders:

xyplot(mpg ~ hp,
       data = mtcars,
       col = "blue",
       main = "MPG vs. HP relation")
```

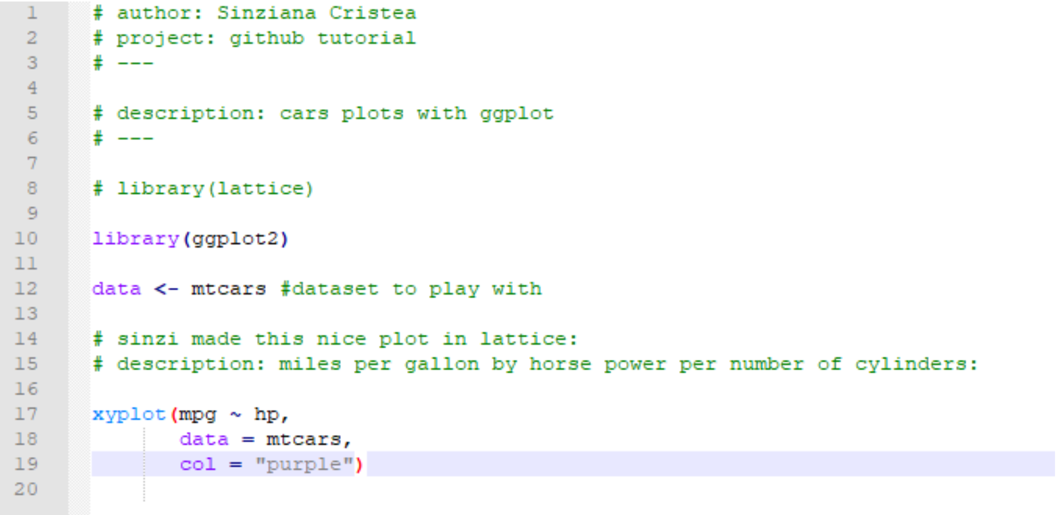
master

Merge



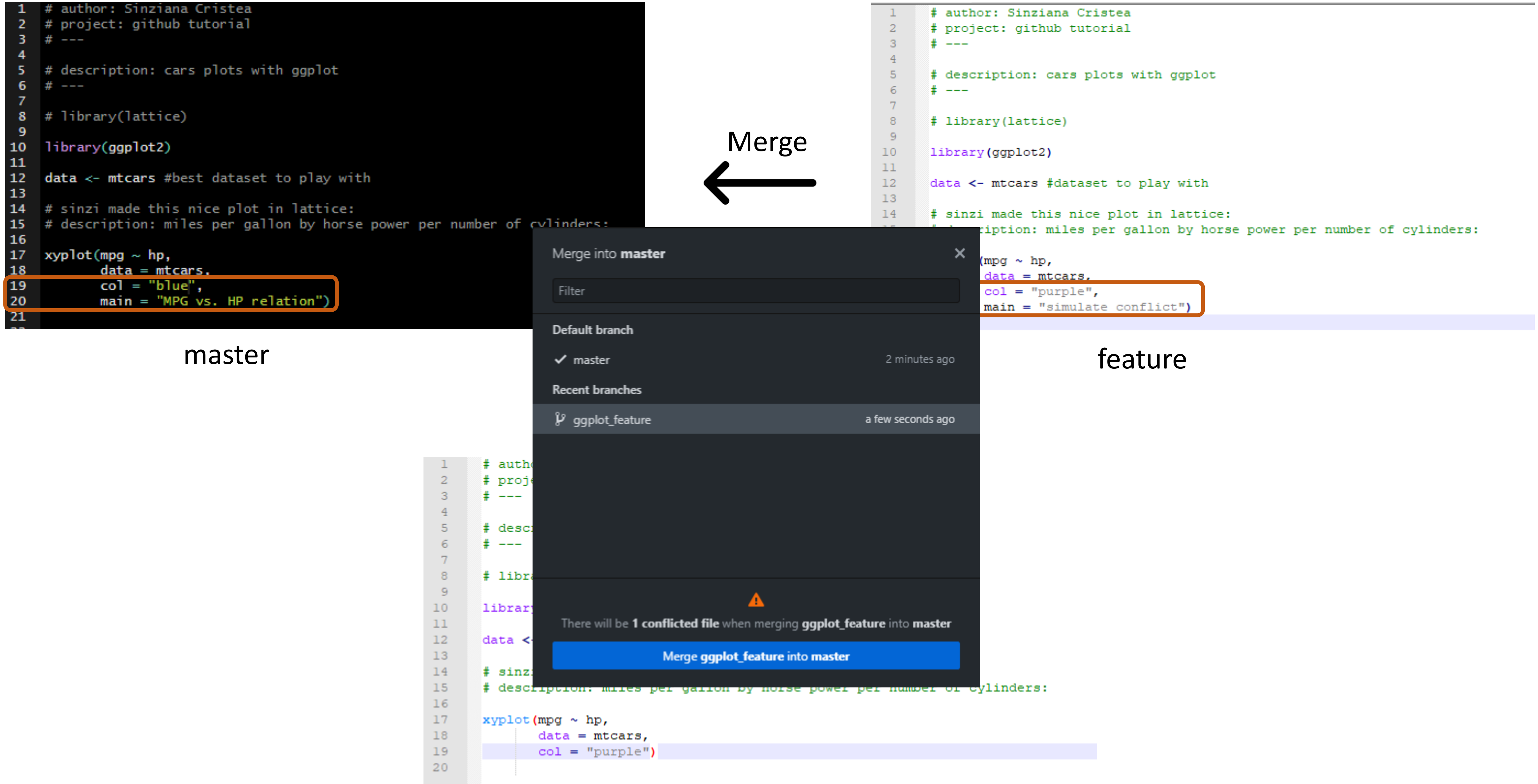
```
# author: Sinziana Cristea
# project: github tutorial
# ---

# description: cars plots with ggplot
# ---

# library(lattice)

library(ggplot2)

data <- mtcars #dataset to play with

# sinzi made this nice plot in lattice:
# description: miles per gallon by horse power per number of cylinders:

xyplot(mpg ~ hp,
       data = mtcars,
       col = "purple",
       main = "simulate conflict")
```

feature



```
# author: Sinziana Cristea
# project: github tutorial
# ---

# description: cars plots with ggplot
# ---

# library(lattice)

library(ggplot2)

data <- mtcars #dataset to play with

# sinzi made this nice plot in lattice:
# description: miles per gallon by horse power per number of cylinders:

xyplot(mpg ~ hp,
       data = mtcars,
       col = "purple")
```

Original file



```
# author: Sinziana Cristea
# project: github tutorial
# ---

# description: cars plots with ggplot
# ---

# library(lattice)

library(ggplot2)

data <- mtcars #best dataset to play with

# sinzi made this nice plot in lattice:
# description: miles per gallon by horse power per number of cylinders:

xyplot(mpg ~ hp,
       data = mtcars,
       col = "blue",
       main = "MPG vs. HP relation")
```

master

Merge

```
# author: Sinziana Cristea
# project: github tutorial
# ---

# description: cars plots with ggplot
# ---

# library(lattice)

library(ggplot2)

data <- mtcars #dataset to play with

# sinzi made this nice plot in lattice:
ription: miles per gallon by horse power per number of cylinders:

(mpg ~ hp,
 data = mtcars,
 col = "purple",
 main = "simulate conflict")
```

feature

Merge into **master**

Default branch

✓ master — 2 minutes ago

Recent branches

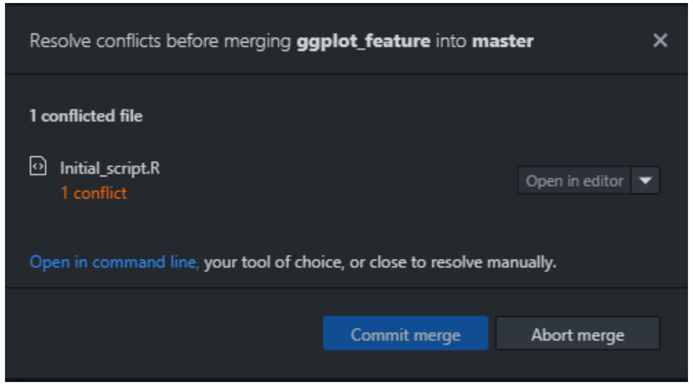
ggplot_feature — a few seconds ago

⚠ There will be **1 conflicted file** when merging **ggplot_feature** into **master**

```
# auth
# proj
# ---

# desc
# ---

# libr

library

data <

# sinz
# description: miles per gallon by horse power per number of cylinders:

xyplot(mpg ~ hp,
       data = mtcars,
       col = "purple")
```

Original file

Solving the conflict



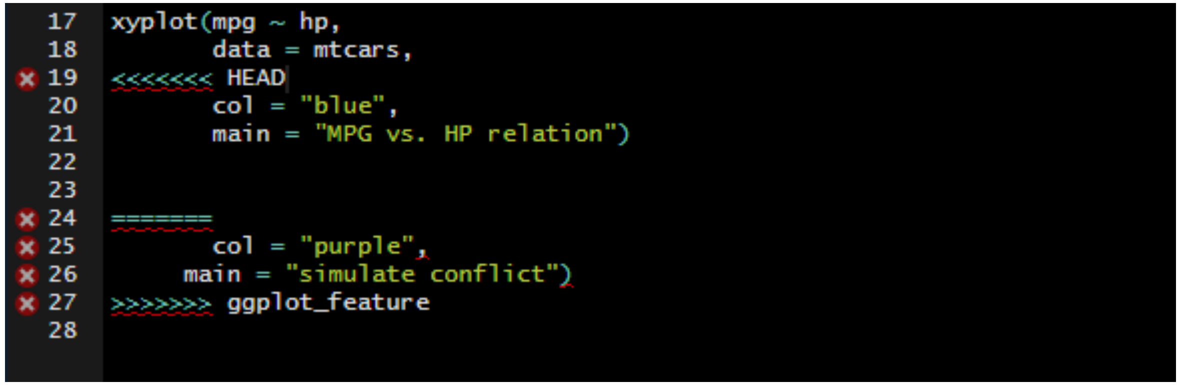
Resolve conflicts before merging **ggplot_feature** into **master**

1 conflicted file

Initial_script.R
1 conflict

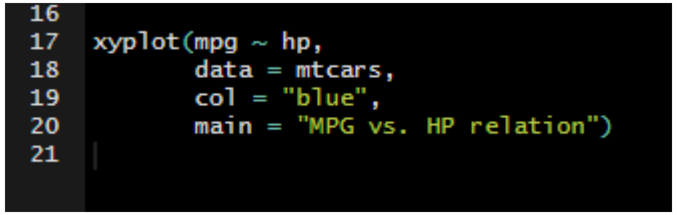
Open in command line, your tool of choice, or close to resolve manually.

Go to editor



```
xyplot(mpg ~ hp,
       data = mtcars,
<<<<<<< HEAD
       col = "blue",
       main = "MPG vs. HP relation")


=======
       col = "purple",
     main = "simulate conflict")
>>>>>>> ggplot_feature
```

Manual edit & save



```
xyplot(mpg ~ hp,
       data = mtcars,
       col = "blue",
       main = "MPG vs. HP relation")
```

Merge successful!

Recap & Take-home

- Basic workflow with git command and GitDesktop
  - Create local and remote repositories
  - Make commits
  - Work with multiple branches:
    - Have more versions of your script
    - Work with others
  - Fork a repository
  - Create a pull request
  - Review and merge a pull request

Additional resources

- Github hello world

  https://guides.github.com/activities/hello-world/#commit
- David Mahler on YouTube: https://www.youtube.com/watch?v=Gg4bLk8cGNo
- Working with the CMD line

  https://github.github.com/training-kit/downloads/github-git-cheat-sheet/
- How repos work:
- https://marklodato.github.io/visual-git-guide/index-en.html
- https://ohshitgit.com/
- http://git-school.github.io/visualizing-git/#free

Fin