

Introduction to Git — Fall 2021

Lecture 6: Working with remotes



HPC2N



SNIC

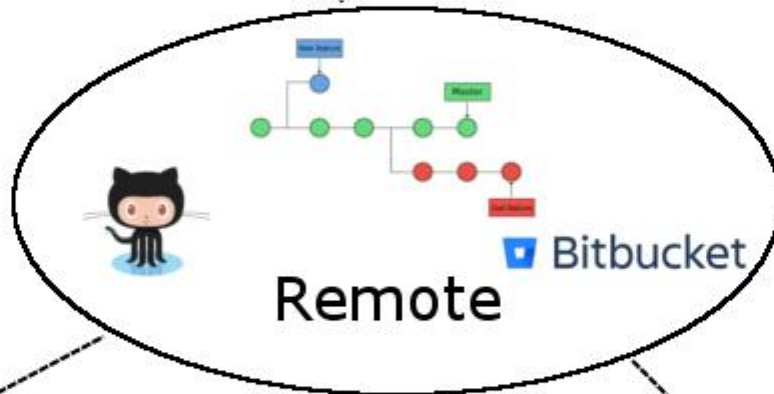
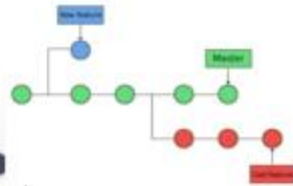
Slides: <https://hackmd.io/@hpc2n-git-2021/L6-remotes#/>

Basic concepts

A remote repository is a version of the project which can be hosted in your local machine, some network, or over the internet[1] where you and your collaborators can push or pull code modifications.

In addition to this, a remote is a way to backup your repository.

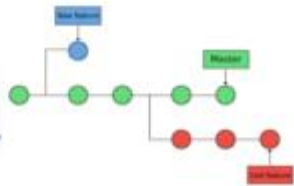
Mirko



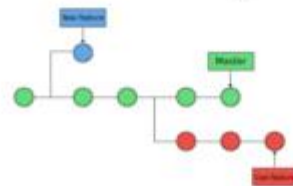
Remote

Bitbucket

Pedro



Birgitte



Basic concepts cont.

The command

```
$ git remote -v  
origin  git@bitbucket.org:arm2011/gitcourse.git (fetch)  
origin  git@bitbucket.org:arm2011/gitcourse.git (push)
```

displays the remotes that are already set up where you can *fetch* and *pull* changes. In this case there is only a single remoted called **origin**.

Adding remotes

A remote repository can be added manually with the command

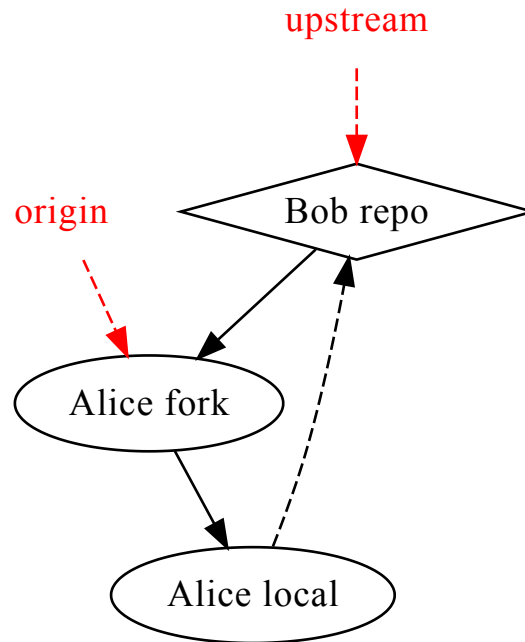
```
$ git remote add remote_name location  
  
$ git remote add origin https://github.com/aliceuser2020/my-first-project.git  
  
$ git remote -v  
origin https://github.com/aliceuser2020/my-first-project.git  
origin https://github.com/aliceuser2020/my-first-project.git
```

where the location of the remote can be an URL or the path if that is in your local machine.

Protocols:

- local -> `git clone /opt/git/project.git`
- SSH -> `git clone ssh://user@server:project.git`
- HTTP -> `git clone`
<http://example.com/gitproject.git>
- Git

Why do we need more than one remote?



```
$ git remote add upstream git@bitbucket.org:bob/gitcourse.git

$ git remote -v
origin  https://github.com/aliceuser2020/my-first-project.git
origin  https://github.com/aliceuser2020/my-first-project.git
upstream      https://github.com/bobuser2020/my-first-proje
upstream      https://github.com/bobuser2020/my-first-proje
```


Working with remotes

One can push or fetch/pull to or from remotes by

```
$ git push  remote_name branch_name  
$ git fetch remote_name branch_name  
$ git pull  remote_name branch_name
```

In case you obtained the repository by cloning an existing one you will have the **origin** remote. You can do push/fetch/pull for this remote with

```
$ git push origin master  
$ git fetch origin master  
$ git pull origin master
```

or

```
$ git push  
$ git fetch  
$ git pull
```

because the remote *origin* and the *master* branch are configured for pushing and pulling by default upon cloning.

The command:

```
$ git pull
```

brings all the changes (branches) that are in the remote and tries to merge them with your local repo. The default behavior of *git pull* is in the *\$GIT_DIR/config* file:

```
[remote "origin"]  
  fetch = +refs/heads/*:refs/remotes/origin/*
```

In fact, *git pull* is a combination of two commands:

```
$ git fetch  
$ git merge
```

The command

```
$ git push
```

will send all the changes (branches) to the remote by default. This can be changed by applying:

```
git config --global push.default matching(default), current,
```

Displaying remote information

```
$ git remote show origin
* remote origin
Fetch URL: git@bitbucket.org:arm2011/gitcourse.git
Push URL: git@bitbucket.org:arm2011/gitcourse.git
HEAD branch: master
Remote branches:
    experiment      tracked
    feature         tracked
    less-salt       tracked
    master          tracked
    nested-feature  tracked
Local branches configured for 'git pull':
    feature         merges with remote feature
    master          merges with remote master
    nested-feature  merges with remote nested-feature
```

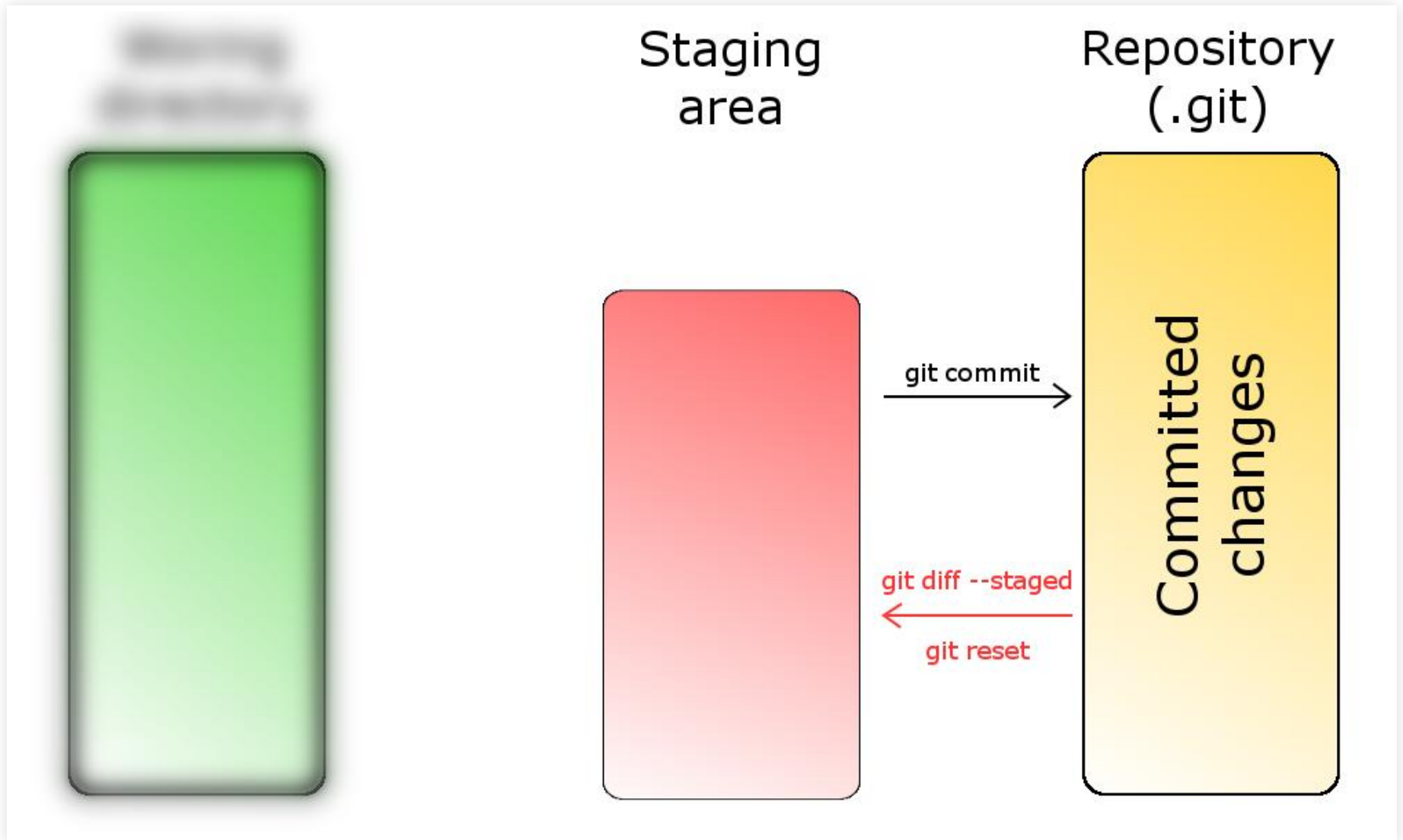
Renaming remotes

```
$ git remote rename initial_name new_name
```

Deleting remotes

```
$ git remote remove remote_name
```

Bare repositories



A bare repository is a repository with no working directory.

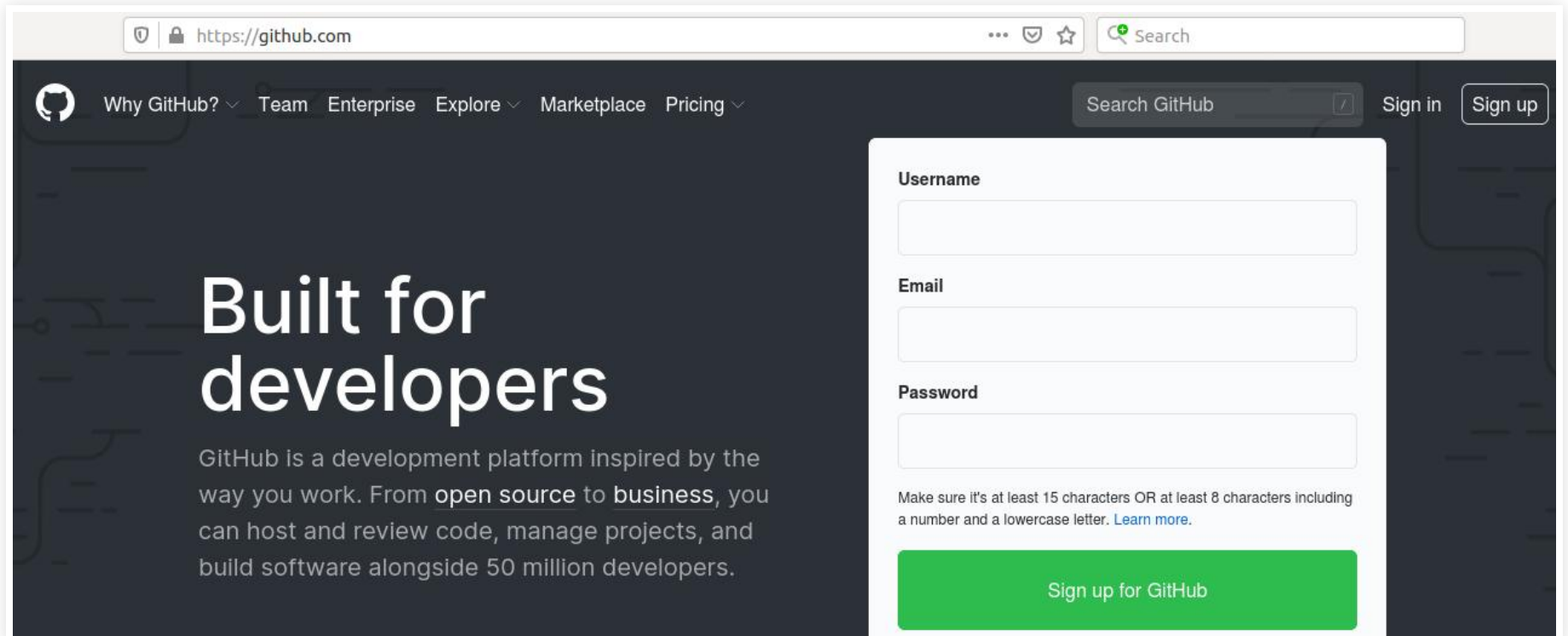
Creating a bare repository

```
$ mkdir bare.git && cd bare.git  
$ git init --bare
```

Cloning a bare repository cont.

```
$ git clone --bare location
```

Using GitHub



The image shows a screenshot of the GitHub website. The browser's address bar displays 'https://github.com'. The navigation bar includes links for 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar is present on the right, along with 'Sign in' and 'Sign up' buttons. The main content area features the GitHub logo and the text 'Built for developers'. Below this, a paragraph describes GitHub as a development platform. A white sign-up form is overlaid on the right side of the page, containing fields for 'Username', 'Email', and 'Password', along with a 'Sign up for GitHub' button.

https://github.com

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 50 million developers.

Username

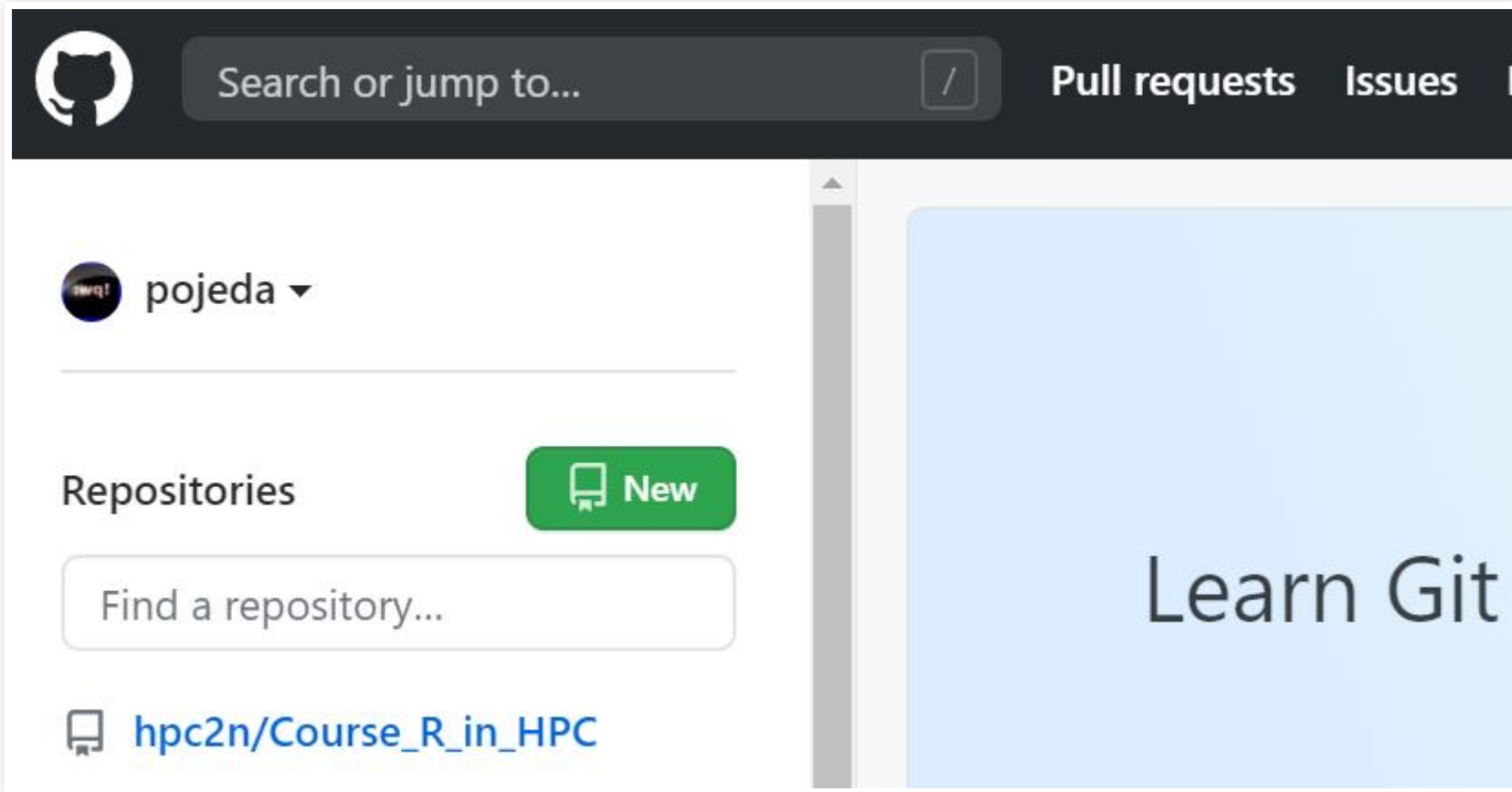
Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

Upon login into your GitHub account you will see the following option to create a new repository



Here, you can choose the type of repository that is appropriate to your needs (public/private), if you want to add *README* and *.gitignore* files and also the type of license for your project,

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *



my-first-repo



Great repository names are short and memorable. Need inspiration? How about **expert-waddle**?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

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

GitHub will suggest some steps that you can take for your brand-new repository:

 Set up in Desktop or `git@github.com:pojeda/my-first-repo.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# my-first-repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin git@github.com:pojeda/my-first-repo.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:pojeda/my-first-repo.git  
git branch -M master  
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Setting ssh-keys

1. `ssh-keygen -t rsa -b 4096 -C`
 `"pedro@gmail.com"`
2. `eval $(ssh-agent -s)`
3. `ssh-add ~/.ssh/id_rsa`
4. `clip < ~/.ssh/id_rsa.pub` (it copies the ssh key that has got generated)

5. Go to your remote repository on github.com and then **Settings** -> **SSH and GPG keys** -> new SSH key -> write a title and paste the copied SSH key and save it
6. check if the key was properly set on github/bitbucket

```
$ ssh -T git@bitbucket.org  
$ ssh -T git@github.com
```

SSH keys / Add new

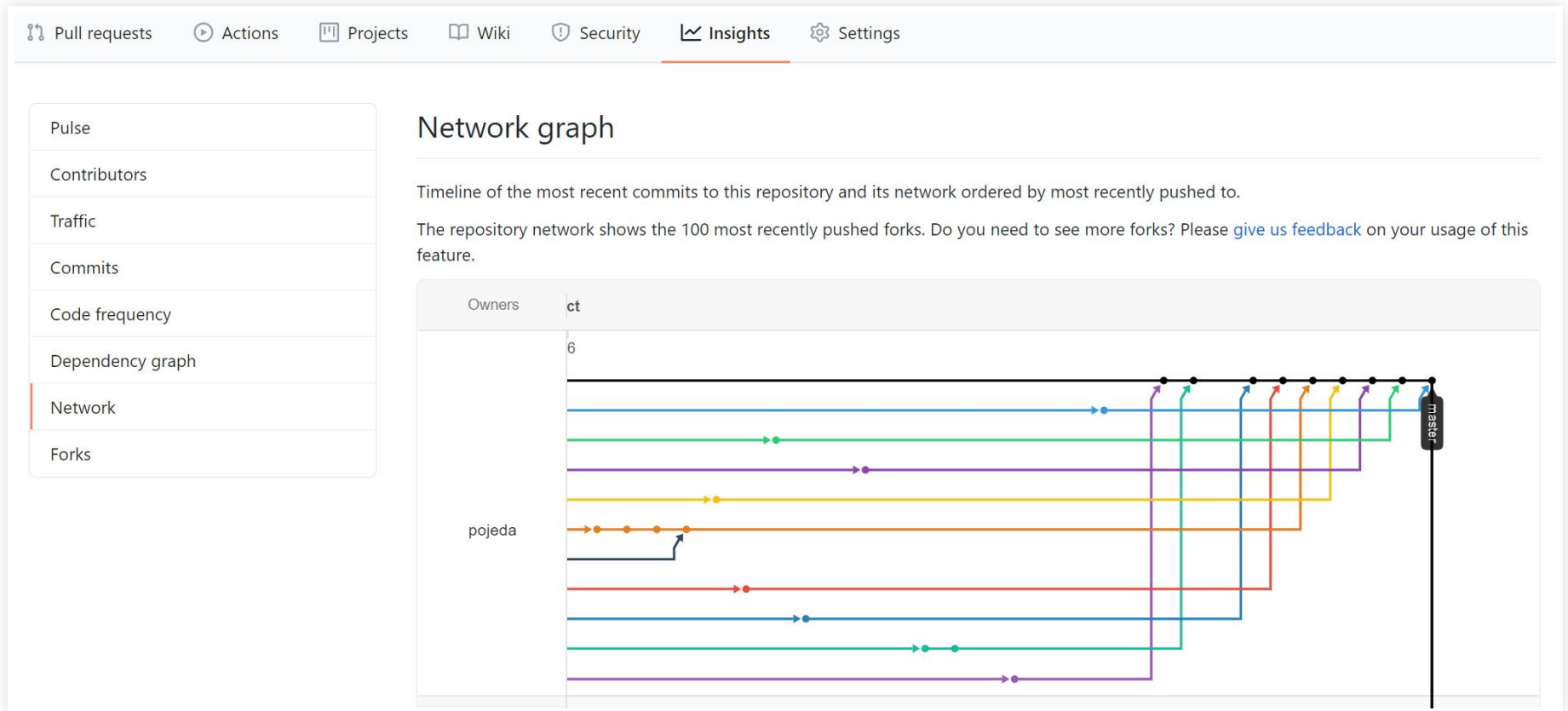
Title

Key

Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

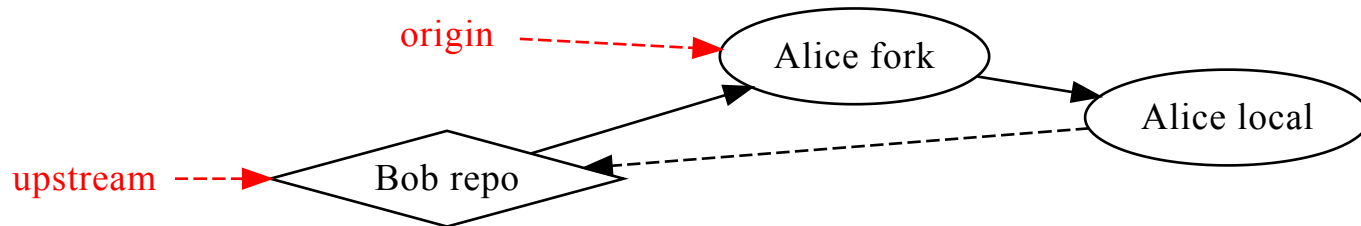
Add SSH key

Network visualization



Pull requests

In the following scenario, a developer, Bob, has its repo on GitHub. Another developer, Alice, finds it useful and forks it. After doing some changes, Alice push them and do a “pull request”



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: bobuser2020/my-first-project base: master ← head repository: aliceuser2020/my-first-project compare: master

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

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

1 commit

1 file changed

0 comments

1 contributor

Commits on Sep 28, 2020

Update README.md

Verified a8902e2

Showing 1 changed file with 3 additions and 1 deletion.

Unified Split

4 README.md

<> [icon] ...

-1 +1,3

Then, Bob receives an email with the pull request information about Alice modifications. On the

GitHub site he sees the request:

bobuser2020 / my-first-project

< Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Update README.md #1

Open aliceuser2020 wants to merge 1 commit into bobuser2020:master from aliceuser2020:master

Conversation 0 Commits 1 Checks 0 Files changed 1



aliceuser2020 commented 11 minutes ago

First-time contributor



change from Alice



Update README.md

Verified

a8902e2

Add more commits by pushing to the **master** branch on **aliceuser2020/my-first-project**.



Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



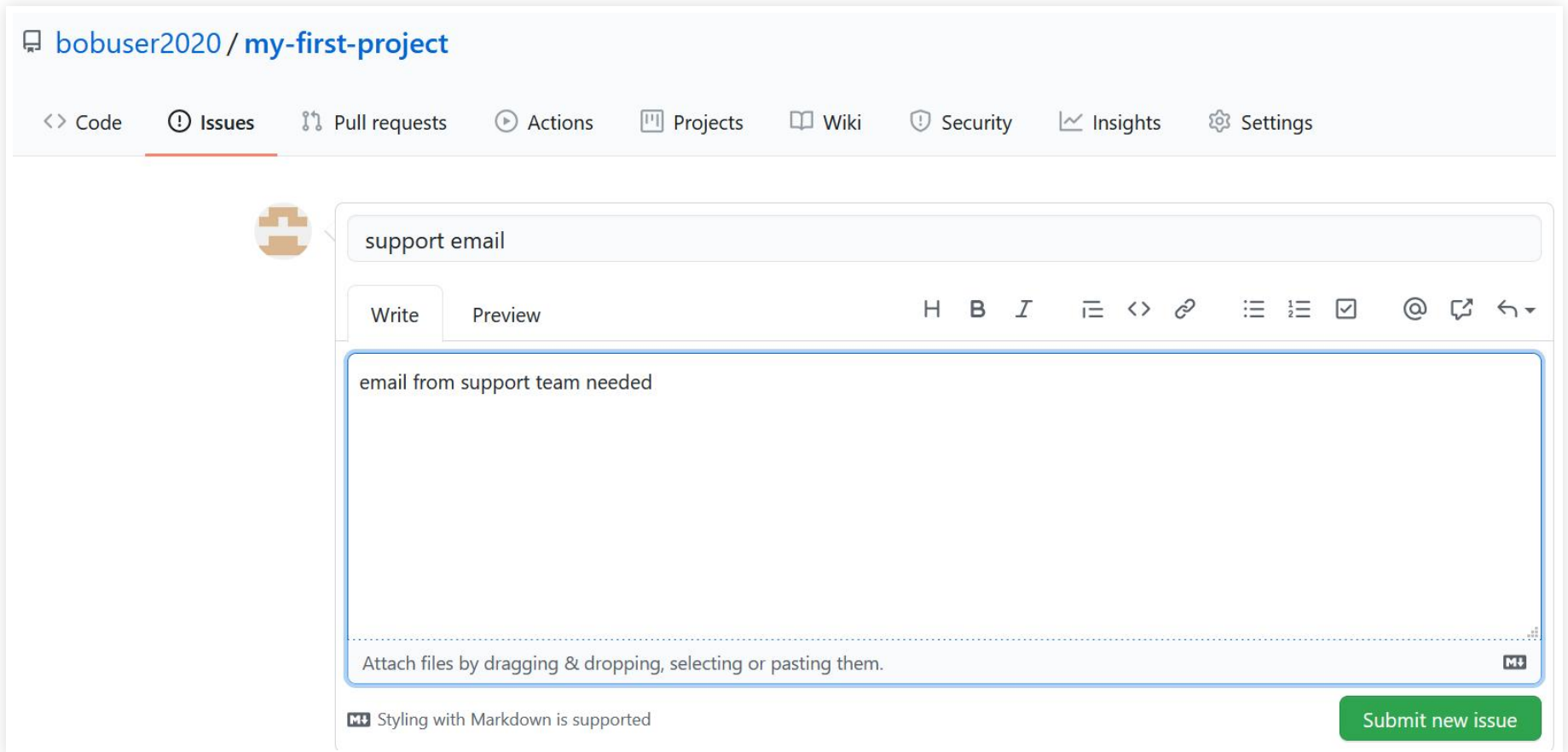
You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Because Bob find the changes from Alice useful and there are no conflicts he can merge them straight away,

The screenshot shows a GitHub pull request interface for the repository `bobuser2020 / my-first-project`. The navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The main heading is "Update README.md #1". A purple "Merged" badge is present, followed by the text "bobuser2020 merged 1 commit into bobuser2020:master from aliceuser2020:master" and a clock icon indicating "5 minutes ago". Below this, a summary bar shows "Conversation 0", "Commits 1", "Checks 0", and "Files changed 1". A comment from `aliceuser2020` (commented 17 minutes ago) says "change from Alice". Below the comment is a commit visualization showing a single commit "Update README.md" with a green "Verified" badge and hash `a8902e2`. At the bottom, a merge summary shows "bobuser2020 merged commit ba421a0 into bobuser2020:master" with a clock icon indicating "5 minutes ago" and a "Revert" button.

Issues








If you find some issues in the files/code you can open an “Issue” on GitHub




bobuser2020 / my-first-project


<> Code **! Issues** Pull requests Actions Projects Wiki Security Insights Settings

support email

Write Preview H B I  <>     @  

email from support team needed

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

 Styling with Markdown is supported

Submit new issue

support email #2

! Open bobuser2020 opened this issue 3 minutes ago · 0 comments



bobuser2020 commented 3 minutes ago

Owner 😊 ...

email from support team needed



Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↗ ↶

Leave a comment

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



Close issue

Comment

You may also assign people to the issues that are more related to that topic.

In future commits you may refer to this issue by using the issue number, [#2](#) in this case. This will allow you to track the evolution of the issue on GitHub.

Best practices

- Talk with your colleagues.
- Some commands such as **git rebase** change the history. It wouldn't be a good idea to use them on public branches.
- Don't accept pull requests right away.