

* Github is a hosting platform for git repos.

* cloning

Get repo not in our machine from URL
URL can be on github, git lab.

git clone <url>

* It is a git command not github

* <url> can be taken by go to github repo and under code button you can get https url.

* Before the do cloning make sure you are not in a git repo using git status command.

* then run git clone <url> command. then create new repo on local machine. Repo is currently on github.

e.g:- git clone https://github.com/gabrielecirulli/2048.git

* Configure SSH keys :- video / 11-github the Basic / 006 Github setup SSH config

See SSH Documentation of github.

repo

* publish a git repo on github currently having in local

1. Create new empty repo in Github

When naming github repo we can't put spaces. it automatically puts - hence of spaces.

git remote -v :- Can see github repo url.
(if currently existing only).

2. git remote add <name> <url>

name is always origin.
it's convention.

above created github repo's url.

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

```
git remote add origin https://github.com/Rumindu/test.git  
git branch -M main  
git push -u origin main
```

This is a guideline from github after creating a repo

3. Confirming using git remote -v

4. Rename and Delete git remote if you needed.

git remote rename <old> <new>

git remote remove <name>

5. Push the local repo to cloud

git push <name> <branch>

↑
origin

e.g:- git push origin master.

* From this command create new branch called "master" in github and push the local branch.

- * Push another branch to git hub
 $\text{git push origin } \underbrace{\text{bench}}_{\text{branch name}}$
- * Push when having differ name between local and github branches

$\text{git push origin <local branch name> : <remote branch>}$
 ex:- $\text{git push origin master:main}$

- * $\text{git push -u origin } \underbrace{\text{master}}_{\text{branch name}}$

After push branch like that we just need git push to push the code. else we need to type $\text{git push origin master}$ for rest of life. (Push അണുമ്പേക്കുന്നത് currently അണുമ്പേക്കുന്ന branch ചില്ല)

Here local branch keep on memory about remote branch which is relevant.

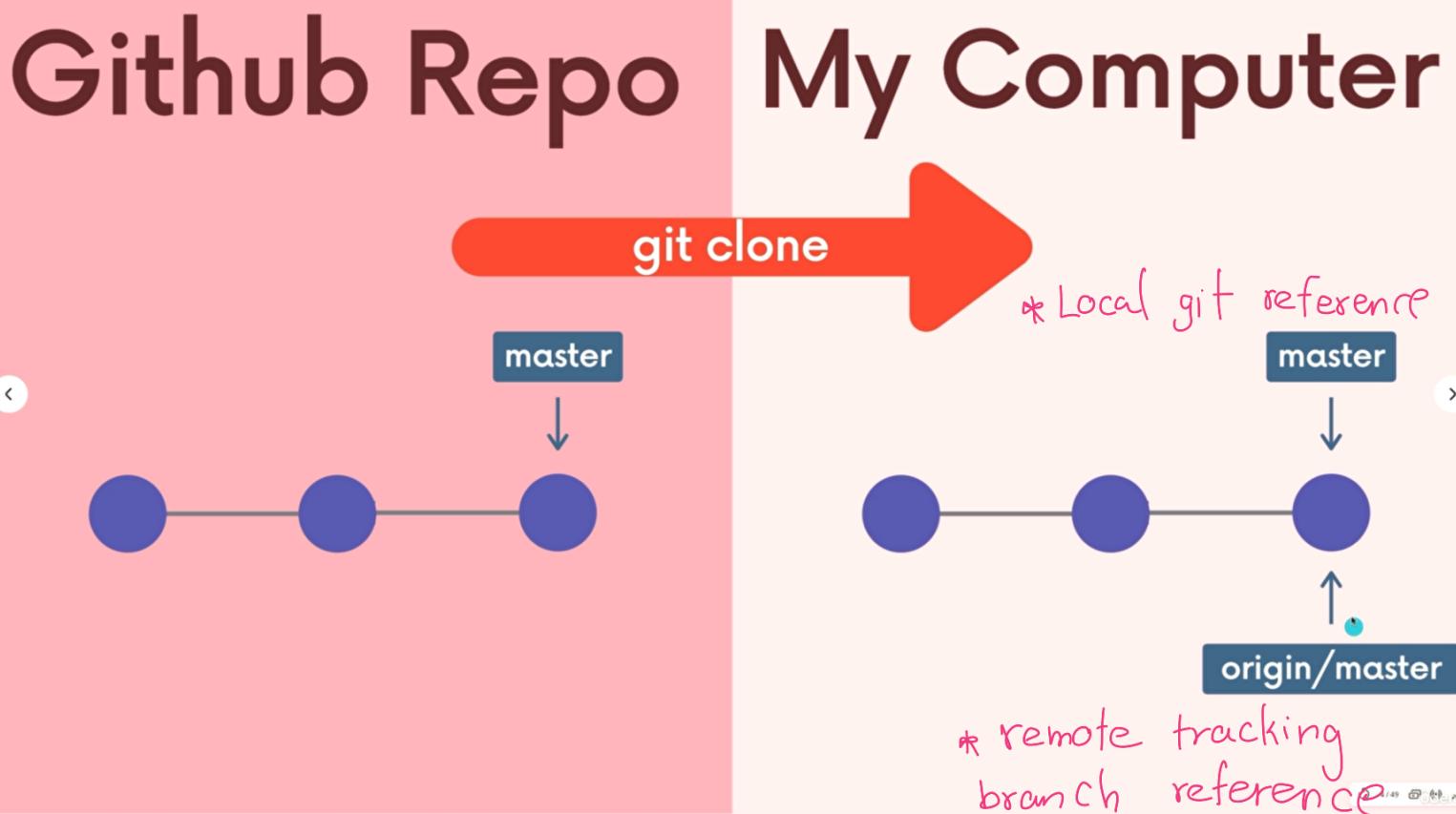
- * When repo is only existing in git hub .
 - i) $\text{git clone <repo URL>}$ * current folder shouldn't contain git hub repo

Here we have pre configer remote "origin"
 we can see it using git remote -v

- * $\text{git branch -M <branch name> : Rename existing branch in git .}$

This command is using because in github default branch is consider as main but in local git default branch name is master. When we create new repo with read me file in github it will create main as default branch. so we can change local branch name using above command.

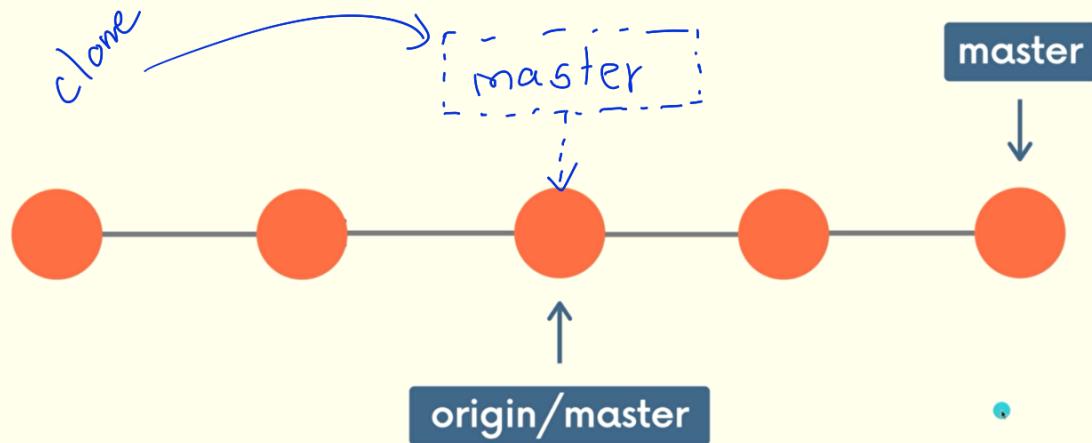
We can change default branch of github through settings.



git branch -r :- Remote tracking branches
(branches in git hub)

My Computer

I make another commit, and the local branch reference moves again.



- * Two commits ahead from git clone

we can see it using git status command.

animals > git status

On branch main

Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

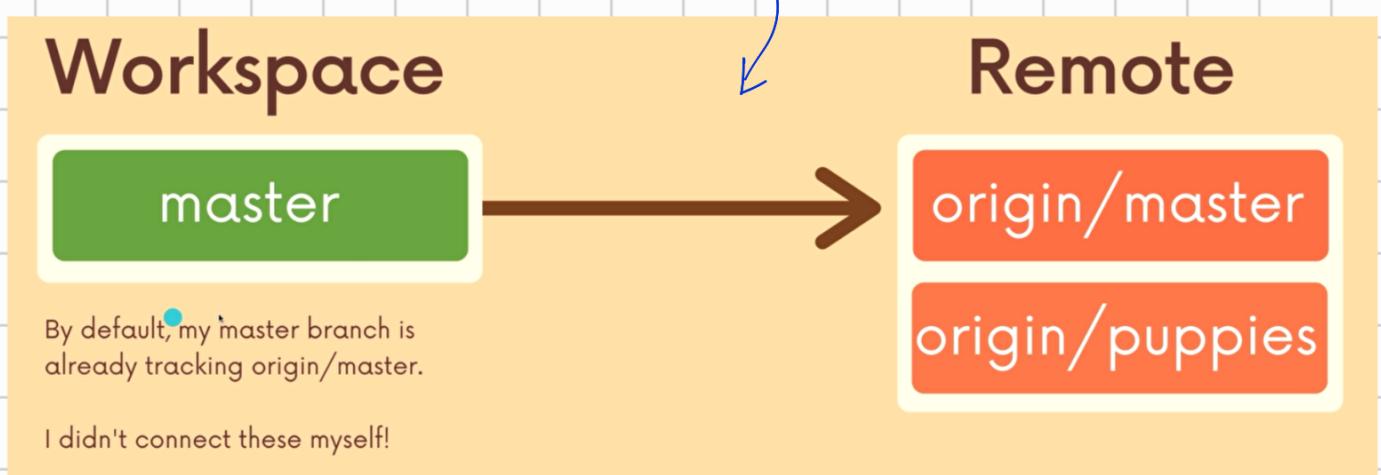
- * git checkout origin/main can checkout "origin/master" which is a reference point of remote -
- * git switch main move to local git reference
- * When we clone github repo we only have master (default) branch in the local machine.
git branch :- can see local branches
git branch -r :- can see remote branches.
- * How to get temporary remote branches content
git checkout origin/<branch name>

- * How to clone other branches with creating new local branch (same name)
- * Want to switch remote branch in the locally

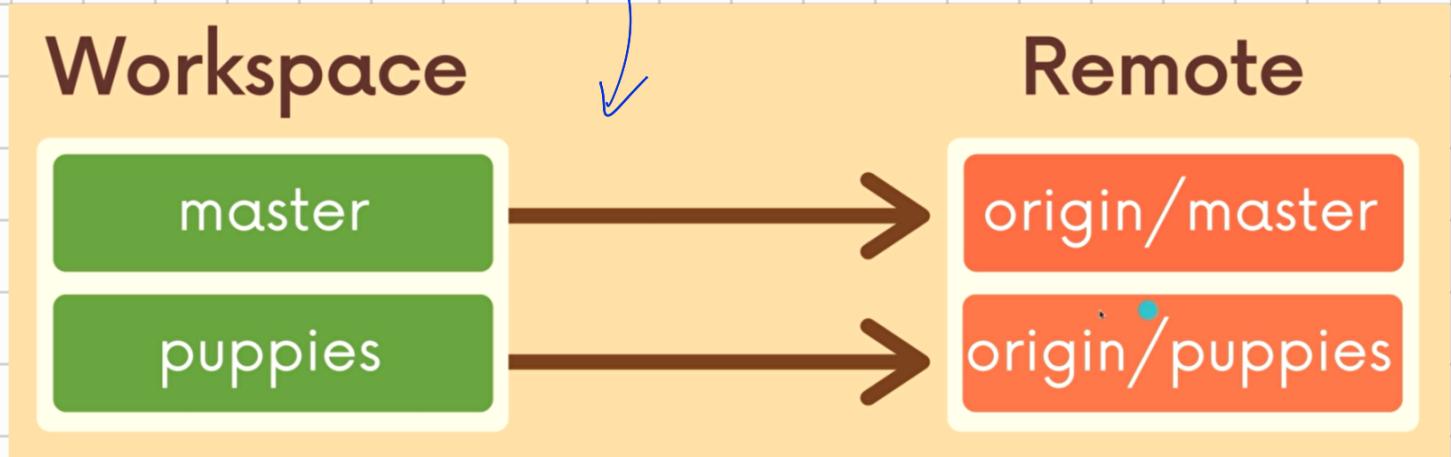

```
git switch <remote-branch-name>
```

ex:- git switch dog .
↳ no need to mention origin

`git clone <repo's URL>`

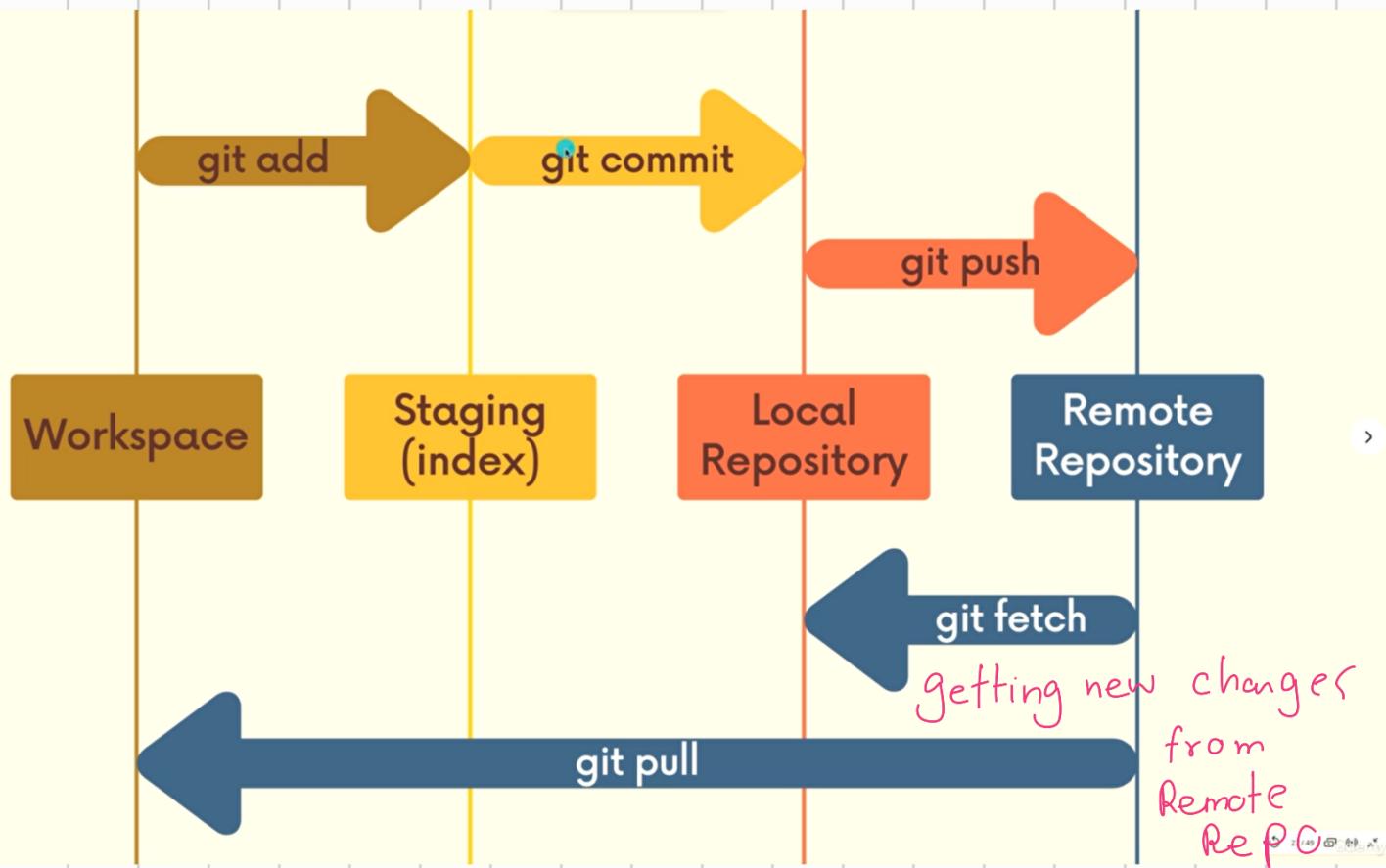


`git switch puppies`



- * Automatically pull the local branch tracking the remote origin & Remote branch.

Fetching and pulling



Fetching

- * Fetching downloads changes from remote Repo But not apply those changes into our working file.
- * Fetching ദിനെ മറ്റ് ദിനകളുടെ പരിപാലനം ചെയ്യുന്നത് without have merge those changes into your local repo .
- * സിൻഗൾ "please go and get the latest information from github , but don't touch my working directory ".

git fetch <remote> :- would fetch all the changes
ex:- git fetch origin from Repo .

git fetch <remote> <branch> :- Would fetch only specific branch
e.g.: git fetch origin main.

Playback paused

Github



Uh oh! The remote repo has changed! A teammate has pushed up changes to the master branch, but my local repo doesn't know!



How do I get those changes???

Local



5 Sec. Forward (by keyframe) / 00:04:00 (70%)



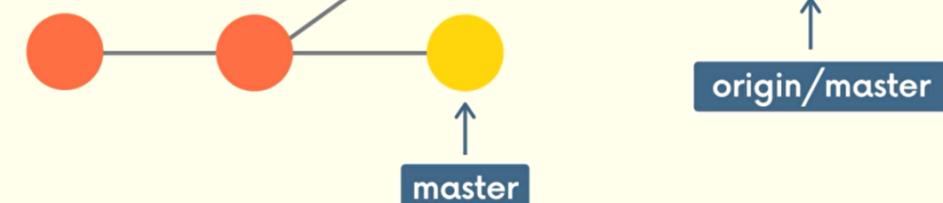
Github



git fetch origin master



Local



32 / 49

33 / 49

- * Changes on my machine. To see those changes
git checkout origin/master.
- * git fetch origin එක ගනාත්තු git status විධ ගෙන්නාතු කෙරීමේ commit ගනාත්තු behind { කියලු .

Pulling

git pull = git fetch + git merge

(update the remote tracking branch with the latest changes from the remote repository) (update my current branch with whatever changes are on the remote tracking branch)

- * git pull <remote> <branch>
ex:- git pull origin master

get all the latest information from the origin's master branch and merge those changes into our current branch

- * Pulls can create merge conflicts.

- * git pull :- "Switch" command එක පවතීම් න්‍යා තිබා ගියාන්න ඇත

Workspace

master

puppies

Remote

origin/master

origin/puppies

When I'm on my local
master branch...

git pull

pulls from origin/master
automatically

Playback paused

Workspace

master

puppies

Remote

origin/master

origin/puppies

When I'm on my local
puppies branch...

git pull

pulls from origin/puppies
automatically

git fetch

- Gets changes from remote branch(es)
- Updates the remote-tracking branches with the new changes
- Does not merge changes onto your current HEAD branch
- Safe to do at anytime

git pull

- Gets changes from remote branch(es)
- Updates the current branch with the new changes, merging them in
- Can result in merge conflicts
- Not recommended if you have uncommitted changes!

- * Change Repo's visibility (Private or public)

Repo settings → general → danger zone

↓
Change repo visibility

Centralized workflow :- Every one works on master
(informal name) branch.

- * When we are going to push the code in to github our local reference and remote reference should be same. It means there shouldn't be any modification after we pull the code. If there any changes on github before the push the code we have to pull the code and solve merge conflict and again push the code.
- * To solve above issue use feature branches.

