



FEBRUARY 23, 2023

DevOps Classroomnotes 23/Feb/2023

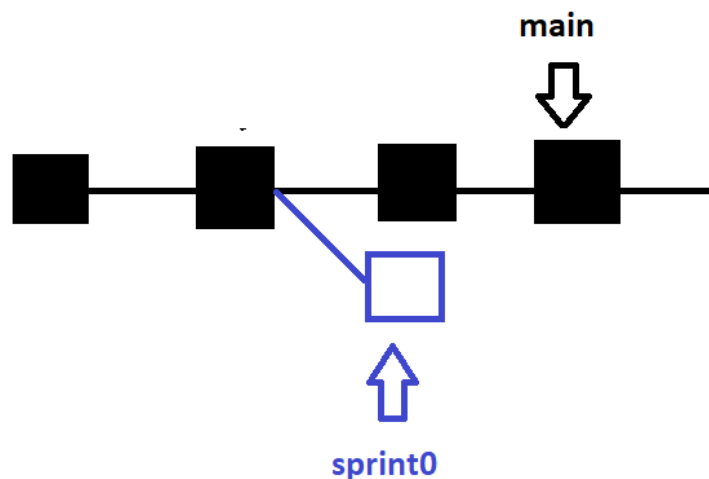
Git Contd

Git Rebase

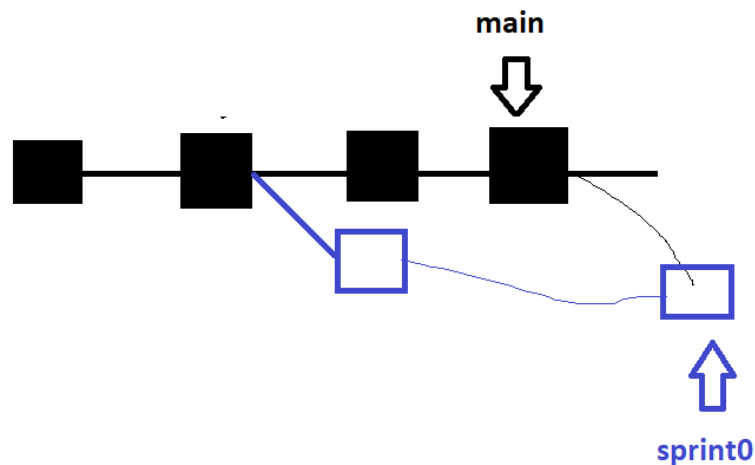
- [Refer Here](#) for docs
- Git rebase is a command used to get changes from one branch onto another by rewriting history
- Consider the following situation where main has two commits after the sprint 0 branch was created

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git log --oneline --graph --all
* 84a2335 (HEAD -> main) added tests
* 99ab5b0 This is change
| * 36c0cb5 (sprint0) Added change
|/
* 2d84516 Added docs
* 649bed7 Added basic folder structure

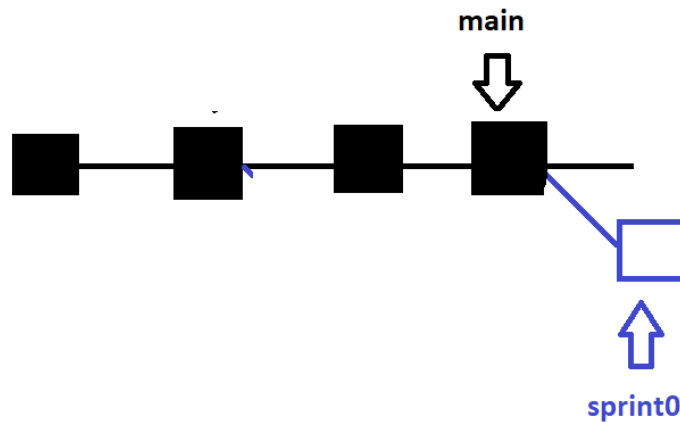
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$
```



- Now we want the changes done in main branch to be part of sprint0 branch
- If we merge the changes from main into sprint 0



- If we rebase



- Rebase is useful for all the child branches but never rebase main/master from child as rebase rewrites history.
- To rebase checkout to sprint0 and rebase from main `git rebase main`

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git checkout sprint0
Switched to branch 'sprint0'

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0)
$ git rebase main
Auto-merging src/main.py
CONFLICT (content): Merge conflict in src/main.py
error: could not apply 36c0cb5... Added change
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 36c0cb5... Added change

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0|REBASE 1/1)
$ git add src/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0|REBASE 1/1)
$ git status
interactive rebase in progress; onto 84a2335
Last command done (1 command done):
  pick 36c0cb5 Added change
No commands remaining.
You are currently rebasing branch 'sprint0' on '84a2335'.
  (all conflicts fixed: run "git rebase --continue")

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0|REBASE 1/1)
$ git rebase --continue

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0|REBASE 1/1)
$ git rebase --continue
[detached HEAD 2aa7cd6] Added change
 1 file changed, 5 insertions(+), 1 deletion(-)
Successfully rebased and updated refs/heads/sprint0.

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0)
$ git log --oneline --graph --all
* 2aa7cd6 (HEAD -> sprint0) Added change
* 84a2335 (main) added tests
* 99ab5b0 This is change
* 2d84516 Added docs
* 649bed7 Added basic folder structure
```

Git cherry pick

- [Refer Here](#) for docs
- If we need to pick specific commits or specific range of commits (sequence) of commits , cherry pick can be used

- Now consider the following

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0)
$ git log --oneline --graph --all
* 43d6b2a (HEAD -> sprint0) added sub
* f3ae24b created unit test docs
* 2aa7cd6 Added change
* 84a2335 (main) added tests
* 99ab5b0 This is change
* 2d84516 Added docs
* 649bed7 Added basic folder structure

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (sprint0)
$
```

- we need the changes from commit id f3ae24b in main branch
- lets cherry pick

```
Switched to branch 'main'

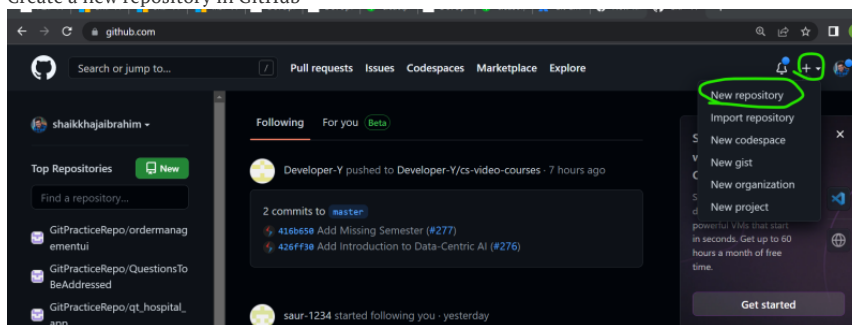
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git cherry-pick f3ae24b
[main 0e04662] created unit test docs
Date: Thu Feb 23 08:32:40 2023 +0530
1 file changed, 2 insertions(+)
create mode 100644 docs/unittest.md

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git log --oneline --graph --all
* 0e04662 (HEAD -> main) created unit test docs
| * 43d6b2a (sprint0) added sub
| * f3ae24b created unit test docs
| * 2aa7cd6 Added change
|/
* 84a2335 added tests
* 99ab5b0 This is change
* 2d84516 Added docs
* 649bed7 Added basic folder structure

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$
```

Remote Repository

- Git Remote Repository is any folder on any server
- To make Git Remote accessible, generally two protocols are widely used
 - http(s)
 - ssh
- There are many popular Git Repositories which are available for free.
- Today lets get started with GitHub
 - Create an account in GitHub [Refer Here](#)
- Create a new repository in GitHub



github.com/new

Create a new repository

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

Repository template
Start your repository with a template repository's contents.

No template -

Owner * shaikkhajaibrahim / **Repository name *** branches_demo ✓

Great repository names are short and memorable. Need inspiration? How about [curly-journey?](#)

Description (optional)
This is for learning purposes

☒ **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.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

📘 You are creating a public repository in your personal account.

Create repository

- Now add this repository as remote to your local repository

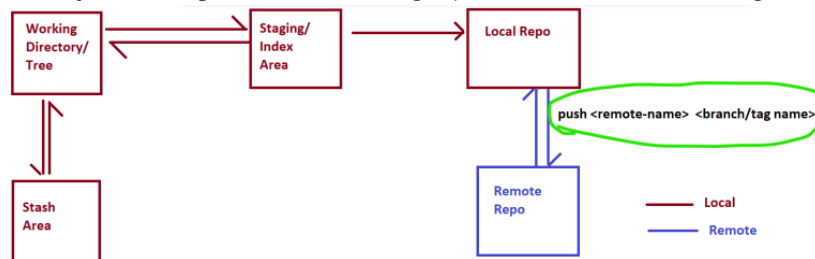
```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git remote add origin https://github.com/shaikkhajaibrahim/branches_demo.git

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git remote --help

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git remote
origin

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$
```

- Now lets push the changes from local to remote git push <remote-name> <branch/tag name>



```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (19/19), 1.53 KiB | 391.00 KiB/s, done.
Total 19 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/shaikkhajaibrahim/branches_demo.git
 * [new branch]      main -> main
```

- Now lets see all the branches in the repo

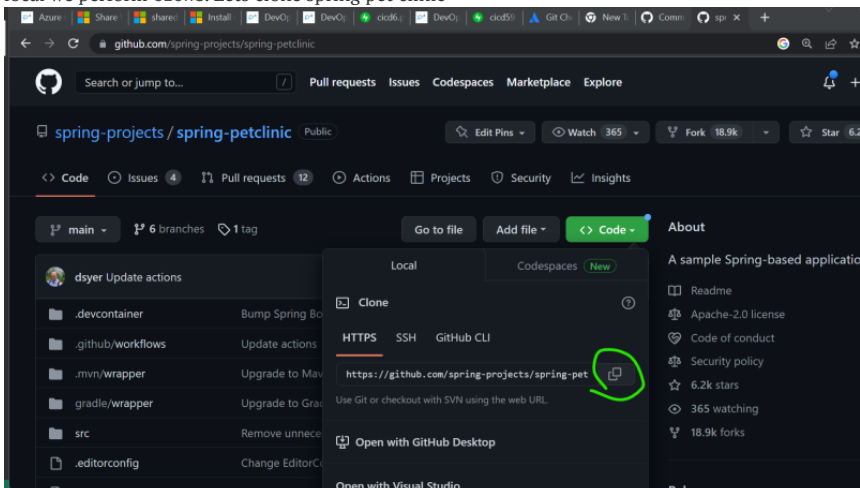
```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git branch --all
* main
  sprint0
  remotes/origin/main
```

- The branches in remote represent remote branches
- Now lets push the sprint 0 branch as well

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git push origin sprint0
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.08 KiB | 554.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'sprint0' on GitHub by visiting:
remote:   https://github.com/shaikkhajaibrahim/branches_demo/pull/new/sprint0
remote:
To https://github.com/shaikkhajaibrahim/branches_demo.git
 * [new branch]      sprint0 -> sprint0

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/branches_demo (main)
$ git branch --all
* main
  sprint0
  remotes/origin/main
  remotes/origin/sprint0
```

- In Organizations we will already have some repositories, Now to get the changes from remote to local we perform clone. Lets clone spring pet clinic



```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice
$ git clone https://github.com/spring-projects/spring-petclinic.git
Cloning into 'spring-petclinic'...
remote: Enumerating objects: 9471, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 9471 (delta 2), reused 13 (delta 1), pack-reused 9453
Receiving objects: 100% (9471/9471), 7.65 MiB | 11.79 MiB/s, done.
Resolving deltas: 100% (3584/3584), done.

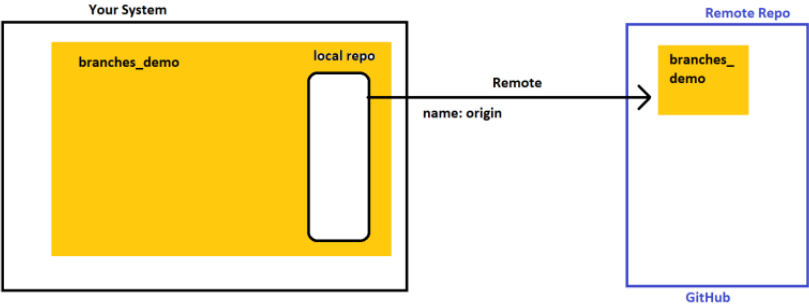
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice
$ ls
backup/  branches_demo/  experiment/  spring-petclinic/  understanding/

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice
$ cd spring-petclinic/

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/spring-petclinic (main)
$ git branch --all
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/efficient-webjars
  remotes/origin/gh-pages
  remotes/origin/hacking/mysql
  remotes/origin/main
  remotes/origin/springboot3
  remotes/origin/wavefront

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/gitpractice/spring-petclinic (main)
$
```

- Summary



Leave a Reply

Enter your comment here...

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)



About continuous learner
devops & cloud enthusiastic learner

[VIEW ALL POSTS](#)

◀ PREVIOUS POST

[Azure Classroomnotes 23/Feb/2023](#)

NEXT POST

[AWS Classroomnotes 23/Feb/2023](#)