Version Controlling

Open and Distance Learning (CODL)
University of Moratuwa

Why Version Controlling ?

Why Version Controlling ?

- History
- Collaboration and syncing with multiple computers
- Feature Branching
- Backup & Restore

Git Version Controlling

- Git is one of the most popular version control systems.
- Git was created by Linus Torvalds in 2005 for development of the Linux kernel.
- Industry adoption is 87.2% in 2018.
- Other version controlling systems
  - Apache Subversion (SVN)
  - Azure DevOps Server
  - Mercurial

Git Hosting Services

- Git ≠ Github
- Hosting services provides home for your Git-based projects on the internet.
- Widely used hosting services
  - Github
  - Bitbucket
  - Gitlab

Installation & Setting up repository

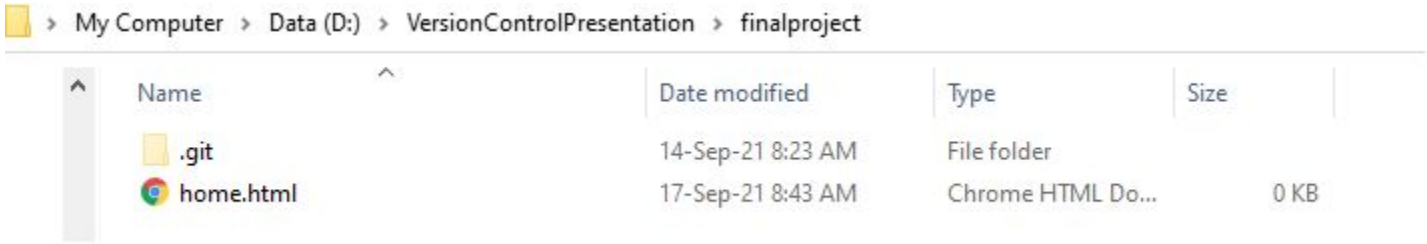
- Install git via https://git-scm.com/downloads
- Configuring Git

```
git config --global user.name "FirstName LastName"
git config --global user.email "your-email@email-provider.com"
```

- **git init** command helps us to initialize a repository to a git repository.

```
D:\VersionControlPresentation>mkdir finalproject

D:\VersionControlPresentation>cd finalproject

D:\VersionControlPresentation\finalproject>git init
Initialized empty Git repository in D:/VersionControlPresentation/finalproject/.git/

D:\VersionControlPresentation\finalproject>
```

Adding files



- **git status -** This command displays the current / working tree repository status.

```
D:\VersionControlPresentation\finalproject>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        home.html

nothing added to commit but untracked files present (use "git add" to track)
```

Staging files

- **git add -** Adds a change in the working directory to the staging area.
- It tells Git that you want to include updates to particular files in the next commit.
- Changes are not actually recorded in repository until you run git commit.
- Git add commands

```
git add <file-name>
git add <file-name> <another-file-name> <yet-another-file-name>
git add .
git add --all
git add -A
```

Staging files

```
D:\VersionControlPresentation\finalproject>git add --all

D:\VersionControlPresentation\finalproject>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   home.html
```

Committing files

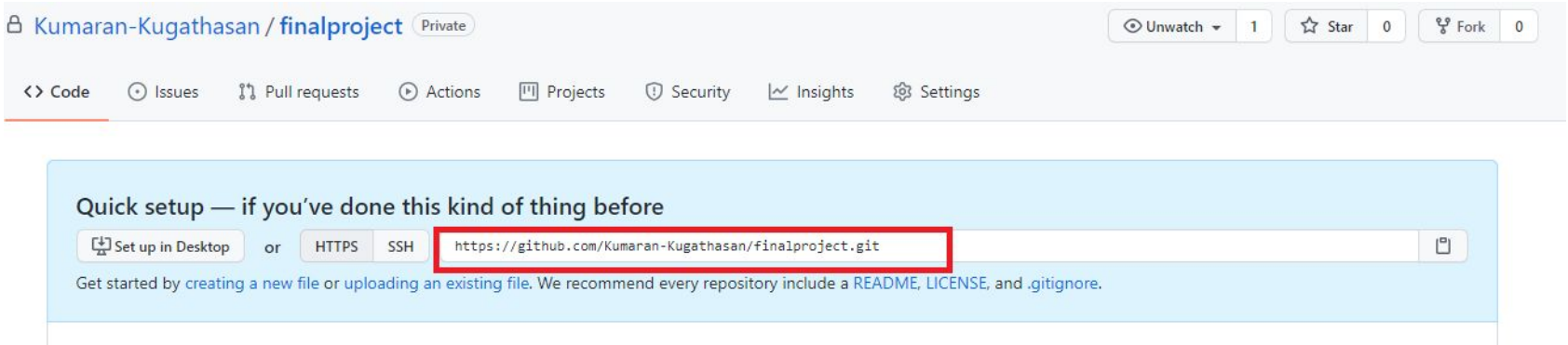
- **git commit** creates a commit, which is like a snapshot of your repository.
- The "commit" command is used to **save your changes to the local repository.**
- Commit with meaningful message.
- Git commit command

```
git commit -m "Add three files"
git reset --soft HEAD^
git commit --amend -m <enter your message>
```

```
D:\VersionControlPresentation\finalproject>git commit -m "homepage added"
[master (root-commit) a812f0f] homepage added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 home.html
```

Remote Repository

- Create a new repository for your project using GitHub, Bitbucket or GitLab
  - **Create a private repository for final year project.**



Binding Local Repo to Remote Repo

- git remote add origin https://github.com/YourUsername/some-small-app.git

```
D:\VersionControlPresentation\finalproject>git remote add origin https://github.com/Kumaran-Kugathasan/finalproject.git
```

  - We tell Git to "add" a repository. The "origin" option is the default name for the server on which your remote repository is located

  - You can't access your code online yet. You just established the link.

- **git push -u origin master**

```
D:\VersionControlPresentation\finalproject>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 54.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Kumaran-Kugathasan/finalproject.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Summary: Local change -> Remote Update

```
D:\VersionControlPresentation\finalproject>git add --all

D:\VersionControlPresentation\finalproject>git commit -m "Added how are you"
[master fb17733] Added how are you
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\VersionControlPresentation\finalproject>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 273 bytes | 273.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Kumaran-Kugathasan/finalproject.git
   9a5693b..fb17733  master -> master
```

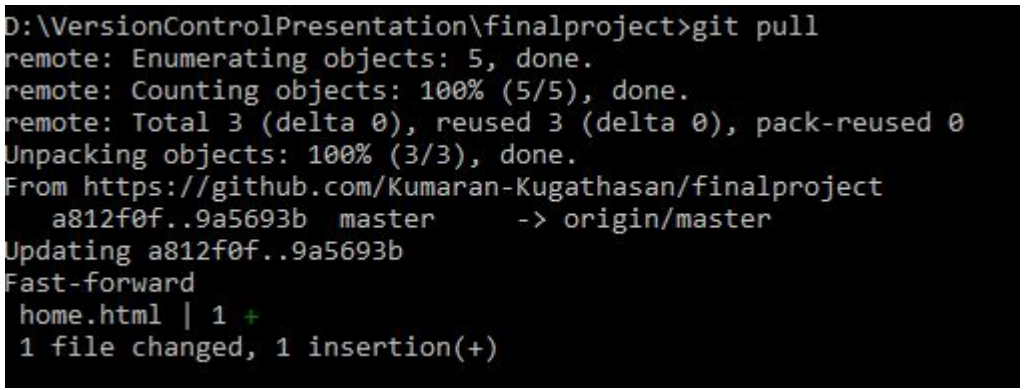
Git Clone

- Clone command is used to obtain copy of the target repository if the project has already been set up in a central repository.

```
D:\VersionControlPresentation\final>git clone https://github.com/Kumaran-Kugathasan/finalproject.git
Cloning into 'finalproject'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Git Pull

- Git pull command is used to pull the contents of a remote repository and update the corresponding local repository.



Git Branching

- Branches, you can actively work on different versions of you projects simultaneously.

- If you have a stable, working application, you don't want to break it when developing a new feature.

```
git branch
git branch <branch-name>
git checkout <branch-name>
git merge <branch-name>
git checkout -b <branch-name>
```

Git Branching

```
D:\VersionControlPresentation\finalproject>git branch
* master

D:\VersionControlPresentation\finalproject>git branch css

D:\VersionControlPresentation\finalproject>git branch
  css
* master

D:\VersionControlPresentation\finalproject>git checkout css
Switched to branch 'css'

D:\VersionControlPresentation\finalproject>git branch
* css
  master
```

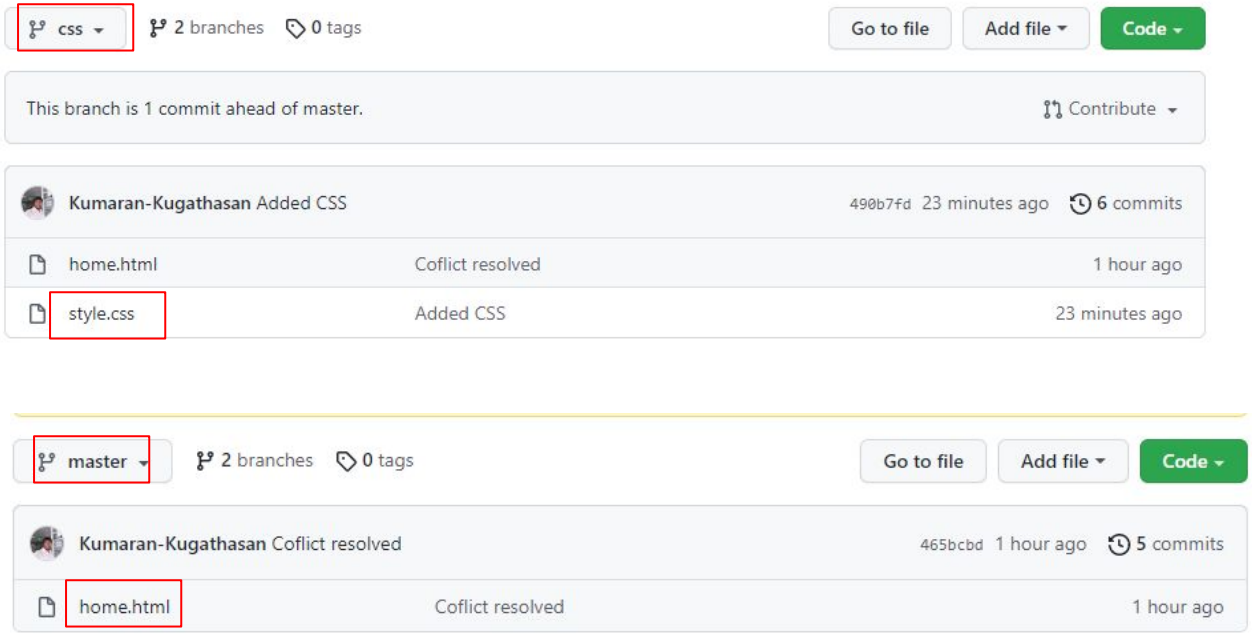
Git Branching

```
D:\VersionControlPresentation\finalproject>git status
On branch css
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        style.css

nothing added to commit but untracked files present (use "git add" to track)

D:\VersionControlPresentation\finalproject>git add --all

D:\VersionControlPresentation\finalproject>git commit -m "Added CSS"
[css 490b7fd] Added CSS
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 style.css
```

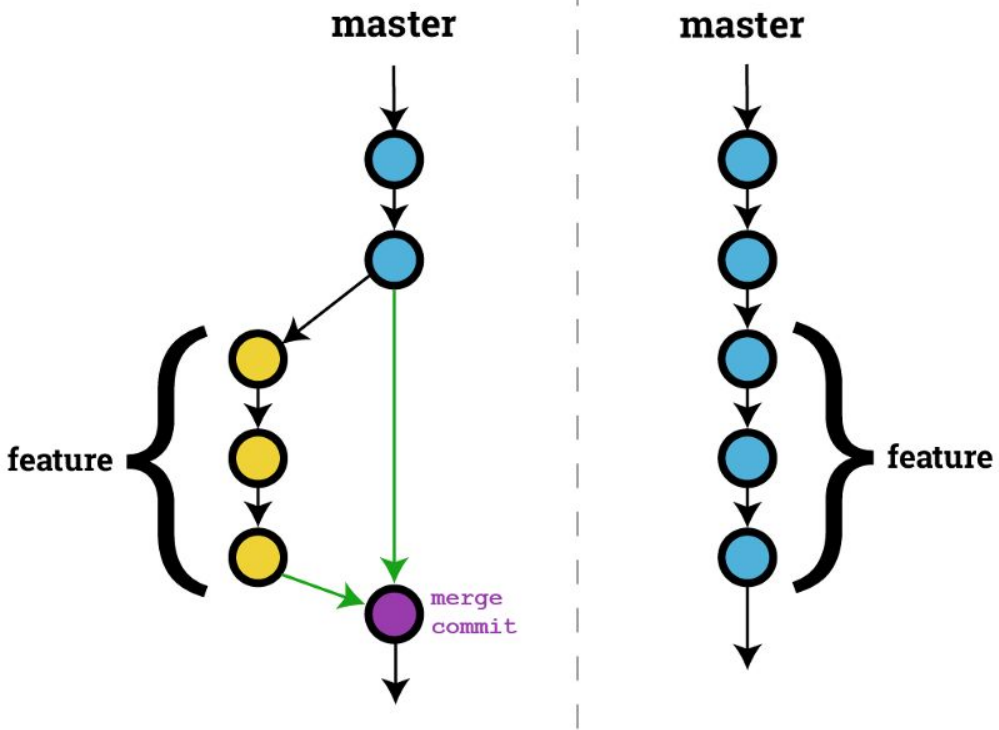
Git Branching

```
D:\VersionControlPresentation\finalproject>git push
fatal: The current branch css has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin css


D:\VersionControlPresentation\finalproject>git push origin css
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 71.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'css' on GitHub by visiting:
remote:      https://github.com/Kumaran-Kugathasan/finalproject/pull/new/css
remote:
To https://github.com/Kumaran-Kugathasan/finalproject.git
 * [new branch]      css -> css
```

Git Branching

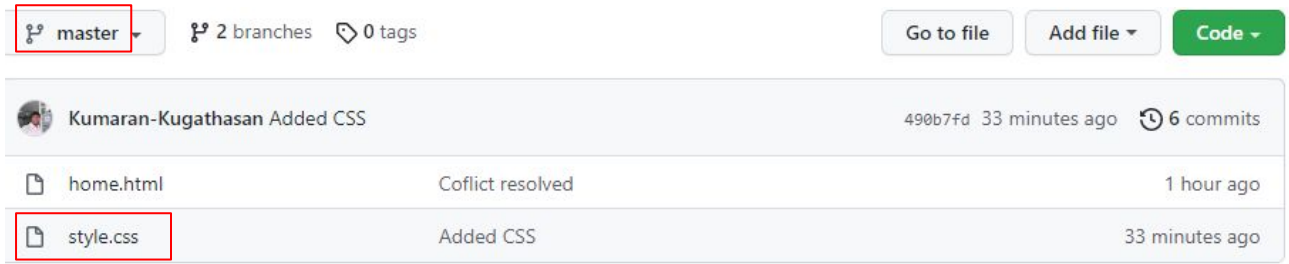


Git Branch Merging



Git Branch Merging

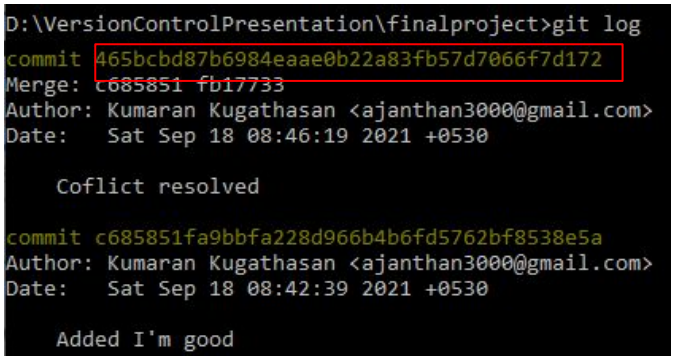
```
D:\VersionControlPresentation\finalproject>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

D:\VersionControlPresentation\finalproject>git merge css
Updating 465bcbd..490b7fd
Fast-forward
 style.css | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 style.css

D:\VersionControlPresentation\finalproject>git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/Kumaran-Kugathasan/finalproject.git
   465bcbd..490b7fd  master -> master
```



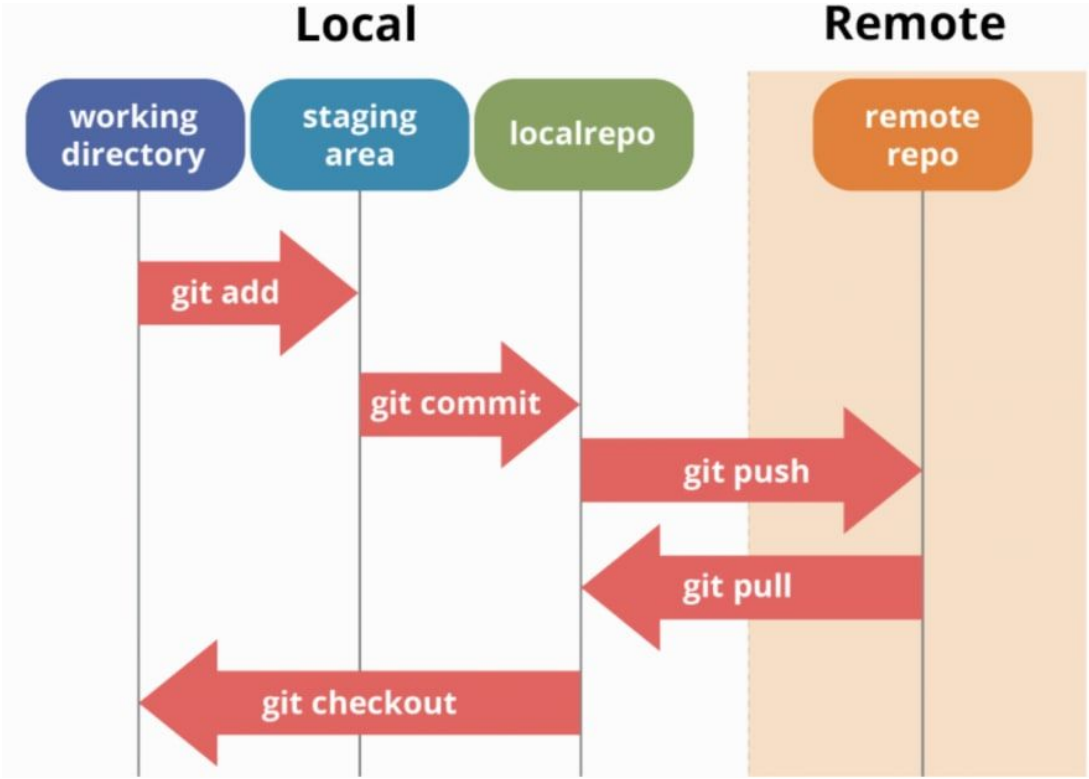
Undoing/Reverting code changes

- Undoing the changes in the file not yet committed.
  - **git checkout -- <file name> (Particular file)**
  - **git checkout -- . (All files)**

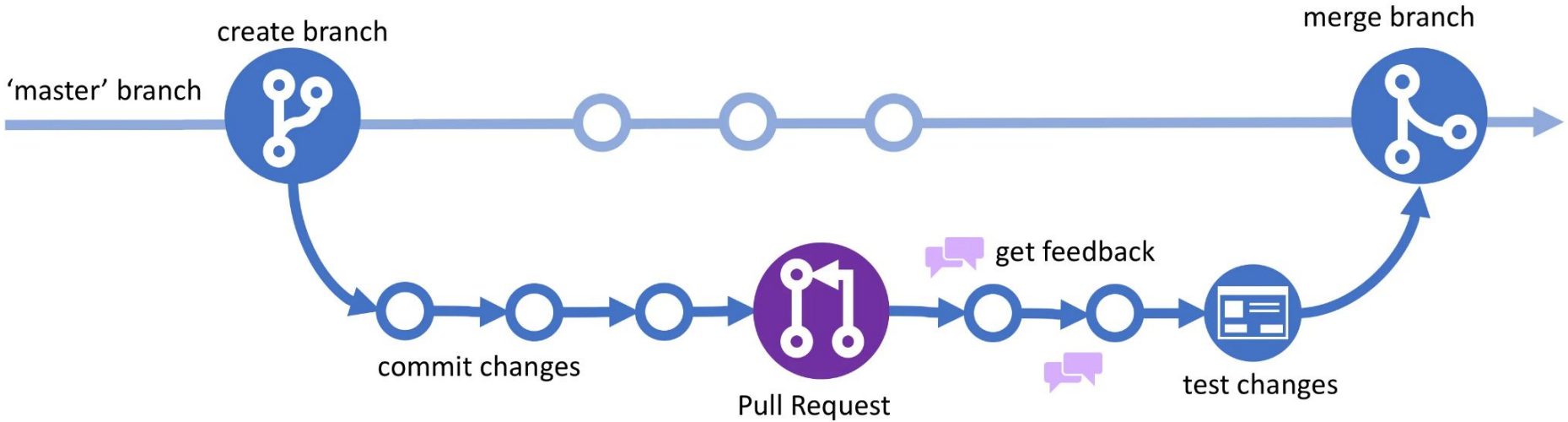
- Reverting a committed changes
  - **git log (List of commits with id)**
  - **Git revert <commit id>**



Summary


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout

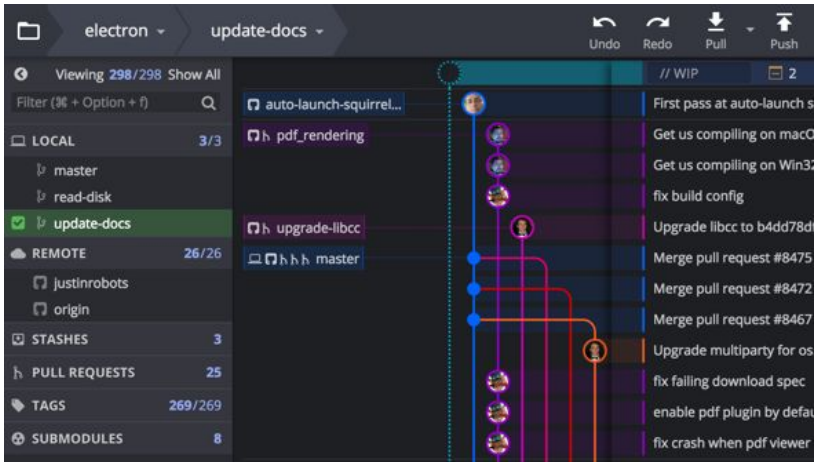
Typical Git Workflow

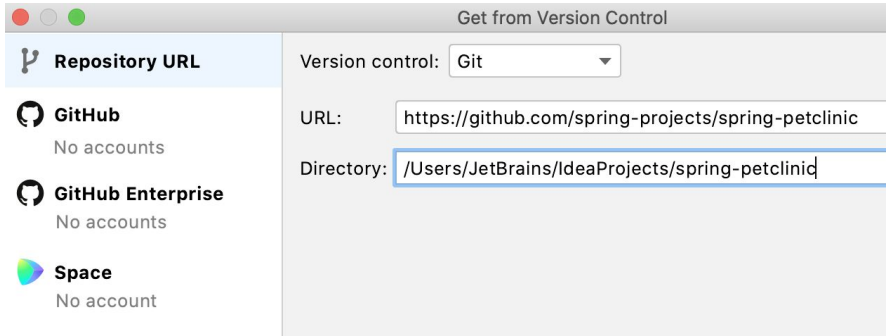


http://buildazure.com

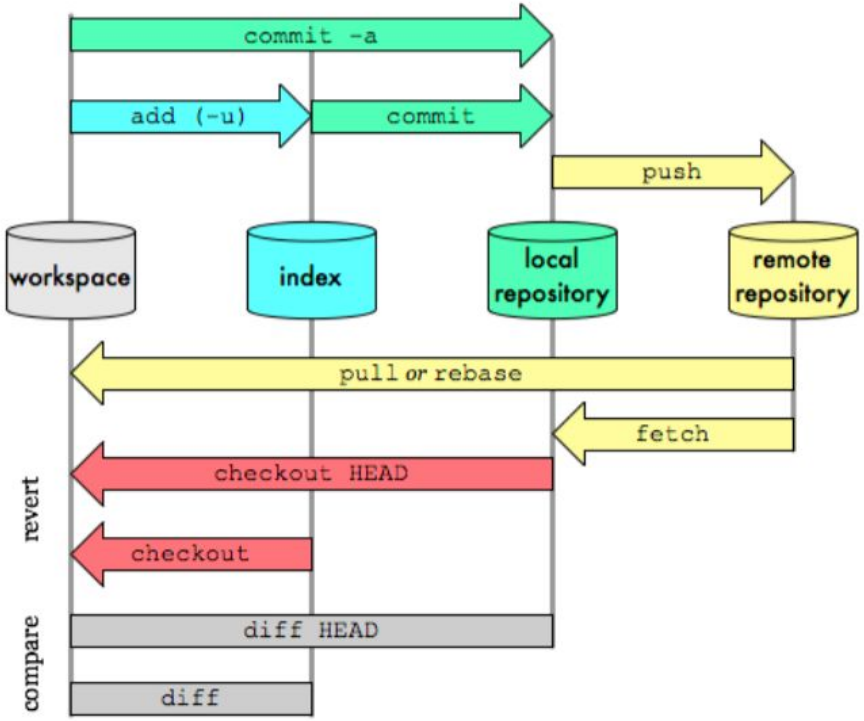
For Your Prototype Evaluation

- Create a remote git repo and push your project to remote repo.
- You can also use Git GUI or Git Support provided by the IDE instead of command prompt.





THANK YOU


commit -a
add (-u)
commit
push
workspace
index
local repository
remote repository
pull or rebase
fetch
checkout HEAD
checkout
diff HEAD
diff
revert
compare