Introduction to Git

A guide by Rafsal Rahim and Sashwat K

Contents

1. Version Control
2. History of version control
3. Git and Various Git hosting services
4. Git installation

5. Creating, cloning, committing and pushing and pulling to Git repository
6. Setting contributors

7. Git log, Status
8. Working with changes.
9. Branches in Git
10. Working with Git branches

11. Working as a team (Contributions)
12. Forking repositories and pull requests
13. Git tags and releases.

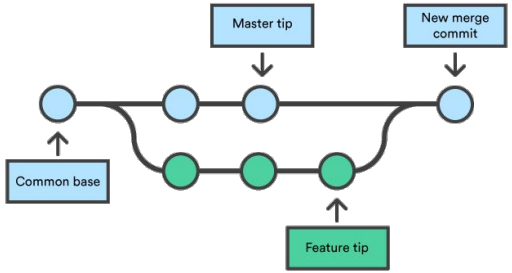
Version Control

A version control a.k.a source control a.k.a revision control is an application that helps to maintain multiple versions of code or files.

Version control works on the principle of graph data structure.

Version control types - local, centralised and distributed.

Popular VCs are git, Apache Subversion and Mercurial.



Advantages

The main advantages of **version control** are:-

- ➔ **Rollback**
  If latest work crashed, go back to previous version.
- ➔ **Compare and work**
  Developer can compare multiple versions to find bugs.
- ➔ **Teamwork & Multiple workflow**
  Supports multiple developers and multiple features simultaneously.
- ➔ **Backup of code**

Important terms

1. Repository: a place to store the code and other files.
2. Truck: Trunk is the master branch of the git repository.
3. Tag: descriptive name given to a specific version of project.
4. Branch: generates multiple development lines.
5. Pull: downloading recent changes.
6. Push: uploading the changes.
7. Committing: records and stores changes.

History of Version Control

The first Version Control System was created in 1972 at Bell Labs where they developed UNIX. The first one was called SCCS (Source Code Control System).

During the development of Linux Kernel, Linus Torvalds introduced Git in 2005 for efficient teamwork.

Fact

Git is a FOSS software developed using C, Python, TCL,Perl and Shell.

Generation	Networking	Operations	Concurrency	Examples
First	None	One file at a time	Locks	RCS, SCCS
Second	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server
Third	Distributed	Changesets	Commit before merge	Bazaar, Git, Mercurial

What is git?

Git is distributed version control application that comes under General Public License 2 (GPL 2). As git comes under distributed version control, the local repository of projects developed using git will have complete history of changes irrespective of internet access or access to the server.

Git Hosting Services

Git hosting services are services that provide online storage of Git repositories.

Famous Git hosting services are:-

1. GitHub [www.github.com]
2. GitLab [www.gitlab.com]
3. Bitbucket [www.bitbucket.org]

Fact

Most used git hosting service is GitHub, followed by GitLab. There are few features in GitLab that GitHub doesn't provide, like create local GitLab server.

Installing Git

Installing Git client on Linux

apt-get install git

dnf install git

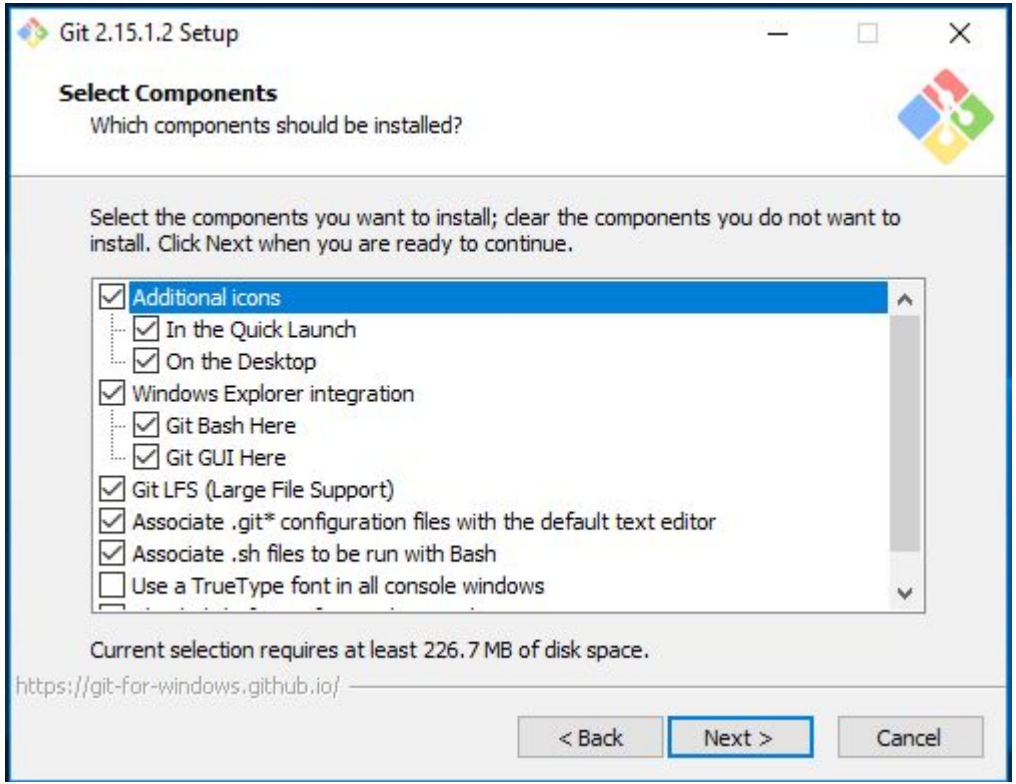
zypper install git

Installing Git client on Windows

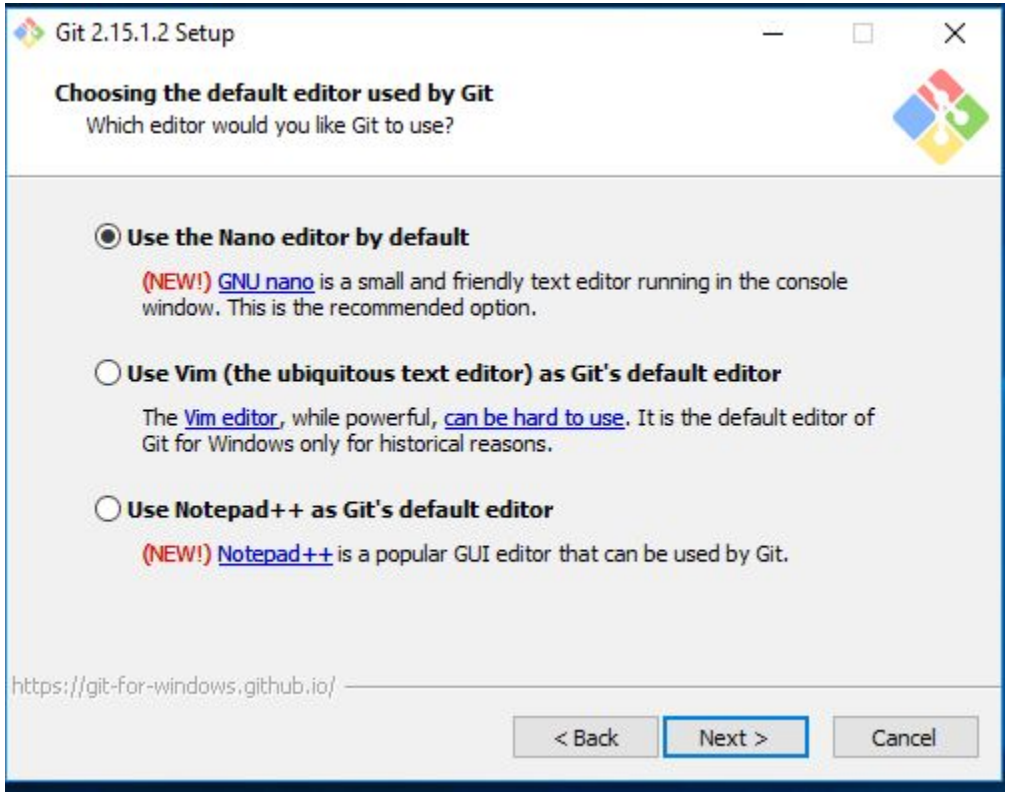
https://git-scm.com/downloads

Installing Git client on Windows

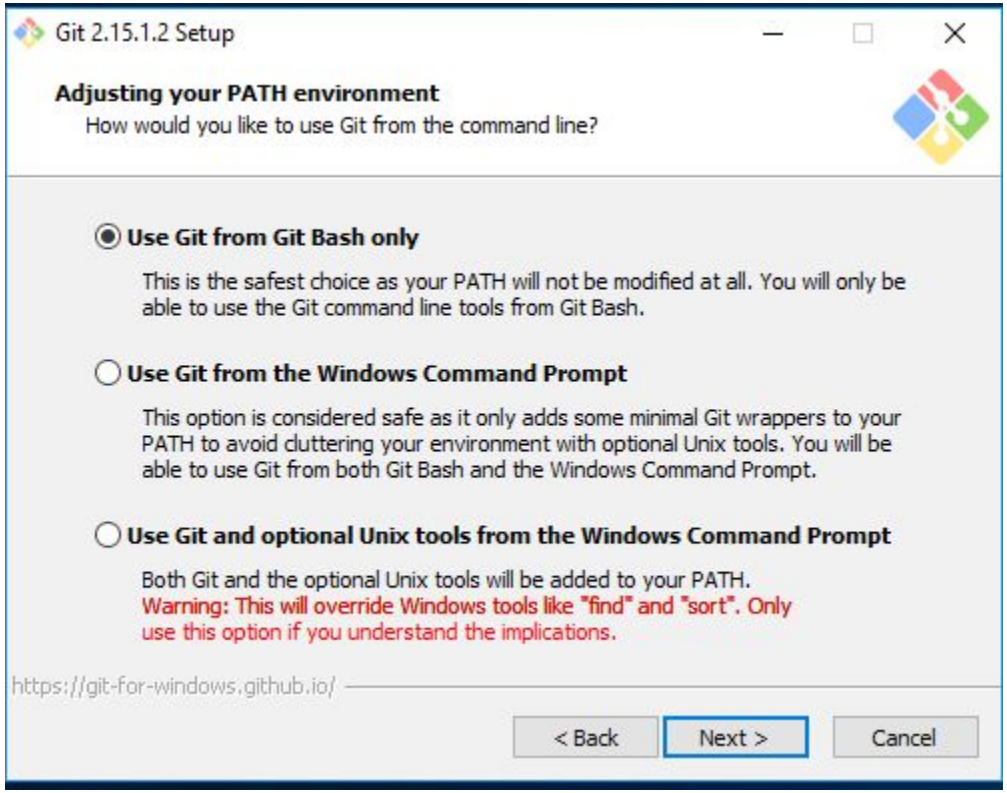
Installing Git client on Windows



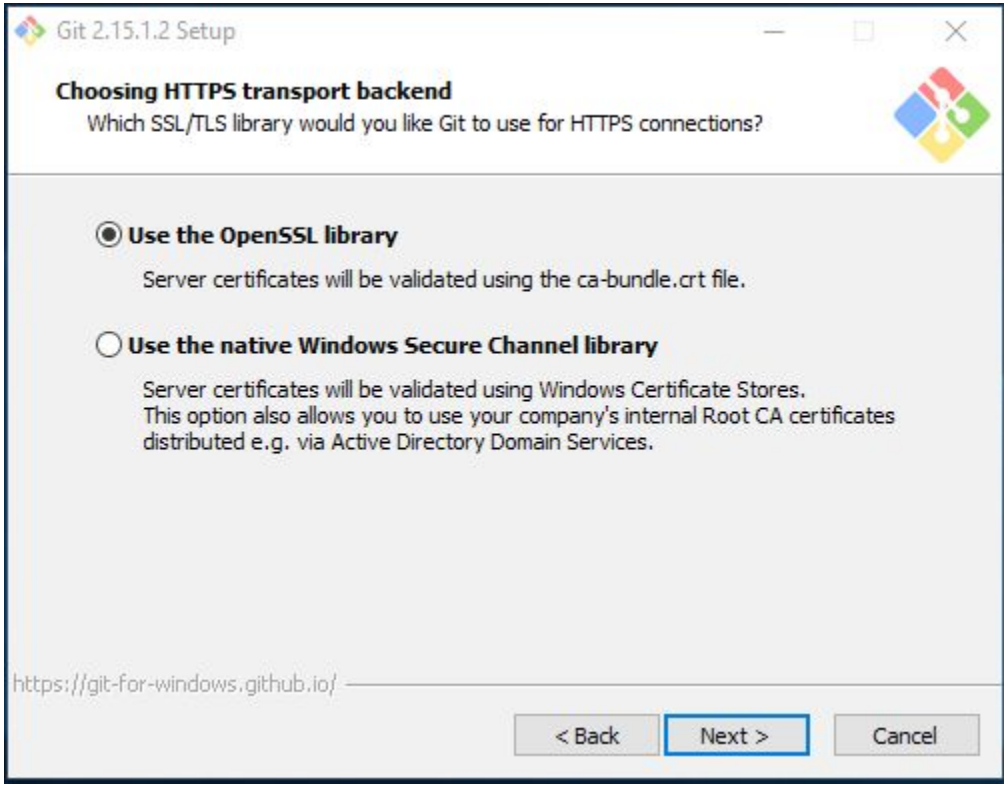
Installing Git client on Windows



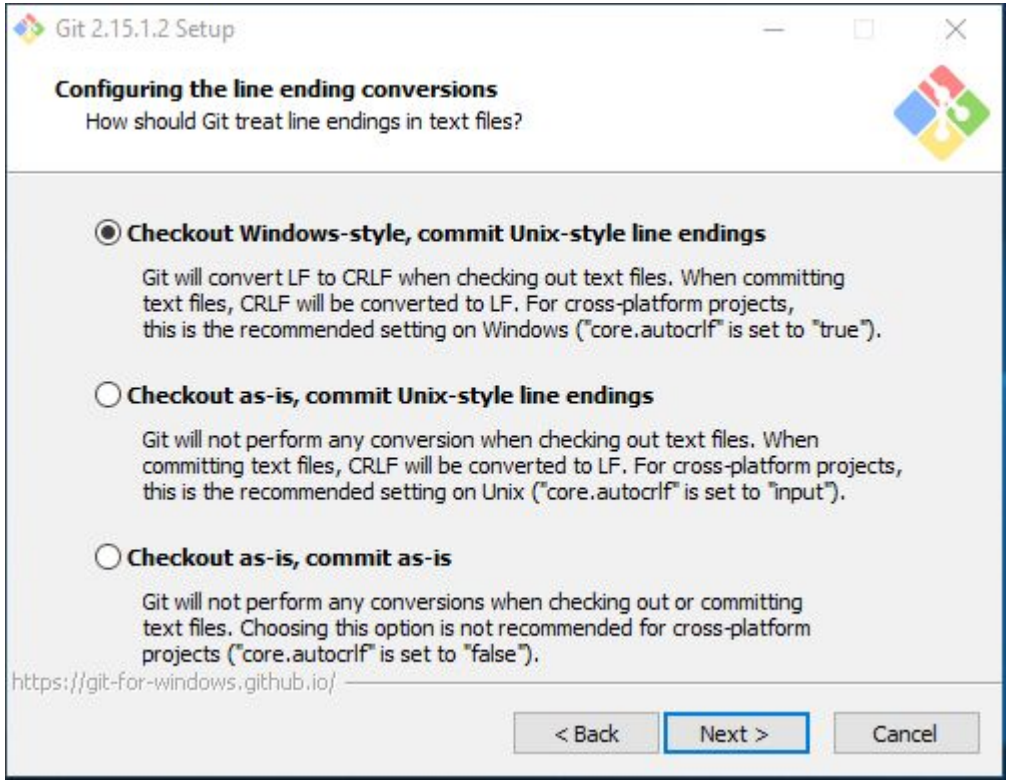
Installing Git client on Windows



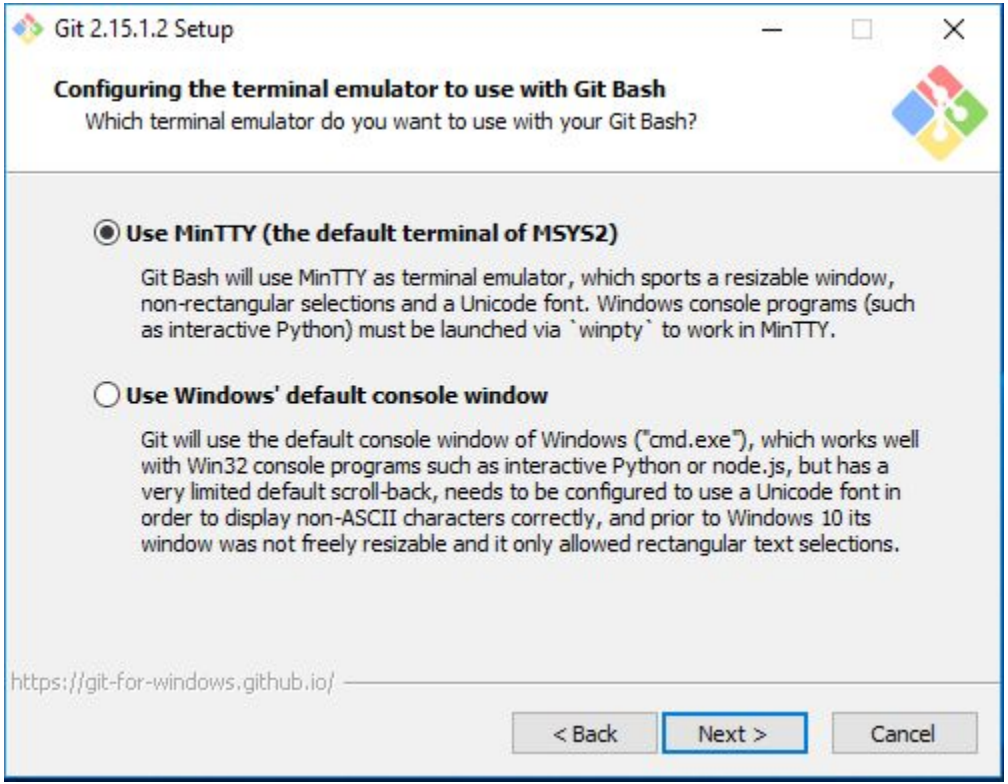
Installing Git client on Windows



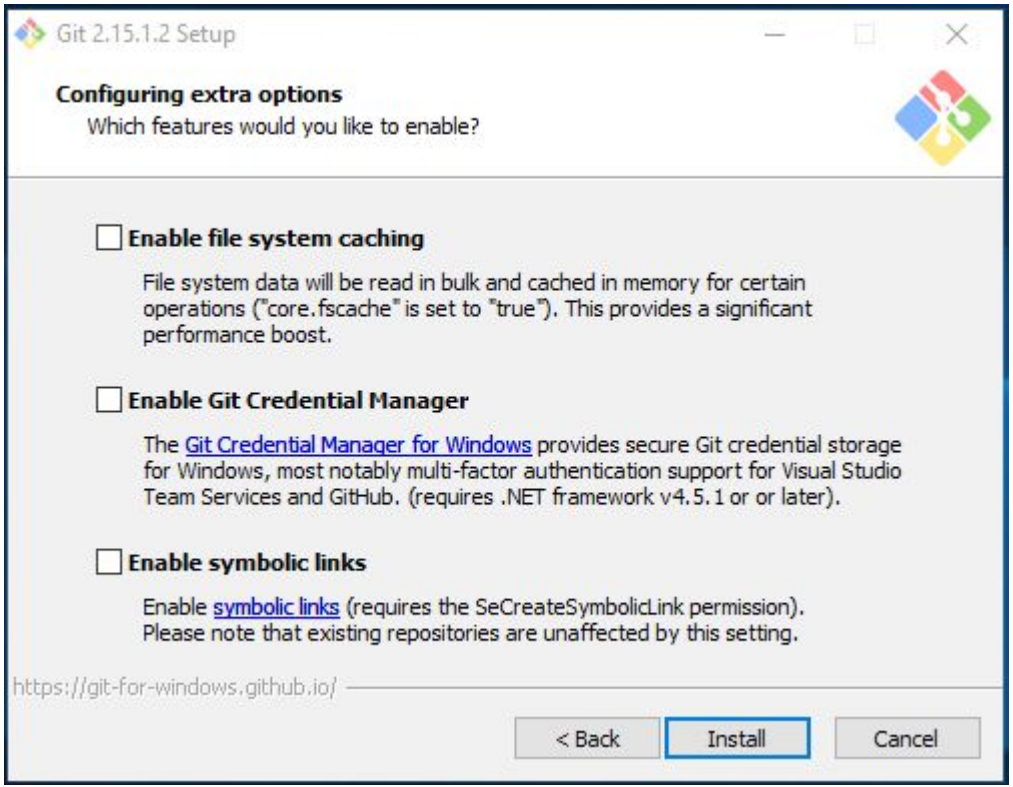
Installing Git client on Windows



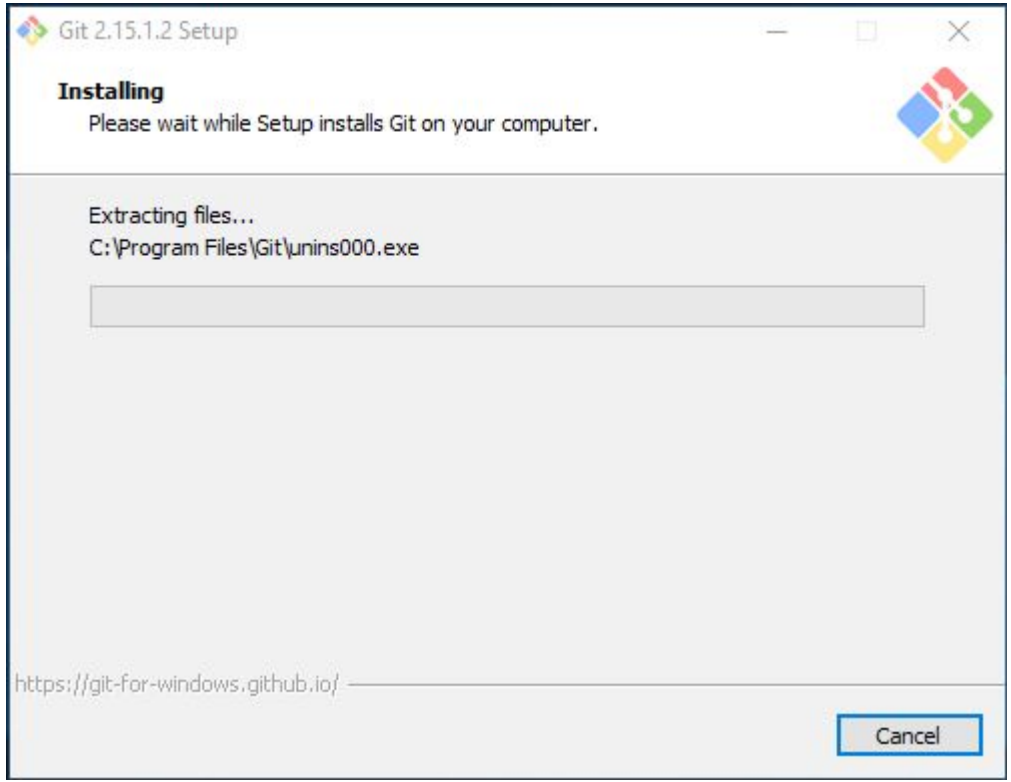
Installing Git client on Windows



Installing Git client on Windows



Installing Git client on Windows



Git Basics

Cloning a repo

git clone https://github.com/user/repo_name.git

Commiting files

“A way to record the changes done to file”

Selecting files for commiting

```
git add filename_1 filename_2 etc.
```

Selecting all files for commiting

```
git add *
```

Commiting changes

```
git commit -m “commit message”
```

Amending commit message

```
git commit --amend
```

Setting contributor info

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

Setting contributor info for current repo

```
git config user.name "John Doe"
git config user.email johndoe@example.com
```

Pushing changes to remote repo

```
git push
```

Keeping the repo updated

```
git pull
```

Listing all the changes

```
git log
```

Get the status of the repo

```
git status
```

See changes in a commit

```
git show commit_id
```

Undoing uncommitted files

Git checkout file_name

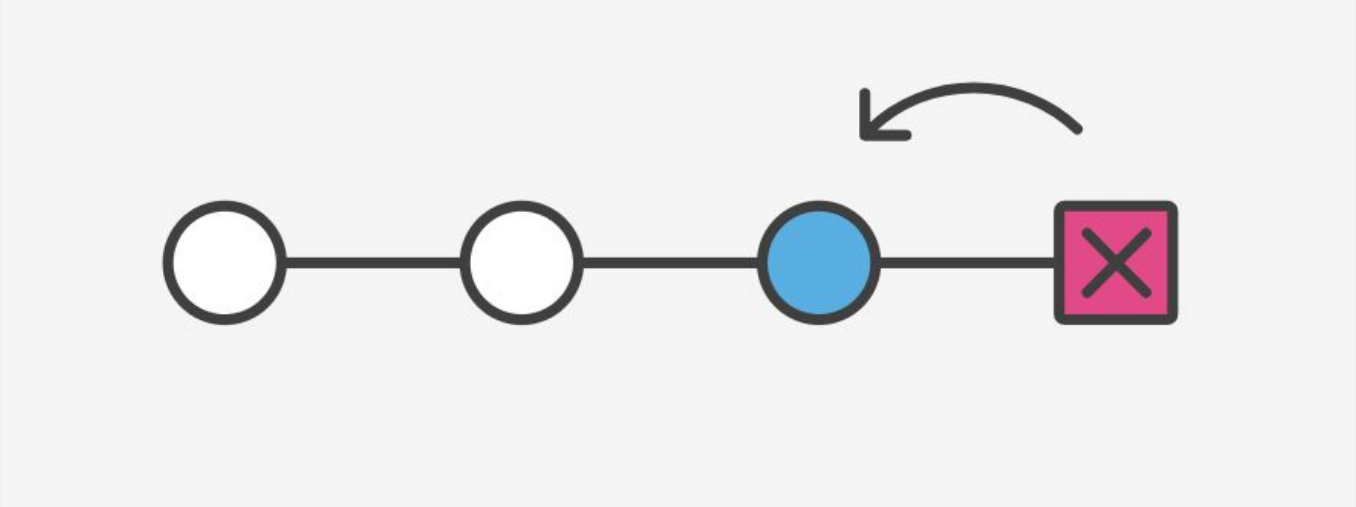
Reverting last commit

```
git revert HEAD
```

Reverting to specific commit

```
git revert commit_id
```

Reverting last commit



Remotes in git

“Remote is a URL which points to another copy of the repo”

Origin

“Origin is the repo from which the local copy originated”

Creating a local repo

```
git init
```

Adding a remote tracking repo

```
git remote add origin https://github.com/user/repo.git
```

Pushing to newly added remote

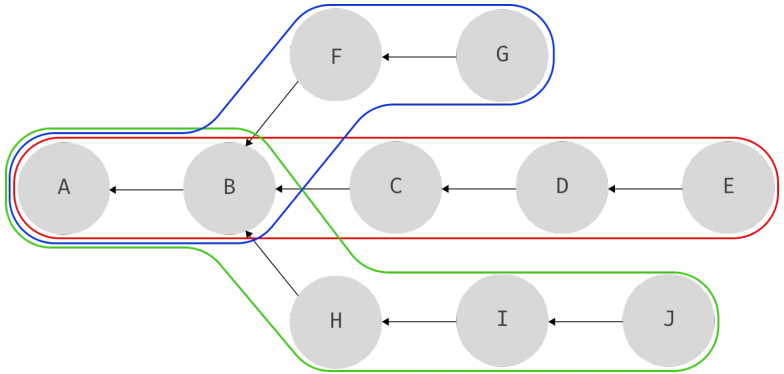
```
git push -u origin master
```

Git Branches

Git Branches

Branches in Git allows creating multiple lines of development,

Eg: If you are working on a product and you can use branches to separate the development of the next version and the current version bug fixes.



Git branches

Create a new branch :-

$git branch <new_branch_name>

If you want to branch after a specific commit id, use

$git branch <new_branch_name> <commit-id>

Switch between branches:-

$git checkout <new_branch_name>

Continue..

If you want to create new branch and switch to new branch.

$git checkout -b <new_branch_name>

If you want to delete a branch.

$git branch -D <branch_name>

Continue..

If you want to rename a branch.

$git branch -m <branch-name> <new-branch-name>

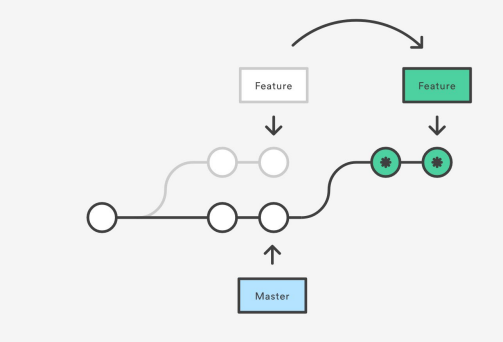
To merge branches.

$git merge origin/<branch-name>

Continue..

When working with Git branches, rebase is one of the main feature of Git to know.

Git rebasing is the process of moving a branch origin or a sequence of commits to a new base.



Continue..

Rebase can be done using the following command.

$git rebase <branch-name>

When working with branches, you can integrate work in 2 ways:-

1. Rebase and merge.
2. Direct merge.

Using git fetch

“Get latest version of origin without affecting your work”

Using git fetch

git fetch

Merging the changes from fetch

```
git merge origin/master
```

Working as a team

There are two methods

1. The owner of the repository can add contributors to the repository directly from repository settings.
2. Each developer can fork the repository and make their changes and create a pull request into the main repository.

Forking repositories

Forking is the process of copying a project. The main advantages are

- **Experimentation**
  The developer can freely experiment with the code without affect the main project.
- **Proposed changes**
  The developer can work of proposed changes.

Pull request

Tell others or the owners about the changes you made to their project.

Info

Pull request is used when contributing to open source projects.

Steps to create pull request

Fork the repository.

Make changes to the project and push the changes to your repository.

Find repository links.

```
$git fetch -v
```

Continue..

Add the parent repository as upstream so as to sync your forked repository with parent repository.

```
$git remote add upstream
https://github.com/original-owner-username/original-repository.git
```

Now sync the repository.

```
$git fetch upstream
```

Continue..

This creates a new branch with name upstream/master.

Now switch to master branch.

```
$git checkout master
```

Now merge upstream master with master

```
$git merge upstream/master
```

Now create a pull request in github or any other service.

Git Tags

Used to capture a point from history. Tag is a branch that doesn't change.

$git tag <tag-name>

To add documentation to tag.

$git tag -a <tag-name> -m “message”

To list tag.

$git tag

Tags doesn’t get pushed automatically, the developer has to push the tag.

Info

To push tags, use the command.

$git push origin <tag-name>

Release is created from existing tags. It also creates release notes and and links to download code.

The .gitignore file

“Avoid git from tracking unwanted files”

The .gitignore file

```
# Compiled class file
*.class

# Compiled bytecode
*.javac

# Log file
*.log
```

Bonus

1. Hacktober fest
2. Host sites using git
3. Jekyll

Additional references

https://www.sashwat.in/devops/git/introduction-to-git/

https://www.sashwat.in/devops/git/working-with-git/

https://github.com/github/gitignore

http://gitready.com/

https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

https://www.atlassian.com/git

https://try.github.iohttp://layervault.tumblr.com/post/102541175774/the-history-of-version-control

Thank you