

# Introduction to git

Sunil Thomas T

College of Engineering Attingal

*suniltt@gmail.com*

December 8, 2016

# Why you need version control

# Installing git

- windows
- linux

## Initial configuration

```
1 git config --global user.name "Dr. Sunil"  
2 git config --global user.email "suniltt@gmail.com"  
3
```

# Initialize git repository

Create a new folder. Move to that folder.

```
1 git init
```

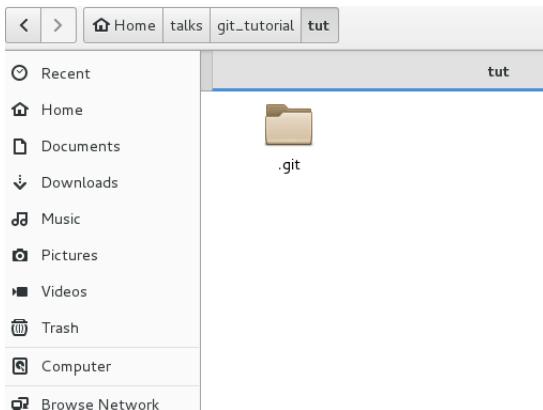
# Initialize git repository

Create a new folder. Move to that folder.

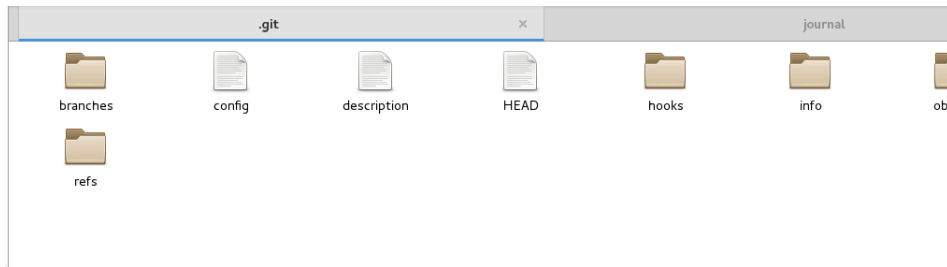
```
1 git init
```

What happens when you do git init.

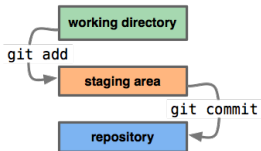
A hidden folder containing the repository is created. Examine that folder. (Control h for viewing hidden files)



# Inside the git repo



# Git work flow



# Examine git status

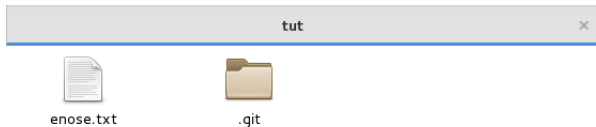
```
1 git status
```

```
unil@vu2swx:~/talks/git_tutorial/tut$ git status
On branch master

nothing to commit (create/copy files and use "git add" to track)
unil@vu2swx:~/talks/git_tutorial/tut$
```



Add a file to your working directory.



```
1 git status
```

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git status
On branch master

Initial commit

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

        enose.txt

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

# Add the file to staging area

```
1 git add enose.txt
2 git status
```

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git add enose.txt
sunil@vu2swx:~/talks/git_tutorial/tut$ git status
On branch master

Initial commit

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

        new file:   enose.txt
```

# Commit changes

```
1 git commit -m "initial commit" enose.txt
```

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git commit -m "Initial commit" enose.t
[master (root-commit) bdf4d0a] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 enose.txt
```

# Add more files



enose.bib



enose.txt



.git

```
1 git add enose.bib
2 git status
3 git commit -m " Added bibliography" enose.bib
```

## Still more files



drawing.fig



enose.bib



enose.txt



.git

# Commit more files

```
1 git add drawing , fig  
2 git status  
3 git commit -m " Added drawing" drawing , fig
```

# Let us see the history

```
1 git log
```

```
unil@vu2swx:~/talks/git_tutorial/tut$ git log
commit cb3946c336ea9d4145950adce46abc017dd10775
author: suniltt <suniltt@gmail.com>
date:   Wed Dec 7 22:01:54 2016 +0530

    add drawing

commit b14403e9f02663079684aa8c689c454430f69bb1
author: suniltt <suniltt@gmail.com>
date:   Wed Dec 7 22:01:32 2016 +0530

    bibliograph

commit bdf4d0a46a5cd582b9a4659626bbd9fe42a383cb
author: suniltt <suniltt@gmail.com>
date:   Wed Dec 7 21:33:19 2016 +0530

    Initial commit
unil@vu2swx:~/talks/git_tutorial/tut$ █
```

# Checkout

Now let us edit `enose.txt`. After some editing we see that we want to restore the saved copy.

```
1 git status
2 git checkout enose.txt
3 git status
```



# Checkout from previous commits

Get the commit commit id

```
1 git log --oneline
```

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git log --oneline
cb3946c add drawing
b14403e bibliograph
bdf4d0a Initial commit
```

You can check out from any previous commits using the commit id

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git log --oneline
cb3946c add drawing
b14403e bibliograph
bdf4d0a Initial commit
sunil@vu2swx:~/talks/git_tutorial/tut$ git checkout bdf4d0a enose.txt
sunil@vu2swx:~/talks/git_tutorial/tut$
```

# Creating a new branch

```
1 git status
2 git branch ieee
3 git checkout ieee
4 git status
```

Do your editing on branch ieee. You can add files. commit and do all sort of operations. Now we want to switch back to master branch.

```
1 git status
2
3 git checkout master
4 git status
```

Check out ieee branch and add few more files to your ieee branch and commit

```
1 git status
2 git checkout ieee
3 git status
```



config



drawing.fig



enose.bib



enose.txt



footer



.git

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git add config
sunil@vu2swx:~/talks/git_tutorial/tut$ git add footer
sunil@vu2swx:~/talks/git_tutorial/tut$ git commit -m "Adding config and footer to
o ieee" footer config
[ieee 26c70bf] Adding config and footer to ieee
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 config
 create mode 100644 footer
sunil@vu2swx:~/talks/git_tutorial/tut$
```

## Check out master branch

```
1 git status  
2 git checkout master  
3 git status
```

See the files from ieee branch vanish.



# Merge branches

## Check out master branch

```
1 git status
2 git checkout master
3 git status
4 git merge ieee
```

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git merge ieee
Updating cb3946c..26c70bf
Fast-forward
 config | 0
 footer | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 config
 create mode 100644 footer
sunil@vu2swx:~/talks/git_tutorial/tut$
```

## Delete a branch

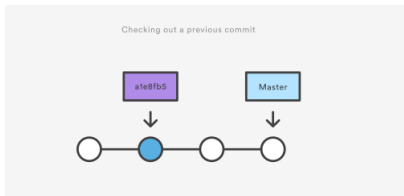
```
1 git status
2
3
4 git branch -d ieee
```

# Reverting Merge

```
sunil@vu2swx:~/talks/git_tutorial/tut$ git log --oneline
26c70bf Adding config and footer to ieee
cb3946c add drawing
b14403e bibliograph
bdf4d0a Initial commit
sunil@vu2swx:~/talks/git_tutorial/tut$
```

```
unil@vu2swx:~/talks/git_tutorial/tut$ git log --oneline
6c70bf Adding config and footer to ieee
b3946c add drawing
14403e bibliograph
df4d0a Initial commit
unil@vu2swx:~/talks/git_tutorial/tut$ git reset --hard cb3946c
EAD is now at cb3946c add drawing
unil@vu2swx:~/talks/git_tutorial/tut$ git log --oneline
b3946c add drawing
14403e bibliograph
df4d0a Initial commit
unil@vu2swx:~/talks/git_tutorial/tut$
```

# Viewing an older version



Get commit id

```
1 git log --online
```

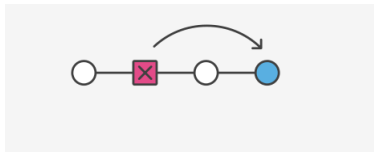
Checkout

```
1 git checkout b14403e
```

To get back the master version

```
1 git checkout master
```

# reverting to older version



Get commit id

```
1 git revert <commit id>
```



# Collaborating

Go to git hub. Create an account. Verify. Create a repository under your login.

# Collaborating

[Pull requests](#) [Issues](#) [Gist](#)

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



mfstvm ▾

Repository name

/ mygit



Great repository names are short and memorable. Need inspiration? How about **improved-octo-meme**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.



**Initialize this repository with a README**

# Collaborating

## Quick setup — if you've done this kind of thing before

or



We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# mygit" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mfstvm/mygit.git
git push -u origin master
```



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

```
git remote add origin https://github.com/mfstvm/mygit.git
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.