Github

By # Suman Gangopadhyay

MSc | CCNP | CCNA-Security | CCNA

"Build software better, together.", "Where software is built"

GitHub

Goals

- What is GitHub ?
- History of GutHub
- Git Initialiazation
- Putting a File and Commiting it into Git repository
- Viewing changes to a file from a Git repo
- Deleting, Moving & Renaming a File in the Git Repo
- Undo changes to the working directory
- Undo changes in the staging area

GitHub

Goals – 2

- Undo Commits
- Retrieving old versions of a File from the GIT repo
- Undo Multiple GIT Commits from the repo
- Soft RESET
- Hard RESET
- Remove untracked files from the working directory of GIT repo
- The **“.gitignore”** file & its relevance !
- Ignore Files Globally

GitHub

Goals – 3

- Create GIT Branches
- Use a New GIT branch
- Create and Switch between GIT branches
- Compare GIT branches
- Rename GIT Branches
- Merging 2 or more GIT branches
- Merge Conflicts
  - Abort merge
  - Resolve the conflicts manually
  - Use any merge tool.

GitHub

Goal – 4

- Create a Remote Repository on github.com
- Remove Remote GIT Repo
- Push Local Repo to Remote GIT Repo
- View Local & Remote GIT Branches
- FETCH remote GIT Repo to your Local machine
- Check Remote GIT Repo
- Delete Remote GIT Repo

GitHub

GITHUB Overview

- It is a Version Control System (VCS)
- Create by Linus Torvalds in 2005
- Unlike other VCS GIT creates independent full-fledged repository with complete version tracking system on the local machines.
- It is a Open-Source software that means completely Free !
- It is licensed under GNU(v2)

GitHub

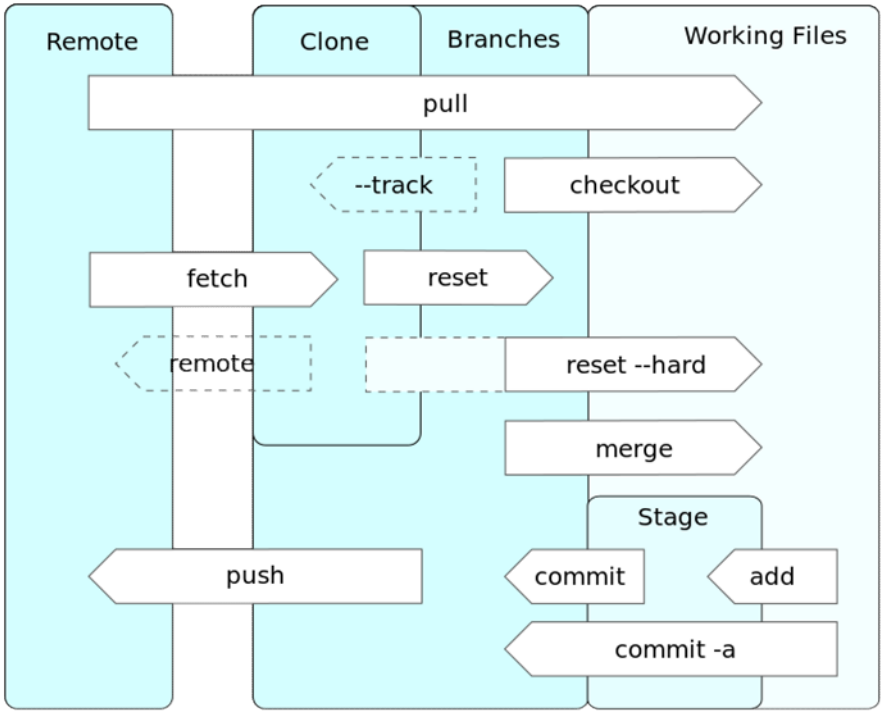
Software Needed !

- Download the GIT software from https://git-scm.com/

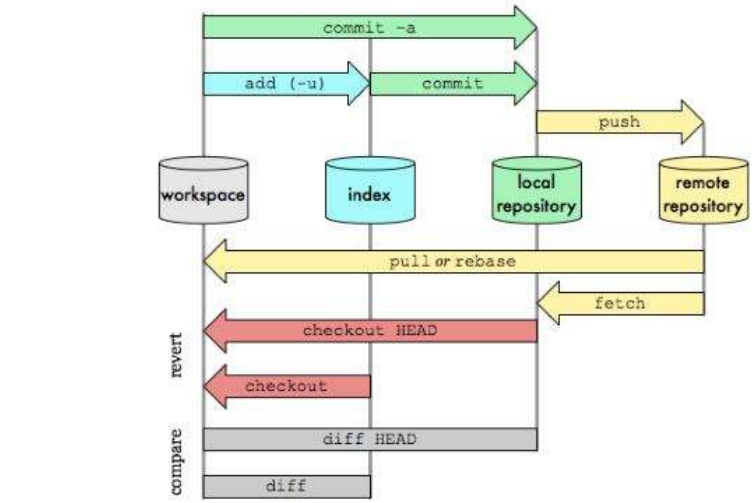
- Choose the correct software depending on your machines
  - OS whether it is Windows / UNIX / LINUX / Mac
  - Either your machine is **x86** or **x64** bit

GitHub

Storage Level in GIT

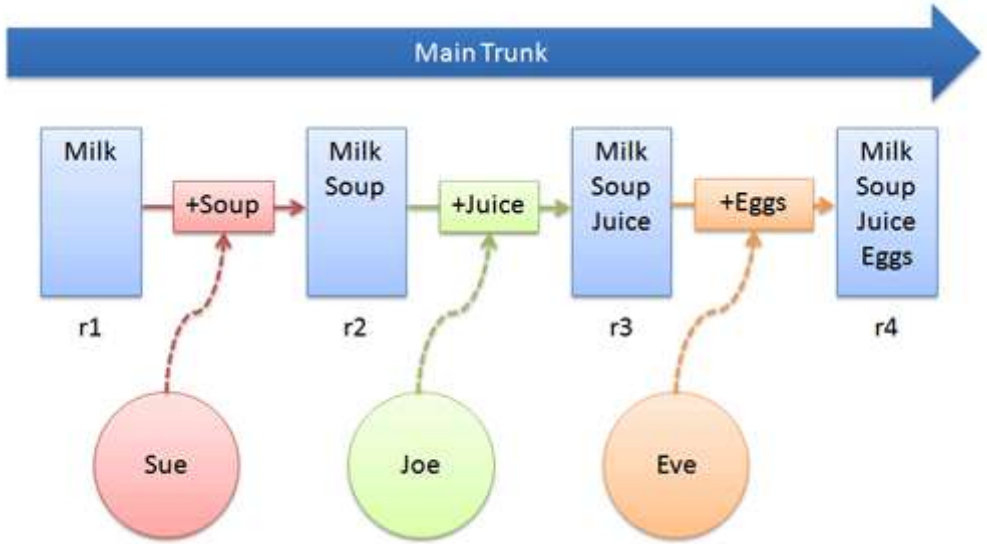


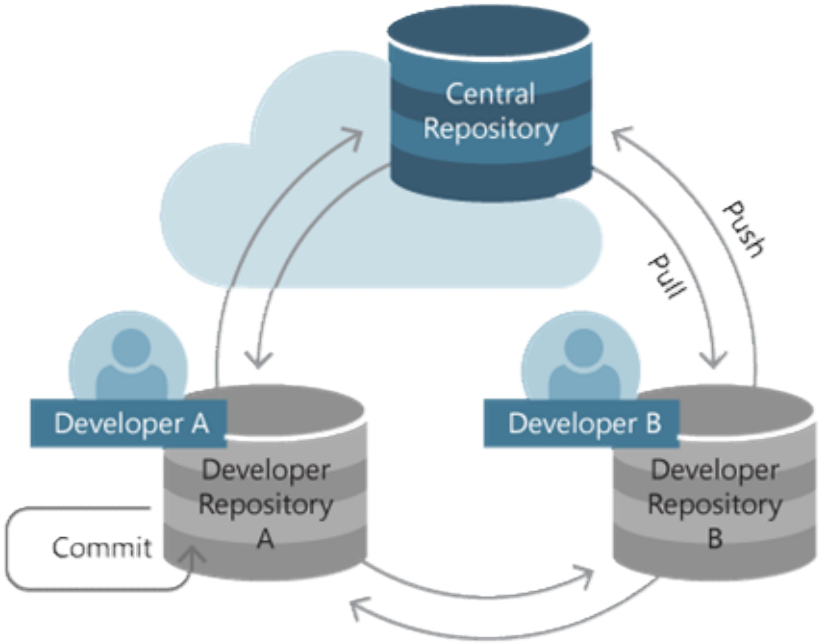
Stages of GIT Working



GitHub

Centralized VCS



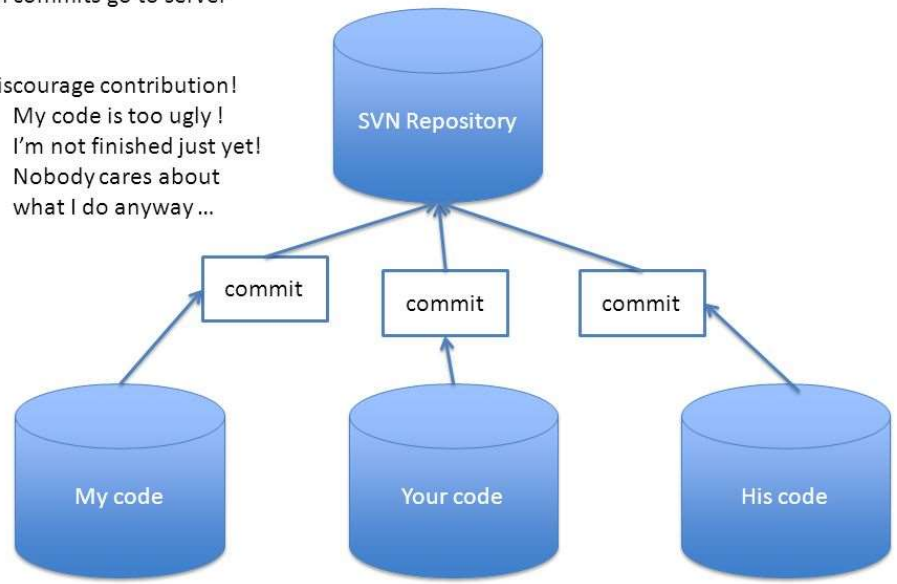


GitHub

Why use Git

SVN :
Centralized server,
All commits go to server

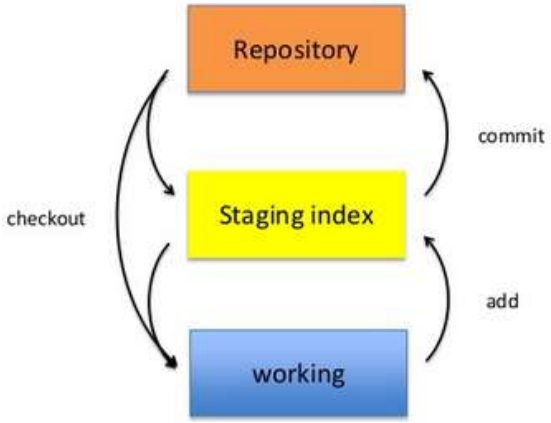
Discourage contribution!
- My code is too ugly !
- I'm not finished just yet!
- Nobody cares about what I do anyway …



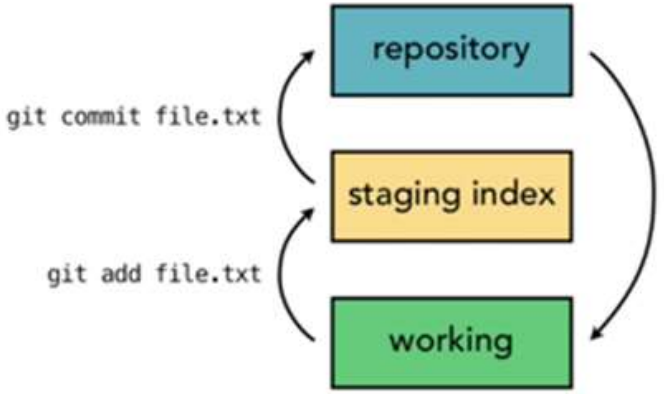
GitHub

GIT Architecture

Git uses a three-tree architecture



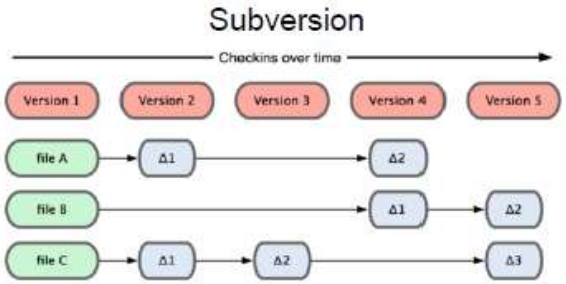
three-tree architecture

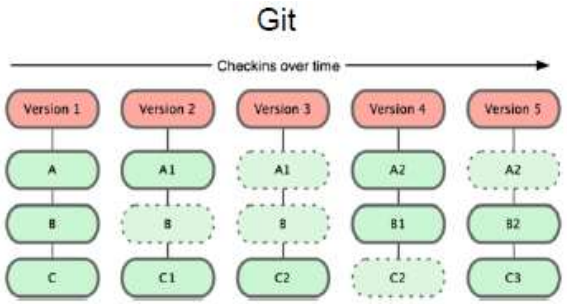


GitHub

GIT Snapshot

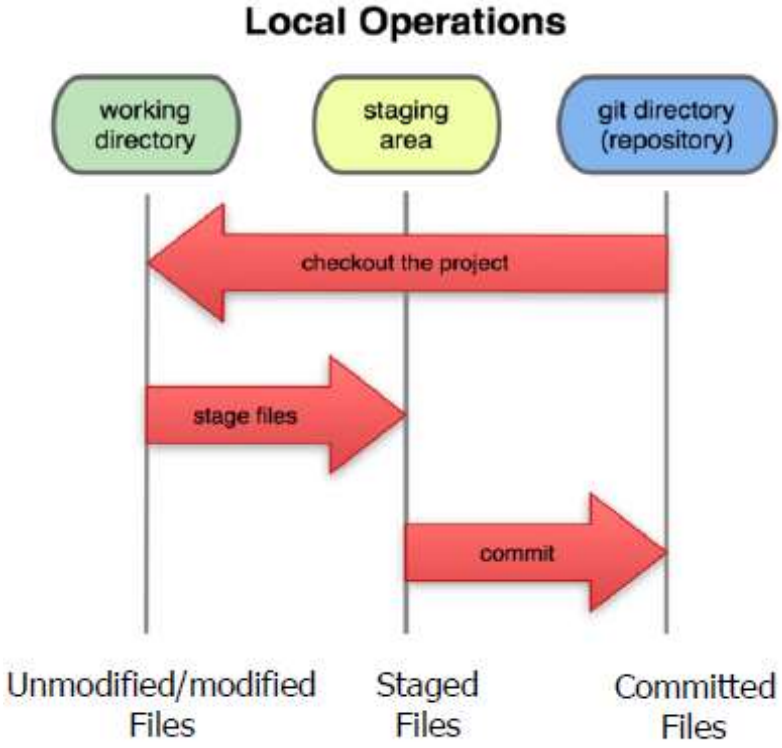
- Centralized VCS track each version of data on each individual file.
- It keeps the “SNAPSHOT” of the entire project state.





GitHub

Local GIT Areas



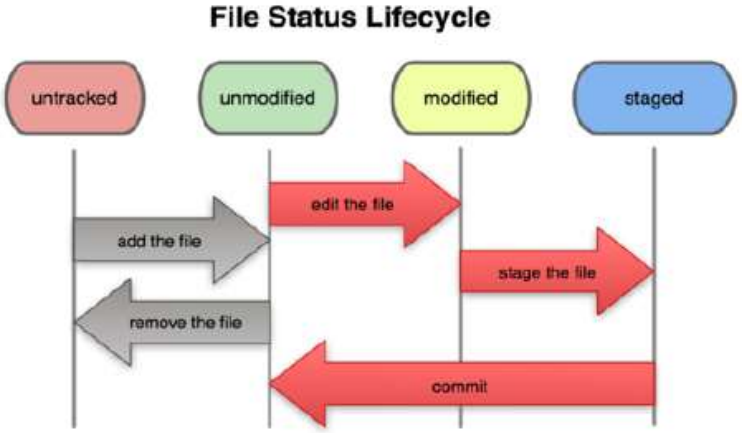
GitHub

GIT Workflow

Modify ➔ Files present in your project directory

Staging Area ➔ Files added to your Staging Area

Commit ➔ Files in your Staging Area and stores permanently into the your GIT repository.



GitHub

GIT Initialization

- To start and create a local Repository you have to initialize a GIT inside your project folder.
- The command you use is :-

```
# git init
# git config user.name “Suman Gangopadhyay”
# git config user.email linuxgurusuman@gmail.com
```

```
C:\Users\suman\ruby-lib>git init
Initialized empty Git repository in C:/Users/suman/ruby-lib/.git/

C:\Users\suman\ruby-lib>git config user.name "Suman Gangopadhyay"

C:\Users\suman\ruby-lib>git config user.email linuxgurusuman@gmail.com
```

GitHub

Adding & Commiting into GIT Repo

Add File into Staging Area

Syntax

```
# git add <Name_of_File>
```

Add File into Staging Area

Syntax

```
# git commit –m “Comment for your File”
```

GitHub

Writing the Commit Messages

- Short single line summary less than 50 characters
- Keep each line less than 72 characters
- Write commit changes in the Present Tense
- Add Ticket “Tracking Number” from bugs to Support Request

```
C:\Users\suman\ruby-lib>git add def.rb

C:\Users\suman\ruby-lib>git commit -m "First save def.rb"
[master (root-commit) bf11d34] First save def.rb
 1 file changed, 19 insertions(+)
 create mode 100644 def.rb
```

GitHub

View GIT Status

```
# git status
```

```
C:\Users\suman\ruby-lib>git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   count_vowel.rb

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

        even_number.rb
        even_number_table.rb
```

GitHub

View Commit Log

```
# git log
```

```
C:\Users\suman\ruby-lib>git log
commit bf11d344c4ab3e415f920db479df9adc0e6880da
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Thu Feb 8 21:00:43 2018 +0600

    First save def.rb
```

```
# git log --author="suman"
```

```
C:\Users\suman\ruby-lib>git log --author="suman"
commit bf11d344c4ab3e415f920db479df9adc0e6880da
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Thu Feb 8 21:00:43 2018 +0600

    First save def.rb
```

GitHub

GIT Help

- To get HELP on any GIT command use the following :-

```
# git help <Name_Of_Command>
```

- If you are using **UNIX / LINUX / MAC** systems you can use the famous man command with the GIT also :-

```
# man git-<Name_Of_Command>
```

GitHub

GIT HASH Values

- GIT generates a MD5# checksum for each change set.
- The Checksum algorithm converts data into a single Hash values.
- These Hash values are unique and distinct.
- The same data ends with same checksum.

```
C:\Users\suman\ruby-lib>git log -n 2
commit 9afe9e18821c3ff5cef1fa1201d3ef10dfea9db5
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Sun Feb 18 13:54:25 2018 +0600

    second revision of def.rb

commit bf11d344c4ab3e415f920db479df9adc0e6880da
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Thu Feb 8 21:00:43 2018 +0600

    First save def.rb
```

GitHub

HEAD Pointer

- It is the reference variable
- It points to the “TIP” of the current branch of the repo.
- Last state of the Repo what was last checked
- Present in `.git/refs/heads/master` file

```
C:\Users\suman\ruby-lib\.git\refs\heads>type master
9afe9e18821c3ff5cef1fa1201d3ef10dfea9db5
```

```
C:\Users\suman\ruby-lib>git log -n 2
commit 9afe9e18821c3ff5cef1fa1201d3ef10dfea9db5
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Sun Feb 18 13:54:25 2018 +0600

    second revision of def.rb

commit bf11d344c4ab3e415f920db479df9adc0e6880da
Author: Suman Gangopadhyay <linuxgurusuman@gmail.com>
Date:   Thu Feb 8 21:00:43 2018 +0600

    First save def.rb
```

GitHub

Viewing Changes To a File

The commands to see the changes to a file are

Syntax

```
# git diff <FileName>
# git diff --staged <FileName>
# git diff --staged
```

Example

```
# git diff def.rb
```

```
C:\Users\suman\ruby-lib>git diff def.rb
diff --git a/def.rb b/def.rb
index 795f25d..bf44f82 100644
--- a/def.rb
+++ b/def.rb
@@ -17,3 +17,4 @@ end
 s=Second_Class.new
 puts s.area_circle(11)
 puts s.HelloWorld()
+puts s.HelloWorld()
\ No newline at end of file
```

GitHub

Delete & Rename Files & Track Them

- It is highly recommend that you delete your Files using GIT.
- If you do not delete your files using GIT then GIT cannot track them in the Repo.
- Delete and Rename are both the same things in GIT or Linux.

Syntax

```
# git rm <FileName>
# git add <FileName>
# git commit –m “Comments to save your File in Repo>
```

GitHub

Move a File

Syntax

```
# git mv <FileName>
# git add <FileName>
# git commit –m “Comments to save your File in Repo>
```

Example

```
# git mv def.rb ruby_class.rb
```

GitHub

```
C:\Users\suman\ruby-lib>git mv def.rb ruby_class.rb

C:\Users\suman\ruby-lib>git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        renamed:    def.rb -> ruby_class.rb

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ruby_class.rb
```

GitHub

UNDO the Changes To Working Directory

Syntax

```
# git checkout -- <FileName>
```

Note :- If the Checkout if it is used with --can option will replace everything on the present directory as well as on the Branch Repo too.

- Try to use it before using the COMMIT else after commit no UNDO can be done.

GitHub

UNDO in the Staging Area

Syntax

```
# git reset HEAD <FileName>
```

Example

git reset HEAD ruby_class.rb

GitHub

UNDO Commits

- We can UNDO the last commit only.

Syntax

```
# git commit --amend -m “Comments”
```

Example

```
# git commit --amend -m “undoing the last commits”
```

GitHub

Retrieving an Old Version from Repo

GitHub