GIT AND GIT HUB

<https://automationstepbystep.com/>

[How to change your git user and email account for per project]

Run the following command:

git config user.name “Saini.ashu90”

git config user.email [Saini.ashu90@gmail.com](mailto:Saini.ashu90@gmail.com)

To verify the current user info:

git config –get user.name

git config –get user.email

====================for global config command===========================

git config –global user.email “saini.ashu---@gmail.com”

git config –global user.name “ashusai”

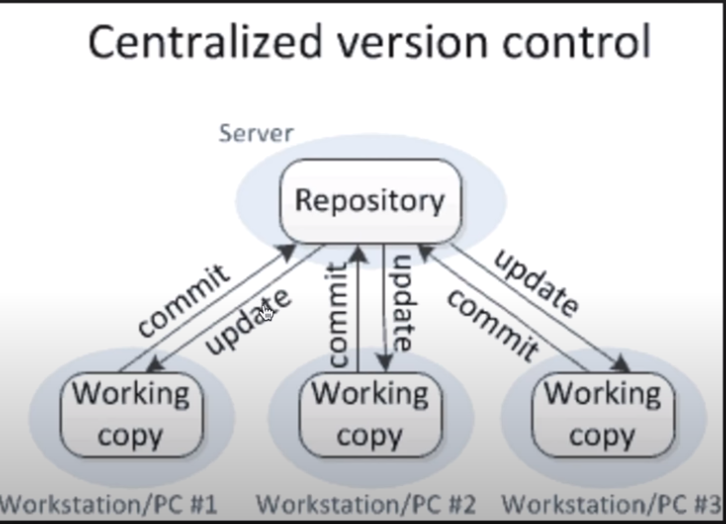
GIT: Git is a version control system [VCS].

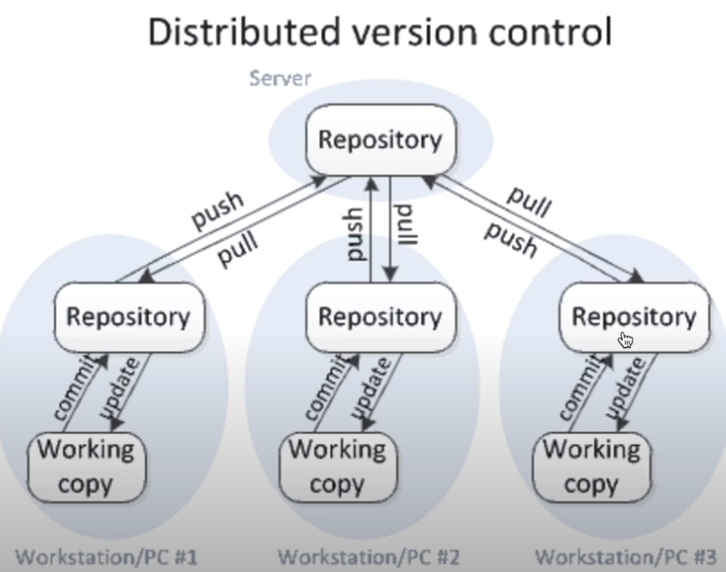
* To track changes in files/folders
* To collaborate in teams.
* Free and open source system

GIT != GITHUB

GIT = DVCS

Centralized VCS | Distributed VCS





GITHUB: website to upload your repositories online

* Provides a backup
* Provides visual interface to your repo.
* Makes collaboration easier

== T-2 Install Git On Windows =

1. Install Git or verify it is already installed
   1. Go to command line and enter “git –version”

**git version 2.17.0.windows.1**

1. **Download and Installed**

GIT BASH: git bash is a git utility which provide helps to write command and provide autocomplete options

RECOMMENDED OPTION: USE GIT FROM THE WINDOWS COMMAND PROMPT

Click next

Use the default selected option: “Checkout windows-style, commit Unix-style line endingss”

Click next

Select: Use windows default console windows

1. Add your project to git
2. Create folder in your local system: GitHub
3. Open a command prompt and Go to the folder location GitHub “E:\GitHub\”

**GIT COMMAND:**

* git init : is to initiate git repository
* git status: check the status
* git add: add new files/folder in the repository
* git add \*.\* : it is add all the files related to the wild card
* git add . : it will add everything which is not yet

ADDING TO REMOTE REPOSITORY

1. Go to github website: <https://github.com/>
2. Create a new repository and make it public
3. Copy the new repo. url <https://github.com/sainiashu/git.git>
4. Go to command line and enter the below command

git remote add origin <https://github.com/sainiashu/git.git>

== T-3 BRANCHING AND MERGING=

1. Create a branch
   1. git branch “branch name”
2. checkout new branch
   1. git checkout “branch name”
3. push your changing on origin of your branch
   1. git push -u origin “branch name”
4. Merge new branch in master branch “BEFORE MERGE INTO MASTER BRANCH FIRST YOU NEED TO CHECK OUT MASTER BRANCH ”
   1. git merge “newbranch\_1”
   2. git push -u origin master : push your changes into master branch.
5. Delete branch
   1. Git branch -d “branch name” : Delete branch from local system
   2. Git push origin –delete “branch name” : delete branch from remote

======== HOW TO SEND EMAIL NOTIFICATION FROM YOUR GIT HUB ACCOUNT===

1. Git Hub – >Repository --> Settings -->
   1. Go to Notification
   2. Enter Address [Saini.ashu90@gmail.com](mailto:Saini.ashu90@gmail.com)
   3. Checked “Active” checked box
   4. Click “Setup notifications”
2. Test and validate by making some change in the project
   1. Made some changes
   2. And follow all the basic stesps
      1. Add/ modify files
      2. Verify the branch status: git status
      3. Git commit -m “add comments”
      4. Git push -u origin “branch name” master

============ Git Tags - what, why, when and how ================

1. Why should I create a tags
   1. To mark release points for your code/data
   2. To create historic restore points
2. When to create a tags
   1. When you want to create a release point for a stable version of your code
   2. When you want to create a historic point for your code and data that you can refer at any future time (to restore your data)
3. How to create tags in GIT
   1. Checkout your branch where you want to create a tag : git checkout “branch name”
   2. Create a tag with some name: git tag “tag name” : for eg: git tag v1.0 (it is a light tag)
      1. You can create a annotated tag using -a command: git tag -a -m “message”:

git tag -a v1.1 -m "annotated tag"

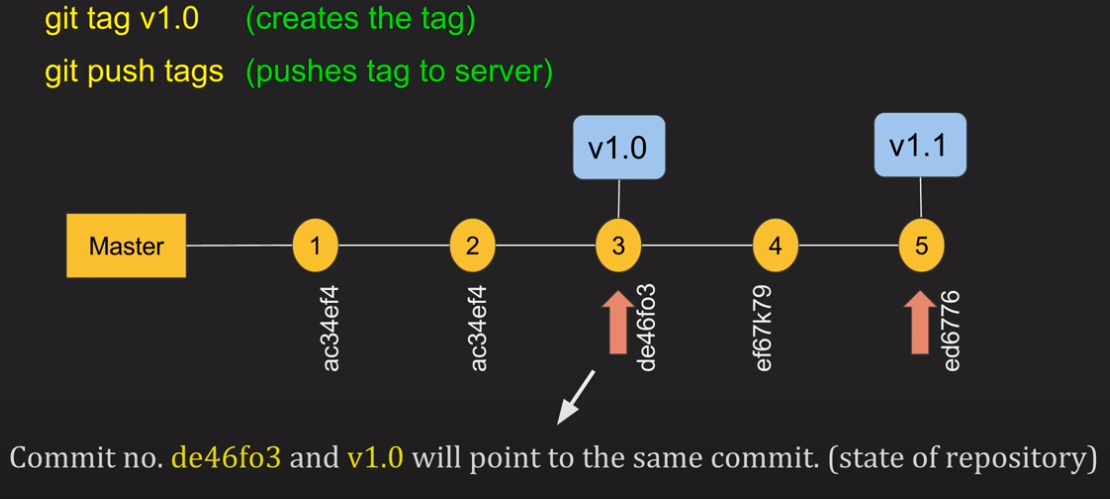
1. Display or show tags
   1. git tag
   2. git show v1.0
   3. git tag -l “v1.\*” ( -l shows the tags)
      1. output: v1.0 , v1.1
2. how to push tags on git hub
   1. First you can check on git hub page 0 release visible on the release tab/icon
   2. Go to command line: **git push origin v1.0**
   3. You can also use command to push all tags in one go:
      1. Git push origin –tags
      2. Git push –tags
3. If you want to delete tag use the following commands:
   1. git tag -d v1.0

or

* 1. git tag –delete v1.0

USE git push origin – d v1.0 : to delete the tag from the repository

PURPOSE IS SHOWN BELOW



\*\*\*\*\* WE CANNOT CHECKOUT TAGS IN GIT

HOW TO CHECKOUT A BRANCH FROM TAG

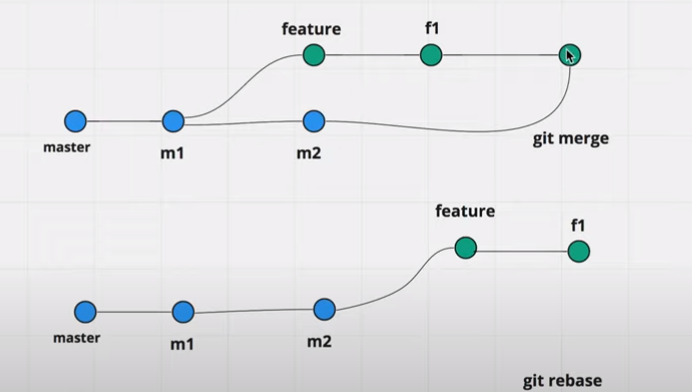
git checkout -b “any branch name” v1.0(v1.0 is tag name from where you want to checkout the branch)

WE CAN CREATE A TAG FOR THE PAST COMMIT:

git tag “tag name” “reference of commit” (No need to enter the 40 digit no. just pass the 5-7 numbers)

git tag “v1.0” “commit id”

============ Git Merge VS Git Rebase ================



Git Merge:

1. is a non-destructive operation
2. existing branches are not changed in anyway
3. creates a new merge commit in the feature branch.

Git Rebase:

1. Moves the entire feature branch to begin on the tip of the master branch.
2. Re-writes the project history.
3. We get much cleaner and linear history.