Install & Config Git

Configuration on AWS Server

Login AWS Account and create two EC2 Instances (Linux)

- 1. Mumbai Location
- 2. Singapore Location.

Step 2 : For Mumbai Location

Open terminal on Mumbai machine

$Ec2-user

Swich user as a root

$sudo su root

Update yum packages

#yum update –y

Install git package

#yum install git –y

show where store git

#which git

show version of git

#git --version

Create user name and email id

#git config - - global user.name sara

#git config - - global user.email sara@gmail.com

Show create user list

#git config - -list

Step 2 : For Singapore Location

Open terminal on Singapore machines

$Ec2-user

Swich user as a root

$sudo su

Update yum packages

#yum update –y

Install git package

#yum install git –y

show where store git

#which git

show version of git

#git --version

Create user name and email id

#git config - - global user.name ricky

#git config - - global user.email ricky@gmail.com

Show create user list

#git config - -list

Step3 : Create Account on github

Github - GitHub, Inc. is an American multinational corporation that provides hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

Github is a web-based platform **used** for version control. Git simplifies the process of working with other people and makes it easy to collaborate on projects. Team members can work on files and easily merge their changes in with the master branch of the project.

Create account on github

https://github.com/

Step 4:How to write and commit code

Create directory for git repository
#mkdir mumbaigit
#ls
#cd mumbaigit
Convert directory into repository
#git init
Create sample code file
#cat > mumbai1
This is my Mumbai git code
Ctrl+d
Check status and transfer code file working area to staging area
#git status
#git add .
#git status
Commit code into local area
#git commit -m "My first commit from Mumbai"
#git status
#git log
For show detail of code with help of commit id
#git show <commit ID paste here>

Step 5:How to push & pull from git

- Go to git hub and login into account
- Now create repository
- Eg. Centralgit

- Copy path of central repo http://...

#git remote add origin https:….
#git push -u origin master
Username : <github username>
Password: <github password>
or
git push https://<GITHUB_ACCESS_TOKEN>@github.com/<GITHUB_USERNAME>/<REPOSITORY_NAME>.git
Eg.
git push https://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@github.com/hackwithabhi/central.git

- Authenticate with github username or password, after press enter code go to the central repo

Step 6: How to pull from git

- #sudo su
- #mkdir singaporegit
- #ls
- #cd singaporegit
- #git init
- #ls -a
- #git remote add origin http …..
- #git pull origin master
- #git log
- #git status
- #git pull origin master

- #git log
- #git show <commit ID>

Git command with explanation

.git init

- The command git init is used to create an empty Git repository.
- After the git init command is used, a .git folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.

git init

Git command with explanation

.git init

- The command git init is used to create an empty Git repository.
- After the git init command is used, a .git folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.

#git init

Git command with explanation

git add

- Add command is used after checking the status of the files, to add those files to the staging area.
- Before running the commit command, "git add" is used to add any new or modified files.

#git add .

Git command with explanation

git commit

- The commit command makes sure that the changes are saved to the local repository.
- The command "git commit –m <message>" allows you to describe everyone and help them understand what has happened.

git commit -m “commit message”

Git command with explanation

git status

- The git status command tells the current state of the repository.
- The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.

#git status

Git command with explanation

git config

- The git config command is used initially to configure the user.name and user.email. This specifies what email id and username will be used from a local repository.
- When git config is used with --global flag, it writes the settings to all repositories on the computer.

#git config --global user.name “any user name”

#git config --global user.email <email id>

Git command with explanation

git pull

- The git pull command is used to fetch and merge changes from the remote repository to the local repository.
- The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.

#git pull <branch_name> <remote URL>

Git command with explanation

git push

- The command git push is used to transfer the commits or pushing the content from the local repository to the remote repository.
- The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

#git push -u origin master

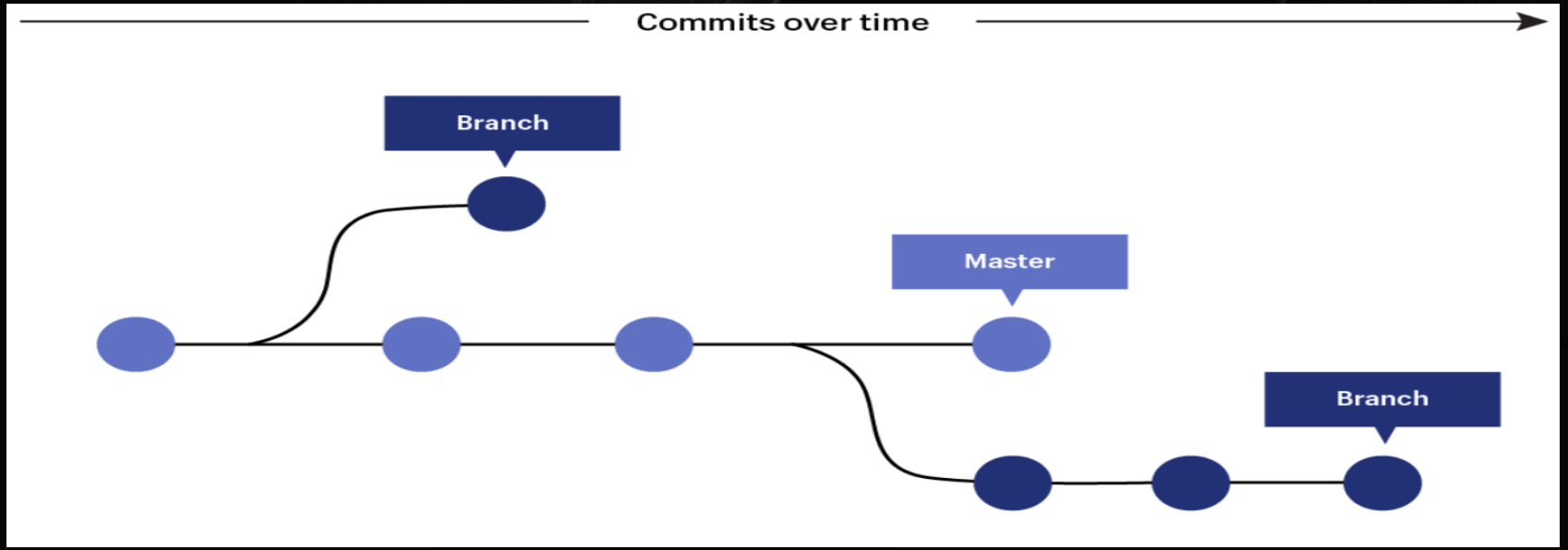
Git command with explanation

git log

- The git log command shows the order of the commit history for a repository.
- The command helps in understanding the state of the current branch by showing the commits that lead to this state.

#git log

How to create Branch, Merge & stash



- The above diagram visualizes a repository two isolated line of development. One for a little features, one for master and one for big features. cause of branches keeps main master branch error free.

How to create Branch, Merge & stash

- Each task has one separate branch
- After done with code, merge other branches with master.
- This concept is usefull for parallel development.
- You can create any no of branches, change are personal to that particular branch.
- File created in workspace will be visible in any of the branch workspace until you commit, once you commit, then that files belong to that particular branch.
- When created new branch, data of the existing branch is copied to new branch.

For show existing commit

Following command show commit on branches

#cd /Mumbai

#git log --oneline

For show branch

#git branch

*Master

(star indicate current working branch)

For create new branch

The git branch command is used to determine what branch the local repository is on.

The command enables adding and deleting a branch.

Create a new branch

#git branch <branch_name>

Eg.

#git branch branch1

Change branch

#git checkout

The git checkout command is used to switch branches, whenever the work is to be started on a different branch.

The command works on three separate entities: files, commits, and branches.

Checkout an existing branch

#git checkout <branch_name>

Eg.

#git checkout branch1

#git branch

For delete branch

The following command deleting a branch.

#git branch -d <branch_name>

For merge branch

git merge

The git merge command is used to integrate the branches together. The command combines the changes from one branch to another branch.

It is used to merge the changes in the staging branch to the stable branch.

git merge <branch_name>

Eg.

#git merge branch1

#ks

#git log --oneline

Transfer update in remote repo

For puch updates in central repo

#git push origin master

Username

Token

Git Conflict

When same file having different content in deferent branches, if you do merge, conflict occurs(Resolve conflict then add and commit).

Conflict occurs when you merge branches

To resolve conflict go to master branch edit manually.

Git stash

git stash

The git stash command takes your modified tracked files and saves it on a pile of incomplete changes that you can reapply at any time. To go back to work, you can use the stash pop.

The git stash command will help a developer switch branches to work on something else without committing to incomplete work.

To transfer file in tash

#git tash

Git Stashing command

Store current work with untracked files

#git stash list

#git stash apply@{0}

Or

#git stash pop

To clear the stash items

#git stash clear

Git Reset

Git reset is a powerful command that is used to undo local changes to the state of a git repo

To reset staging area

#git reset <filename>

To resent changes from both staging area and working directory at a time

#git reset --hard

Git Reset eg.

Create one file in working area

#cat > demofile

#git add .

#git status

#git reset .

#git status

Now it remove from staging are, it available at workspace

Git Reset eg.

To resent changes from both staging area and working directory at a time

#cat > demofile2

#git add .

#git status

#git reset --hard

#git status

It remove from staging area as well as workspace

Git revert

The revert command help you undo an existing commit.

It does not delete any data in the process instead rather git create new commit with the included file reverted to their previous state, you version

```
#cd    /mumbaigit
#cat   >  demofile3
Hi,
This is new code
ctrl+d
#git   add  .
```

Git Revert

#git commit -m “code1”

#git log --oneline

Again add more code and commit file

#cat >> demofile3

New another code

ctrl+d

#git add .

#git commit -m “wrong code”

#git log --oneline

Finally after commit think we commit wrong code

Git Revert

One time read content of file

#cat demofile3

#git log

#git revert <commit-id> (wrong commit)

(please ignore previous commit)

:wq

#ls

#cat demofile3

#git log --oneline

Remove Untrack files

We can remove all untrack files (which not add into staging or not commited.)

#cd /mumbaigit

#touch file1 file2 file3

#ls

#git status

#git clean -n (it show what will be remove)

#git clean -f (it remove untrack file)

#ls

#git status

Git Tag

Tag operation allow to give meaningful names to a specific version in the repository

#git tag

#git tag -a importantcode -m “this is imp code” <commit-id>

#git tag

#git show importantcode

#git tag -d importantcode

#git tag

#git log --oneline

Git Reset eg.

To resent changes from both staging area and working directory at a time

```
#git  > demofile2
#git  add  .
#git  status
#git   reset  --hard
#git  status
```

It remove from staging area as well as workspace

Git Reset eg.

To resent changes from both staging area and working directory at a time

```
#git  > demofile2
#git  add  .
#git  status
#git   reset  --hard
#git  status
```

It remove from staging area as well as workspace

Git Ignore

Git ignore is use to ignore some files while committing.

Create one hidden file .gitignore and enter the file extension which we want to ignore.

Eg.

#vim .gitignore

*.yml

*.txt

:wq

#git addd .gitignore

#git commit -m “ignore the yml and txt file”

#git status

Git ignore

Now create some files with .yml and .txt and then run git add

#touch file{1..5}.yml

#touch note{1..5}.txt

#ls

#git status

#git add.

#git status

#git commit -m “my html file only”

Git Rebase

- Git rebase is a command that allows developers to integrate changes from one branch to another.
- Rebasing is a process to reapply commits on top of another base trip. It is used to apply a sequence of commits from distinct branches into a final commit. It is an alternative of git merge command. It is a linear process of merging.
- In Git, the term rebase is referred to as the process of moving or combining a sequence of commits to a new base commit. Rebasing is very beneficial and it visualized the process in the environment of a feature branching workflow.

Git rebase

If there are some conflicts in the branch, resolve them, and perform below commands to continue changes:

#git rebase **<branch** name**>**

#git rebase --continue

- The above command is used to continue with the changes you made. If you want to skip the change, you can skip as follows:

Git Clone

- In Git, cloning is the act of making a copy of any target repository. The target repository can be remote or local. You can clone your repository from the remote repository to create a local copy on your system.

#git clone **<repository** URL**>**