

14.12Apr.GitAdv(Merge - Conflict & Non-Conflict, Stash)

12 April 2023 06:47 PM

- 1) File .gitignore
- 2) Git Area
 - a. Working Dir.(Changes) Area
 - b. Staging Area
 - c. Local Repo. Area
 - d. Remote Area(Push, Pull, Clone)
- 3) Creating Bulk File/Dir/Folders
 - a. Windows Power shell:
 - b. Unix Shell(touch PreFixFileName{RANGE}.ext) :
- 4) **L1: Push Remote Repo.**
 - a. Install git on AWS Ubuntu(My Laptop)
 - b. Create an Empty Repo
 - c. IDE(VSCode/Eclipse)
 - d. .gitignore excludes files
 - a. *.class
 - b. *.settings
 - c. *.projects
 - e. Source Code --> Staging Area
 - f. Staging Area --> Local Repo
 - g. Create Repo on Github
 - h. PUSH
- 5) Merge No Conflict
 - a. Different Files Changes
 - b. Merge
- 6) Merge Conflict
 - a. Same Files Changes
 - b. Conflict
 - c. Create Remote Repo --> Master Branch
 - d. Master\$ Create master.txt file
 - e. Create Branch F1\$ Create F1.txt file & F12.txt
 - f. Create Branch F2\$ Create F2.txt file & F12.txt
 - g. Merge with Master Branch

```
$ echo Welcome >> README.md
$ git add README.md
$ git commit -m "MESSAGE"
OR
$ git commit -a -m "MESSAGE"
```

1. Create 2 Branch(master, B1)

2. Change in 1st Branch(master)
 - a. Added extra word in mango.txt(abc)
3. Change in 2nd Branch(B1)
 - a. Added extra word in mango.txt(xyz)
4. Checkout to 1st Branch(master)

1)Branch: master

File: index.txt

2) Create Branch B2

\$ git branch B2

Branch: B2

Update File: index.txt

master --> index.txt abc

B2 --> index.txt xyz

Merge B2(Incoming) --> master (Current) [Conflict]

\$ git checkout master # 1st Branch(master)

\$ git merge B2 # 2nd Branch(B2)

Resolve --> Integrator(TL/Architecture)

\$ git commit -m "MESSAGE"

\$ git push -u origin master

master --> B1 (C1, C2, C3, C4)

\$ git stash

OR

\$ git stash save "MESSAGE"

\$ git stash list

STASH1

STASH2

STASH3

\$ git stash **pop** stash@{n}

OR

\$ git stash **apply** stash@{n}

\$ git stash **clear** #Clear Everything

\$ git stash **drop** <stash_id>

\$ git stash **show** <stash_id>

\$ git stash **show** stash@{1} --patch

OR

```
$ git stash show stash@{1} -p
```

```
$ git stash branch branch_2 stash@{0}
```

```
$ git stash create "sample stash"
```