

Git and Git HUB

What is Git : Git is source code management system and maintain the version control.

There were 2 types of version control

- 1) Centralized version controlling
- 2) Distributed version controlling

- 3) Note: Centralized version controlling system was depreciated.

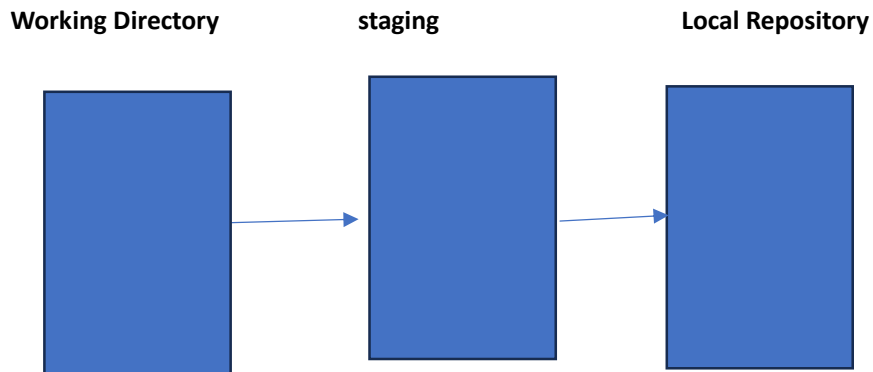
What is Git bash: Git bash is terminal to run the git commands.

How to install Git

- 1) Download the software from Git official site
[Git - Downloading Package \(git-scm.com\)](https://git-scm.com)

Install the Git bash with default options as per on-screen instructions.

Git Process:



How make working directory: Git init command make was working directory called as master directory.

Git init

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

kotai@KotaiiahS MINGW64 /c/git
• $ git init
Initialized empty Git repository in C:/git/.git/

kotai@KotaiiahS MINGW64 /c/git (master)
○ $
```

How to move files from master to staging

Command : git add < file name >

Git add .

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

kotai@KotaiiahS MINGW64 /c/git (master)
• $ touch b

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git status
On branch master

No commits yet

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

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

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git add "b"

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git status
On branch master

No commits yet

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

kotai@KotaiiahS MINGW64 /c/git (master)
○ $
```

How to move the file from staging to local repository

Command : `git commit -m "name "`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git commit -m "my first commit"
[master (root-commit) 086ec7a] my first commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 a
 create mode 100644 b

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git status
On branch master
nothing to commit, working tree clean

kotai@KotaiiahS MINGW64 /c/git (master)
○ $
```

Git Branching : To maintain source code from different projects recommended to create the branches.

Command :

- `git branch test`
- `Git checkout test`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git branch test

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git branch
* master
  test

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git checkout test
Switched to branch 'test'

kotai@KotaiiahS MINGW64 /c/git (test)
○ $
```

Merge Branch: git merge is used to combine two branches.

When we merge the child branch to parent branch, the merge showing as latest commit.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git branch test

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git branch
* master
  test

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git checkout test
Switched to branch 'test'

kotai@KotaiiahS MINGW64 /c/git (test)
• $ touch f

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git add .

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git commit -m "f"
[test 2eaffbf] f
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f

kotai@KotaiiahS MINGW64 /c/git (test)
• $ touch g

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git add .

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git commit -m "g"
[test b7918c8] g
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 g

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git log --oneline
b7918c8 (HEAD -> test) g
2eaffbf f
086ec7a (master) my first commit

kotai@KotaiiahS MINGW64 /c/git (test)
• $ git checkout master
Switched to branch 'master'

kotai@KotaiiahS MINGW64 /c/git (master)
• $ git branch merge test
```

```

kotai@KotaiiahS MINGW64 /c/git (test)
● $ git commit -m "g"
[test b7918c8] g
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 g

kotai@KotaiiahS MINGW64 /c/git (test)
● $ git log --oneline
b7918c8 (HEAD -> test) g
2eaffbf f
086ec7a (master) my first commit

kotai@KotaiiahS MINGW64 /c/git (test)
● $ git checkout master
Switched to branch 'master'

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git branch merge test

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git branch
* master
merge
test

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git log --oneline
086ec7a (HEAD -> master) my first commit

kotai@KotaiiahS MINGW64 /c/git (master)
○ $

```

.git ignore : when we compiling or validating source code like “java, terraform” will be creating extra file like .tfstate, .sh and these file are not pure source code.

To push the only pure source code into local repository required to use .git ignore file to exclude unwanted files.

Steps : 1) create .gitignore file (touch .gitignore)

2) Update the wild card for unused file (vi .gitignore)

```

kotai@KotaiiahS MINGW64 /c/git (master)
$ touch .gitignore

kotai@KotaiiahS MINGW64 /c/git (master)
$ vi .gitignore

kotai@KotaiiahS MINGW64 /c/git (master)
$ cat .gitignore
.sh
.satte

kotai@KotaiiahS MINGW64 /c/git (master)
$

```

Git Rebase: Git rebase called as fast forward merge

Commits are picked up from child branch and added into master branch as latest commits.

Advantages of rebase : 1) picked the top commits

2)Rearrange commit order

3) Merge the commits

Command : checkout sub branch (git branch <branch Name>

- Git rebase master
- Git checkout master
- Git merge test1

```
kotai@KotaiiahS MINGW64 /c/git (test1)
● $ git commit -m "nn"
[test1 243f21b] nn
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nn

kotai@KotaiiahS MINGW64 /c/git (test1)
● $ git log --oneline
243f21b (HEAD -> test1) nn
1dc3f49 mm
ffa25cc i
85ffeb0 hh
6402e22 gg
086ec7a my first commit

kotai@KotaiiahS MINGW64 /c/git (test1)
● $ git rebase master
Successfully rebased and updated refs/heads/test1.

kotai@KotaiiahS MINGW64 /c/git (test1)
● $ git checkout master
Switched to branch 'master'

kotai@KotaiiahS MINGW64 /c/git (master)
⊗ $ git branch merge test1
fatal: a branch named 'merge' already exists

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git merge test1
Updating 173950c..b9c4140
Fast-forward
 mm | 0
 nn | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mm
 create mode 100644 nn

kotai@KotaiiahS MINGW64 /c/git (master)
● $ git log --oneline
b9c4140 (HEAD -> master, test1) nn
ac79a3b mm
173950c zz
832ac18 xx
ffa25cc i
85ffeb0 hh
6402e22 gg
086ec7a my first commit

kotai@KotaiiahS MINGW64 /c/git (master)
○ $
```

Re arrange the commits: to change the commit order.

We cannot change the first commit order.

Command – git rebase -I HEAD~2(commits)

Merge Commits : to remove the unnecessary commits.

Steps : squash

- Git rebase -i HEAD~1
- Replace the word with “squash” against commit ID

Pickup selective commits from child branch:

- Collect commit id from child branch
- Checkout to master branch
- Run command “ git cherry-pick <commit ids>

Amend : Amend command used changes amend to existing commit when modify the source code.

Git commit -amend -m “commit name>

Rollback to previous version :

- Collect the commit id
- Git reset –hard <commit id>

GitHub : GitHub is remote repository and common centralized repository called as GitHub.

- Create github account
- Sign in
- Create repository