# GIT Exercises

Subject: GIT Practice

Author: minhld@smartosc.com

Name: ???

Date: Aug 28,2019

Revision: 1.0

Skype Contact: leduyminh2521989@gmail.com

## Contents

# I. Git config (Global, system, local)

## 1. Set global configs

```
git config --global user.name “Minh Le”
```

```
git config --global user.email “minhld@smartosc.com”
```

```
git config --global core.filemode false git config --global core.autocrlf true
```

[image]

## 2. View system configs

```
git config --system -l
```

[image]

## 3. View all configs

```
git config -l
```

[image]

# II. Getting and Creating Projects

## 1. Create a Git repository:

git init - Create an empty Git repository or reinitialize an existing one

git remote add origin https://github.com/leduyminh1989/git-training-demo.git - add new remote origin (to update use git remote set-url origin <new-URL>)

git remote -v (view remote)

[image]

## 2. Clone a repository:

```
git clone https://github.com/leduyminh1989/git-training-demo.git <folder-name>
```

[image]

# III. Git basic

## 1. Add & commit:

git add <file-name> or git add * (add file contents to the index )

git status (show the working tree status)

git commit -m “commit message” (record changes to the repository)

[image]

## 2. Push:

git push -u origin <current branch name> (send those changes to your remote repository, use in the fist time and for the next time push use git push)

[image]

## 3. Create branch & switch branch:

git checkout -b feature/branch1 (create a new branch named "feature/branch1" and switch to it using)
Note: a branch is not available to others unless you push the branch to your remote repository

[image]

git checkout <branch name> (to switch to an existing branch)

[image]

## 4. Delete branch

git branch -a (list all branch)

git branch -d <branch name>(delete a branch on local, or git branch -D shortcut for -d -f )

git push origin :<branch name> (find a ref that matches <branch name> in the origin repository and delete it)

[image]

## 5. Git fetch:

git fetch (download objects and refs from another repository)

Note: create a new branch on Git website before running “fetch” command to display clear results.

[image]

## 6. Git log:

git log (Show commit logs)

[image]

## 7. Merge branch

*Add more details if you can*

Modify and change README from feature/branch1

[image]

Merge branch master into feature/branch1:

git fetch (update index from remote repository)

git merge <branch> (merge branch master into feature/branch1)

[image]

README file in conflict state.

[image]

Way 1: We need to resolve it manually.

[image]

Run git status to check status. Commit and push README after resolving conflicts.

[image]

Way 2: We need to resolve it by other Git GUI Client Tool.

[image]

Create a merge request on website (*you should add more details*)

[image]

## 8. Rebase (Learn more)

Modify and commit a few times on master and feature/branch1.

git pull (fetch and replay the changes from the remote repository)

git rebase origin/master (configured master for feature/branch1) Manually resolve conflicts then add file using git add <file name>

git rebase --continue (restart the rebasing process after having resolved a merge conflict)

git rebase --continue until the last commit

[image]

After resolving every conflicts, push to remote using "-f"

git push -f

[image]

**Notes:**

If you want to synthetic the commits use git rebase -i HEAD~2

"-i" to make a list of the commits which let the user edit before rebase.

"HEAD~n" – synthetic from the last commit to HEAD~n, n is the index of the commit

## 9. Reset (Learn more)

Use git log to show commit logs.

[image]

git reset --hard <commit id> ( Reset current HEAD to the specified state )

[image]

**Notes:**

Use "--hard" to reset index and source.

Use "--soft" to reset just index.

Use “HEAD^” to specify the last commit, “HEAD~n” where n is the index of the commit from the last commit, or just use commit ID
[image]

git push -f (push updated commits to origin)

**Note**: must use “-f” (force push) after git reset
[image]