**--push my repository**

git add --all

git commit -am "<commit message>"

git push

**--Push code first time to git:**

mkdir my\_project

cd my\_project

touch .gitignore

git init

git add .

git commit -m "Initial commit"

git remote add origin youruser@yourserver.com:/path/to/my\_project.git

git push origin master

**--clone repo branch only**

git clone u project\_name.git -b branch\_name /your/folder

**--push branch or only folder**

git add foldername

git commit -m "commit operation"

git push

**--without affecting local repo (safe)**

git fetch

**--comparing two different branches**

git diff branch\_1...branch\_2

**Change working branch :**

$ git checkout [name\_of\_your\_new\_branch]

Push the branch on github :

$ git push origin [name\_of\_your\_new\_branch]

git add .

git commit -m "Initial commit"

git push -u origin [name\_of\_your\_new\_branch]

When you want to commit something in your branch, be sure to be in your branch. Add -u parameter to set upstream.

**Delete a Local GIT branch**

To delete the *local* GIT branch we can try one of the following commands:

git branch -d branch\_name  
git branch -D branch\_name

You can see all branches created by using :

$ git branch

**Delete a remote branch:**

$ git push origin --delete feature/login

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Merge development branch code into master branch?**

If you have remote-tracking branches set up locally, it's as simple as:

git checkout production

git merge development

git push origin production

git fetch origin

git checkout branch # or `git checkout -b production origin/production` if you haven't set up production branch locally

git merge origin/development

git push origin master

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**switching to new release branch of DevOps using git command**

$ git checkout -b release-1.2 develop

**--add changed file..**

git add a

git commit a -m "bugfix, in

git push

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Clone master into develop branch:**

>git clone https://github.com/nshehzad/MyTestRepository.git Develop

tag release file into master branch

----------------------------------------------------------------------------------------------------------------------------------

$ git checkout master

**Switched to branch 'master'**

$ git merge --no-ff release-1.2

Merge made by recursive.

(Summary of changes)

$ git tag -a 1.2

**Create new branch/push stuff from your local master branch to new branch**

git checkout -b newBranch

git push -u origin newBranch

**Create new branch/push stuff to new branch whatever you have locally**

git checkout -b newbranchName

git add --all

git commit -am "my new files to push to new branch"

git push -u origin Nadeem1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tags:**git tag v0.1.0-rc1

git push --tags

**Delete tags:**

**$> git push origin :refs/tags/old\_tag**

**$> git tag -d old\_tag**

# **Pushing to GitLab using Git-Bash**

$ git config --global http.sslVerify false

$ git remote add origin <https://….ES124.git>

$ git config --global credential.helper cache

$ git push -u origin --all

git: 'credential-cache' is not a git command. See 'git --help'.

Counting objects: 775, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (368/368), done.

Writing objects: 100% (775/775), 6.00 MiB | 210.00 KiB/s, done.

Total 775 (delta 395), reused 775 (delta 395)

remote: Resolving deltas: 100% (395/395), done.

To <https://git.ca.state.sbu/NCS-Dev/AES124.git>

\* [new branch]      master -> master

Branch master set up to track remote branch master from origin.

Optional useful commands

Remove

$ git remote remove origin