

Jenkins, Git and GitHub (Continued)

Git Commands:

1) For finding the git version

- `git --version`

2) Tell Git who you are :

- `git config --global user.name "Arun Motoori"`
- `git config --global user.email "arun@qafox.com"`
- `git config --list` (All the configurations with their values will be listed)

3) Get help with any git command

- `git config --help`
- `git add --help`
- `git commit --help`

4) Create a new local repository and initialize it to a git repository:

- `git init`

3) Checkout a repository

- `git clone clonesshurl`

4) Find the local code changes:

- `git status`

5) Understand where you are ?

- Are you at working directory, Staging area or remote repository

6) Add Files to staging area

- `git add abc.txt`
- `git add .`
- `git add -A`

7) Reset the added files back to working directory

- `git reset`
- `git reset abc.txt`

6) Commit the changes

- `git commit -m "first commit"`

7) Adding the github repository location

- `git remote add origin clonesshurl`

8) Generating the ssh key (Only required to establish secured connection between local machine and github)

- `ssh-keygen -t rsa`

7) Validating the proper establishment of secured connection

- `ssh -T git@github.com`

8) Send changes to the master branch

- `git push origin master`

9) Update the local code with the changes done at remote repository

- `git pull`
- `git pull origin master`

10) For forcefully reverting the previous commit

- `git reset --hard`
- `git pull`

11) For clearing the changes in local

- `git stash`

14) Tracking the commits

- `git log`

15) View information about remote repository

- `git remote -v`

16) Finding the changes made

- `git diff`

17) To know different branches in the current repository

- `git branch`

- `git branch -a` (Shows all the local and remote branches)

18) Creating a new branch

- `git branch branchname`

19) Changing the branch

- `git checkout branchname`

Do some changes to local code and push to new branch
Check the changes in the github.com

20) Switch back to master branch

- `git checkout master`

21) To find whether all the branches have been merged to master

- `git branch --merged`

22) Merge the branch code to master branch

- `git merge branchname`

Now push the merged changes to the remote master branch

- `git push origin master`

Now recheck whether all the branches have been merged to master

- `git branch --merged`

23) Delete the branch

- `git branch -d branchname`

Now check the available branches

- `git branch`
- `git branch -a`

24) Delete the branch from remote repository

- `git push origin --delete branchname`

