***Git – Basics***

Git is basically a version control system which allows the developers to do the check of their code and maintain a complete history of their work.

Advantages of Version Control System:

* Allows developers to maintain a track of the code.
* Will not give a chance to overwrite others checked in code.
* Has got the history of every check-in.

Let’s see how we install Git Client for this lets use UBUNTU os for reference,

* sudo apt-get update
* sudo apt-get install git-all
* git --version 🡪 this gives the installed version

Git provide git config file for editing the configurations by setting the configuration variables. These global configurations are stored in ***.gitconfig*** file. to set these configuration values as global, we need to add the --global option and if not with this --global option then that file is considered for the current git repo only.

***/etc/gitconfig*** file, provides the config for every user and repo on the system. to set the username and useremailid under the global permissions then we need to see the below mentioned code.

* git config --global user.name “DevOps Forum”
* git config --global user.email “Devopsforum@gmail.com”
* git config --list 🡪 this will result the settings of the system present in the system and this you can see the below sample.
* git config --global –e 🡪 this command will open the config file in the nano editor so that you can edit the user details fed on the config file.

***Git Life Cycle:***

Git has different stage in making the file to reach back to the repo by getting it into the local machine and sending back by doing the modifications as per the code requirements to the repo.

* step1: we clone the Git repo as a working copy into our local machine.
* step2: here we do the modifications of the code files like add/edit and then make it ready for committing back again into the repo.
* step3: here we do an update of the file before doing a commit with the new changes. With this update of the file it will reflect the new incoming changes to the file. (i.e. updates from other developers.).
* step4: we need compare the incoming changes with our code changes so that there will not be any merge issues before committing.
* step5: final step you will commit the code files without any merge issues and then do the final push to the repo.

Now let’s start the task on the machine using GIT and hope you all installed ***git*** in your machines. so lets begin the show,

* first lets create a user group and under this user group lets create a user as well, for this use the below mentioned commands,
  + - * + groupadd DevOps 🡪 this will add new group
        + useradd --G dev --d /home/DevOpsuser --m --s /bit/bash DevOpsuser 🡪 this will create a user DevOpsuser
        + passwd DevOpsuser 🡪 this will allow to give the user password.
        + mkdir forum.git 🡪 this will crate a directory named forum
        + cd forum.git/
        + git --bare init 🡪 initializes empty git repo in /home/DevOpsuser-m/forum.git/ (this steps helps us to create the repo).
      * Till here we have seen how to create a user group, add the user to the group, and initialize a bare-repo. Now lets see how to establish the connection to make the user authentication.
      * ssh-keygen 🡪 use this command to generate the key and .ssh directory.
      * Now let’s use the generated public key across the machines(i.e. user machines) using the below command,

ssh-copy-id -i ~/.ssh/id\_rsa.pub DevOpsuser@git-server.com

**Show the result**

We need to do the above copy step for all users on the respective machine repo

now user1 has created a file and need to push it into the server for this we need to follow the below steps.

**mkdir user1\_repo**

**cd user1\_repo**

**git init**

**echo ‘update: add the code for the codefile’ > CODEFILE**

**git status –s**

**git add .**

**git status –s**

**git commit –m ‘Initial commit’**

**git remote add origin https://github.com/<repository>/<proj-name>.git 🡪** this will add the repo that needs to udpate

**git push –u origin master 🡪** this will push the files to GIT gui

**git log 🡪** this will give the log file showing all changes list of the commits, additions, push, etc.