**GITHUB**

GitHub is the single largest host for Git repositories, and is the central point of collaboration for millions of developers and projects. A large percentage of all Git repositories are hosted on GitHub, and many open-source projects use it for Git hosting, issue tracking, code review, and other things.

**GIT INSTALLATION**

* $ sudo yum install git

**GIT CONFIGURATION**

* $ git config --global user.email "FVENKAT007@GMAIL.COM"
* $ git config --global user.name "VENKATESH"
* $ git config --list

Default editor

* $ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noplugin"
* $ git config --system core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noplugin"
* P4MERGE INSTALL
* Download P4MERGE tool from <https://www.perforce.com/downloads/visual-merge-tool>
* $ git config –global diff.tool p4merge
* $ git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"
* $ git config --global difftool.prompt false
* $ git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"
* $ git config --global mergetool.prompt false
* Alias
* $ git config --global alias.**hist** "log --oneline --graph --decorate --all"
* $ git **hist**

**Git Basics**

* $ git init
* $ git add <filename>
* $ git commit -m <message>
* $ git status
* $ git log
* $ git show
* $ git commit -am <message>
* $ git restore <filename> ……………undo in working directory
* $ git restore --staged <filename> ……….undo stage file
* $ git log --oneline --graph
* $ git log --oneline --graph --decorate --all

**Git Advanced**

$ git diff <hash id> <HEAD>

$ git difftool <hashID> <HEAD>

**Branching & Merging**

HEAD >>>>> points to last commit of current branch >> can be moved

$ git branch

$ git branch <branch name>

**GITHUB**

**Account Setup and Configuration**

**1] Sign up github**

**2] Creating a Github Repo**

**3] Linking to our Github Repo**

**…create a new repository on the command line**

* echo "# dummy" >> README.md
* git init
* git add README.md
* git commit -m "first commit"
* git branch -M main
* git remote add origin git@github.com:fvenkat/dummy.git
* git push -u origin main

### …or push an existing repository from the command line

* git remote add origin git@github.com:fvenkat/dummy.git
* git branch -M main
* git push -u origin main

**Authentication SSH vs HTTPS**

* **We have ssh & https authentication in order to push the changes up. Https is perfectly fine , but there are few drawbacks. So whener you push the changes , Github will prompt me for username & password. For private repo it may prompt you each time you need to do any network-based communication, including pulling or fetching. SSH is little more secure & it convenient as well.**

**SSH Authentication**

Key generation in local machine

* $cd ~
* $mkdir .ssh
* $cd .ssh/
* $ssh-keygen -t rsa -C [fvenkat007@gmail.com](mailto:fvenkat007@gmail.com)
* $cat id\_rsa.pub >>>>>copy the public key

**Github**

* Profile settings >> SSH & GPG Keys >> New SSH key >> paste the public & add

**SSH Verfication**

* $ssh -T [git@github.com](mailto:git@github.com)

**Create a Local copy with clone**

* $git clone [git@github.com:fvenkat/github\_test.git](mailto:git@github.com:fvenkat/github_test.git)
* $git clone <https://github.com/fvenkat/github_test.git>

**Seeding the repo with sample content**

* Download bootstrap content from <http://www.initializr.com/> & move it to repo
* $git add .
* $git commit -m “added sample content”
* $git push origin main

**Updating Remote repo name & Remote References**

* Do rename the repo name in Github & then you need to modify the refe url in git local.
* $git remote set-url [git@github.com:fvenkat/github\_Demo\_house.git](mailto:git@github.com:fvenkat/github_Demo_house.git)

Closing the issue with commit

$git commit -am “completed the issue, close #4”

$git push

Using mentions with issues

In comment   
 #3