**GitHub Commands**

**Git Client Installation**

**Install git:** <https://git-scm.com/download>

**Desktop GUI version:** https://desktop.github.com

**Help:** git --help

**Particular command help page (e.g. to find** add **command details):** git add --help

**Version check**: git--version

**Where git is installed:** which git

**Initial Configuration**

**Checking all local git configuration:** git config --list

**Configure a user name:** git config --global user.name <@user\_name> (e.g. git config --global user.name "Mo Faruqe")

**Configure a user email:** git config --global user.email <@user\_email> (e.g. git config --global user.email mfaruqe@example.com)

**To read a particular config value:** git config <@value\_to\_be\_read> (e.g. To find the user name configured for git, run: git config user.name)

**To have output color coded for easier read:** git config --global color.ui true

**To ignore file mode:** git config --global core.filemode false

**To set a particular command line editor:** git config --global core.editor <@editor\_name>(e.g. to set “Nano” as the editor: git config --global core.editor nano)

**To deal with line ends do this during initial configuration:** Mac/Linux: git config --global core.autocrlf input, Windows: git config --global core.autocrlf true

**Check your local .ssh directory:**

Mac:

cd ~/.ssh

ls

Windows:

cd %home%\.ssh

dir \*.\*

**If .ssh directory doesn’t exist create it and navigate to it:**

Mac:

mkdir ~/.ssh

cd ~/.ssh

Windows:

mkdir %home%\.ssh

cd %home%\.ssh

**Now generate a ssh key pair from inside the directory:**

ssh-keygen -t rsa -C "your\_email@example.com"

(Give a filename or keep default “id\_rsa” and passphrase is optional – you can keep it empty or put a password to make it more secure.)

**Copy the SSH public key from your local system:**

Mac: cat ssh\_public\_key\_file\_name (e.g. cat id\_rsa.pub) and select & copy the contents.

Windows: Open the .pub file with a text editor and copy the contents.

It will start with “ssh-rsa” and ends with your email, something like “ssh-rsa xxxx…xxxx your\_email@example.com”.

**Add the SSH key in your GitHub account:**

Go to “Settings” (from down arrow key near your profile picture)>>”SSH and GPS Keys”>>”New SSH Key”: Enter a “Title” related to your particular machine name e.g. “My Home MacBook Pro” and paste the SSH public key into the “Key” text box. Press “Add SSH Key” and it will be added to your account. All the previous keys, if there is any, and the just added key will be shown in this area.

**Test from the terminal window in your local system:**

ssh -T git@github.com

If successful you will get a message like “Hi your\_name! You've successfully authenticated, but GitHub does not provide shell access.”

**Create and Manage Repository**

**Create a new repository in your GitHub account and clone it to your local system:**

Go to “Your repositories” (from down arrow key near your profile picture). Click “New”, enter a “Repository name”, set “Public” or “Private”, don’t check “Initialize this repository with a README”. Finally click “Create repository”. Now copy the url from https or SSH in the next screen and clone it to your local system:

git clone <@repo\_url> (e.g. git clone <https://github.com/faruqem/hello-world>)

If you already have a SSH key established, you can also use:git clone git@github.com:faruqem/hello-world.git

**Create a local README file and push it to GitHub for testing:**

nano README.md (create the new README file using editor Nano)

git status

git add --all

git commit -m “first commit”

git push -u origin master

**Optionally without cloning, you can create a local repository and link it with the remote GitHub repository or push an existing repository to the remote repository:**

mkdir test

cd test

git init (initialize the directory for git)

ls -a (it will show a .init folder)

nano README.md

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:faruqem/hello-world.git

git push -u origin master

**Repository status check:** git status

**To replace your local repository contents with GitHub’s** (**Warning:** It will replace all your local code, only run if you are ok to lose any local changes):

git fetch --all

git reset --hard origin/master

(Replace “origin” with your remote’s name if needed)