1. GIT Setup
2. Using GIT, when creating a new repo
3. Using GIT when cloning a project to your local system.
4. Using GIT, in everyday use.

1. GIT Setup

1. Git is already installed on your system. Confirm it by typing the below command

**$ git --version**

Else install git - **$ sudo apt-get -y install git-core**

1. Create an account on github.com

gaurav-netsol / gaurav.gargi@netsolutionsindia.com

1. Git configuration

**$ git config --global color.ui true**

**$ git config --global user.name "Gaurav Gargi"**

**$ git config --global user.email "gaurav.gargi@netsolutionsindia.com"**

1. Generate SSH Key

**$ ssh-keygen -t rsa -C "**[**gaurav.gargi@netsolutionsindia.com**](mailto:gaurav.gargi@netsolutionsindia.com)**"**

**$ Enter the file for saving the key ( /home/gaurav/.ssh/id\_rsa ) :**

**Passphrase: Netsolutionsindia**

**Re-enter passphrase: Netsolutionsindia**

1. Add the key to SSH agent

**$ eval "$(ssh-agent -s)"**

**$ ssh-add ~/.ssh/id\_rsa**

1. Copy the key to clipboard

**$ sudo apt-get install xclip**

**$ xclip -sel clip < ~/.ssh/id\_rsa.pub**

1. Goto<https://github.com/settings/ssh> and click to add a new SSH key

ctrl+V in the 'key' text-area, and save the key.

1. Test the setup-

**$ ssh -T git@github.com**

References -

<https://help.github.com/articles/generating-ssh-keys/#platform-linux>

2. Using GIT, when setting up a repository

Create a new repo at<https://github.com/new>

After creating the repo, select the SSH link

e.g. git@github.com:gaurav-netsol/hello\_world.git

**$ cd *my\_rails\_app***

**$ git init**

**$ git add .**

**$ git commit -m “Vanilla app”**

**$ git remote add origin git@github.com:gaurav-netsol/hello\_world.git**

**$ git push origin master**

If it asks for the password, enter the Passphrase entered during GIT setup.

3. Using GIT, to clone an existing repo on your local system

**$ cd workspace**

**$ git clone** [***git@github.com***](mailto:git@github.com)***:gaurav-netsol/hello\_world.git***

**$ cd *my\_rails\_app***

**$ bundle install**

**$ rake db:migrate**

4. Using GIT, in everyday use

At the beginning of each day, verify that you are on master branch

**$ git checkout master**

Do a latest pull from the master branch.

**$ git pull origin master**

Move to your working branch for the day

**$ rake db:migrate**

**$ git checkout mybranch**

Or create a new branch

**$ git checkout -b mybranch**

Merge master branch into your branch

**$ git merge master**

Resolve manual conflicts (you are in mybranch), if any, by appropriately modifying the conflicted files.

Start working on your branch!

**$ git status** #To see what changes have been done by you since the previous commit.

**$ git add <folders>** #To add folders for committing

**$ git add <files>** #To add files for committing

**$ git diff --cached** #To view the changes done

**$ git commit -m “Message for commit”** #To commit the changes

End of Day merging -

**$ git checkout master** #Move to master branch

**$ git pull origin master** #Get latest changes of master

**$ rake db:migrate**

**$ git checkout mybranch** #Move to your working branch

**$ git merge master** #Merge master branch into mybranch

Resolve manual conflicts (you are in mybranch), if any, by appropriately modifying the conflicted files.

**$ git add <conflicted\_files>**

**$ git diff --cached** #To view the changes done

**$ git commit -m “Resolving conflicts”**

**$ git push origin mybranch** #Push changes of mybranch on git

**$ git checkout master** #Move to master branch

**$ git merge mybranch** #Merge mybranch into master branch

Resolve manual conflicts (you are in master branch), if any, by appropriately modifying the conflicted files.