Setup Git Hooks

This document describes how to configure the hooks in local and server repository for preventing when user push the unwanted commits to server.

* First, We need to fetch the Git repository of your project.

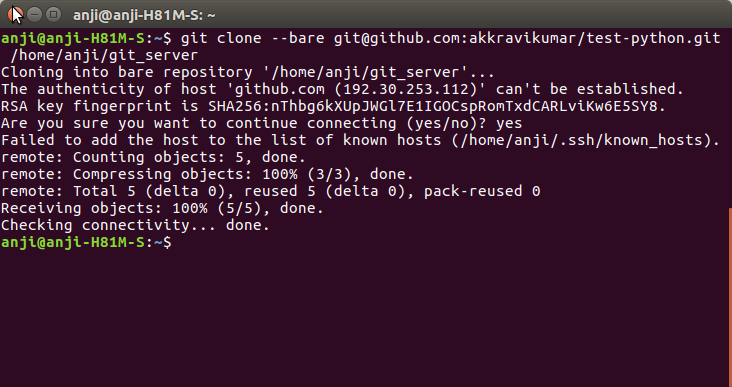
**In Your Server,**

* Make sure the server ssh key is added in Git else it blocks the git to clone..
* If not we need to add ssh key in Git Settings -> SSH and GPG Keys -> Click New SSH Key -> Paste your ssh key --> and click Add ssh key button.
* Create a bare repo in a server anywhere in the location
* Clone with ssh url, open Terminal, and type as follows:

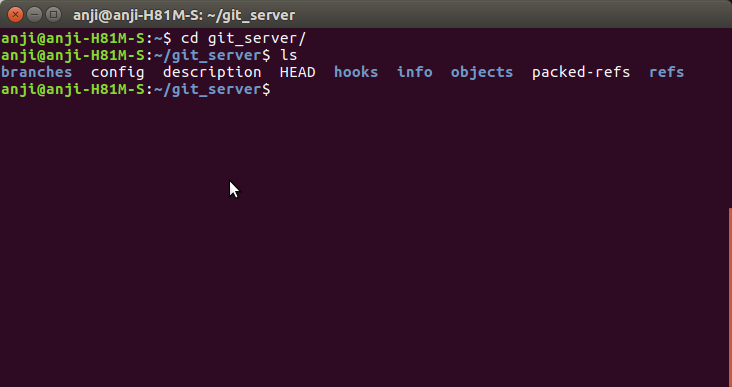
> git clone --bare git@github/user/my-project /home/user/server/my-project

If clone from same location, then we need to specify (.) dot as follows

> git clone --bare git@github/user/my-project .

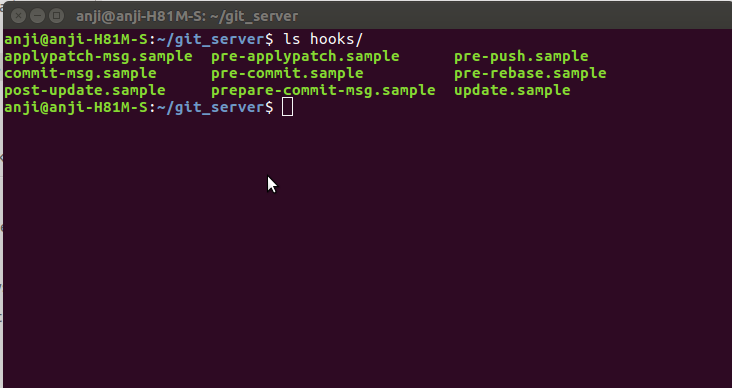


When executing above commends it has cloned git repository files as follows



### Installing Hooks

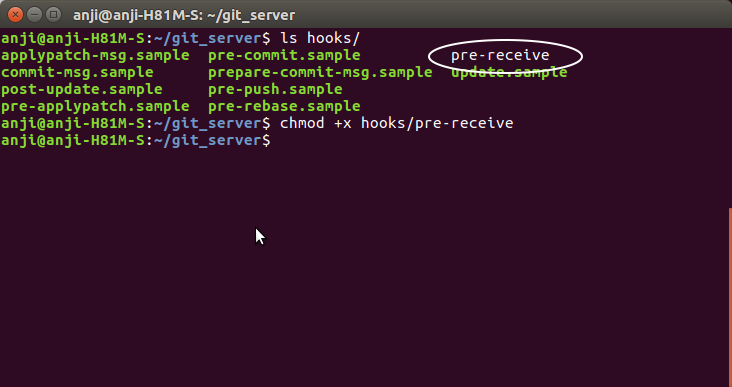
Hooks reside in the .git/hooks directory of every Git repository. Git automatically populates this directory with example scripts when you initialize a repository. If you take a look inside .git/hooks, you’ll find the following files:



These represent most of the available hooks, but the .sample extension prevents them from executing by default. To “install” a hook, all you have to do is remove the .sample extension. Or, if you’re writing a new script from scratch, you can simply add a new file matching one of the above filenames, minus the .sample extension.

* We should add pre-receive file inside the hooks folder.
* Make permission to pre-receive file as follows

> chmod +x hooks/pre-receive

****

Hooks need to be executable, so you may need to change the file permissions of the script if you’re creating it from scratch. For example, to make sure that prepare-commit-msg is executable.

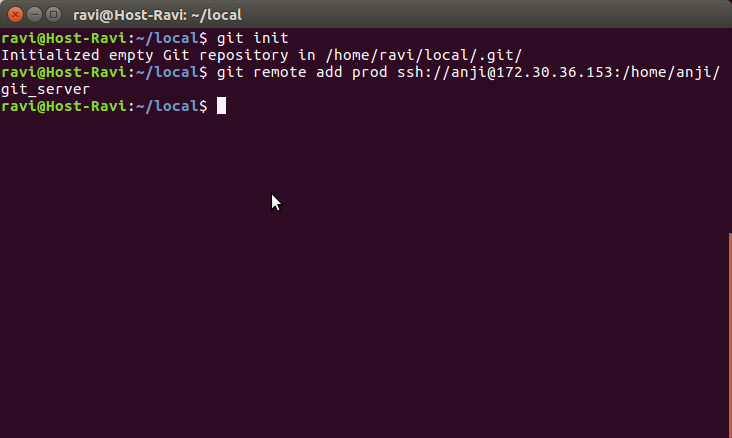
**In Your local machine**

* On your local machine, add a new remote to deploy on this freshly new repository in any location.
* Make sure git is initialized in the current folder.

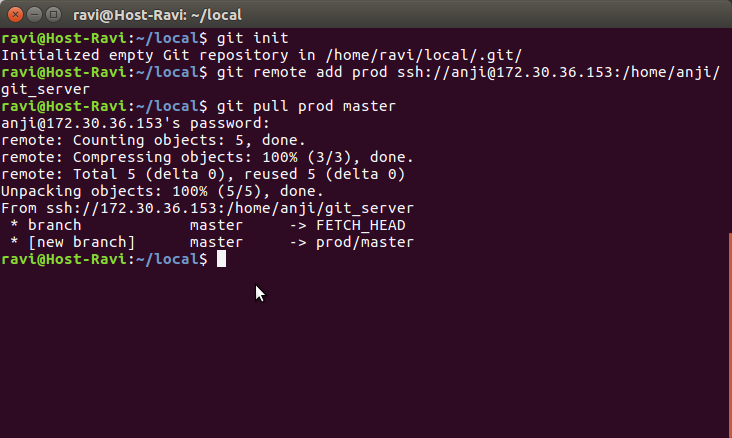
> git init

> git remote add prod ssh:[//me@my\_server](mailto://me@my_server):port/home/user/local

**Note**: Port 22 is default. So not need to mention the port number as below image.

* + Pull the repository from server as follow

> git pull prod master

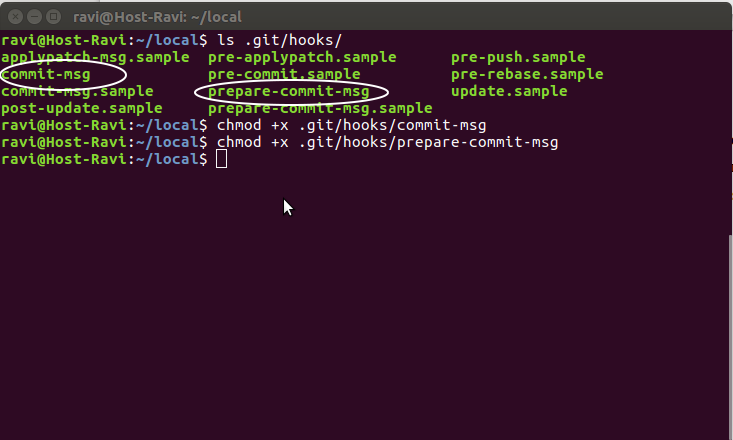


This is not the default configured remote for the current branch, we must specify a branch on the command line.

* We need to add commit-msg and prepare-commit-msg files inside the .git/hooks folder for validating messages.
* Make permission to pre-receive file as follows

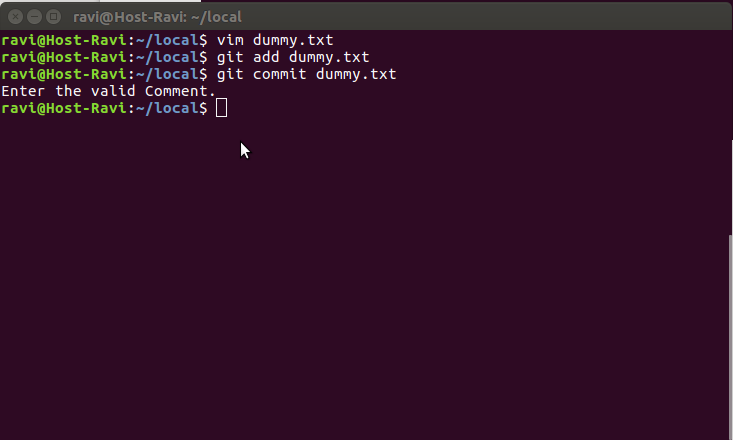
> chmod +x hooks/commit-msg

> chmod +x hooks/prepare-commit-msg



Now successfully configured the git hook both local and server.

The following screens are how it works



We must provide the valid Bug Id followed by comment. For instance:

“Bug Id:<bug\_id>-your\_comment\_here”

