

[HOME](#)[HELP FORUM](#)[SUBSCRIBE!](#)[VIDEOS](#)[WRITE FOR US](#)[PRIVACY](#)[TERMS](#)[Q](#)

Git • Ubuntu

# Configure Git Server with SSH on Ubuntu

4 months ago • by Shahriar Shovon

If you have a small number of team members working on some projects, then you can setup a Git server via SSH on your office and work on projects as a team very easily. You don't have to use GitHub or any other services in that case. SSH based Git server is really easy to setup and use. In this article, I am going to show you how to configure a Git server with SSH on Ubuntu and how to use it. So, let's get started.

## Configuring Git Server:

In this section, I am going to show you how to configure an Ubuntu server as a SSH accessible Git server.

First, update the APT package repository cache with the following command:

```
$ sudo apt update
```

```
shovon@linuxhint-s20:~$ sudo apt update
```

The APT package repository cache should be updated.

```
shovon@linuxhint-s20:~$ sudo apt update
[sudo] password for shovon:
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
155 packages can be upgraded. Run 'apt list --upgradable' to see them.
shovon@linuxhint-s20:~$ _
```



Now, install OpenSSH server and Git with the following command:

```
$ sudo apt install openssh-server git
```

```
showon@linuxhint-s20:~$ sudo apt install openssh-server git _
```

Now, press **Y** and then press **<Enter>** to confirm the installation.

```
showon@linuxhint-s20:~$ sudo apt install openssh-server git
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 154 not upgraded.
Need to get 3,907 kB of archives.
After this operation, 32.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

OpenSSH server and Git should be installed.

```
showon@linuxhint-s20:~$ sudo apt install openssh-server git
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 154 not upgraded.
Need to get 3,907 kB of archives.
After this operation, 32.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.4 [3,907 kB]
Fetched 3,907 kB in 0s (13.9 MB/s)
Selecting previously unselected package git.
(Reading database ... 66030 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.4_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.4) ...
Setting up git (1:2.17.1-1ubuntu0.4) ...
showon@linuxhint-s20:~$
```

Now, create a new user **git** with the following command:

```
$ sudo useradd --create-home --shell /bin/bash git
```

```
showon@linuxhint-s20:~$ sudo useradd --create-home --shell /bin/bash git
```



All the Git repositories will be saved in the home directory of the **git** user **/home/git**.

Now, login as the **git** user with the following command:

```
$ sudo su - git
```

```
shovan@linuxhint-s20:~$ sudo su - git
git@linuxhint-s20:~$
```

Now, create a new directory **.ssh** with the following command:

```
$ mkdir .ssh
```

```
git@linuxhint-s20:~$ mkdir .ssh _
```

Now, allow only **git** user to have read, write, exec permissions on the directory **.ssh/** as follows:

```
$ chmod 700 .ssh/
```

```
git@linuxhint-s20:~$ chmod 700 .ssh/ _
```

As you can see, the **git** user only has read (r), write (w), execute (x) permissions on the **.ssh/** directory.

```
$ ls -ld .ssh/
```

```
git@linuxhint-s20:~$ ls -ld .ssh/
drwx----- 2 git git 4096 Sep  9 16:32 .ssh/
git@linuxhint-s20:~$ _
```

Now, create a new empty file **.ssh/authorized\_keys** as follows:

```
$ touch .ssh/authorized_keys
```

```
git@linuxhint-s20:~$ touch .ssh/authorized_keys _
```

Only allow read and write to the file from the **git** user as follows:

```
$ chmod 600 .ssh/authorized_keys
```

```
git@linuxhint-s20:~$ chmod 600 .ssh/authorized_keys _
```

As you can see, only the **git** user has read (r) and write (w) permissions to the 1 ^

## **.ssh/authorized\_keys.**

```
git@linuxhint-s20:~$ ls -ld .ssh/authorized_keys
-rw----- 1 git git 0 Sep  9 16:34 .ssh/authorized_keys
git@linuxhint-s20:~$ _
```

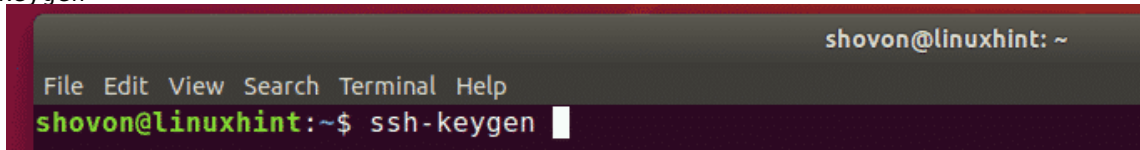
In the **.ssh/authorized\_keys** file, you have to add the public key of the users whom you want to access the Git repositories on the Git server.

## **Adding Client Public Key to the Git Server:**

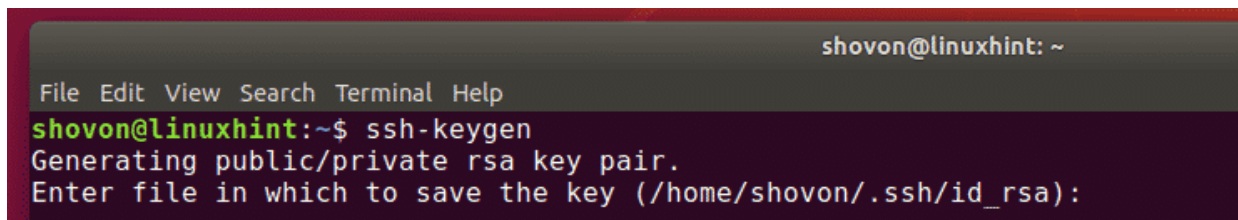
To access the Git repositories on the Git server, the client must add his/her public key to the Git server.

The client can generate a public-private key pair as follows:

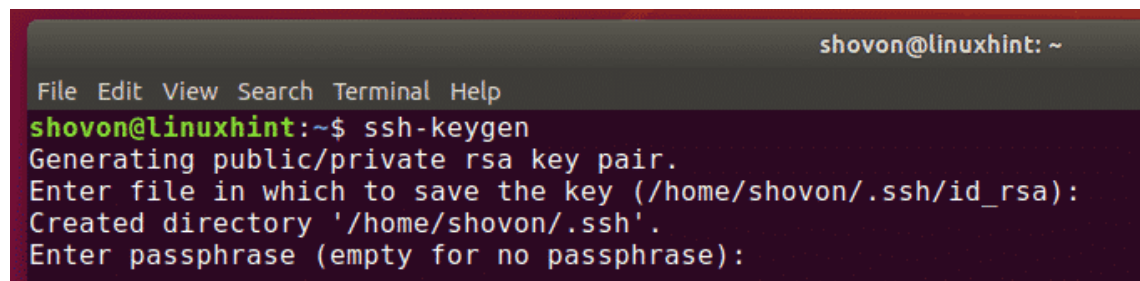
```
$ ssh-keygen
```

A terminal window titled 'shovon@linuxhint: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'shovon@linuxhint:~\$' and the command 'ssh-keygen' is entered, with a cursor at the end.

Press **<Enter>**.

A terminal window titled 'shovon@linuxhint: ~' with a menu bar. The prompt is 'shovon@linuxhint:~\$' and the command 'ssh-keygen' has been executed. The output is: 'Generating public/private rsa key pair. Enter file in which to save the key (/home/shovon/.ssh/id\_rsa):'. The cursor is at the end of the prompt.

Press **<Enter>**.

A terminal window titled 'shovon@linuxhint: ~' with a menu bar. The prompt is 'shovon@linuxhint:~\$' and the command 'ssh-keygen' has been executed. The output is: 'Generating public/private rsa key pair. Enter file in which to save the key (/home/shovon/.ssh/id\_rsa): Created directory \'/home/shovon/.ssh\'. Enter passphrase (empty for no passphrase):'. The cursor is at the end of the prompt.

Press **<Enter>**.

```
shovon@linuxhint: ~  
File Edit View Search Terminal Help  
shovon@linuxhint:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/shovon/.ssh/id_rsa):  
Created directory '/home/shovon/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Press **<Enter>**.

[illegible]

Now, the client can find his/her public key as follows:

```
$ cat ~/.ssh/id_rsa.pub
```

```
shovon@linuxhint: ~  
File Edit View Search Terminal Help  
shovon@linuxhint:~$ cat ~/.ssh/id_rsa.pub
```

Client's public key should be printed. Now, the client can send this public key to the manager (who manages the Git server). The manager can then add the public key to the Git server. Then the client can access the Git server.

```
shovon@linuxhint: ~
File Edit View Search Terminal Help
shovon@linuxhint:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD1xlr...bEd2xED2ndWgx79UYzevB1gN94GsM+X6aJkozpfTY8eC
tpaxNo4ZBeHD3JkvesCD3P...COPTXSG681PCHzn1Kh4U+xpLiqEboDFU70/adAcWTEwRYDBeqn
gwN4+NYnZLDWA9S6YdyuNyjbRQBkg88Y...ZAFN1X0VBJ05WWo2Fm+BVk0Tf0g8u26NyNxwMuUD/PjvHcZf0DrVdsNJ2RE
wRxE6bPI/lrB0TiCeik0gWl9hm5oytoFuv+BPH9IuvbLcfzquZi2KtdJ shovon@linuxhint
shovon@linuxhint:~$
```

Let's say, the client sent his/her public key to the Git server manager. The manager uploaded the public key to **/tmp/shovon-key.pub** file on the Git server.

```
git@linuxhint-s20:~$ cat /tmp/shovon-key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD1xlr...bEd2xED2ndWgx79UYzevB1gN94GsM+X6aJko
zpfTY8eCtpaxNo4ZBeHD3JkvesCD3P...COPTXSG681PCHzn1Kh4U+xpLiqEboDFU70
/adAcWTEwRYDBeqn/gwN4+NYnZLDWA9S6YdyuNyjbRQBkg88Y...ZAFN1X0VBJ05WWo2Fm+BVk0Tf0g8u26NyNx
wMuUD/PjvHcZf0DrVdsNJ2REwRxE6bPI/lrB0TiCeik0gWl9hm5oytoFuv+BPH9IuvbLcfzquZi2KtdJ shovon@linuxhint
git@linuxhint-s20:~$
```

Now, the Git server manager can add the public key of the client as follows:

```
$ cat /tmp/shovon-key.pub >> ~/.ssh/authorized_keys
```

```
git@linuxhint-s20:~$ cat /tmp/shovon-key.pub >> ~/.ssh/authorized_keys _
```

Now, the **.ssh/authorized\_keys** file should have the public key of the client.

```
git@linuxhint-s20:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD1xlr...bEd2xED2ndWgx79UYzevB1gN94GsM+X6aJko
zpfTY8eCtpaxNo4ZBeHD3JkvesCD3P...COPTXSG681PCHzn1Kh4U+xpLiqEboDFU70
/adAcWTEwRYDBeqn/gwN4+NYnZLDWA9S6YdyuNyjbRQBkg88Y...ZAFN1X0VBJ05WWo2Fm+BVk0Tf0g8u26NyNx
wMuUD/PjvHcZf0DrVdsNJ2REwRxE6bPI/lrB0TiCeik0gWl9hm5oytoFuv+BPH9IuvbLcfzquZi2KtdJ shovon@linuxhint
git@linuxhint-s20:~$
```

## Creating Git Repositories on the Server:

The clients can't create new Git repositories on the server. The Git server manager must create a repository on the server. Then, the clients can clone, push/pull from the repository.

Now, create a new empty Git repository **testrepo** on the Git server as follows:

```
$ git init --bare testrepo
```

```
git@linuxhint-s20:~$ git init --bare testrepo
Initialized empty Git repository in /home/git/testrepo/
git@linuxhint-s20:~$
```



Now, the client only needs to know the IP address of the Git server in order to access the **testrepo** Git repository.

The Git server manager can find this information as follows:

```
$ ip a
```

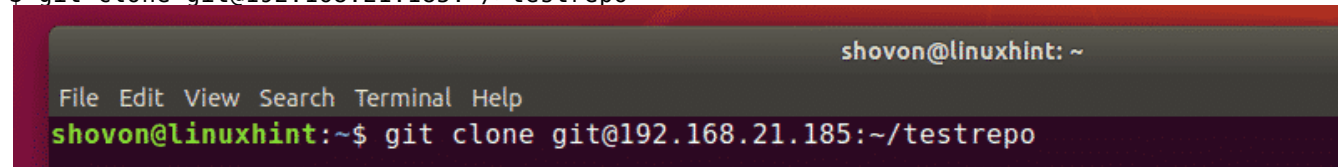
As you can see, the IP address of the Git server is **192.168.21.185**. Now, the server manager can tell it to the clients who will be working on the project.

```
git@linuxhint-s20:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:37:ce:94 brd ff:ff:ff:ff:ff:ff
    inet 192.168.21.185/24 brd 192.168.21.255 scope global dynamic ens33
        valid_lft 1619sec preferred_lft 1619sec
    inet6 fe80::20c:29ff:fe37:ce94/64 scope link
        valid_lft forever preferred_lft forever
git@linuxhint-s20:~$ _
```

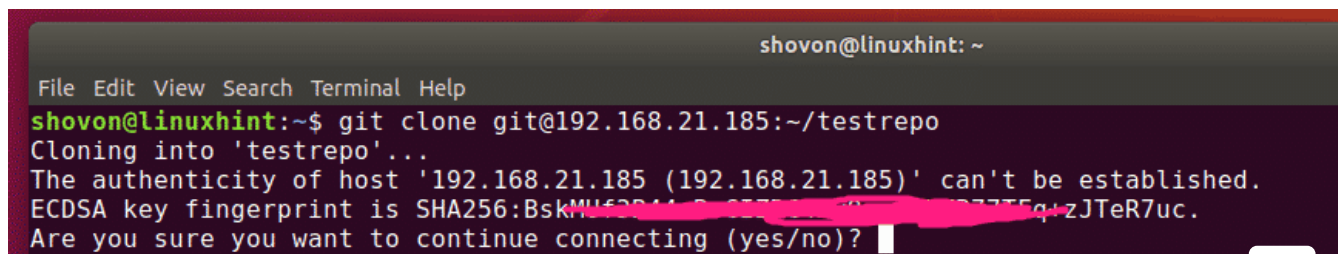
## Cloning Git Repository from the Server:

Once the client knows the IP address and the Git repository name, he/she can clone it to his/her computer as follows:

```
$ git clone git@192.168.21.185:~/>testrepo
```

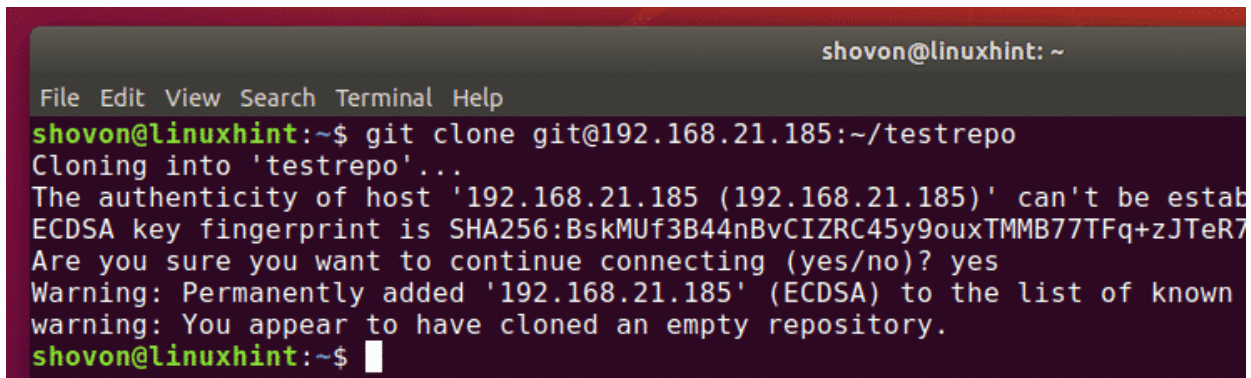
A terminal window titled 'shovon@linuxhint: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'shovon@linuxhint:~\$ git clone git@192.168.21.185:~/testrepo' is entered and highlighted in green.

Now, type in **yes** and press **<Enter>**. You will need to do this once, only the first time.

A terminal window titled 'shovon@linuxhint: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'shovon@linuxhint:~\$ git clone git@192.168.21.185:~/testrepo' is entered. The output shows 'Cloning into 'testrepo'...' followed by a warning: 'The authenticity of host '192.168.21.185 (192.168.21.185)' can't be established. ECDSA key fingerprint is SHA256:BskMUf32M1...'. The prompt 'Are you sure you want to continue connecting (yes/no)?' is shown with a cursor.

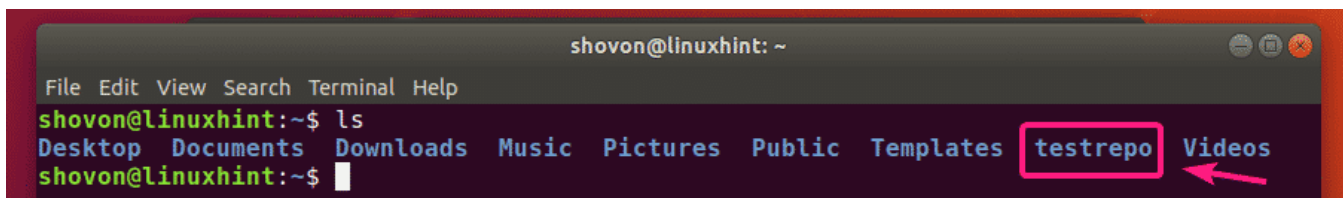


The **testrepo** Git repository should be cloned from the server.



```
shovon@linuxhint: ~  
File Edit View Search Terminal Help  
shovon@linuxhint:~$ git clone git@192.168.21.185:~/testrepo  
Cloning into 'testrepo'...  
The authenticity of host '192.168.21.185 (192.168.21.185)' can't be established  
ECDSA key fingerprint is SHA256:BskMUf3B44nBvCIZRC45y9ouxTMMB77TFq+zJTeR7  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.21.185' (ECDSA) to the list of known hosts  
warning: You appear to have cloned an empty repository.  
shovon@linuxhint:~$
```

A new directory **testrepo** should be created.

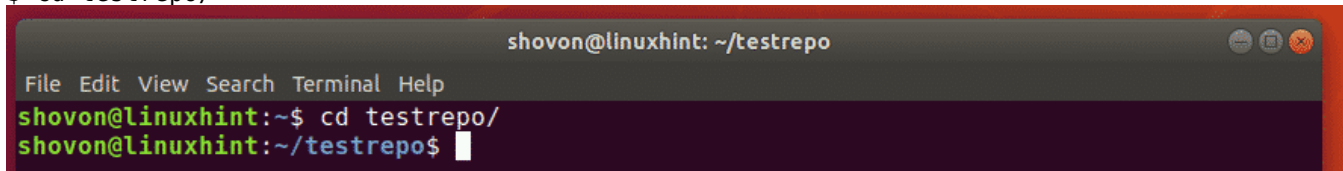


```
shovon@linuxhint: ~  
File Edit View Search Terminal Help  
shovon@linuxhint:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates testrepo Videos  
shovon@linuxhint:~$
```

## Making Changes and Pushing Changes to Git Server:

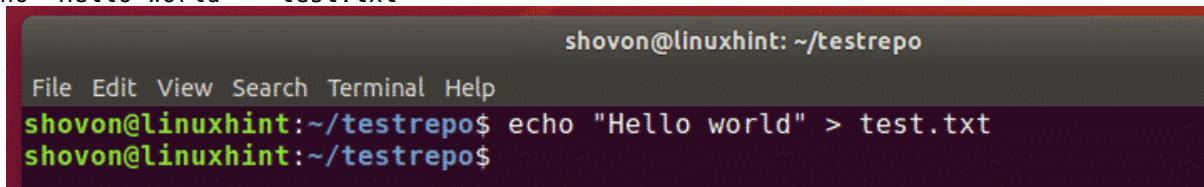
Now, the client can add commits to the **testrepo/** repository and push the changes to the Git server.

```
$ cd testrepo/
```



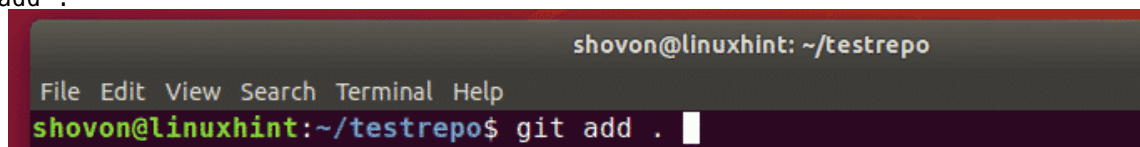
```
shovon@linuxhint: ~/testrepo  
File Edit View Search Terminal Help  
shovon@linuxhint:~$ cd testrepo/  
shovon@linuxhint:~/testrepo$
```

```
$ echo "Hello world" > test.txt
```



```
shovon@linuxhint: ~/testrepo  
File Edit View Search Terminal Help  
shovon@linuxhint:~/testrepo$ echo "Hello world" > test.txt  
shovon@linuxhint:~/testrepo$
```

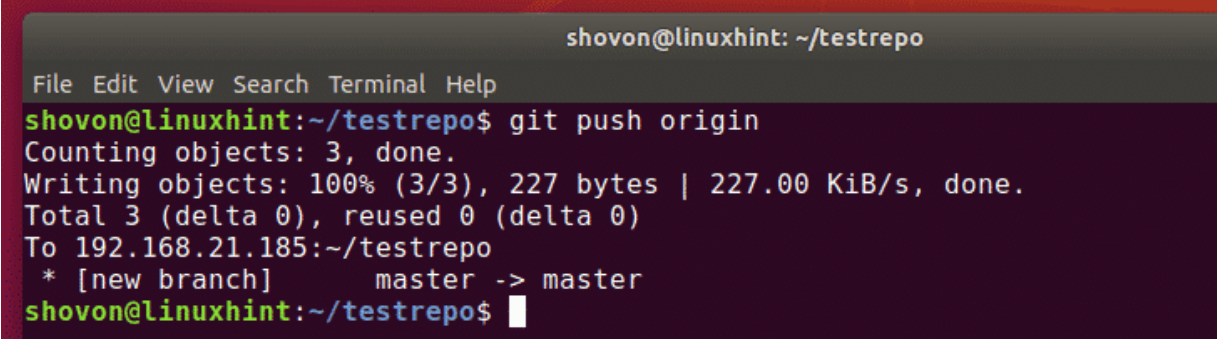
```
$ git add .
```



```
shovon@linuxhint: ~/testrepo  
File Edit View Search Terminal Help  
shovon@linuxhint:~/testrepo$ git add .
```



```
$ git commit -m 'initial commit'
[/cc[
<a href="https://linuxhint.com/wp-content/uploads,
[cc lang="bash"]
$ git push origin
```

A terminal window titled 'shovon@linuxhint: ~/testrepo' showing the output of a git push command. The output indicates that 3 objects were counted, 227 bytes were written, and the push was successful to the master branch.

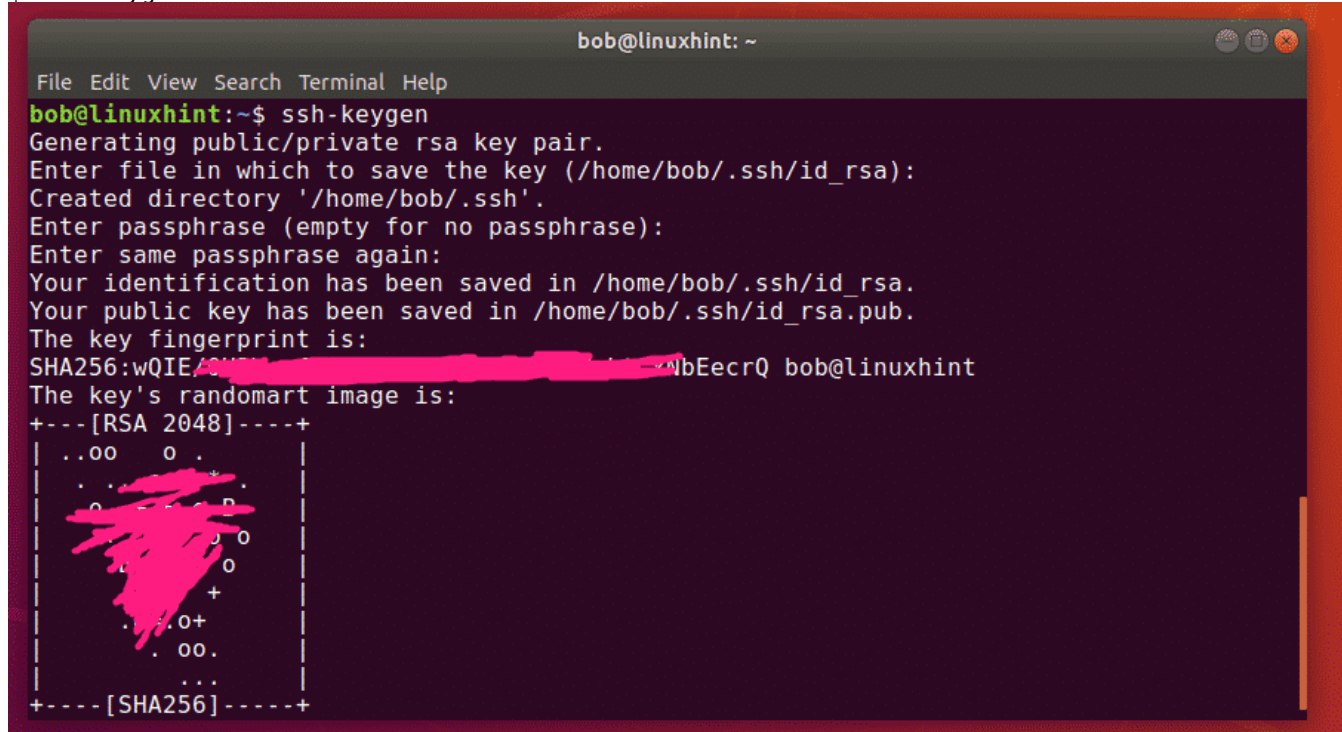
```
shovon@linuxhint: ~/testrepo
File Edit View Search Terminal Help
shovon@linuxhint:~/testrepo$ git push origin
Counting objects: 3, done.
Writing objects: 100% (3/3), 227 bytes | 227.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To 192.168.21.185:~/testrepo
* [new branch]      master -> master
shovon@linuxhint:~/testrepo$
```

## Adding a New Team Member:

Now, let's say, **bob** wants to contribute to the **testrepo** Git repository.

All he has to do is generate a SSH key pair and send the public key to the Git server manager.

```
$ ssh-keygen
```

A terminal window titled 'bob@linuxhint: ~' showing the output of the ssh-keygen command. It prompts for a file to save the key, a passphrase, and displays the key fingerprint and a randomart image.

```
bob@linuxhint: ~
File Edit View Search Terminal Help
bob@linuxhint:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bob/.ssh/id_rsa):
Created directory '/home/bob/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bob/.ssh/id_rsa.
Your public key has been saved in /home/bob/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:wQIE/...NbEecrQ bob@linuxhint
The key's randomart image is:
+---[RSA 2048]---+
| ..00 0 .
| . . . .
| . . . .
| . . . .
| . . . .
| . . . .
| . . . .
| . . . .
| . . . .
| . . . .
+---[SHA256]-----+
```

Once the Git server manager has the public key of **bob**, he can upload it to the Git server and add it to the **.ssh/authorized\_keys** file as follows:

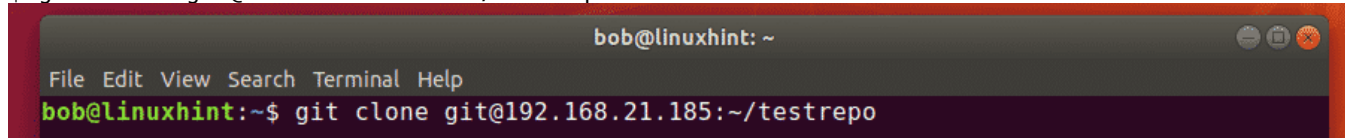


```
$ cat /tmp/bob-key.pub >> ~/.ssh/authorized_keys
```

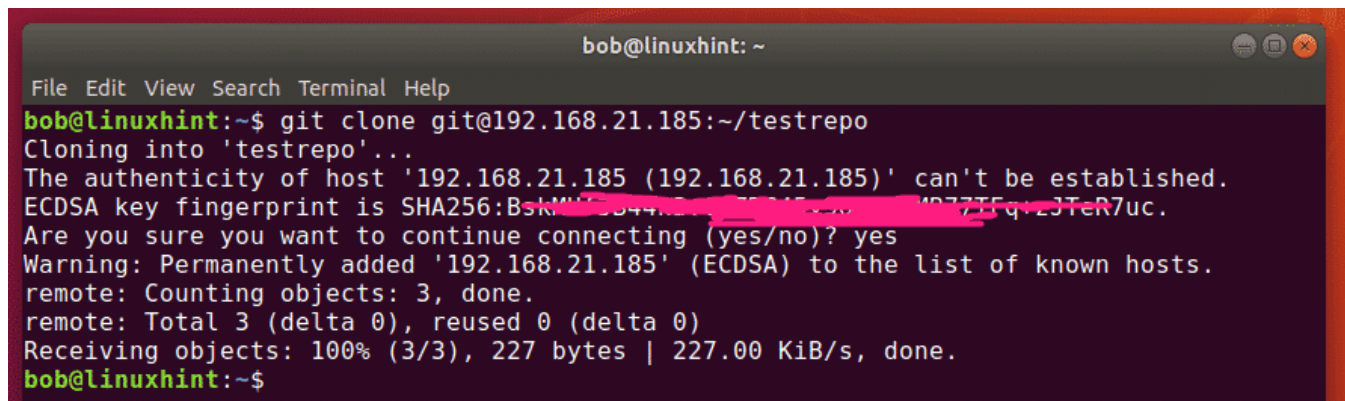
```
git@linuxhint-s20:~$ cat /tmp/bob-key.pub >> ~/.ssh/authorized_keys
```

Now, **bob** can clone the **testrepo** Git repository from the server as follows:

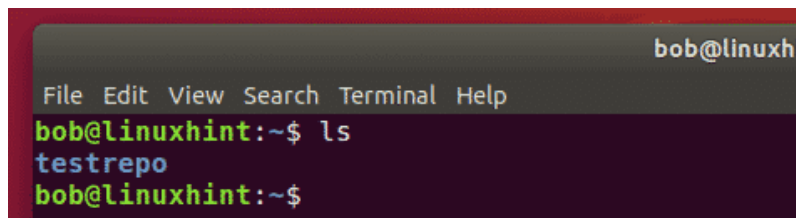
```
$ git clone git@192.168.21.185:~/testrepo
```

A terminal window titled 'bob@linuxhint: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'git clone git@192.168.21.185:~/testrepo' has been executed successfully, and the prompt is now 'bob@linuxhint:~\$'.

**testrepo** should be cloned.

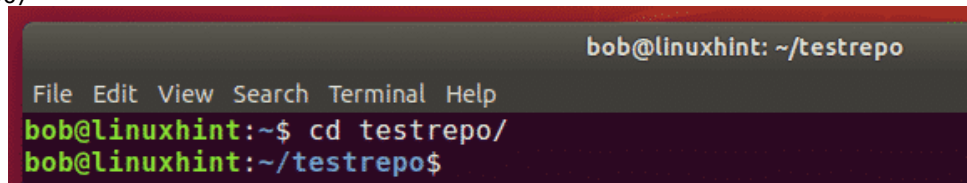
A terminal window titled 'bob@linuxhint: ~' showing the cloning process. The output includes: 'Cloning into 'testrepo'...', 'The authenticity of host '192.168.21.185 (192.168.21.185)' can't be established. ECDSA key fingerprint is SHA256:BsK... 4P77IFq+2JTeR7uc.', 'Are you sure you want to continue connecting (yes/no)? yes', 'Warning: Permanently added '192.168.21.185' (ECDSA) to the list of known hosts.', 'remote: Counting objects: 3, done.', 'remote: Total 3 (delta 0), reused 0 (delta 0)', 'Receiving objects: 100% (3/3), 227 bytes | 227.00 KiB/s, done.', and the final prompt 'bob@linuxhint:~\$'.

A new directory **testrepo** should be created in bob's computer.

A terminal window titled 'bob@linuxhi' showing the command 'ls' being executed. The output is 'testrepo', and the prompt is now 'bob@linuxhint:~\$'.

Now, bob can navigate to the Git repository as follows:

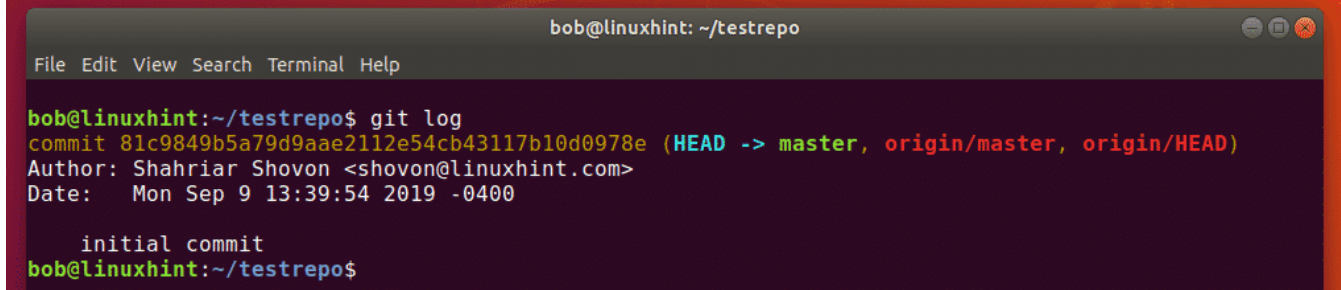
```
$ cd testrepo/
```

A terminal window titled 'bob@linuxhint: ~/testrepo' showing the command 'cd testrepo/' being executed. The output is 'bob@linuxhint:~/testrepo\$', and the prompt is now 'bob@linuxhint:~/testrepo\$'.

He should find some existing commits.



```
$ git log
```

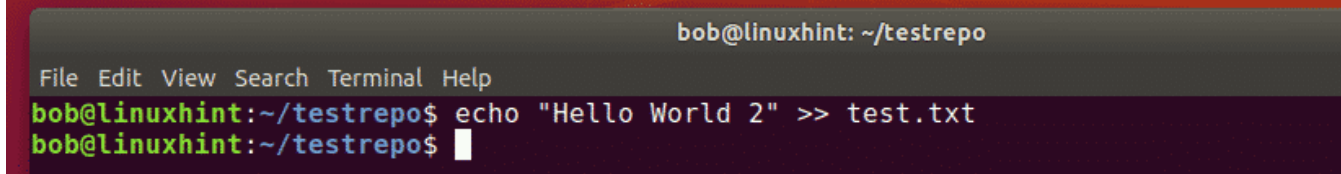
A terminal window titled 'bob@linuxhint: ~/testrepo' showing the output of the 'git log' command. The output displays a single commit with hash 81c9849b5a79d9aae2112e54cb43117b10d0978e, labeled as HEAD -> master, origin/master, and origin/HEAD. The author is Shahriar Shovon <shovon@linuxhint.com> and the date is Mon Sep 9 13:39:54 2019 -0400. The commit message is 'initial commit'.

```
commit 81c9849b5a79d9aae2112e54cb43117b10d0978e (HEAD -> master, origin/master, origin/HEAD)
Author: Shahriar Shovon <shovon@linuxhint.com>
Date:   Mon Sep 9 13:39:54 2019 -0400

    initial commit
bob@linuxhint:~/testrepo$
```

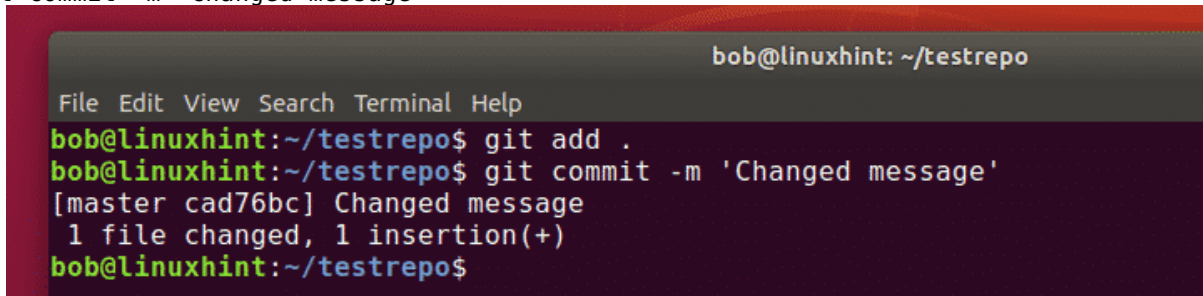
Now, **bob** can do his own work and commit it. Then, push the changes to the server.

```
$ echo "Hello World 2" >> test.txt
```

A terminal window titled 'bob@linuxhint: ~/testrepo' showing the execution of the 'echo' command to append text to a file.

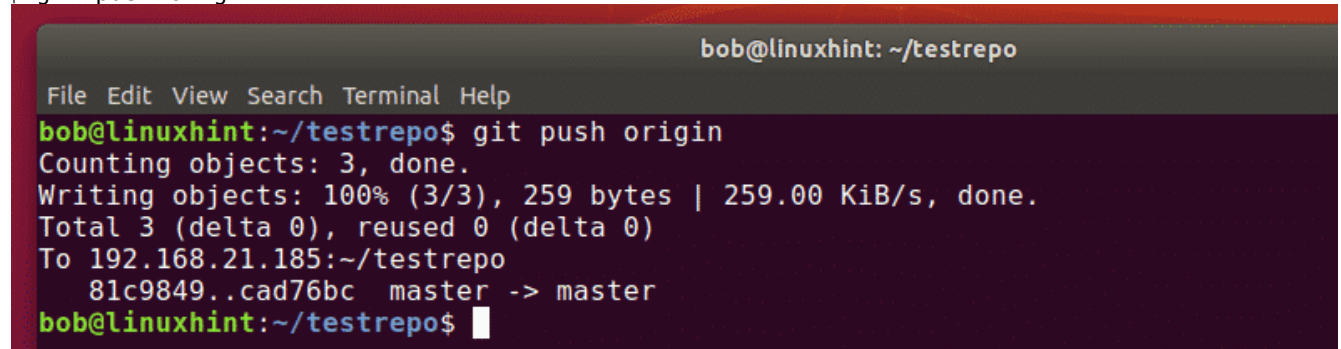
```
bob@linuxhint:~/testrepo$ echo "Hello World 2" >> test.txt
bob@linuxhint:~/testrepo$
```

```
$ git add .
$ git commit -m 'Changed message'
```

A terminal window titled 'bob@linuxhint: ~/testrepo' showing the execution of 'git add .' and 'git commit -m \'Changed message\''. The output shows the commit was successful, with 1 file changed and 1 insertion.

```
bob@linuxhint:~/testrepo$ git add .
bob@linuxhint:~/testrepo$ git commit -m 'Changed message'
[master cad76bc] Changed message
1 file changed, 1 insertion(+)
bob@linuxhint:~/testrepo$
```

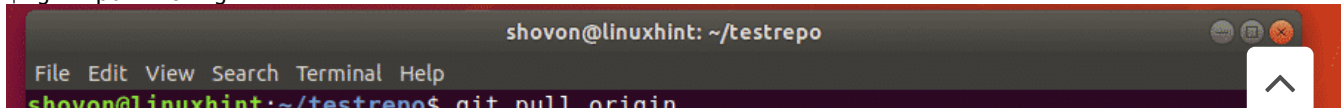
```
$ git push origin
```

A terminal window titled 'bob@linuxhint: ~/testrepo' showing the output of 'git push origin'. It indicates that 3 objects were counted, 259 bytes were written, and the push was successful to the master branch.

```
bob@linuxhint:~/testrepo$ git push origin
Counting objects: 3, done.
Writing objects: 100% (3/3), 259 bytes | 259.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To 192.168.21.185:~/testrepo
    81c9849..cad76bc  master -> master
bob@linuxhint:~/testrepo$
```

Now, other people working on the same repository can pull the changes as follows:

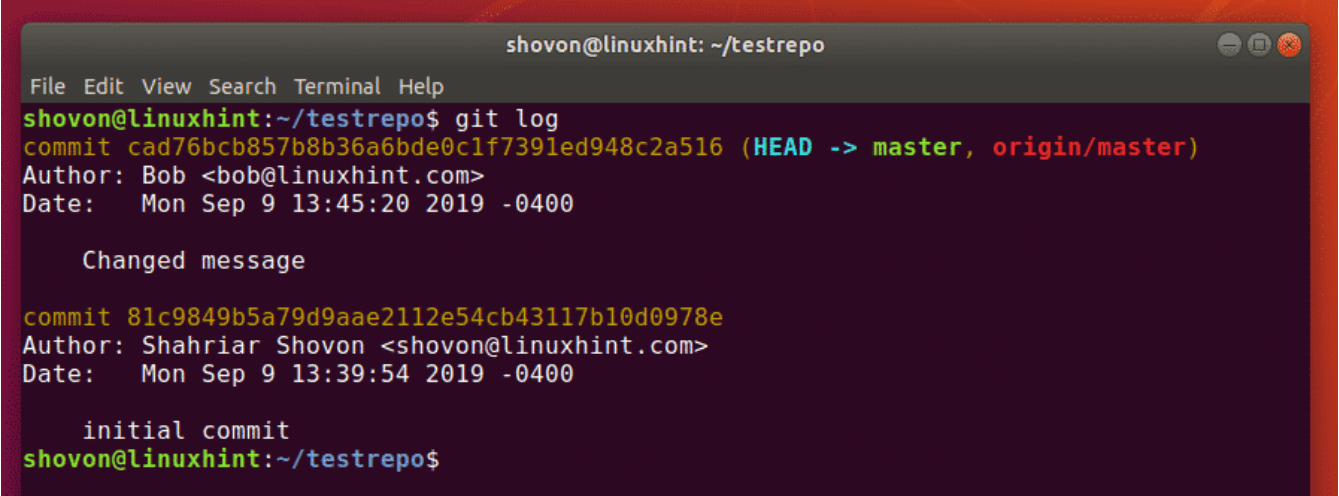
```
$ git pull origin
```

A terminal window titled 'shovon@linuxhint: ~/testrepo' showing the execution of the 'git pull origin' command. The window is partially visible, showing the command being entered.

```
shovon@linuxhint:~/testrepo$ git pull origin
```

```
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From 192.168.21.185:~/testrepo
 81c9849..cad76bc master    -> origin/master
Updating 81c9849..cad76bc
Fast-forward
 test.txt | 1 +
 1 file changed, 1 insertion(+)
shovon@linuxhint:~/testrepo$
```

He/she should find the commits that **bob** made.



```
shovon@linuxhint: ~/testrepo
File Edit View Search Terminal Help
shovon@linuxhint:~/testrepo$ git log
commit cad76bcb857b8b36a6bde0clf7391ed948c2a516 (HEAD -> master, origin/master)
Author: Bob <bob@linuxhint.com>
Date:   Mon Sep 9 13:45:20 2019 -0400

    Changed message

commit 81c9849b5a79d9aae2112e54cb43117b10d0978e
Author: Shahriar Shovon <shovon@linuxhint.com>
Date:   Mon Sep 9 13:39:54 2019 -0400

    initial commit
shovon@linuxhint:~/testrepo$
```

So, this is how you configure a Git Server with SSH on Ubuntu and use it. Thanks for reading this article.

---

ABOUT THE AUTHOR

---



## Shahriar Shovon

Freelancer & Linux System Administrator. Also loves Web API development with Node.js and JavaScript. I was born in Bangladesh. I am currently studying Electronics and Communication Engineering at Khulna University of Engineering & Technology (KUET), one of the demanding public engineering universities of Bangladesh.

---

[View all posts](#)

---

Linux Hint LLC  
1669 Holenbeck Ave, #2-244, Sunnyvale,  
CA 94087  
[editor@linuxhint.com](mailto:editor@linuxhint.com)

---

AN ELITE CAFEMEDIA PUBLISHER

