Git - SSH Login

Lab: Cloning and Working with a Remote Git Repository over SSH (Ubuntu)

User: tiago-paquete@Ubuntu1
Remote Platform: GitHub
Repository Name: T-Paquete

Goal:
Set up SSH access
Clone the repo using SSH
Verify changes using Git commands

Step 1: Check for Existing SSH Keys

ls -al ~/.ssh

Verify:
Look for id_rsa and id_rsa.pub.

If none found, proceed to Step 2.

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ ls -la ~/.ssh
==========================================================================
total 16
drwx------  2 tiago-paquete tiago-paquete 4096 May  6 08:24 .
drwxr-x--- 22 tiago-paquete tiago-paquete 4096 May 21 15:22 ..
-rw-------  1 tiago-paquete tiago-paquete    0 Sep 20  2024 authorized_keys
-rw-------  1 tiago-paquete tiago-paquete 1120 May  6 08:33 known_hosts
-rw-r--r--  1 tiago-paquete tiago-paquete  142 May  6 08:24 known_hosts.old
==========================================================================
```

Step 2: Generate a New SSH Key (If Needed)

ssh-keygen -t rsa -b 4096 -C “your_email@example.com"

Press Enter to save in default location (~/.ssh/id_rsa)

Optionally enter a passphrase

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$  ssh-keygen -t rsa -b
4096 -C “user_email"
============================================================================
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tiago-paquete/.ssh/id_rsa):
rsa_github
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in rsa_github
Your public key has been saved in rsa_github.pub
The key fingerprint is:
SHA256:oyLoks+NJbkBAr1PVKtkR0IafCz5G6lntXk6xFxDnXw user_email
The key's randomart image is:
+---[RSA 4096]----+
| ..+o o  o .     |
| .+oo+ .. + E    |
|. o++.o.   .     |
|.  =+o. o        |
|o ..o* +S.       |
|.o.++ *...       |
|..=o+..o         |
|+. 0 .o          |
|.o= .  .         |
+----[SHA256]-----+
============================================================================

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ mv rsa_github ~/.ssh/
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ mv rsa_github.pub ~/.ssh/
```

Command Breakdown

Part	Meaning
ssh-keygen	The main command to generate a new SSH key pair (public/private).
-t rsa	Specifies the **type** of key to generate. rsa stands for the RSA algorithm. This is widely supported and secure.
-b 4096	Sets the **number of bits** in the key. 4096 is stronger than the default 2048 and offers better security.
-C "user_email"	Adds a **comment** to the key. This comment (often an email) is embedded in the .pubfile and helps identify the key.

What It Does in Practice
Running this command:

- Creates a **private key** file (by default: ~/.ssh/id_rsa)
- Creates a **public key** file (by default: ~/.ssh/id_rsa.pub)
- Appends the comment "user_email" to the public key for identification

Step 3: Start SSH Agent and Add Key

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ eval "$(ssh-agent -s)"
==========================================================================
Agent pid 11132
==========================================================================

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ ssh-add ~/.ssh/id_rsa
==========================================================================
/home/tiago-paquete/.ssh/id_rsa: No such file or directory
==========================================================================

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ ssh-add ~/.ssh/
==========================================================================
Identity added: /home/tiago-paquete/.ssh/rsa_github (user_email)
==========================================================================
```

Verify:

```
ssh-add -l
```

Expected output includes your new SSH key fingerprint.

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ ssh-add -l
==========================================================================
4096 SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx user_email (RSA)
==========================================================================
```

Component Breakdown

eval "$(ssh-agent -s)"
Purpose:
Starts the SSH authentication agent (ssh-agent) in the background and sets up the environment so your terminal can communicate with it.

Details:

- ssh-agent is a background program that **holds your private keys** in memory.
- -s outputs environment variables like SSH_AUTH_SOCK.
- eval runs those environment variable assignments in your current shell.

ssh-add ~/.ssh/id_rsa
Purpose:
Adds your **private key** (id_rsa) to the running SSH agent so it can be used for authentication.

What happens:

- The private key is loaded into memory.
- The agent can now use it to authenticate without re-typing your passphrase every time.

Optional:
If your key is encrypted with a passphrase, you will be prompted to enter it once when you run this command.

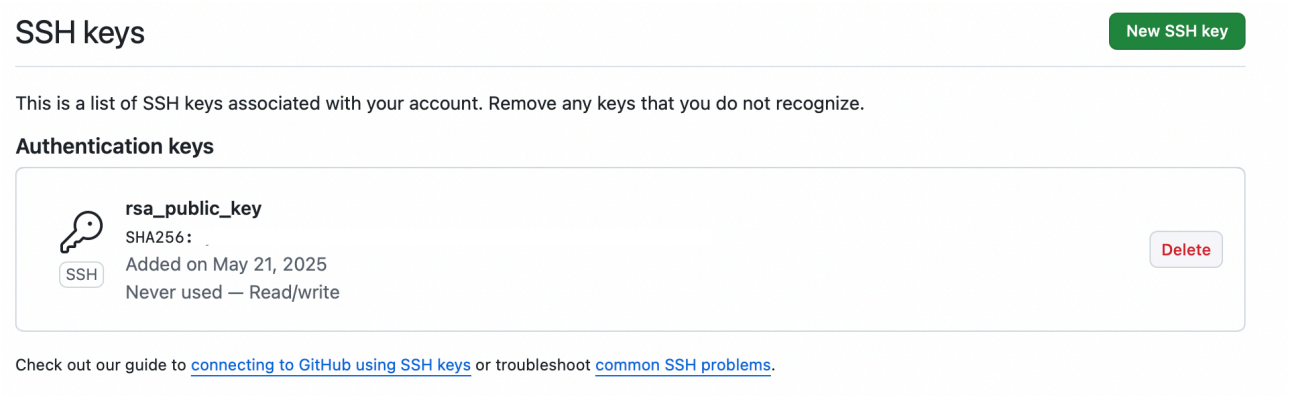
Step 4: Add Public Key to GitHub

Display your public key:
cat ~/.ssh/id_rsa.pub

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ cat ~/.ssh/id_rsa.pub
================================================================================
cat: /home/tiago-paquete/.ssh/id_rsa.pub: No such file or directory
================================================================================

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ cat ~/.ssh/
rsa_github.pub
================================================================================
ssh-rsa xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx
================================================================================
```

Copy the key and go to GitHub:
Settings → SSH and GPG Keys → New SSH Key



Step 5: Test SSH Authentication with GitHub

ssh -T git@github.com

Verify:
Expected output:

Hi tiago-paquete! You've successfully authenticated, but GitHub does not provide shell access.

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ ssh -T git@github.com
================================================================================
The authenticity of host 'github.com (140.82.121.4)' can't be
established.
Xxxxxx key fingerprint is SHA256:xxxxxxxxxxxxxxx.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (xxxxxxx) to the list of known
hosts.
Hi T-Paquete! You've successfully authenticated, but GitHub does not
provide shell access.
================================================================================
```

Command Breakdown

Component	Meaning
ssh	The **Secure Shell** command-line tool for connecting to remote systems.
-T	Tells SSH to **disable pseudo-terminal allocation** (you don't need an interactive terminal). This is common for automation or scripted operations.
git@github.com	The **SSH user and host** you're connecting to:

git is the GitHub SSH user (not your GitHub username)
github.com is the remote host |

Purpose

- This command checks:
- Whether your SSH agent is running
- Whether your **SSH key is correctly loaded**
- Whether GitHub **recognizes your public key**

Expected Output (if everything works):
Hi tiago-paquete! You've successfully authenticated, but GitHub does not provide shell access.

This confirms:

- Your SSH key is working
- GitHub has your public key on file
- You’re authenticated and can now push/pull via SSH

Step 6: Clone the Repository via SSH

Navigate to where you want the clone to live:

cd ~/GitHubProjects

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ mkdir clonerepo
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete$ cd clonerepo
```

Clone the repository using SSH:

git clone git@github.com:T-Paquete/testfiles.git

Confirm SSH Key Is Loaded:

```
tiago-paquete@Ubuntu1:~/GitHubProjects$ ssh-add -l
========================================================================
4096 SHA256: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx user_email (RSA)
========================================================================
```

Confirm SSH Works with GitHub:

```
tiago-paquete@Ubuntu1:~/GitHubProjects$ ssh -T git@github.com
========================================================================
Hi T-Paquete! You've successfully authenticated, but GitHub does not
provide shell access.
========================================================================
```

Double-Check You Uploaded the Public Key

```
tiago-paquete@Ubuntu1:~/GitHubProjects$ cat ~/.ssh/rsa_github.pub
========================================================================
Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx
Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx Xxxxxxxxxx
========================================================================
```

```
tiago-paquete@Ubuntu1:~/GitHubProjects$ git clone git@github.com:T-
Paquete/testfiles.git
========================================================================
Cloning into 'testfiles'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 30 (delta 8), reused 24 (delta 5), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), done.
Resolving deltas: 100% (8/8), done.
========================================================================
```

Step 7: Verify Repository Setup

git remote -v

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git remote
-v
==========================================================================
origin  https://github.com/T-Paquete/testfiles.git (fetch)
origin  https://github.com/T-Paquete/testfiles.git (push)
==========================================================================
```

Step 8: Make a Test Change in the Clone

Edit a file or create a new one:
echo "SSH setup verified" >> file1

Stage and commit:
git add file1
git commit -m "Test commit over SSH”

Push the changes:
git push

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ echo "SSH
setup verified" >> file1

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git add
file1

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git commit
-m "Test commit over SSH”
================================================================================
[main 2e63b5e] Test commit over SSH
 1 file changed, 1 insertion(+)
 create mode 100644 clonerepo/file1
================================================================================

tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git push
================================================================================
Username for 'https://github.com': T-Paquete
Password for 'https://T-Paquete@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 347 bytes | 347.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/T-Paquete/testfiles.git
   9cdc72c..2e63b5e  main -> main
================================================================================
```

Verify:
git log --oneline -n 1

Check GitHub UI for your new commit.

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git log --
oneline -n 1
================================================================================
2e63b5e (HEAD -> main, origin/main) Test commit over SSH
================================================================================
```

Step 9: Final SSH Operation Check

Pull the latest changes to confirm SSH is used:

git pull

No password prompt should appear.

```
tiago-paquete@Ubuntu1:~/GitHubProjects/T-Paquete/clonerepo$ git pull
==========================================================================
Username for 'https://github.com': T-Paquete
Password for 'https://T-Paquete@github.com':
Already up to date.
==========================================================================
```