# How to set up SSH for Git?

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username or password at each visit.

Before you generate an SSH key, you can check to see if you have any existing SSH keys.

Open Terminal. Enter "ls -al ~/.ssh" to see if existing SSH keys are present:

```
$ ls -al ~/.ssh
# Lists the files in your .ssh directory, if they exist
```

Check the directory listing to see if you already have a public SSH key. By default, the filenames of the public keys are one of the following:

```
id_dsa.pub
id_ecdsa.pub
id_ed25519.pub
id_rsa.pub
```

If you see an existing public and private key pair listed that you would like to use on your Bitbucket, GitHub (or similar) account you can copy the contents of the id_*.pub file.

If not, you can create a new public and private key pair with the following command:

```
$ ssh-keygen
```

Press the Enter or Return key to accept the default location. Enter and re-enter a passphrase when prompted, or leave it empty.

Ensure your SSH key is added to the ssh-agent. Start the ssh-agent in the background if it's not already running:

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

Add you SSH key to the ssh-agent. Notice that you'll need te replace id_rsa in the command with the name of your private key file:

```
$ ssh-add ~/.ssh/id_rsa
```

If you want to change the upstream of an existing repository from HTTPS to SSH you can run the following command:

```
$ git remote set-url origin ssh://git@bitbucket.server.com:7999/projects/your_pro
```

In order to clone a new repository over SSH you can run the following command:

```
$ git clone ssh://git@bitbucket.server.com:7999/projects/your_project.git
```

<< Previous Program