To set up SSH keys and connect them to your GitHub account, you can follow the steps below. This procedure assumes you are using a Unix-based system like Linux or macOS, but similar steps apply for Windows, using tools like Git Bash.

### Step 1: Check for Existing SSH Keys

Before creating a new SSH key, you should check whether you have existing SSH keys.

```sh

ls -al ~/.ssh

```

If you see files named `id\_rsa.pub`, `id\_ed25519.pub` or similar, you already have an SSH key.

### Step 2: Generate a New SSH Key

If you don’t have an SSH key or want to create a new one, use the following command. When prompted, you can press Enter to accept the default file location.

```sh

ssh-keygen -t ed25519 -C "your\_email@example.com"

```

This command will create a new SSH key, using the provided email as a label. If your system doesn't support `ed25519`, you can use `rsa` instead:

```sh

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

```

### Step 3: Start the SSH Agent

Run the following commands to start the SSH agent in the background.

```sh

eval "$(ssh-agent -s)"

```

### Step 4: Add SSH Key to SSH Agent

If you created an `ed25519` key, run:

```sh

ssh-add ~/.ssh/id\_ed25519

```

For an `rsa` key, run:

```sh

ssh-add ~/.ssh/id\_rsa

```

### Step 5: Add SSH Key to GitHub Account

1. Run the following command to copy your SSH public key to the clipboard. For `ed25519` key:

```sh

pbcopy < ~/.ssh/id\_ed25519.pub

```

Or for an `rsa` key:

```sh

pbcopy < ~/.ssh/id\_rsa.pub

```

On Linux, you may need to use `xclip` instead of `pbcopy`:

```sh

xclip -sel clip < ~/.ssh/id\_ed25519.pub

```

2. Go to [GitHub](https://github.com/).

3. In the upper-right corner of any page, click your profile photo, then click \*\*Settings\*\*.

4. In the user settings sidebar, click \*\*SSH and GPG keys\*\*.

5. Click \*\*New SSH key\*\* or \*\*Add SSH key\*\*.

6. In the "Title" field, add a descriptive label for the new key.

7. Paste your key into the "Key" field.

8. Click \*\*Add SSH Key\*\*.

### Step 6: Test Connection

You can test that your connection works by connecting to GitHub via SSH with this command:

```sh

ssh -T git@github.com

```

You should see a message saying you've successfully authenticated.

### Using SSH with Git

Now, when cloning a new repository, use the SSH URL instead of HTTPS. The SSH URL can be found on the main page of the repository and it looks like this:

```sh

git@github.com:username/repo.git

```

So, for example, when cloning a repo, you would run:

```sh

git clone git@github.com:username/repo.git

```

This will use SSH to interact with GitHub instead of HTTPS, meaning you won’t have to enter your GitHub username and password every time you pull or push to the repo.