github

Generate ssh key

ssh-keygen -t rsa -C "rajeevkumar.kp100@gmail.com" -f "RajeevPal458"

**Create SSH keys for all accounts**

First make sure your current directory is your **.ssh** folder.

$ cd ~/.ssh

Syntax for generating unique ssh key for ann account is:

ssh-keygen -t rsa -C "your-email-address" -f "github-username"

here,

**-C** stands for comment to help identify your ssh key

**-f** stands for the file name where your ssh key get saved

**Now generating SSH keys for my two accounts**

ssh-keygen -t rsa -C "my\_office\_email@gmail.com" -f "github-rahul-office"

ssh-keygen -t rsa -C "my\_personal\_email@gmail.com" -f "github-rahul-personal"

Create a Config File and Make Host Entries

The ~/.ssh/config file allows us specify many config options for SSH.

If config file not already exists then create one (make sure you are in ~/.ssh directory)

touch config

The commands below opens config in your default editor....Likely TextEdit, VS Code.

open config

Now we need to add these lines to the file, each block corresponding to each account we created earlier.

Host github.com-codecurious2023

HostName github.com

User git

IdentityFile ~/.ssh/id\_rsa\_codecurious2023

Host github.com-RajeevPal458

HostName github.com

User git

IdentityFile ~/.ssh/id\_rsa\_RajeevPal458

For Example: I am making a repository on my personal github account and naming it **TestRepo** Now for cloning the repo use the below command:

git clone git@github.com-{your-username}:{owner-user-name}/{the-repo-name}.git

[e.g.] git clone git@github.com-rahul-personal:rahul-personal/TestRepo.git

**Finally**

From now on, to ensure that our commits and pushes from each repository on the system uses the correct GitHub user — we will have to configure **user.email** and **user.name** in every repository freshly cloned or existing before.

To do this use the following commands.

git config user.email "my\_office\_email@gmail.com"

git config user.name "Rahul Pandey"

git config user.email "my-personal-email@gmail.com"

git config user.name "Rahul Pandey"

Pick the correct pair for your repository accordingly.

To push or pull to the correct account we need to add the remote origin to the project

git remote add origin git@github.com-rahul-personal:rahul-personal

git remote add origin git@github.com-rahul-office:rahul-office

Now you can use:

git push

git pull