**How to handle multiple GitHub accounts on your Windows PC.**

GitHub is a vital developer tool that offers version control and a collaboration platform. If you're new to GitHub or looking to manage multiple GitHub accounts(maybe work, personal) on the same PC, this guide will walk you through the process.

Setting Up GitHub on Your PC for the first time

1. **Install Git:**

Download Git from the official git website (https://git-scm.com/downloads).

Follow the installation instructions for your operating system (Windows, macOS, or Linux).

1. **Create a GitHub Account:**

Sign up for a free GitHub account at (https://github.com/).

1. **Generate SSH Keys:**

Open your terminal (Git Bash on Windows).

Generate a new SSH key by running:

`ssh-keygen -t rsa -b 4096 -C [your\_email@example.com](mailto:your_email@example.com)`

*(replace your\_email@example.com with the email you used to create your GitHub account).*

When prompted, save the key to the default location (`/home/you/.ssh/id\_rsa` or `C:\Users\you\.ssh\id\_rsa`).

1. **Add SSH Key to GitHub:**

Copy the SSH key to your clipboard using the “cat” command:

`cat ~/.ssh/id\_rsa.pub`

Log in to GitHub and go to Settings >> SSH and GPG keys >> New SSH key.

Paste your key and give it a title.

1. **Test the Connection:**

Run the following command to test your connection:

`ssh -T git@github.com`

**Managing Multiple GitHub Accounts**

Managing multiple GitHub accounts on the same PC requires careful configuration to avoid conflicts. Here's how you can do it:

1. **Create SSH Keys for Each Account:**

Generate a separate SSH key for each account:

`ssh-keygen -t rsa -b 4096 -C "your\_email@example1.com" -f ~/.ssh/id\_rsa\_account1`

`ssh-keygen -t rsa -b 4096 -C "your\_email@example2.com" -f ~/.ssh/id\_rsa\_account2`

1. **Add SSH Keys to GitHub:**

Follow the earlier steps to add each key to the respective GitHub accounts.

1. **Configure SSH:**

Open or create (if it doesn’t exist) the `~/.ssh/config` file:

`nano ~/.ssh/config`

Add the following configuration:

# account 1

Host github-account1

HostName github.com

User git

IdentityFile ~/.ssh/id\_rsa\_account1

# account 2

Host github-account2

HostName github.com

User git

IdentityFile ~/.ssh/id\_rsa\_account2

1. **Cloning Repositories:**

Use the custom hostnames when cloning repositories:

Using SSH

git clone [git@github-account1:username/repo.git](about:blank)

git clone [git@github-account2:username/repo.git](about:blank)

1. **Switching Accounts:**

To switch accounts for existing repositories, update the remote URL:

`git remote set-url origin git@github-account1:username/repo.git`

`git remote set-url origin git@github-account2:username/repo.git`

**Conclusion**

By following these steps, you can easily connect your PC to GitHub and manage multiple GitHub accounts without any hassle. Whether you have separate accounts for personal and work projects or multiple personal accounts, this setup will help you keep everything organized and streamlined.

(Cool, I think you should implement what you wrote, grab the screenshots where important and add them to the article, for a newbie, that may be all the help he needed. Also, please add at the conclusion that they can check out our website for a comprehensive devops course that covers this extensively, link the website too.)