Coordonix

Be Empowered Online

USER MANUAL

GitHub Setup and Configuration

www.coordonix.com

Last Udpated: 2025/01/16

Introduction

GitHub is a vital tool for developers. This guide shows how to set up GitHub with Windows Subsystem for Linux *(WSL)* and configure SSH keys for secure access. By the end, you'll be ready to clone, push, and pull repositories seamlessly.

Prerequisites

Before you begin, ensure you have the following:

1. **Windows Subsystem for Linux (WSL)** installed and set up on your system.

2. **Git** installed with your WSL environment. You can install it by running:

   ```
   sudo apt install git
   ```

3. A **GitHub Account:** Sign up at `www.github.com` if you don't already have one.

Installation Process

Follow these steps to install and configure your environment:

1. Open WSL by clicking Start and typing `wsl`. Hit `Enter` to launch.

2. Update your WSL distribution by running: `sudo apt update && sudo apt upgrade`

3. Generate your SSH key pair by running: `ssh-keygen -t ed25519 -C "your.email@example.com"`
   Replace *your.email@example.com* with the email address associated with your GitHub account.

4. When prompted for the file location and key name enter `/home/USERNAME/.ssh/NEWNAME`.

5. You can set a passphrase for extra security or hit enter for no passphrase.

6. Copy your public key to the clipboard. Enter `cat ~/.ssh/KEYNAME.pub`
   Select and copy the information displayed.

Post-Installation Configuration

GitHub Setup

1. Go to `www.github.com` and sign in.

2. Click on your profile in the top right and click Settings.

3. Click SSH and GPG Keys in the left menu.

4. Click New SSH Key.

5. Add a descriptive title and paste the copied key into the Key field.

6. Click Add SSH key.

.bashrc.local Setup

1. Back in your WSL terminal, run `cd` to be back in your home directory.
2. Create your .bashrc.local run `vim .bashrc.local`
3. Hit `I` to enter Insert Mode and type the following:

```
eval "\$(ssh-agent -s)" > /dev/null
ssh-add ~/.ssh/KEYNAME 2> /dev/null
```

4. Hit `Esc`, type `:wq`, and then hit `Enter` to save and exit the file.

.bashrc Setup

Now that we've created our .bashrc.local file, we need WSL to source that file when we launch.

1. Back your WSL terminal, run `cd` to be back in your home directory.
2. Open your .bashrc by running `vim .bashrc`
3. Hit `Shift + G` to go to the end of the file.
4. Hit `Shift + A` to go to the end of the line and hit `Enter` twice to add a new space at the end of the document.
5. Now enter the following to source your .bashrc.local

```
if [ -f ~/.bashrc.local ]; then
        source ~/.bashrc.local
fi
```

6. Hit `Esc` and type `:wq` and hit `Enter` to save and exit the file.
7. Once back on the terminal run the `exit` command to close WSL.

Verify Installation

Test your SSH Connection

1. Open WSL by clicking Start and typing `wsl`. Hit `Enter` to launch.
2. Test the connection to GitHub by typing `ssh -T git@github.com`. You should get the following return:

```
Hi USERNAME! You’ve successfully authenticated, but GitHub
   does not provide shell access.
```