Instruction Sheet for Git Setup

Docker- Automating Docker Builds

Spoken Tutorial Team

IIT Bombay

This sheet describes the procedure to set up Git on Windows and Linux (Ubuntu) systems, as well as configuring it to work with GitHub. Additionally, instructions to initialize a Git repository and push the code to GitHub are also provided.

Click on the relevant links to jump to a specific section.

[Instructions to install and configure Git on Windows systems](#_p2fbds2csb5o)

[Instructions to install and configure Git on Ubuntu systems](#_7bsjpspjma8o)

[Instructions to configure Git](#_h0jbphqwqkd2)

[Generate SSH keys](#_mo4xb5oqpvkq)

[Add SSH keys to ssh-agent](#_fryg78fl12pm)

[Add public key to GitHub](#_7sopctwgqix)

[Creating a repository on GitHub](#_qgw2rd1qfoon)

[Pushing code to GitHub](#_oyiqd3ny39r2)

# **Instructions to install Git on Windows systems**

1. Getting ready for Git installation
   1. To follow the installation procedure, you need to be connected to the internet.
2. Download the Git installer executable file
   1. Go to the website: [Git - Downloading Package (git-scm.com)](https://git-scm.com/downloads/win)
   2. Under the **Standalone Installer** section, choose the appropriate download option based on your system type (64-bit or 32-bit).
   3. Download the appropriate installer for your system.
   4. Go to the directory where you have downloaded the **.exe** installer file (default **Downloads**).
   5. Run the installer by double clicking on the file.
   6. Select 'Yes' in the User Account Control prompt that appears.
   7. The installer window will open.
   8. Click on the **Next** button on the **GNU General Public License Page.**
   9. In the **Select Components** page, keep the default components selected and click on the **Next** button.
   10. In the **Choose the default editor used by Git** screen, choose the editor from the dropdown list that you wish to use. Click on **Next**.
   11. In the **Adjusting the name of the initial branch in new repositories** page. Select **Override the default branch name for new repositories**.
   12. In the **Specify the name "git init" should use for the initial branch** field, type **main.** Then click on **Next.**
   13. In the adjusting your **PATH** environment page, select **Git from the command line and also from 3rd party software**. Click on **Next.**
   14. In the **Choosing the SSH executable** page, select **Use bundled OpenSSH** option and click on **Next.**
   15. In the **Choosing HTTPS transport protocol** page, select **Use the OpenSSL library** option and click on **Next**.
   16. In the **Configuring the line ending conversions**, select **Checkout Windows-style, commit Unix-style** line endings. Click on **Next**.
   17. In the **configuring the terminal emulator to use with Git Bash** page, select **Use MinTTY** and click on **Next.**
   18. In the **Choose the default behavior of 'git pull'** page, select **Fast-forward or merge** and click on **Next.**
   19. In the **Choose a credential helper,** select **Git Credential Manager** and click on **Next.**
   20. In the **Configuring extra options** page, select **Enable file system caching** and click on **Next.**
   21. In the **Configuring experimental options** leave all options **unchecked** and click on **Next.**
   22. Click on the **Install** button

# **Instructions to install Git on Ubuntu systems**

1. Getting ready for Git installation
   1. To follow the installation procedure, you need to be connected to the internet.
2. Installing Git from command line
   1. Update the repositories using the command **sudo apt-get update**.
   2. Install Git using the command **sudo apt-get install git**.

# **Instructions to configure Git (Com**mon to Windows and Linux)

1. Set Git username using the command **git config --global user.name '<your username>'** (username in quotes)
2. Set Git email using the command **git config --global user.email '<your email>'** (email in quotes)

# Generate SSH keys

SSH keys are required for authentication to push code from our systems to a remote repository. Follow these steps to generate SSH key pair of public and private keys:

1. Type the command **ssh-keygen -t ed25519 -C '<your email>'** (email in quotes)
2. In the prompt **Enter file in which to save the key,** press **Enter** to keep the default file name and location
3. In the **Enter passphrase** prompt, type a secure password for the private key.
4. Enter the password again in the **Enter same passphrase again** prompt.

# Add SSH keys to ssh-agent

To avoid having to enter the password for the private key every time, we can save the keys in the ssh agent.

Follow these steps to add the key to the ssh agent:

1. Type the command **eval "$(ssh-agent -s)"** and press **Enter.** You will get the ID of the agent. This will start the agent in the background.
2. Type the command **ssh-add ~/.ssh/id\_ed25519** and press **Enter.** Enter your password for the **Private Key** set earlier when prompted. This will add the private key to the agent.

**Note:** The above commands assume your private key is named **id\_ed25519** and it is saved in the default location **~/.ssh.** If either of these is different, change the command accordingly.

# Add public key to GitHub

To authenticate while working with GitHub from the command line, we need to add the public key to our GitHub account. If you do not have a GitHub account yet, go to the following link and create an account. The email ID should be the same as the one used to configure Git.

[Join GitHub · GitHub](https://github.com/signup)

Follow these steps to add the public key to your **GitHub** account:

1. Type **cat ~/.ssh/id\_ed25519.pub** and press **Enter.** Copy the output.
2. Go to your **GitHub** account settings by clicking on the profile picture on the top right and in the sidebar click on **Settings.**
3. In the settings page, on the sidebar, click on **SSH and GPG Keys**.
4. Under **SSH Keys,** click on the New **SSH key** button.
5. Give a title for the key in the **Title** field.
6. Under **Key type** choose **Authentication key**.
7. In the **Key** text box, paste the public key copied previously.
8. Click on the **Add SSH key** button.

**Note:** The above commands assume your private key is named **id\_ed25519** and it is saved in the default location **~/.ssh.** If either of these is different, change the command accordingly.

# Creating a repository on GitHub

To create a new repository on GitHub, follow these steps:

1. Go to **GitHub.**
2. Click on the **Repositories** tab on the top ribbon.
3. Click on the **New** button.
4. In the **Create a new repository** page, type a name for the repository in the **Repository name** field.
5. Change other options if required.
6. Click on the **Create repository** button.

# Pushing code to GitHub

To push code to GitHub using command line, follow the steps below:

1. Get the URL for the repository from the repository page on GitHub. Copy the SSH URL. It should look like the following: **git@github.com:<username>/<repository-name>.git**
2. In the terminal, switch to the **node-express** directory.
3. Initialize an empty git repository in the directory using the command **git init**.
4. Set the origin for the repository to the **GitHub** repository created previously using the command **git remote add origin <Github repository URL>**
5. Switch to the **main** branch using the command **git branch -M main**.
6. Add the code files to be pushed using the command **git add .** (period after **add** to add all the files in the current directory).
7. Commit the changes using the command **git commit -m "<commit message>"**(Replace commit message between the quotes)
8. Push the code using the command **git push -u origin main**.
9. You may get a prompt asking to add **github.com** to the list of known hosts. Type **yes** and press **Enter**.