

GIT INSTALLATION

Install and setup #Git on #Ubuntu 22.04



Git is a widely used version control system. It is easy to install and configure on Ubuntu. Here are the steps to install Git on Ubuntu:

Installation Linux

1. Update your local package index.

```
```bash
```

```
sudo apt update
```

```
...
```

2. Install Git by running the following command:

```
```bash
```

```
sudo apt install git
```

```
...
```

3. Once the installation is complete, you can check the Git version by running the following command:

```
```bash
```

```
git --version
```

```
...
```

# Setting Up Git

1. Next, you should set up your Git identity. This includes your name and email address, which Git will use to associate your commits with you. Run the following commands to set your name and email address:

```
```bash
```

```
git config --global user.name "Your Name"
```

```
git config --global user.email "youremail@domain.com"
```

```
...
```

2. You can also configure Git to use a default text editor for editing commit messages. Run the following command to set your preferred editor (in this case, nano):

```
```bash
```

```
git config --global core.editor "nano"
```

```
...
```

3. You can display all the configuration items

```
```bash
```

```
git config --list
```

```
...
```

Configure Git with your key

1. Lists the files in `.ssh` directory, if they exist

```
```bash
```

```
ls -al ~/.ssh
```

```
...
```

2. You will need to generate an SSH key pair and add the public key to your Git hosting service (e.g., GitHub, GitLab, Bitbucket). Here are the steps to generate an SSH key pair:

```
```bash
```

```
ssh-keygen -t ed25519 -C "youremail@example.com"
```

...

or

```
```bash
```

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

...

### 3. Start the ssh-agent in the background

```
```bash
```

```
eval "$(ssh-agent -s)"
```

...

4. Add your SSH private key to the ssh-agent.

```
```bash
```

```
ssh-add ~/.ssh/id_ed25519
```

...

or

```
```bash
```

```
ssh-add ~/.ssh/id_rsa
```

...

5. Copy the SSH public key to your clipboard.

```
```bash
```

```
cat ~/.ssh/id_ed25519.pub
```

...

or

```
```bash
```

```
clip < ~/.ssh/id_ed25519.pub
```

...

Installing Git on Windows:

1. Visit the official Git website: <https://git-scm.com/>.
2. Download the latest version of Git for Windows.
3. Run the installer and follow the on-screen instructions.
4. During the installation, you can choose the default options or customize them according to your preferences.
5. Once the installation is complete, you can access Git through the Git Bash terminal or other supported terminals.
6. To check the git version:

```
git --version
```

Installing Git on Linux (Ubuntu/Debian):

1. Open the terminal.
2. Run the following command to install Git:

```
```sh
```

```
sudo apt update
```

```
sudo apt install git
```

3. Provide your password if prompted, and Git will be installed on your Linux system.

## \*\*Configuring Global User Information:\*\*

After installing Git, you should configure your global user information. Open a terminal or Git Bash and run the following commands, replacing "Your Name" and "[your.email@example.com](mailto:your.email@example.com)" with your own name and email address:

```
```sh
```

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

These commands set your name and email address to be associated with your Git commits.

Basic Git Commands:

1. **Clone**: Clone a repository from a remote server to your local machine.

```
git clone <repository-url>
```

2. **Add**: Add changes to the staging area before committing.

```
git add <file-name>          # Add a specific file  
git add .                   # Add all files in the current directory
```

3. **Commit**: Commit your changes with a descriptive message.

```
git commit -m "Commit message"
```

4. **Push**: Push your committed changes to a remote repository.

```
git push origin <branch-name>
```

5. **Pull**: Update your local repository with the latest changes from the remote repository.

```
git pull origin <branch-name>
```

6. **Branch:** Create, list, or switch between branches.

```
git branch          # List all branches  
git branch <branch-name>    # Create a new branch  
git checkout <branch-name>  # Switch to an existing branch
```

7. **Add Remote URL:** Add a remote URL to your local repository.

```
git remote add origin <remote-url>
```

These are some of the basic Git commands to get you started. There are many more commands and options available depending on your specific needs. You can explore additional Git documentation and resources to learn more about the advanced features and workflows.