

# Setup Manual

## Read the instruction/text carefully and follow each and every step!!!

These are the steps to perform after you login to the newly installed Ubuntu system. If you are performing these steps in the workshop class avoid the step 2 and step 3 as it might take some time and we will have limited time. Do perform it later at your own time as it helps keep your system up to date.

1. Press CTRL+ALT+T to open terminal. You can also press the window key and search for it
2. In the terminal run the following command: `sudo apt update`

`sudo` keyword is used to provide root access...this will ask you for a password enter your system password

`apt` is a command line utility that will be used to install any new software in your system from here onwards

`update` will get any updates regarding the software repository from remote

```
ardent@ardent:~$ sudo apt update
[sudo] password for ardent:
Hit:1 http://np.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://np.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
0% [2 InRelease 35.7 kB/126 kB 28%]
```

3. Run `sudo apt upgrade`

this will upgrade all the software that have updates available

```
ardent@ardent:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
```

Enter y when/if prompted

After this our system is up-to-date with any pending being applied. Now we will install other software that we will require throughout the course

Initially, let us figure out the python version using below command

```
ardent@ardent:~$ python3 --version
Python 3.12.3
ardent@ardent:~$
```

Now install the virtual environment relevant to the python version

```
ardent@ardent:~$ sudo apt install python3.12-venv
[sudo] password for ardent:
Reading package lists... Done
Building dependency tree... Done
```

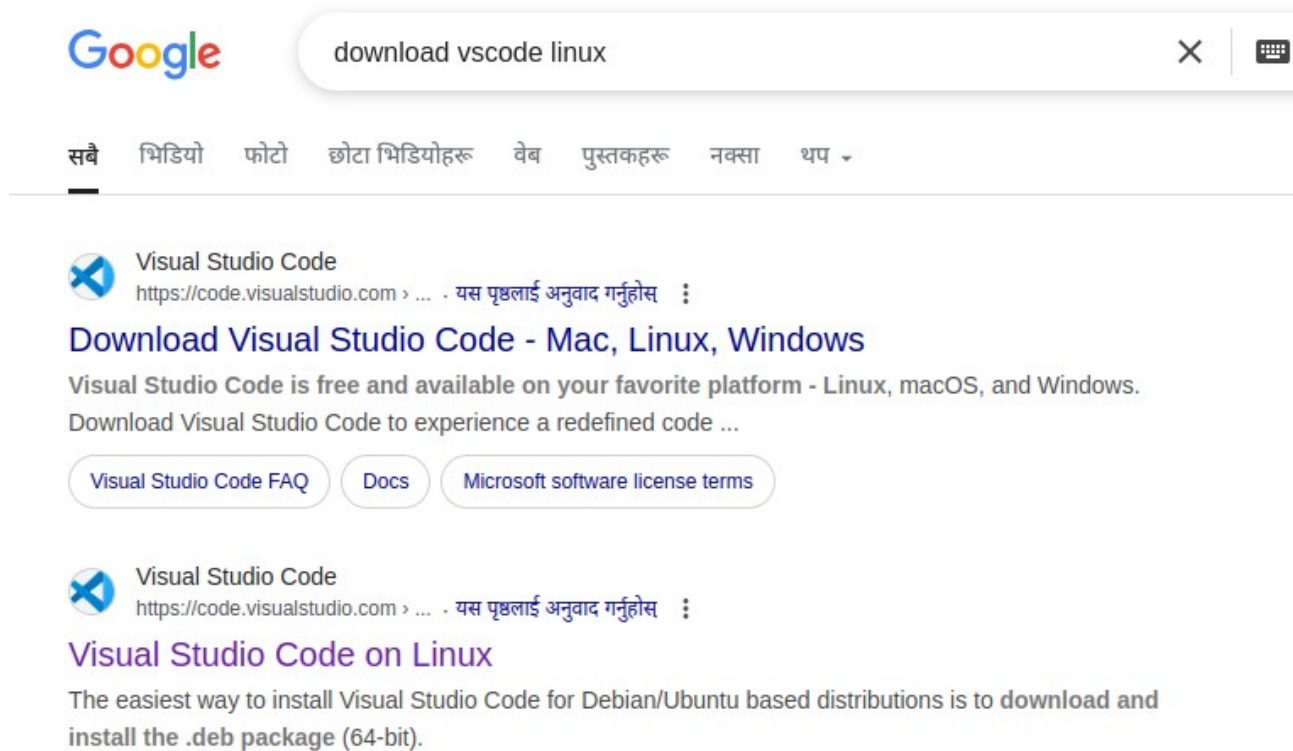
Install git and other useful libraries using the below command. Notice how multiple tools can be downloaded at once separated by space. Vim is optional but is a good command line text editor if anyone wants to learn.

```
ardent@ardent:~$ sudo apt install git vim curl openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

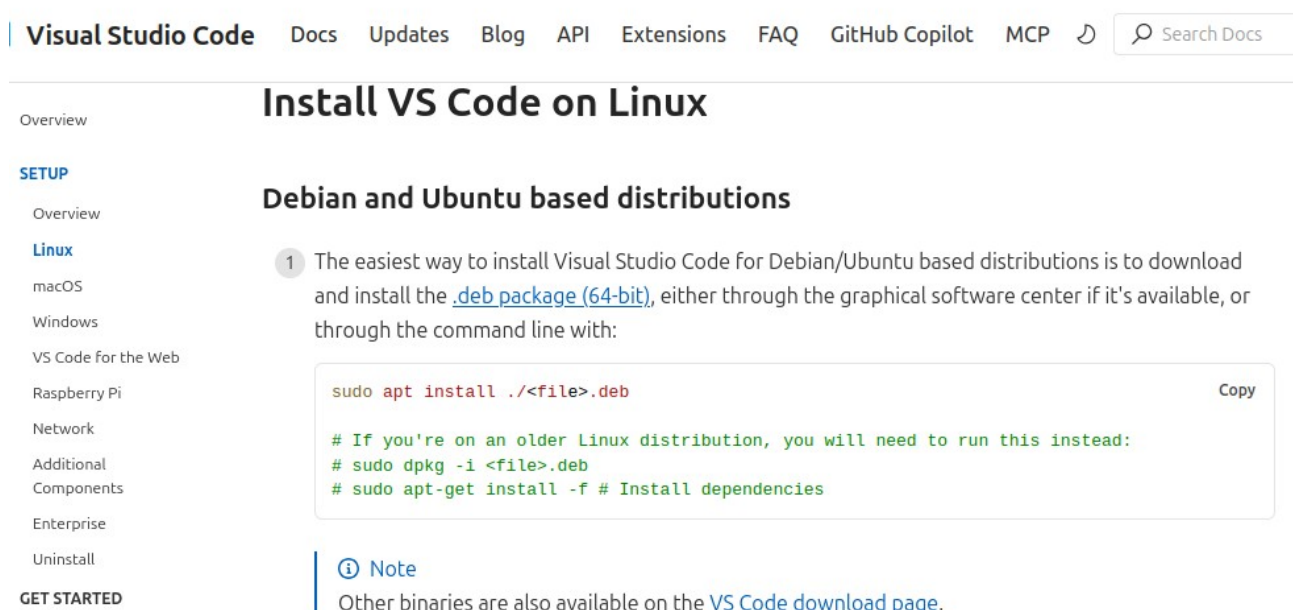
## VSCode Installation

Now, it is time to install vscode. Since the ubuntu repository does not have location of vscode prebuilt to it. We will have to do this manually.

1: Go to your browser search bar and type download vscode linux



2. Open the link that has “visual studio code on linux” written on it and scroll down until you see link for code written as “.deb package (64-bit),”



This will automatically download .deb file to your system

3. Change directory to Downloads as below:

```
ardent@ardent:~$ cd ~/Downloads
ardent@ardent:~/Downloads$
```

Notice the ~ after the cd command this means start from the user home directory

4. You can either run ls command and view all the files or use grep to quickly figure out if code has been downloaded successfully. If the code is not available there, it has either not downloaded properly or is in some other directory

```
ardent@ardent:~/Downloads$ ls -l
total 511992
drwxrwxr-x 2 ardent ardent 4096 Jun 23 15:06 archive
drwxrwxr-x 2 ardent ardent 4096 Jun 23 15:19 'archive(1)'
-rw-rw-r-- 1 ardent ardent 201984462 Jun 23 15:19 'archive(1).zip'
-rw-rw-r-- 1 ardent ardent 8571542 Jun 23 14:05 archive.zip
-rw-rw-r-- 1 ardent ardent 108605322 Jun 22 21:17 code_1.101.1-1750254731_amd64.deb
-rw-rw-r-- 1 ardent ardent 80536530 Jun 22 20:30 obsidian_1.8.10_amd64.deb
-rwxrwxr-x 1 ardent ardent 116781756 Jun 22 20:29 Obsidian-1.8.10.AppImage
-rw-rw-r-- 1 ardent ardent 7763096 Jun 22 21:19 OneDrive_2025-06-22.zip
drwxrwxr-x 3 ardent ardent 4096 Jun 22 21:20 'summer class - data pyspark'
ardent@ardent:~/Downloads$ ls | grep code
code_1.101.1-1750254731_amd64.deb
ardent@ardent:~/Downloads$
```

5. Run the following command to install vscode in your system

```
ardent@ardent:~/Downloads$ sudo dpkg -i code_1.101.1-1750254731_amd64.deb
(Reading database ... 168275 files and directories currently installed.)
Preparing to unpack code_1.101.1-1750254731_amd64.deb ...
```

You can also use apt to install as apt internally uses dpkg

```
ardent@ardent:~/Downloads$ sudo apt install ./code_1.101.1-1750254731_amd64.deb
Reading package lists... Done
Building dependency tree... Done
```

## Git and GitHub SSH Setup

Now, we have most of the tools we will need throughout the course. Let us configure git and GitHub to work with ssh key

1. Let us first create an ssh key using below command: Make sure to set the email you have a github account for

```
ssh-keygen -t ed25519 -C "your_email@gmail.com"
```

```
upgraded, 0 newly installed, 0 to remove and 3 not upgraded.  
ardent@ardent:~/Downloads$ ssh-keygen -t ed25519 -C "ardent.sharma@islingtoncollege.edu.np"
```

2. You can set the filename and passphrase to use but for now leave it **empty** press enter 3 times

```
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/ardent/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

You should output similar to below:

```
Enter same passphrase again:  
Your identification has been saved in /home/ardent/.ssh/id_ed25519  
Your public key has been saved in /home/ardent/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:hPbf00p/eBYOMskj5G8WFJ2/iuEh6qdUrIUBoGMM  
The key's randomart image is:  
+--[ED25519 256]--+  
|E   ...      .  . |  
|*  .   ..   .  o  |  
| *    o..   .   . |  
|. . . o+. .   .   |  
|      .S+o .   .   |  
|      += @ o o   |  
|      o. * % =..   |  
|      .. . B =..+  |  
|      .oo o . .+   |  
+-----[SHA256]-----+
```

3. Change directory to where the key were saved using below command. Use ls command to view the files in that directory. If you left blank space above and did not provide any file name the names

will be same as below: id\_ed25519 and id\_ed25519.pub

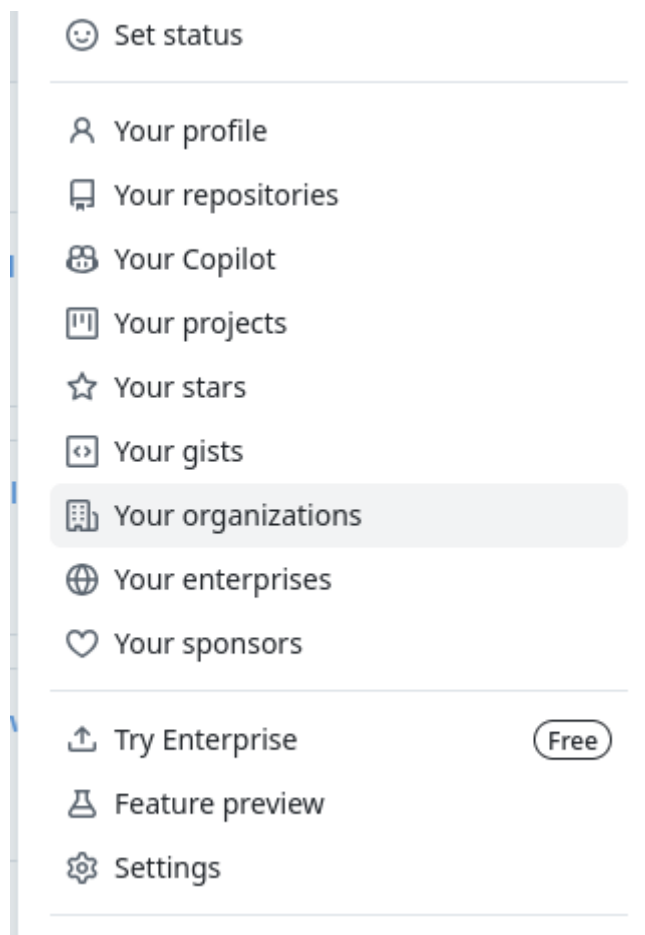
id\_ed25519.pub is the public key and will be used in GitHub to setup ssh. Do not share these key with anyone else especially the private key

```
ardent@ardent:~/Downloads$ cd ~/.ssh  
ardent@ardent:~/ssh$ ls  
authorized_keys  id_ed25519  id_ed25519.pub  
ardent@ardent:~/ssh$
```

4. View the content of the pub file using cat and copy the text as it is using CTRL+SHIFT+C

```
ardent@ardent:~/ssh$ cat id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDKDRnB2E408u  
ardent@ardent:~/ssh$
```

5. Log in to GitHub and goto settings, from the profile icon



6. In the left hand side you should see Access section. Click on SSH and GPG keys then New SSH Key on the right side

The screenshot shows the GitHub 'SSH keys' management interface. On the left is a sidebar with navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access (expanded), Billing and licensing, Emails, Password and authentication, Sessions, SSH and GPG keys (selected), Organizations, Enterprises, and Moderation. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below the title is a message: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the heading 'Authentication keys', there are two listed keys: 'personal-pc' (added May 15, 2024) and 'personal-windows' (added Jun 22, 2025). Each key entry shows its SHA256 fingerprint and a 'Delete' button. At the bottom of the main area, there is a link to a guide on connecting to GitHub using SSH keys. Below the main content is a section for 'GPG keys' with a 'New GPG key' button.

7. Give a title and paste the content you copied from the terminal to Key section. Note that the start of key must be ssh-ed25519 and end must be your email. Finally, click on Add SSH Key button

The screenshot shows the 'Add new SSH Key' form. It has three main sections: 'Title' with a text input field containing 'summer-class'; 'Key type' with a dropdown menu set to 'Authentication Key'; and 'Key' with a large text area containing the SSH key 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDKDRnB2E4O8u7l0tf3G'. At the bottom of the form is a green 'Add SSH key' button.

8. Finally, to let git know who you are use the following command. User your own email address

```
ardent@ardent:~/Github/summerclass$ git config --global user.email "ardent.sharma@islingtoncollege.edu.np"
ardent@ardent:~/Github/summerclass$
```



## Pyspark Setup

Let us now install pyspark in our system in a virtual environment

1. Create a new directory Workspace and cd into that directory. Create this directory in your home

```
ardent@ardent:~$ mkdir Workspace && cd Workspace
ardent@ardent:~/Workspace$
```

2. Clone the following git repo inside the Workspace directory: it contains script to install pyspark

git clone <https://github.com/neotheobserver/pyspark-install.git> pyspark

```
ardent@ardent:~/Workspace$ git clone git@github.com:neotheobserver/pyspark-install.git pyspark
Cloning into 'pyspark'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
ardent@ardent:~/Workspace$
```

3. Now cd into the newly created pyspark directory and ls to verify if the files have been downloaded

```
ardent@ardent:~/Workspace$ cd pyspark/
ardent@ardent:~/Workspace/pyspark$ ls
pyspark-installation.sh  README.md
ardent@ardent:~/Workspace/pyspark$
```

4. Run the installation script using below command

```
bash pyspark-installation.sh 3.3.1 1.12.99
```

Here, 3.3.1 is the pyspark version and 1.12.99 is the aws bundle. Make sure the same version is installed or else there might be dependency issues

```
ardent@ardent:~/Workspace/pyspark$ bash pyspark-installation.sh 3.3.1 1.12.99
Python installed version: 3.12
Activating virtual environment....
Installing Pyspark...
Collecting pyspark==3.3.1
```



If there is any error during installation make sure the internet is working fine. If the issue persists some tool might be missing: install them using apt

```
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total     Spent    Left     Speed
100  850k    100  850k    0      0   557k      0  0:00:01  0:00:01 --:--:--  557k
Downloading aws bundle: https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk-bundle/1
2.99.jar
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total     Spent    Left     Speed
100  235M    100  235M    0      0  12.6M      0  0:00:18  0:00:18 --:--:-- 13.9M
JAVA_HOME variable is not set
Make sure the java version installed is compatible with the pyspark version installed
ardent@ardent:~/Workspace/pyspark$
```

Most of us will get the above output. Notice JAVA\_HOME variable is not set issue shown here. We have already installed openjdk-11-jdk before, we need to configure it in our .bashrc file. .bashrc is located in the home directory and it is the file that is run by the system everytime we open a terminal. So making the changes there will be reflected in every terminal. Let us first figure out the java installation directory

5. Use the below command to find the installed directory. We will take the path before the bin

`readlink -f $(which java)`

```
ardent@ardent:~/Workspace/pyspark$ readlink -f $(which java)
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
ardent@ardent:~/Workspace/pyspark$
```

Copy the path before bin using CTRL+SHIFT+C

6. Change directory to home and open the .bashrc file in a text editor of your choice. It could either be vim, nano, or vscode

The file should be opened in the text editor

```
ardent@ardent:~/Workspace/pyspark$ cd ~
ardent@ardent:~$ code .bashrc
ardent@ardent:~$
```

7. Go to the end of the file and add the following line

```
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

Make sure there is no spaces in either the path or the assignment variables and operators

```
$ .bashrc x
home > ardent > $ .bashrc
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
120
121
```

Save the file and close the text editor. The changes will be applied after we restart the terminal. Alternatively, we can tell the bash to reload the file using the source command as below

```
ardent@ardent:~/Workspace/pyspark$ cd ~
ardent@ardent:~$ code .bashrc
ardent@ardent:~$ source .bashrc
ardent@ardent:~$
```

8. Let us move back to the directory we setup pyspark virtual environment and activate the virtual environment.

```
cd ~/Workspace/pyspark/
source venv/bin/activate
```

Since we have installed pyspark in a virtual environment we need to activate it everytime before running any pyspark code

```
ardent@ardent:~$ cd Workspace/pyspark/
ardent@ardent:~/Workspace/pyspark$ ls
pyspark-installation.sh  README.md  venv
ardent@ardent:~/Workspace/pyspark$ source venv/bin/activate
(venv) ardent@ardent:~/Workspace/pyspark$
```

9. Enter pyspark in the terminal. You should get the below output which verifies that pyspark has been installed properly

```
ardent@ardent:~/Workspace/pyspark$ source venv/bin/activate  
(venv) ardent@ardent:~/Workspace/pyspark$ pyspark  
Python 3.12.3 (main, Jun 18 2025, 17:59:45) [GCC 13.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
25/06/24 17:37:09 WARN Utils: Your hostname, ardent resolves to a loopback address: 127.0.1.1; us  
ead (on interface enpi0)  
25/06/24 17:37:09 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
25/06/24 17:37:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform..  
asses where applicable  
Welcome to
```



```
 version 3.3.1  
  
Using Python version 3.12.3 (main, Jun 18 2025 17:59:45)  
Spark context Web UI available at http://192.168.122.50:4040  
Spark context available as 'sc' (master = local[*], app id = local-1750765930730).  
SparkSession available as 'spark'.  
>>>
```

Congratulations! Your system is ready to run pyspark jobs and you can now perform git operations from your local system using ssh authentication.