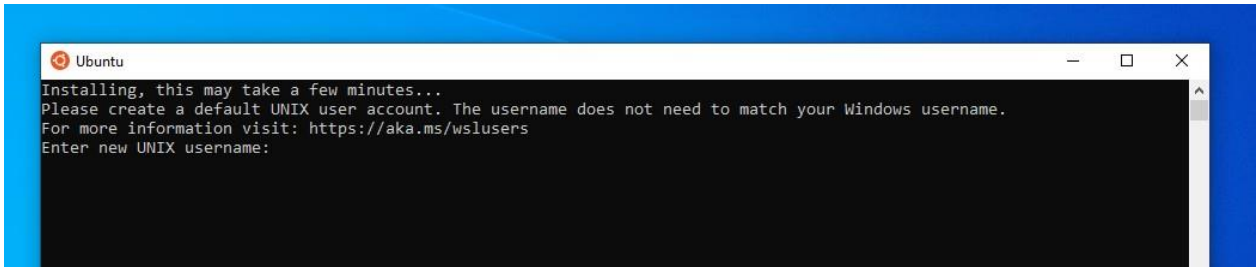
WSL Installation (**only for windows users**)

1. You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11.

   To check your Windows version and build number, select **Windows logo key + R**, type **winver**, select OK.
2. Enter the following command in an **administrator** PowerShell or Windows Command Prompt and then restart your machine.
   `wsl –install` This command will enable the features necessary to run WSL and install the Ubuntu distribution of Linux.
   After restart, Ubuntu should start installing

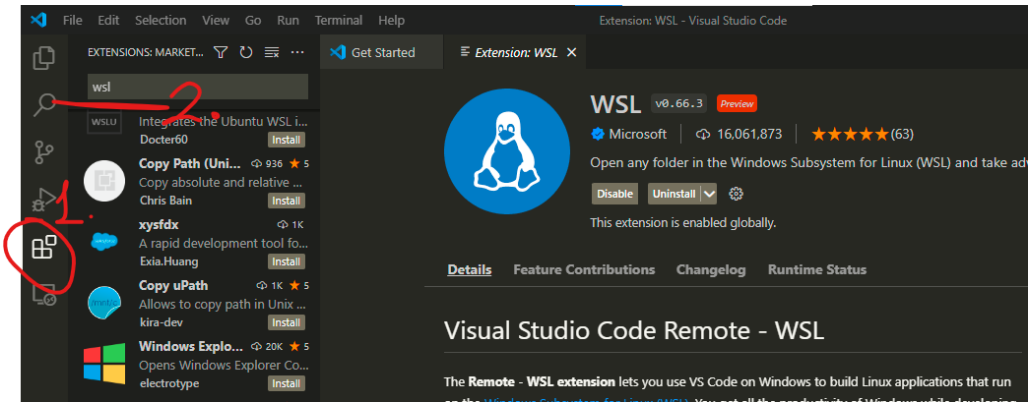


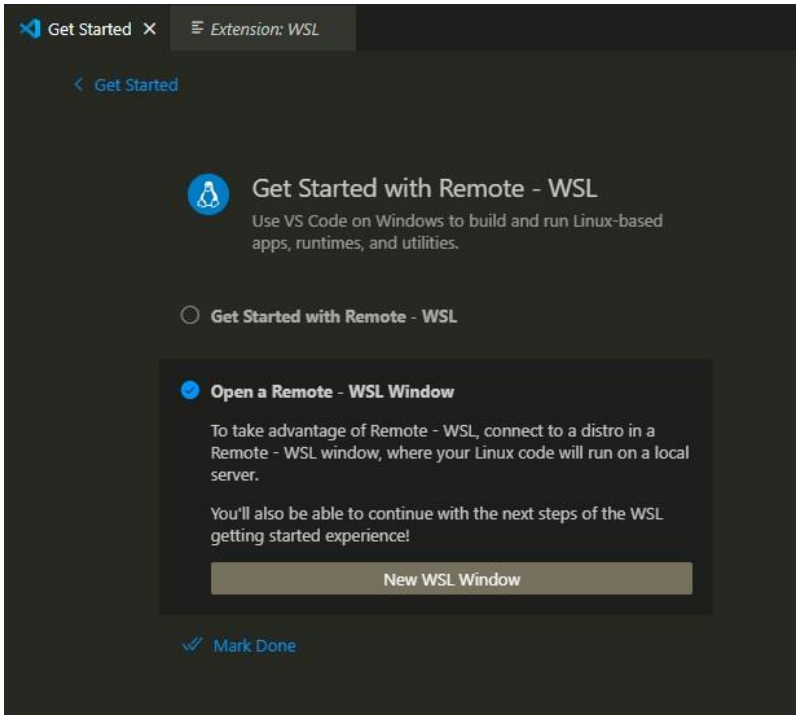
3. Create an account on Ubuntu by entering username and password.
   While entering your password you won't see any changes in prompt. And please **DO NOT FORGET** this password.
   You can also download Windows Terminal from here for a better terminal on windows.

Vscode

1. Download vscode from here if you do not have already installed.
   Windows users download vscode to windows.

2. (**this step is only for windows users**)
   - Open vscode, then open extensions menu (number 1 in picture below).
   - Search for `wsl` (number 2 in picture below).
   - Install the extension.





In the **Get Started** page of WSL, click **Open a Remote** (this will install vscode on ubuntu, so that you can later open files in ubuntu from vscode installed on windows).

Github account and SSH key (**for everybody**)

1. Open a github account if you don't have it already. (You can use your sabanci usernames if you are opening a new account, if you already have a github account that's also fine.)
2. Fill out this form, so we can match your github accounts with sabanci accounts.

Adding SSH key to github account

Windows users, for the following part "terminal" means ubuntu terminal (please **DO NOT** use PowerShell or Windows Command Prompt)

1. Open terminal
2. Paste the command below, substituting in your GitHub email address.
   ssh-keygen -t ed25519 -C "your_email@example.com"

   Note: If you are using a legacy system that doesn't support the Ed25519 algorithm, use:
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

   This creates a new SSH key, using the provided email as a label.
   > Generating public/private algorithm key pair.

3. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.
   > Enter a file in which to save the key (/home/umit/.ssh/id_ed25519):[Press enter]

4. At the prompt, type a secure passphrase. (Also **DO NOT FORGET** this)
   > Enter passphrase (empty for no passphrase): [Type a passphrase]
   > Enter same passphrase again: [Type passphrase again]

5. Set up the necessary environment using command:

   eval `ssh-agent -s`

6. Add your SSH private key to the ssh-agent.
   ssh-add ~/.ssh/id_ed25519
   This is the **file from step 3**, if you changed that change file location here accordingly.
   This command will ask for a passphrase, it is the **passphrase you typed at step 4**.

7. Open the SSH public key you just created, and copy it
   ~/.ssh/id_ed25519.pub
   You can find this file at the location you decided at **step 3.**

8. Open github from the web browser and login.
   a. In the upper-right corner of any page, click your profile photo, then click **Settings**.
   b. In the "Access" section of the sidebar, click **SSH and GPG keys**.
   c. Click **New SSH key** or **Add SSH key**.
   d. In the "Title" field, add a descriptive label for the new key.

e. Select the type of key as **authentication**.
f. Paste your key (SSH public key you copied at **step 6**) into the "Key" field.
g. Click Add SSH key.

9. Testing your ssh connection to github
    a. Open terminal
    b. Enter the following
    ssh -T git@github.com
    > *Attempts to ssh to GitHub*

    You may see a warning like this:
    > The authenticity of host 'github.com (IP ADDRESS)' can't be established.
    > RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
    > Are you sure you want to continue connecting (yes/no)?

    Type “yes” and press Enter

    c. If you see a similar response to the following, you have successfully set SSH with github.
    > *Hi username! You've successfully authenticated, but GitHub does not*
    > *provide shell access.*