

Guide to install Docker

- [Installation](#)
 - [Windows](#)
 - [Enable WSL](#)
 - [Install WSL](#)
 - [Install Docker and Docker Desktop on your Windows machine](#)
 - [Activate the WSL Integration](#)
- [Configuration](#)
 - [Windows](#)
 - [Generate a New SSH Key](#)
 - [Add Your Key to the SSH Agent](#)
 - [Add Your Key to GitHub](#)
 - [Display Your Branch Name in the Terminal](#)
- [\(**Optional\)](#)
- [Step 1 \[PassphareAsking\]](#) [How to stop git asking you for passphrase](#)
 - [Step 2 \[PassphareAsking\]](#)
- [Step 3 \[PassphareAsking\]](#)
- [Step 4 \[PassphareAsking\]](#)
- [Step 5 \[PassphareAsking\]](#)
- [Step 6 \[PassphareAsking\]](#)
- [Step 7 \[PassphareAsking\]](#)
- [Step 8 \[PassphareAsking\]](#)
- [Step 9 \[PassphareAsking\]](#)
- [Step 10 \[PassphareAsking\]](#)
- [Step 11 \[PassphareAsking\]](#)
- [Step 12 \[PassphareAsking\]](#)
- [Step 13 \[PassphareAsking\]](#)
- [Step 14 \[PassphareAsking\]](#)
- [Step 15 \[PassphareAsking\]](#)
- [Step 16 \[PassphareAsking\]](#)
- [Step 17 \[PassphareAsking\]](#)
- [Step 18 \[PassphareAsking\]](#)
- [Step 19 \[PassphareAsking\]](#)
- [Step 20 \[PassphareAsking\]](#)
- [Step 21 \[PassphareAsking\]](#)
- [Step 22 \[PassphareAsking\]](#)
- [Step 23 \[PassphareAsking\]](#)
- [Step 24 \[PassphareAsking\]](#)
- [Step 25 \[PassphareAsking\]](#)
- [Step 26 \[PassphareAsking\]](#)
- [Step 27 \[PassphareAsking\]](#)
- [Step 28 \[PassphareAsking\]](#)
- [Step 29 \[PassphareAsking\]](#)
- [Step 30 \[PassphareAsking\]](#)
- [Step 31 \[PassphareAsking\]](#)
- [Step 32 \[PassphareAsking\]](#)
- [Step 33 \[PassphareAsking\]](#)
- [Step 34 \[PassphareAsking\]](#)
- [Step 35 \[PassphareAsking\]](#)
- [Step 36 \[PassphareAsking\]](#)
- [Step 37 \[PassphareAsking\]](#)
- [Step 38 \[PassphareAsking\]](#)
- [Step 39 \[PassphareAsking\]](#)
- [Step 40 \[PassphareAsking\]](#)
- [Step 41 \[PassphareAsking\]](#)
- [Step 42 \[PassphareAsking\]](#)
- [Step 43 \[PassphareAsking\]](#)
- [Step 44 \[PassphareAsking\]](#)
- [Step 45 \[PassphareAsking\]](#)
- [Step 46 \[PassphareAsking\]](#)
- [Step 47 \[PassphareAsking\]](#)
- [Step 48 \[PassphareAsking\]](#)
- [Step 49 \[PassphareAsking\]](#)
- [Step 50 \[PassphareAsking\]](#)
- [Step 51 \[PassphareAsking\]](#)
- [Step 52 \[PassphareAsking\]](#)
- [Step 53 \[PassphareAsking\]](#)
- [Step 54 \[PassphareAsking\]](#)
- [Step 55 \[PassphareAsking\]](#)
- [Step 56 \[PassphareAsking\]](#)
- [Step 57 \[PassphareAsking\]](#)
- [Step 58 \[PassphareAsking\]](#)
- [Step 59 \[PassphareAsking\]](#)
- [Step 60 \[PassphareAsking\]](#)
- [Step 61 \[PassphareAsking\]](#)
- [Step 62 \[PassphareAsking\]](#)
- [Step 63 \[PassphareAsking\]](#)
- [Step 64 \[PassphareAsking\]](#)
- [Step 65 \[PassphareAsking\]](#)
- [Step 66 \[PassphareAsking\]](#)
- [Step 67 \[PassphareAsking\]](#)
- [Step 68 \[PassphareAsking\]](#)
- [Step 69 \[PassphareAsking\]](#)
- [Step 70 \[PassphareAsking\]](#)
- [Step 71 \[PassphareAsking\]](#)
- [Step 72 \[PassphareAsking\]](#)
- [Step 73 \[PassphareAsking\]](#)
- [Step 74 \[PassphareAsking\]](#)
- [Step 75 \[PassphareAsking\]](#)
- [Step 76 \[PassphareAsking\]](#)
- [Step 77 \[PassphareAsking\]](#)
- [Step 78 \[PassphareAsking\]](#)
- [Step 79 \[PassphareAsking\]](#)
- [Step 80 \[PassphareAsking\]](#)
- [Step 81 \[PassphareAsking\]](#)
- [Step 82 \[PassphareAsking\]](#)
- [Step 83 \[PassphareAsking\]](#)
- [Step 84 \[PassphareAsking\]](#)
- [Step 85 \[PassphareAsking\]](#)
- [Step 86 \[PassphareAsking\]](#)
- [Step 87 \[PassphareAsking\]](#)
- [Step 88 \[PassphareAsking\]](#)
- [Step 89 \[PassphareAsking\]](#)
- [Step 90 \[PassphareAsking\]](#)
- [Step 91 \[PassphareAsking\]](#)
- [Step 92 \[PassphareAsking\]](#)
- [Step 93 \[PassphareAsking\]](#)
- [Step 94 \[PassphareAsking\]](#)
- [Step 95 \[PassphareAsking\]](#)
- [Step 96 \[PassphareAsking\]](#)
- [Step 97 \[PassphareAsking\]](#)
- [Step 98 \[PassphareAsking\]](#)
- [Step 99 \[PassphareAsking\]](#)
- [Step 100 \[PassphareAsking\]](#)

Installation

Windows

We highly recommend running Docker from WSL (Windows Subsystem for Linux) to avoid performance issues. Follow the links below for additional information.

Enable WSL

In some cases WSL is not enabled to enable it in PowerShell run this command.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Loading personal and system profiles took 721ms.
PS C:\Windows\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.19041.844

Image Version: 10.0.19044.2006
```

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

```
PS C:\Windows\system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.19041.844

Image Version: 10.0.19044.2006

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
PS C:\Windows\system32>
```

```
Enable-WindowsOptionalFeature -Online -FeatureName
VirtualMachinePlatform -NoRestart
```

```
PS C:\Windows\system32> Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform -NoRestart
WARNING: Restart is suppressed because NoRestart is specified.

Path      :
Online    : True
RestartNeeded : True
```

Now restart your entire laptop.

Install WSL

Some Windows versions have WSL 1 instead of WSL 2. Run the following command in PowerShell to switch to WSL 2: `wsl --set-default-version 2`.

```
Administrator: Windows PowerShell

PS C:\WINDOWS\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/ws12
The operation completed successfully.
PS C:\WINDOWS\system32>
```

Confirm your version by running `wsl --status`. You should see "Default Version: 2."

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Loading personal and system profiles took 642ms.
PS C:\WINDOWS\system32> wsl --status
Default Version: 2

Windows Subsystem for Linux was last updated on 7/27/2022
The Windows Subsystem for Linux kernel can be manually updated with 'wsl --update', but automatic updates cannot occur due to your system settings.
To receive automatic kernel updates, please enable the Windows Update setting: 'Receive updates for other Microsoft products when you update Windows'.
For more information please visit https://aka.ms/wsl2kernel.

Kernel version: 5.10.16
```

Update your kernel version by running `wsl --update`. This may take a few minutes. Once complete, you should see the new version to be updated. In this case, we're updating from 5.10.16 to 5.10.102. (if receive an error, try running powershell not as admin)

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Loading personal and system profiles took 642ms.
PS C:\WINDOWS\system32> wsl --status
Default Version: 2

Windows Subsystem for Linux was last updated on 7/27/2022
The Windows Subsystem for Linux kernel can be manually updated with 'wsl --update', but automatic updates cannot occur due to your system settings.
To receive automatic kernel updates, please enable the Windows Update setting: 'Receive updates for other Microsoft products when you update Windows'.
For more information please visit https://aka.ms/wsl2kernel.

Kernel version: 5.10.16
PS C:\WINDOWS\system32> wsl --update
Checking for updates...
Downloading updates...
Installing updates...
This change will take effect on the next full restart of WSL. To force a restart, please run 'wsl --shutdown'.
Kernel version: 5.10.102.1
PS C:\WINDOWS\system32>
```

Finish the update by running `wsl --shutdown`. Wait about a minute, then run `wsl --status` again to verify your kernel version.

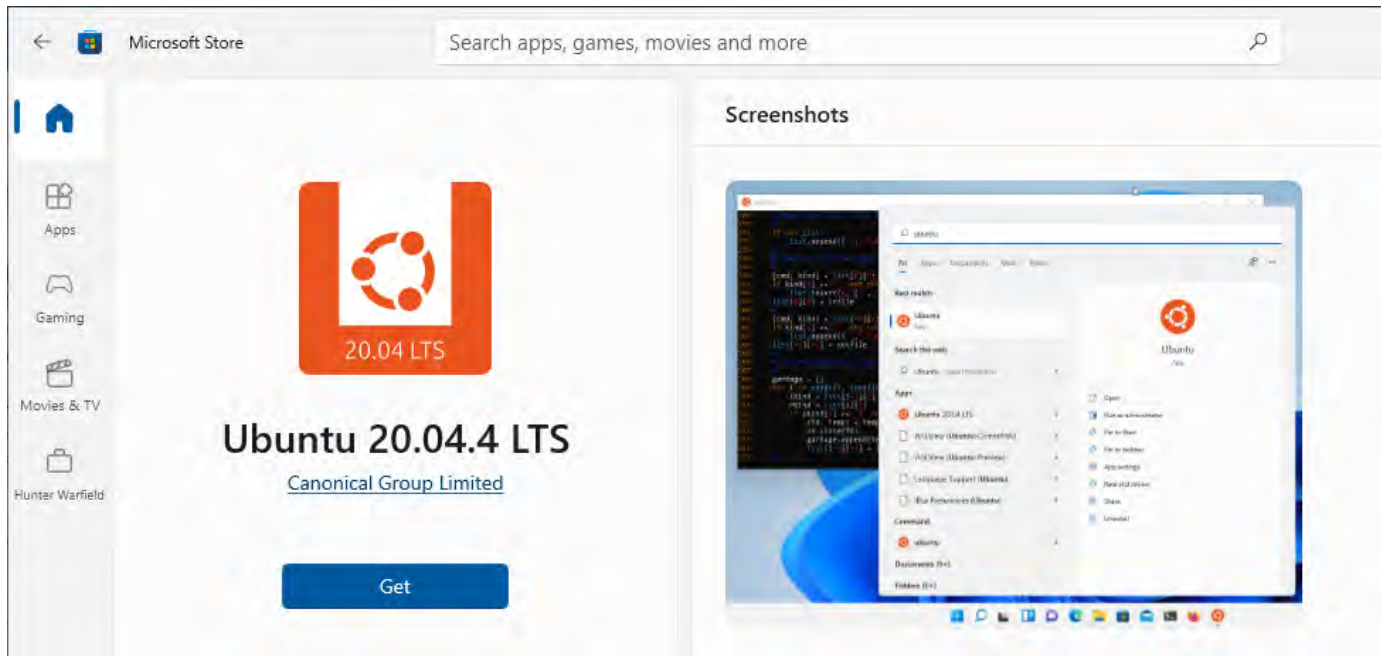
```
Administrator: Windows PowerShell

PS C:\WINDOWS\system32> wsl --status
Default Version: 2

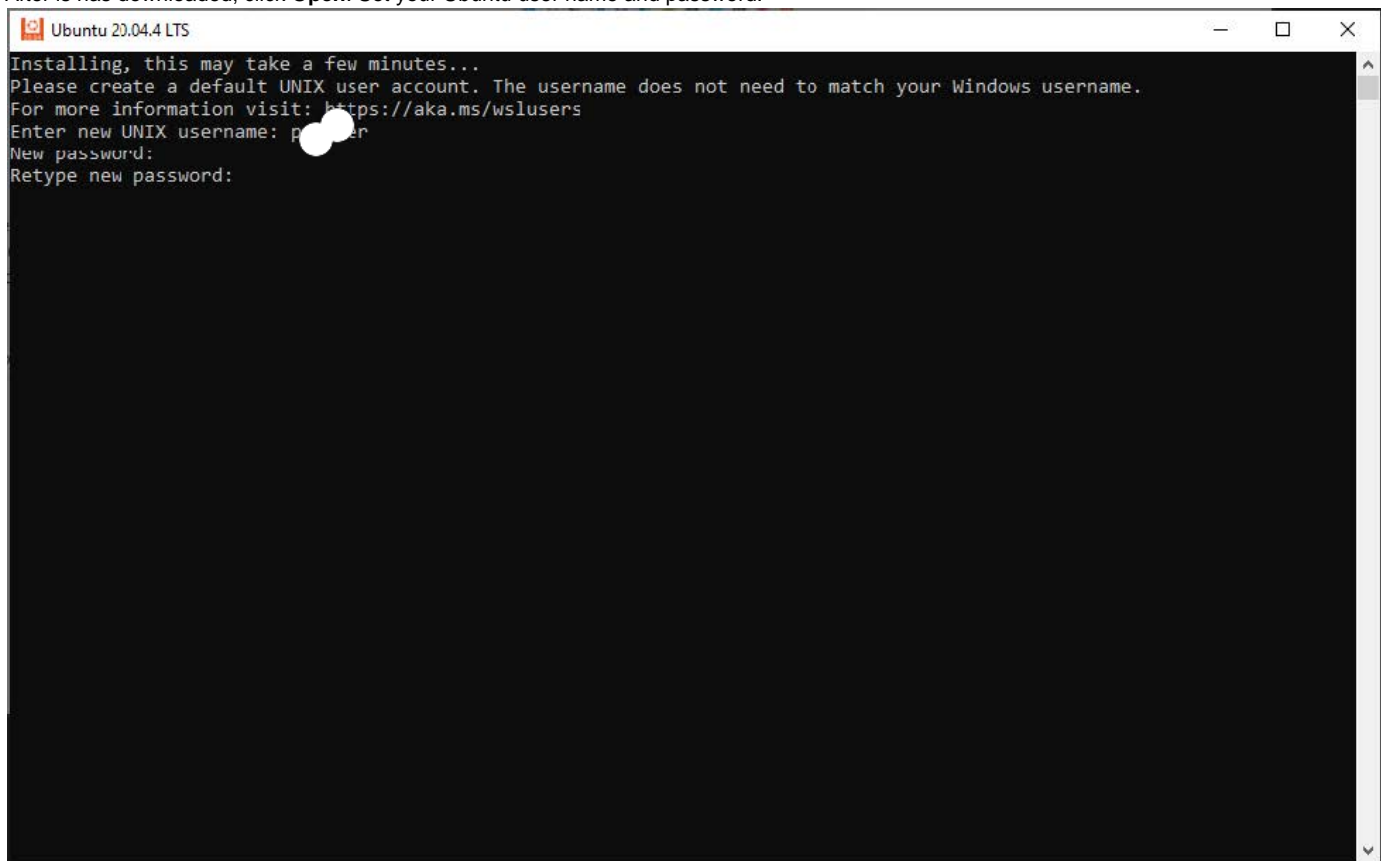
Windows Subsystem for Linux was last updated on 7/27/2022
The Windows Subsystem for Linux kernel can be manually updated with 'wsl --update', but automatic updates cannot occur due to your system settings.
To receive automatic kernel updates, please enable the Windows Update setting: 'Receive updates for other Microsoft products when you update Windows'.
For more information please visit https://aka.ms/wsl2kernel.

Kernel version: 5.10.102.1
PS C:\WINDOWS\system32>
```

Download the latest Ubuntu LTS version from the Windows store.



After it has downloaded, click **Open**. Set your Ubuntu user name and password.



Run `sudo apt-get update` to apply any pending updates.

Install Docker and Docker Desktop on your Windows machine

Follow this guide: [Install Docker Desktop on Windows](#).

Here's a walk-through of Step 5 (adding your user to the `docker-users` group), since it differs a bit for those of us using Resident Interface laptops.

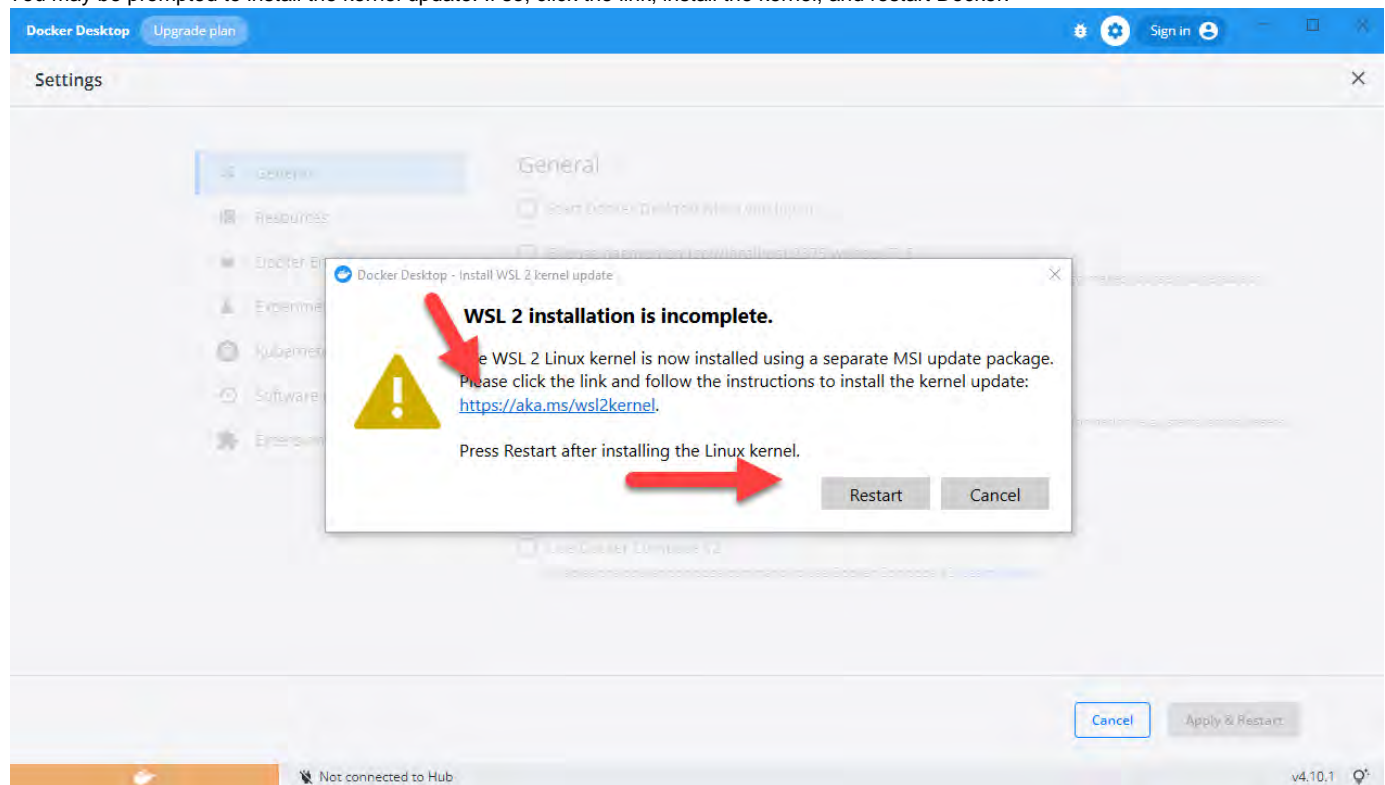
Click **OK** again, then close Computer Management. Log out of Windows and back in for the permissions to take effect

Activate the WSL Integration

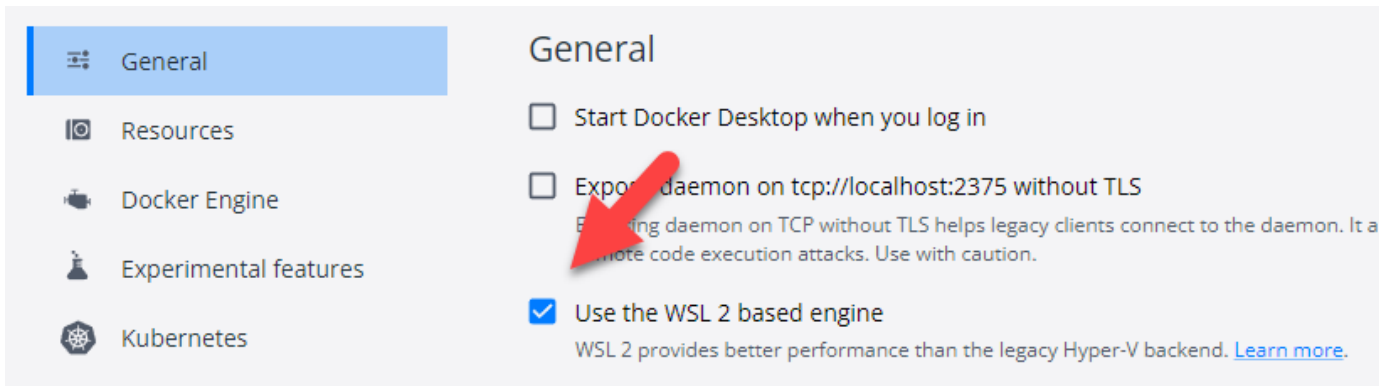
Start Docker Desktop from the Windows Start menu.

From the Docker menu, select **Settings > General**.

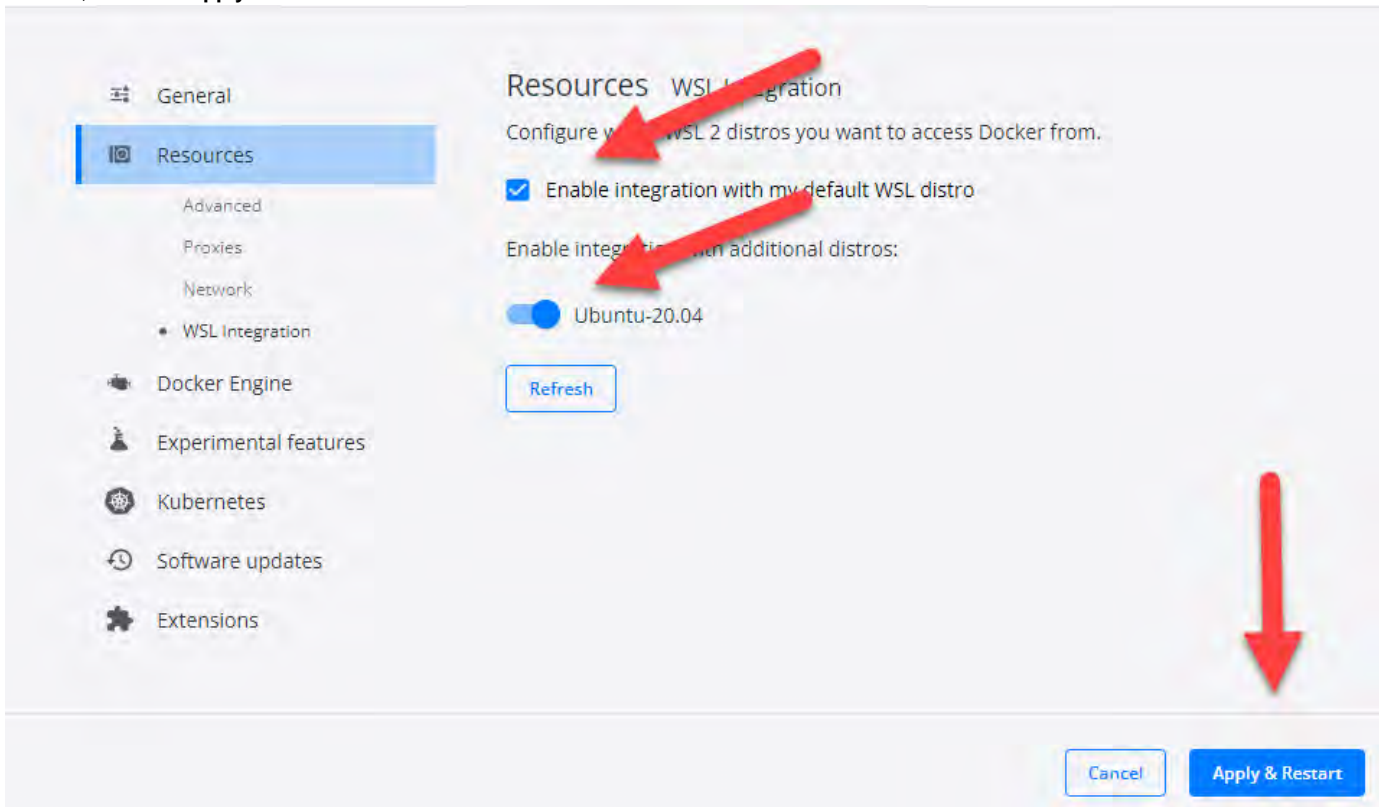
You may be prompted to install the kernel update. If so, click the link, install the kernel, and restart Docker.



Click **Settings > General** and verify "Use the WSL 2 based engine" is checked



Next, click **Resources > WSL Integration** and verify “Enable integration with my default WSL distro” is checked. Select the Linux distribution you installed, then click **Apply & Restart**.

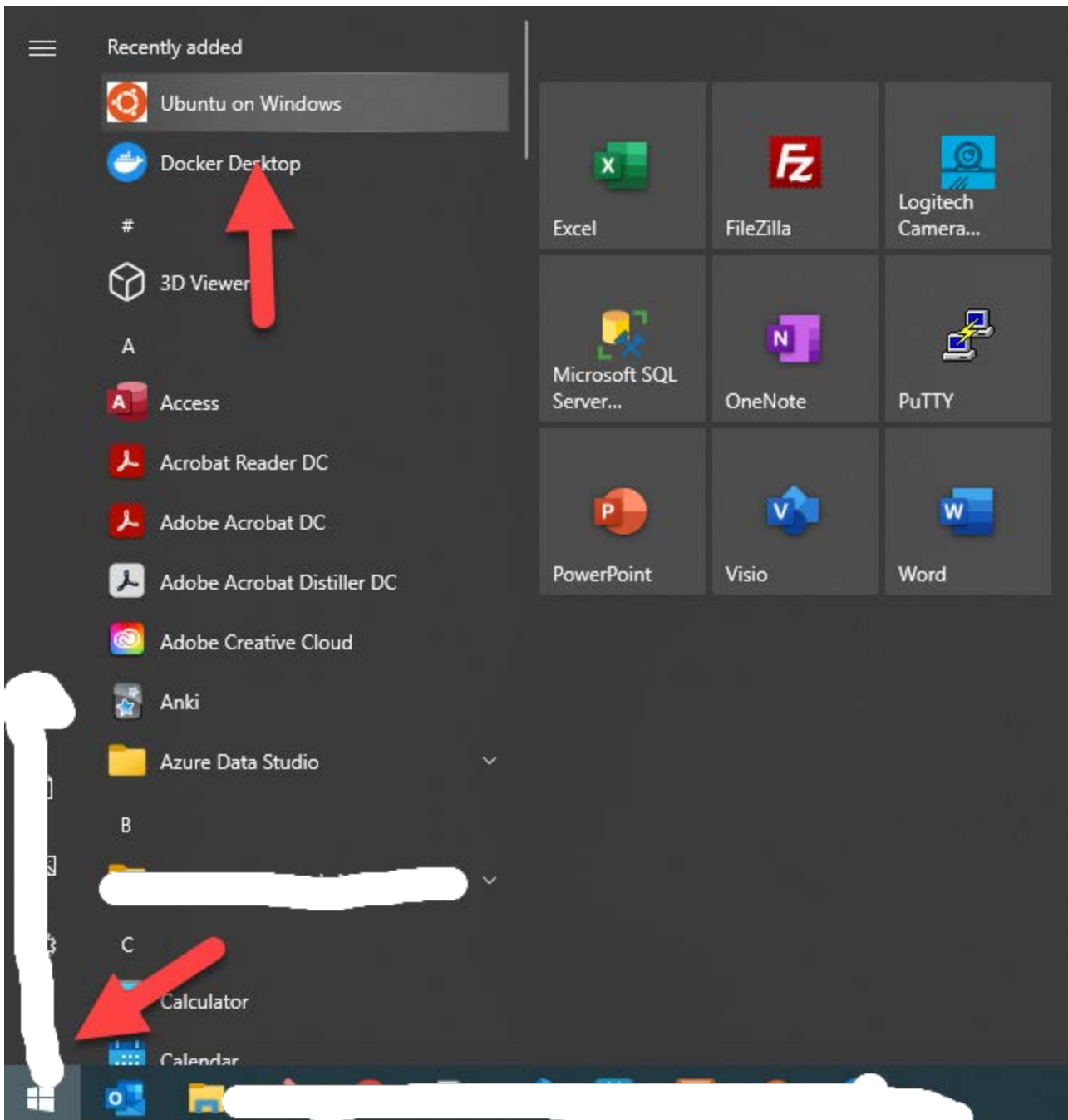


Configuration

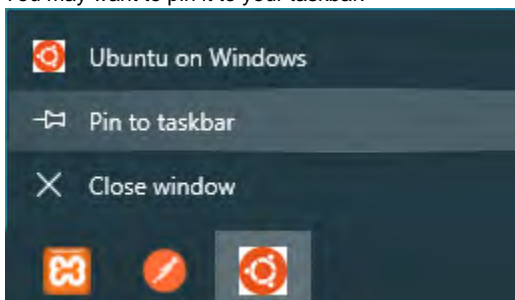
Windows

For performance reasons, you'll want to store your local copies of repositories in Linux, rather than Windows.

To open your WSL instance, click **Start**, then **Ubuntu on Windows**.



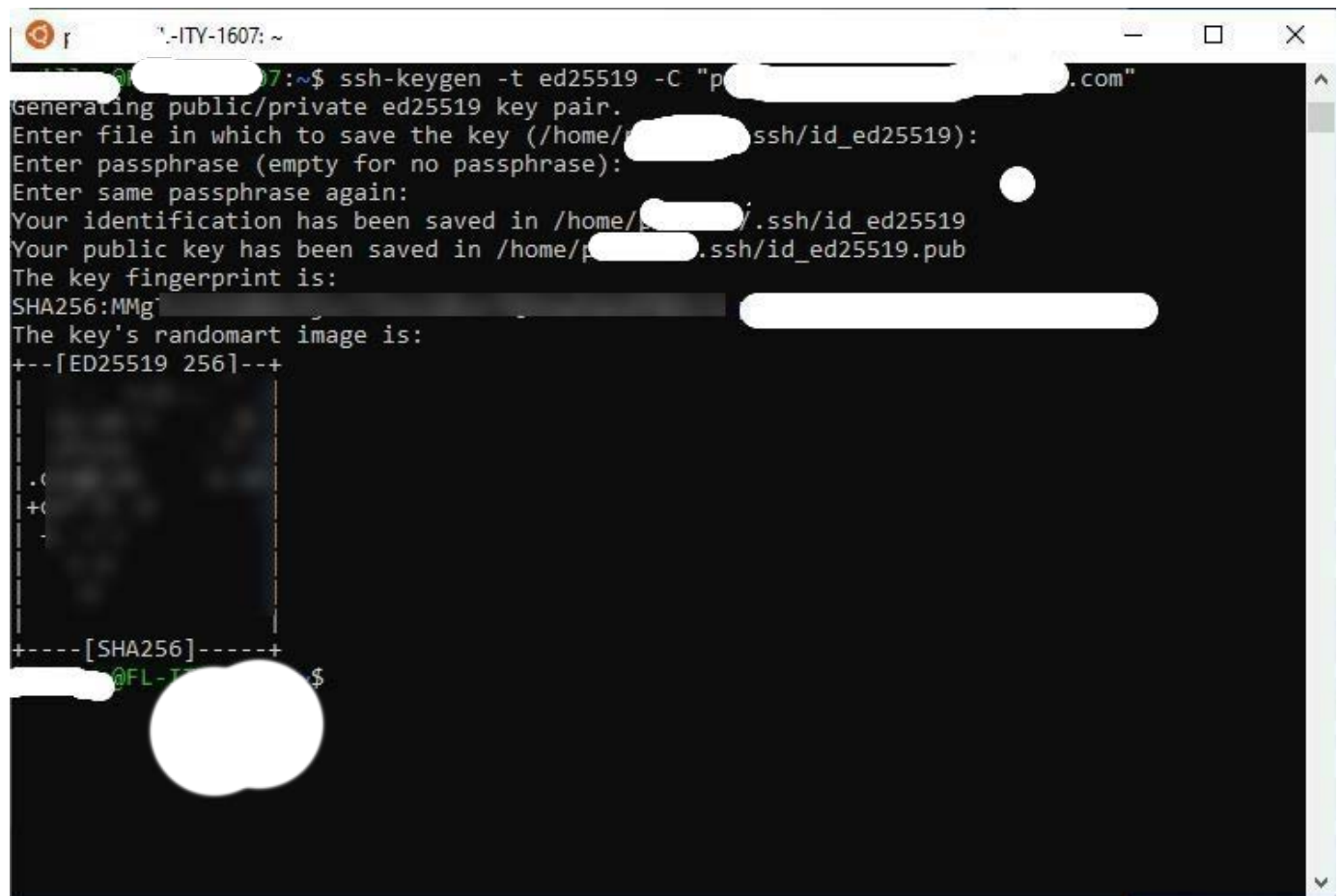
You may want to pin it to your taskbar.



Generate a New SSH Key

Source: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Run `ssh-keygen -t ed25519 -C "youremail@company.com"` to generate a new key, substituting your email address. Accept or alter the options as you see fit, although we ask that you use a passphrase. When you're done, the output should look something like this:

A terminal window with a dark background and light-colored text. The window title is ".-ITY-1607: ~". The prompt is "7:~\$". The command entered is "ssh-keygen -t ed25519 -C 'p[redacted].com'". The output shows the generation of an ed25519 key pair, prompts for a file name and passphrase, and the saving of the key files. It also displays the key's fingerprint and a randomart image. The randomart image is a square with a grid of characters. The terminal ends with a prompt "7:~\$".

```
7:~$ ssh-keygen -t ed25519 -C "p[redacted].com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/[redacted]/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/[redacted]/.ssh/id_ed25519
Your public key has been saved in /home/[redacted]/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:MMg[redacted]
The key's randomart image is:
+--[ED25519 256]--+
|
|
|
|
|
|
|
|
|
|
+-----[SHA256]-----+
[redacted]@FL-T[redacted]$
```

Add Your Key to the SSH Agent

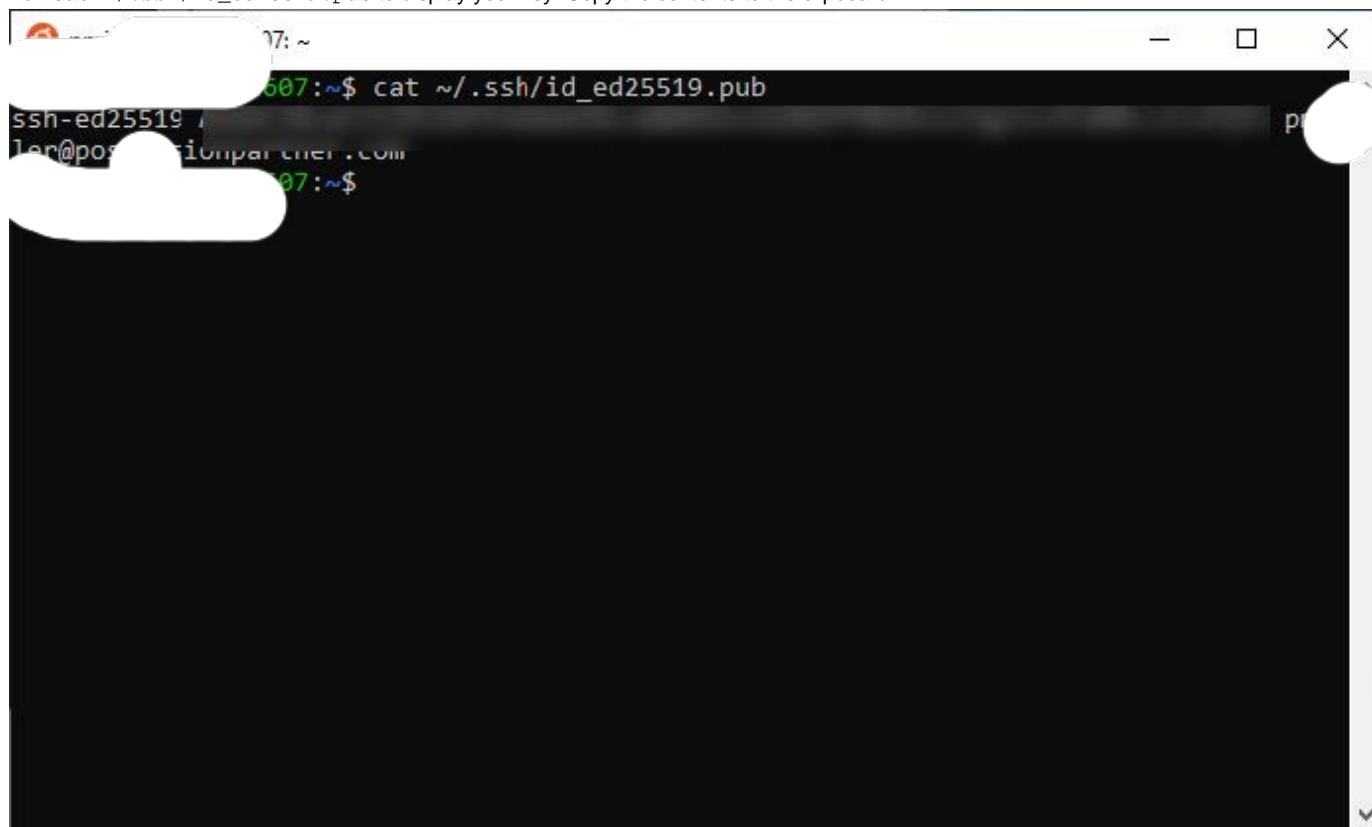
Start the SSH agent by running `eval "$(ssh-agent -s)"`.

Add your SSH private key to the ssh-agent by running `ssh-add ~/.ssh/id_ed25519`. If you created your key with a different name, or if you are adding an existing key that has a different name, replace `id_ed25519` in the command with the name of your private key file.

Add Your Key to GitHub

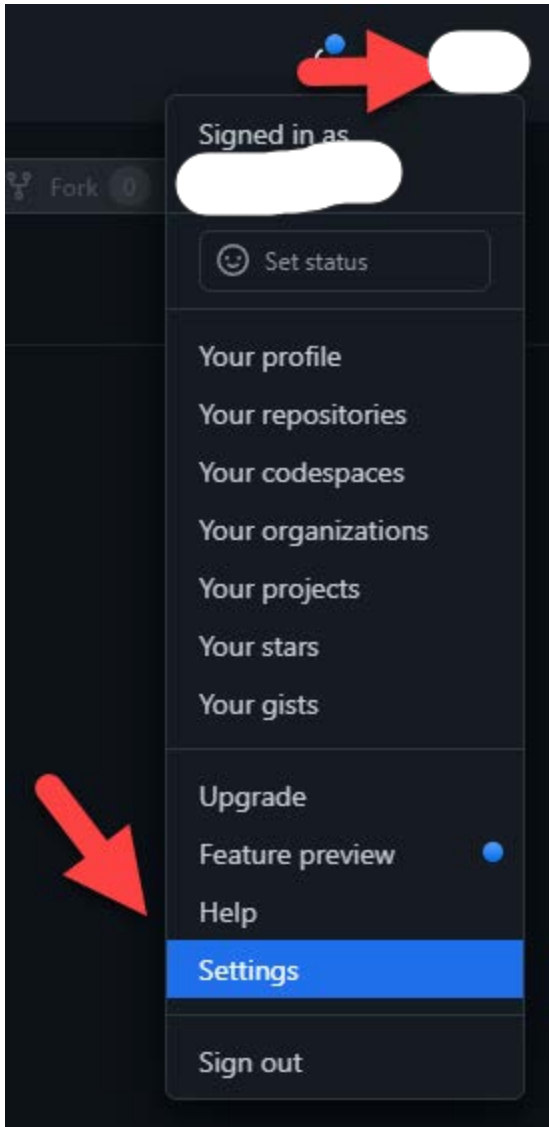
Source: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

Run `cat ~/.ssh/id_ed25519.pub` to display your key. Copy the contents to the clipboard.

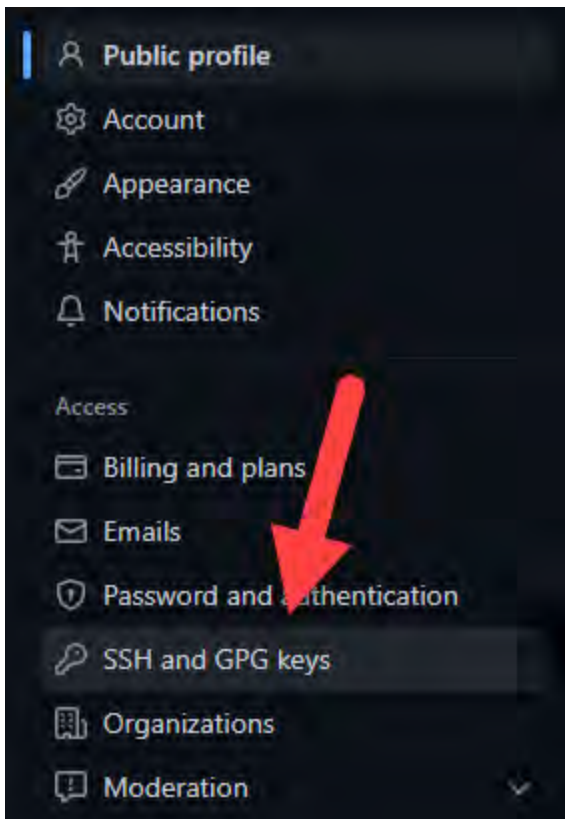
A terminal window with a black background and green text. The prompt is '607:~\$'. The command 'cat ~/.ssh/id_ed25519.pub' has been entered. The output shows the SSH key content, which is partially obscured by a white redaction box. The terminal window has a title bar with standard window controls (minimize, maximize, close) and a small icon on the left. The window is titled '607:~'.

Log into your GitHub account.

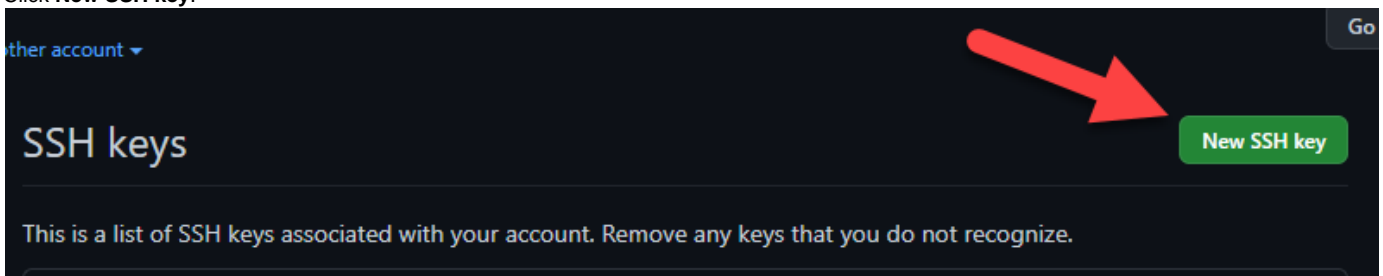
Click your icon, then click **Settings**.



In the "Access" section of the sidebar, click **SSH and GPG keys**.



Click **New SSH key**.



In the "Title" field, add a descriptive label for the new key. Paste your key into the "Key" field, then click **Add SSH key**.

SSH keys / Add new

Title

[redacted] WSL

Key

```
ssh-ed25519 [redacted]@[redacted].com
```

Add SSH key

Create a directory to store your source code. In this case, we're creating a directory named `source` in our user's home (`~`) directory by running `mkdir ~/source`:

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Jul 27 06:37:00 EDT 2022

System load:  0.52      Users logged in:      0
Usage of /home: unknown  IPv4 address for eth2: 192.168.1.217
Memory usage: 45%      IPv6 address for eth2: 2603:9000:8c00:60d4::723
Swap usage:   0%       IPv4 address for eth3: 172.24.144.1
Processes:    7

0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/username/.ushlogin file.
root@507:~# mkdir ~/source
root@507:~#
```

Display Your Branch Name in the Terminal

Get your current prompt setting by running `echo $PS1`.

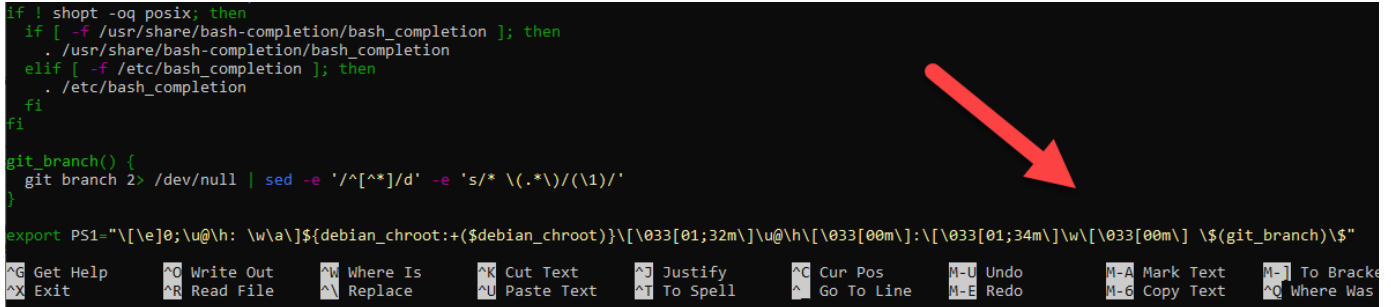
```
root@507: ~/source/platform
root@507:~/source/platform$ echo $PS1
\[\033[01;32m\]\u@h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$
root@507:~/source/platform$
```

Copy the value to your clipboard.

Execute `nano ~/.bashrc` to open your bash configuration file. Add the following function at the bottom of the file:

```
git_branch() {
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
```

Next, create a line that begins with `export PS1=`. Paste the value of `PS1` you copied earlier. Add `(git_branch)\$` to the end. Your final value should look something like this:



```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

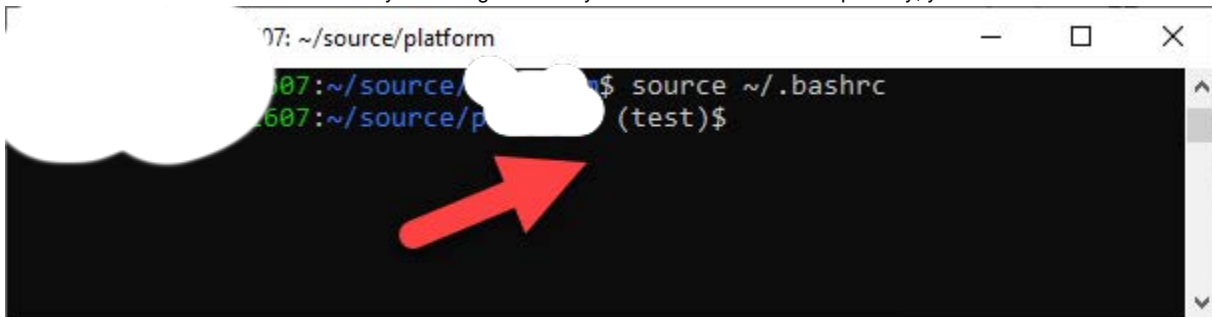
git_branch() {
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}

export PS1="\[\e]0;\u@h: \w\[\e]0m\${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\] \$(git_branch)\$"
```

You'll see that I added a space between `\[\00m\]` and `\$` to make the display a little easier to read.

Save your changes and close the file.

Execute `source ~/.bashrc` to reload your configuration. If you are in a folder with a repository, you should now see the branch name.



```
07: ~/source/platform
07:~/source/platform$ source ~/.bashrc
07:~/source/platform (test)$
```

(**Optional)

Step 1 [PassphareAsking] How to stop git asking you for passphrase

1. Star ubuntu shell
2. Edit your nano `~/.bashrc` file
3. Add the following lines to the file at the end

```

SSH_ENV=$HOME/.ssh/environment

# start the ssh-agent
function start_agent {
    echo "Initializing new SSH agent..."
    # spawn ssh-agent
    /usr/bin/ssh-agent | sed 's/^echo/#echo/' > ${SSH_ENV}
    echo succeeded
    chmod 600 ${SSH_ENV}
    . ${SSH_ENV} > /dev/null
    /usr/bin/ssh-add
}

if [ -f "${SSH_ENV}" ]; then
    . ${SSH_ENV} > /dev/null
    ps -ef | grep ${SSH_AGENT_PID} | grep ssh-agent$ > /dev/null || {
        start_agent;
    }
else
    start_agent;
fi

```

Step 2 [PassphareAsking]

Sometimes is a little annoying to keep linux asking you for passphrase, in the scenario you like to enter your credentials once you may open the file called `nano ~/.ssh/config` and then add the following lines

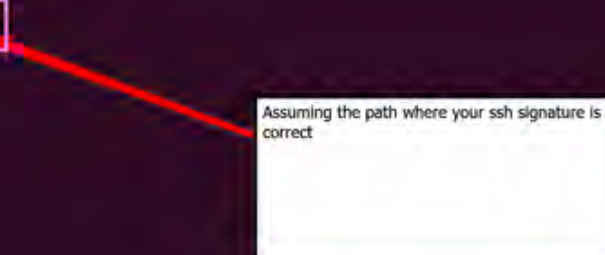
```

Host *
    IgnoreUnknown AddKeysToAgent,UseKeychain
    AddKeysToAgent yes
    UseKeychain yes
    IdentityFile ~/.ssh/id_ed25519

```



```
GNU nano 6.2 /home/lsilva/.ssh/config *
host *
  IgnoreUnknown AddKeysToAgent,UseKeychain
  AddKeysToAgent yes
  UseKeychain yes
  IdentityFile ~/.ssh/id_ed25519
```



Assuming the path where your ssh signature is correct

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo

Breaking news. Unfolding now.

Once you save the file, run one more time the command to tell git our signature.

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

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

Check if the key is added (parameter is a lowercase L):

```
$ ssh-add -l
```

```
lsilva@MSI:~/source/... (test)$git pull
Enter passphrase for key '/home/lsilva/.ssh/id_ed25519':

lsilva@MSI:~/source/... (test)$eval "$(ssh-agent -s)" ①
Agent pid 963653
lsilva@MSI:~/source/... (test)$ssh-add ~/.ssh/id_ed25519 ②
Enter passphrase for /home/lsilva/.ssh/id_ed25519:
Identity added: /home/lsilva/.ssh/id_ed25519 (lsilva@msi)
lsilva@MSI:~/source/... (test)$git pull ③
remote: Enumerating objects: 24, done.
reused 0
Un
Fr
Up
Fa
1 file changed, 1 insertion(+), 1 deletion(-)
lsilva@MSI:~/source/... (test)$
```

Now git pull did not required the password

then **restart** your terminal and **passphrase** will be asked just once!. Enjoy it!