

How To Set Up Docker

🕒 Created	@October 13, 2022 6:18 PM
☰ Tags	Command Line Robocode
📎 Files & media	
☰ Level	Beginner
🔗 Related Resources	
⚙️ Status	In progress

How to download, install, and set up Docker to run the ros image. Also an example program to help you start to learn docker

Part 1: Setting up Docker

Step 1: Enable Windows Subsystem for Linux (wsl) and virtual platform in windows

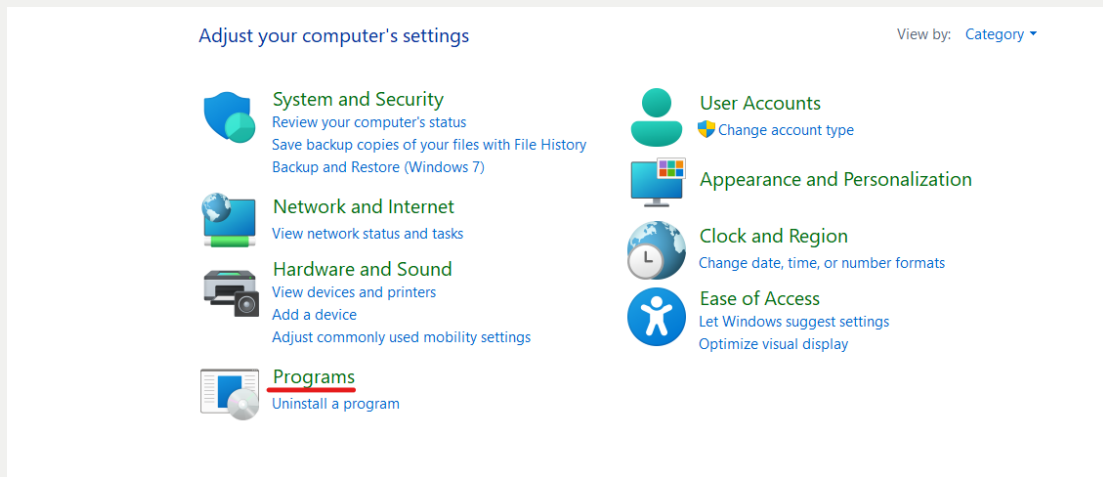


The best way to Enable WSL2 is by launching a `cmd` by and entering the `wsl --install` command. This will turn on all of the windows features and g

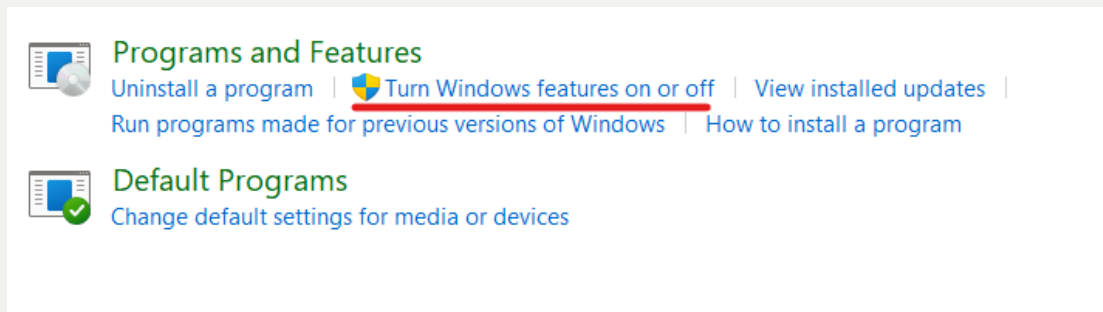
The first step to get Docker working on Windows is turning on Windows Subsystem for Linux and enabling virtual machine platform.



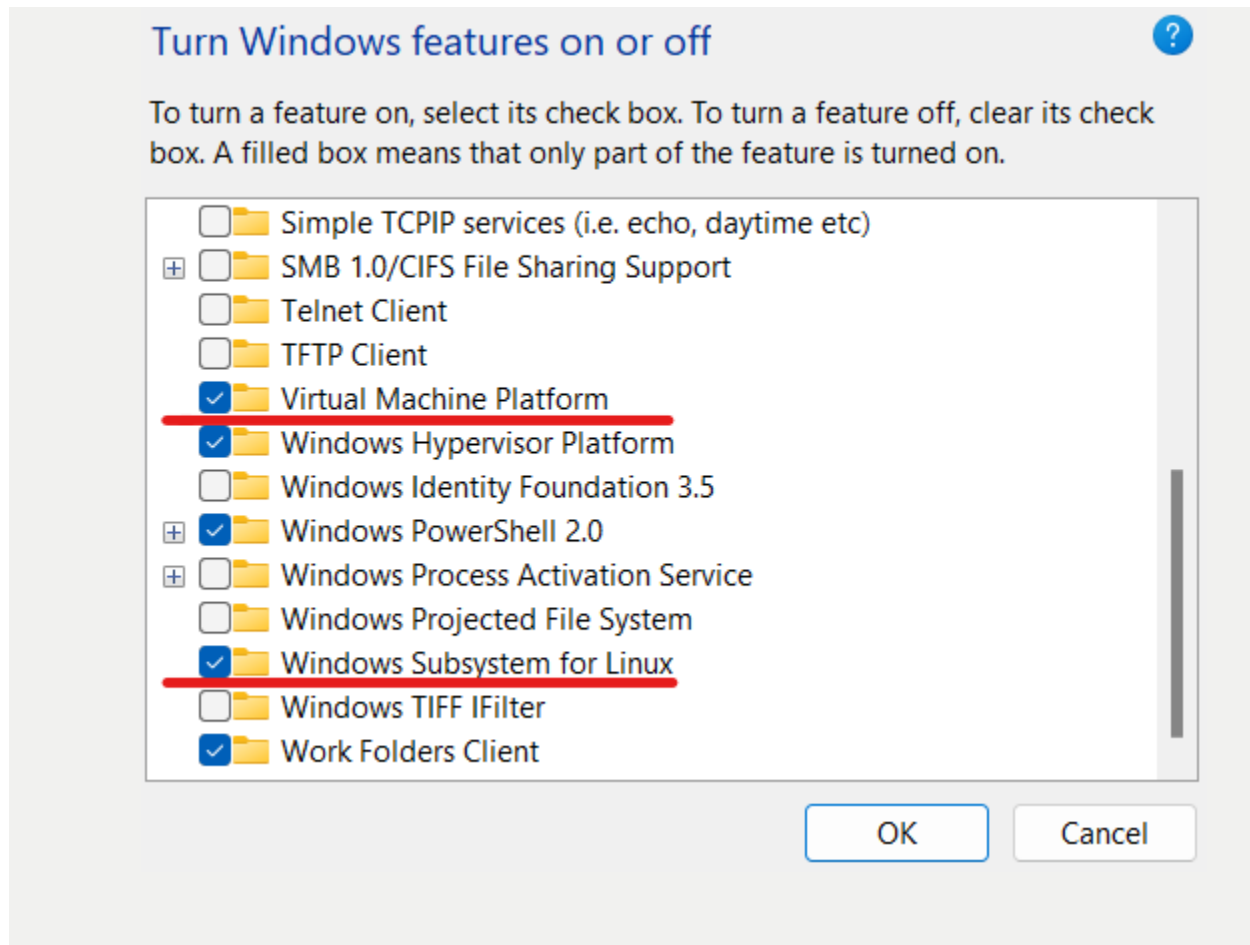
1. Go to the control panel and select the “Programs” option.



2. Select the “Turn windows features on or off” option below the “Programs and Features” option.



3. Select the “Windows Subsystem for Linux” and “virtual machine platform” options. This will require your system to reboot



Step 2: Update WSL2 and Install Ubuntu

After Turning on wsl and the virtual machine platform you are going to want to check that you have the latest version of wsl. Once we know that wsl is up to date, we will install Ubuntu onto our system.


Also for the best experience managing your different bash terminals, it's recommended that you use Windows Terminal. However, it is not necessary for this tutorial and can be done in PowerShell, command line, or whatever your favorite terminal is.

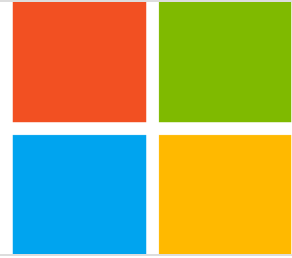
Installing Windows Terminal

This will be useful when we get started on using the robot operating system (ros/ros2). To install Windows Terminal along with a bunch of basics you can find it here:

Windows Terminal installation

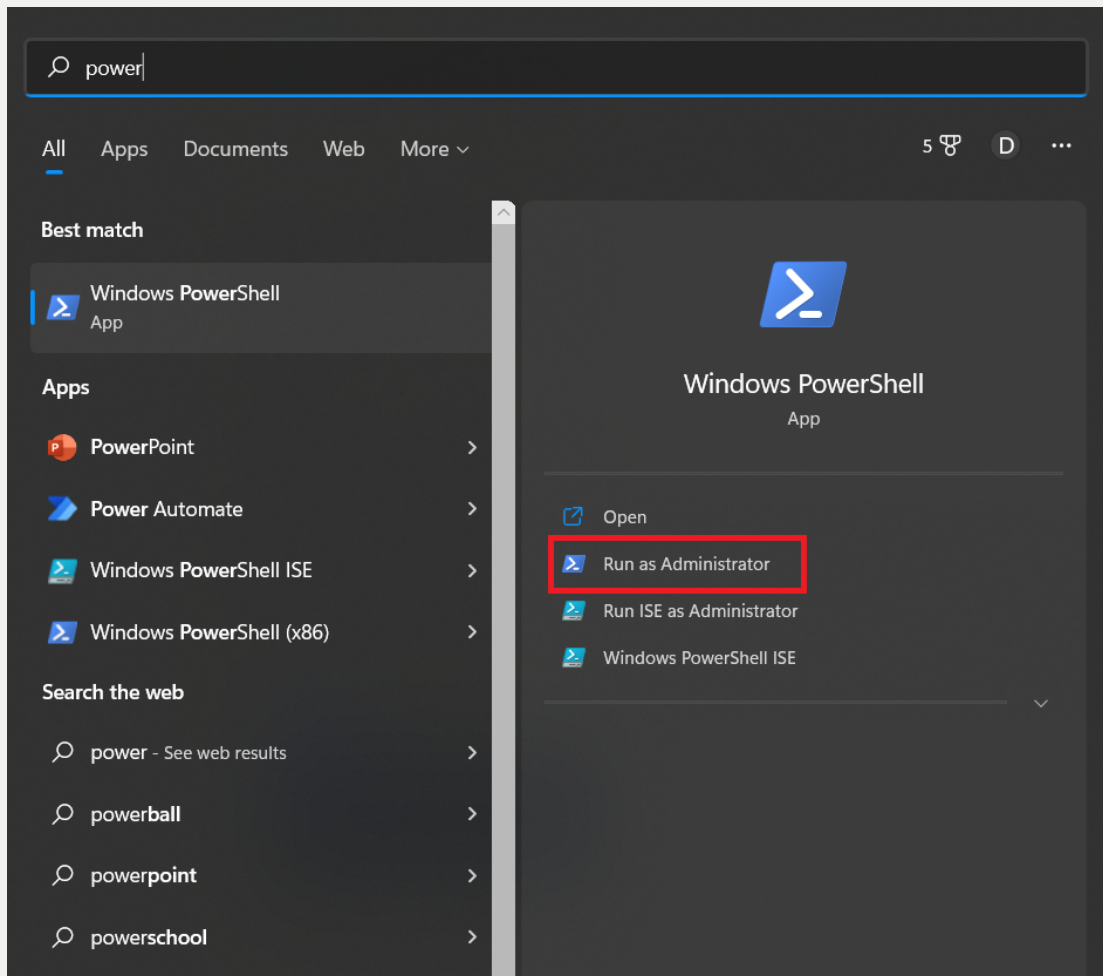
To try the latest preview features, you may also want to install Windows Terminal Preview. Note If you don't have access to the Microsoft Store, the builds are published on the GitHub releases

 <https://learn.microsoft.com/en-us/windows/terminal/install>

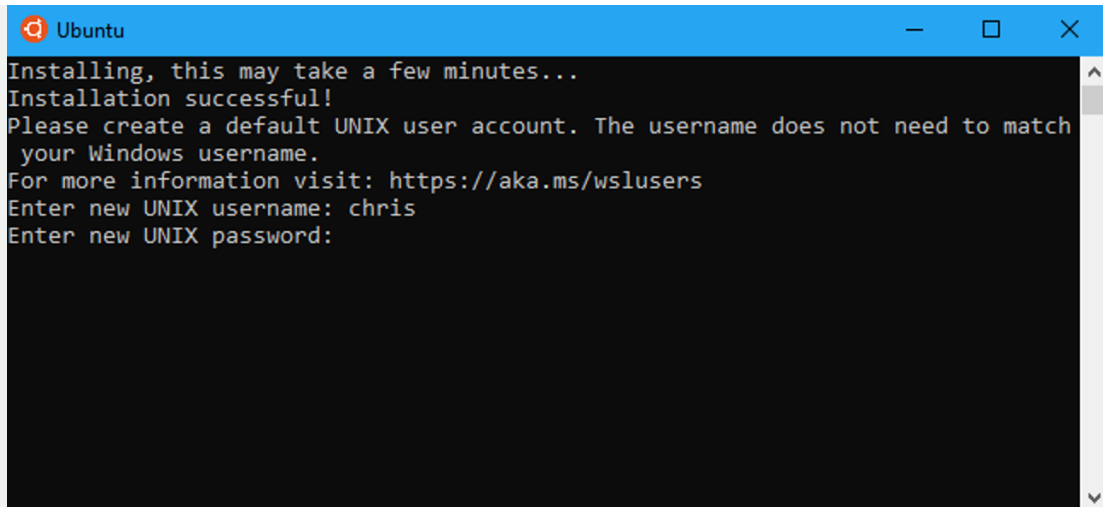




1. To start open a terminal as an administrator and type `wsl --update`



2. To install ubuntu with `wsl --install ubuntu-20.04`
3. Once ubuntu has installed, a terminal will pop up asking for unix credentials (it will look something like this)



```
Ubuntu
Installing, this may take a few minutes...
Installation successful!
Please create a default UNIX user account. The username does not need to match
your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: chris
Enter new UNIX password:
```

Once you have entered your credentials the terminal should close (if it doesn't you can just manually close it).

4. In your PowerShell set the wsl version and default distribution with `wsl --version ubuntu-20.04 2`

The 2 at the end of the command sets the version of wsl to 2 (default version). You can change to a 1 to downgrade to wsl 1 (but in most cases you would want to stick with wsl 2).

Step 3: Install Docker

Now we need to install Docker. This can be done by going to the link below and following the page's instructions.

<https://docs.docker.com/desktop/install/windows-install/>

Part 2: Getting familiar with Docker

Step 4: Running the ROS Docker Image

The last step is to install docker and have it ready to start working with our robot code is to run the ros/ros2 image.



1. run this in your terminal: `docker run osrf/ros:humble-desktop-full`

Step 5: Create an internal network and two containers

Now we are going to get familiar with how to use the `osrf/ros:humble-desktop-full` image. We are going to create a couple of nodes (containers) and have the nodes communicate with each other over an internal network.

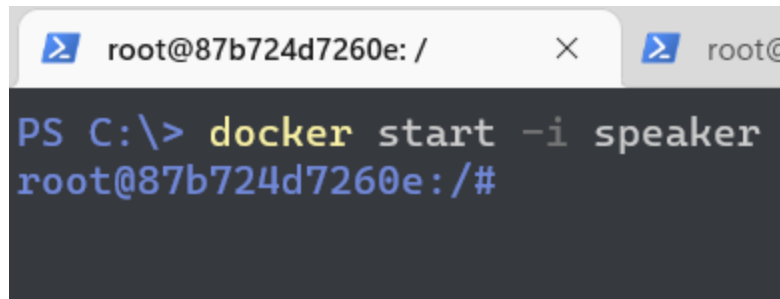
1. We are going to create an internal network so that our containers can communicate with each other with `docker network create ros-internal`.
2. To show your current networks enter `docker network list`. This will list all of your docker networks and the new `ros-internal` network should be shown on that list.

```
PS C:\> docker network list
NETWORK ID        NAME                DRIVER              SCOPE
b86b300fd2cc      bridge              bridge              local
40bcc20aed01      host                host                local
6869f7dff8ef      none                null                local
ab78786e8a59      ros-internal        bridge              local
PS C:\>
```

3. Now that we know we have created an internal network to use for this test we are going to create two new containers. To do this enter `docker run -it --network ros-internal --name listener osrf/ros:humble-desktop-full` This will launch an interactive terminal (`-it`) on our `ros-internal` network. This container's name will be `listener` (important for later). You'll know that you created the container

because instead of showing your file path on the command line it will show

```
root@<containerID> .
```



```
PS C:\> docker start -i speaker
root@87b724d7260e: /#
```

4. Now open another terminal and enter `docker run -it --network ros-internal --name speaker osrf/ros:humble-desktop-full`. The only difference in this is the name. The name for this container is `speaker`.
5. Now that we have two containers we are going to go to our listener container and run `ros2 run demo_nodes_cpp listener`
6. Next change to your speaker container and run `ros2 run demo_nodes_cpp talker`. It will begin pinging a hello world message. If you change back to the listener container you will see the same messages with the same numbers.

```
root@87b724d7260e: / × root@385900ac54ea: / + v
[INFO] [1666224685.097257000] [listener]: I heard: [Hello World: 45]
[INFO] [1666224686.097175200] [listener]: I heard: [Hello World: 46]
[INFO] [1666224687.097154200] [listener]: I heard: [Hello World: 47]
[INFO] [1666224688.097357700] [listener]: I heard: [Hello World: 48]
[INFO] [1666224689.098355600] [listener]: I heard: [Hello World: 49]
[INFO] [1666224690.097401200] [listener]: I heard: [Hello World: 50]
[INFO] [1666224691.097317800] [listener]: I heard: [Hello World: 51]
[INFO] [1666224692.097358900] [listener]: I heard: [Hello World: 52]
[INFO] [1666224693.098416500] [listener]: I heard: [Hello World: 53]
[INFO] [1666224694.097365000] [listener]: I heard: [Hello World: 54]
[INFO] [1666224695.096917700] [listener]: I heard: [Hello World: 55]
[INFO] [1666224696.097126900] [listener]: I heard: [Hello World: 56]
[INFO] [1666224697.098627600] [listener]: I heard: [Hello World: 57]
[INFO] [1666224698.097410500] [listener]: I heard: [Hello World: 58]
[INFO] [1666224699.098669100] [listener]: I heard: [Hello World: 59]
[INFO] [1666224700.097386000] [listener]: I heard: [Hello World: 60]
[INFO] [1666224701.097283900] [listener]: I heard: [Hello World: 61]
[INFO] [1666224702.097452800] [listener]: I heard: [Hello World: 62]
[INFO] [1666224703.099574600] [listener]: I heard: [Hello World: 63]
[INFO] [1666224704.096877600] [listener]: I heard: [Hello World: 64]
[INFO] [1666224705.097010900] [listener]: I heard: [Hello World: 65]
[INFO] [1666224706.097414300] [listener]: I heard: [Hello World: 66]
[INFO] [1666224707.097815500] [listener]: I heard: [Hello World: 67]
[INFO] [1666224708.094690000] [listener]: I heard: [Hello World: 68]
[INFO] [1666224709.095484700] [listener]: I heard: [Hello World: 69]
[INFO] [1666224710.096104300] [listener]: I heard: [Hello World: 70]
[INFO] [1666224711.094826500] [listener]: I heard: [Hello World: 71]
[INFO] [1666224712.095130800] [listener]: I heard: [Hello World: 72]
[INFO] [1666224713.095051300] [listener]: I heard: [Hello World: 73]
```

7. To exit type ctrl+c on each container.
