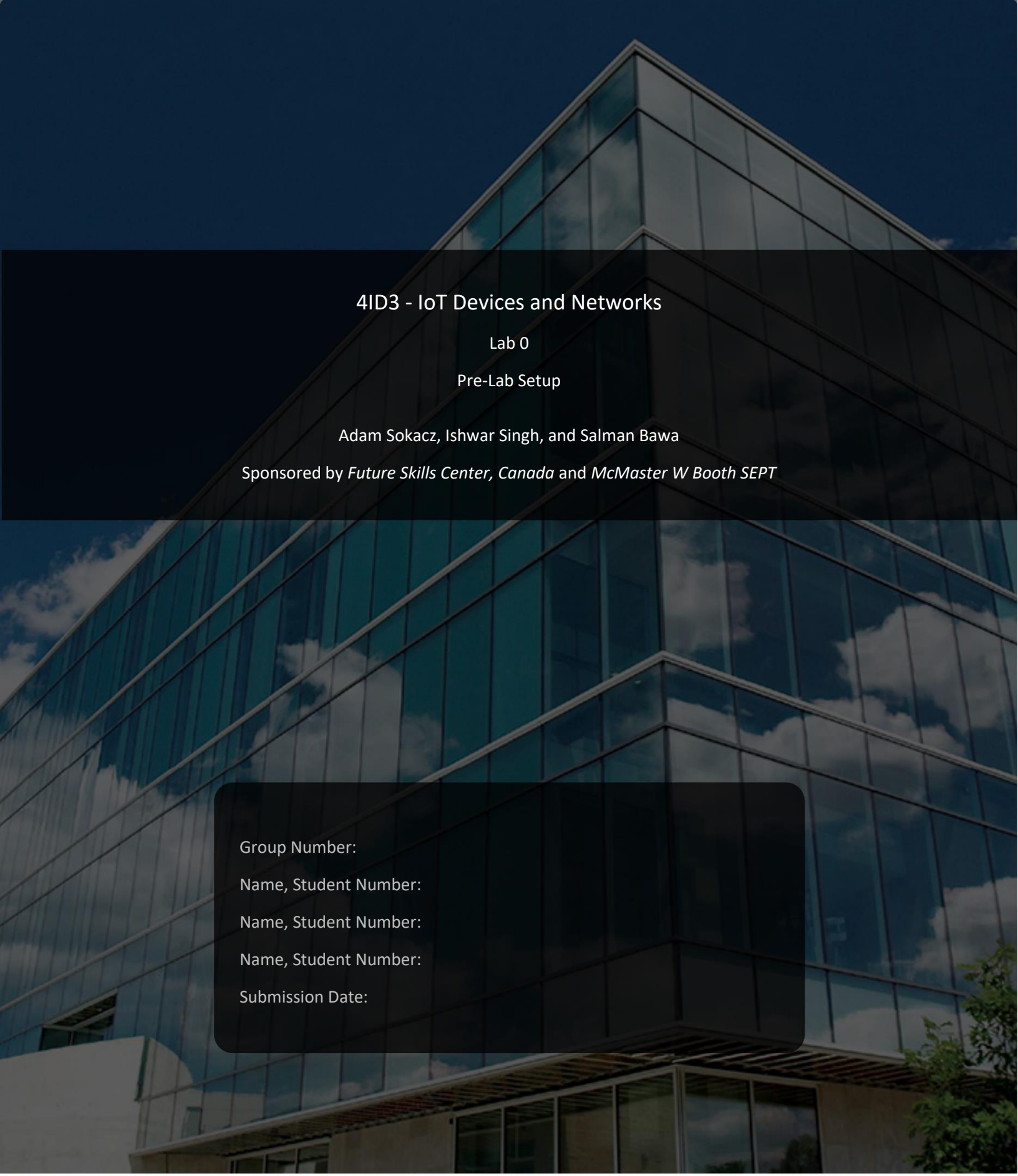


4ID3 - IoT Devices and Networks

Lab 0

Pre-Lab Setup

Adam Sokacz, Ishwar Singh, and Salman Bawa

Sponsored by *Future Skills Center, Canada* and *McMaster W Booth SEPT*

Group Number:

Name, Student Number:

Name, Student Number:

Name, Student Number:

Submission Date:

W BOOTH SCHOOL OF ENGINEERING PRACTICE AND TECHNOLOGY

McMaster-Mohawk Bachelor of Technology Partnership

McMaster University ENGINEERING

mohawk

Objective

To install the software that will be used in upcoming labs.

Contents

Feedback

Q1 - What would you rate the difficulty of this lab?

(1 = easy, 5 = difficult)

1 **2** **3** **4** **5**

Comments about the difficulty of the lab:

Q2 - Did you have enough time to complete the lab within the designated lab time?

YES **NO**

Q3 - How easy were the lab instructions to understand?

(1 = easy, 5 = unclear)

1 **2** **3** **4** **5**

List any unclear steps:

Q4 - Could you see yourself using the skills learned in this lab to tackle future engineering challenges?

(1 = no, 5 = yes)

1 **2** **3** **4** **5**

Pre-Lab Questions

Post-Lab Questions

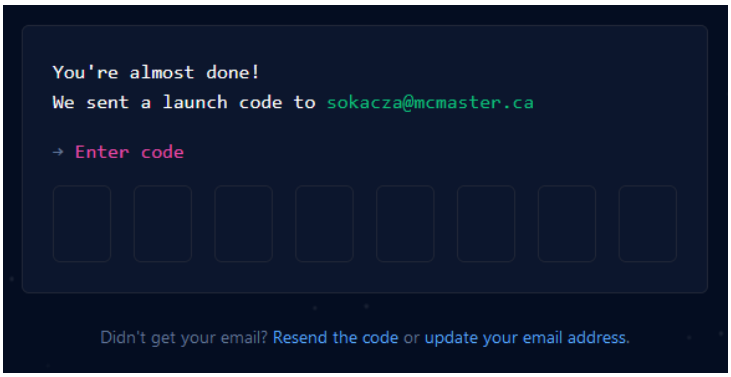
Setting Up a Software Portfolio

To gain a competitive advantage over other applicants when pursuing software positions, it is important to be able to demonstrate and articulate your past experiences, projects, and skills. During this class and other classes, it is highly recommended that you collect all of your coding assignments and projects into a GitHub that you can showcase on your resume and LinkedIn profile. This section will guide you step-by-step to setting up a GitHub account, initializing a repository, cloning an existing repository, and pushing your local changes to the remote.

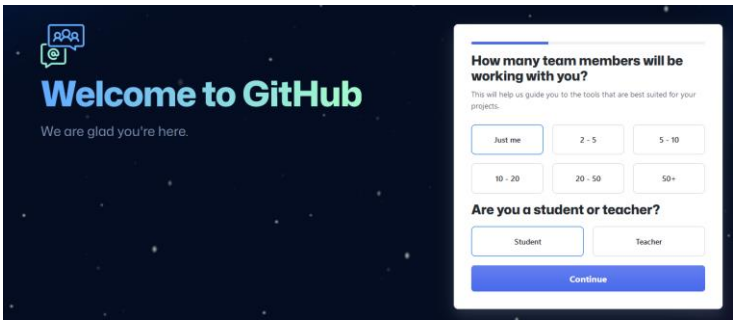
Navigate to the following website:

https://github.com/signup?source=login

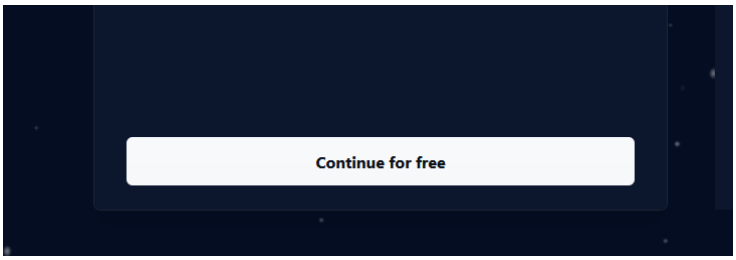
Follow the step-by-step instructions to creating an account using your McMaster email address.



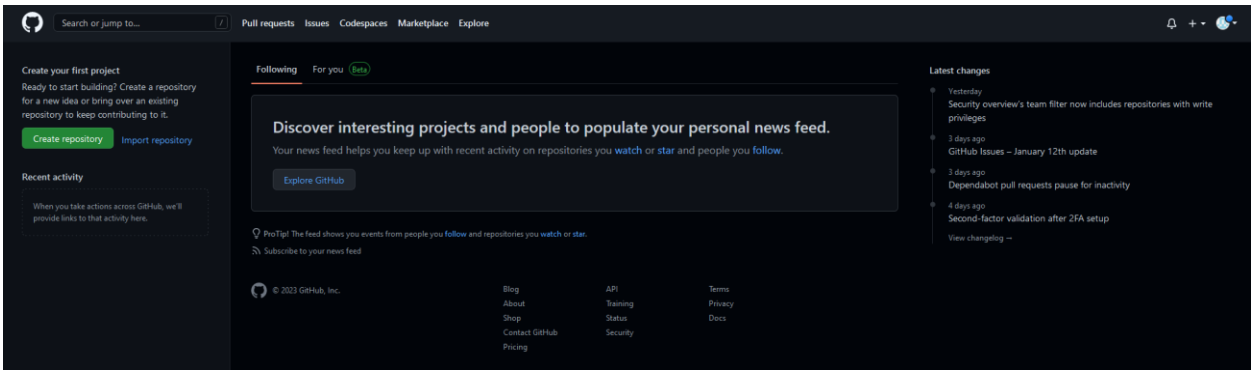
Press **continue**.



Press **continue for free**.

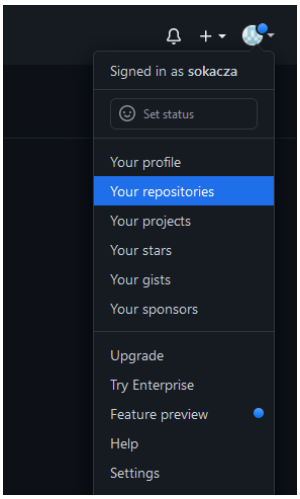


You should be shown to the account homepage.

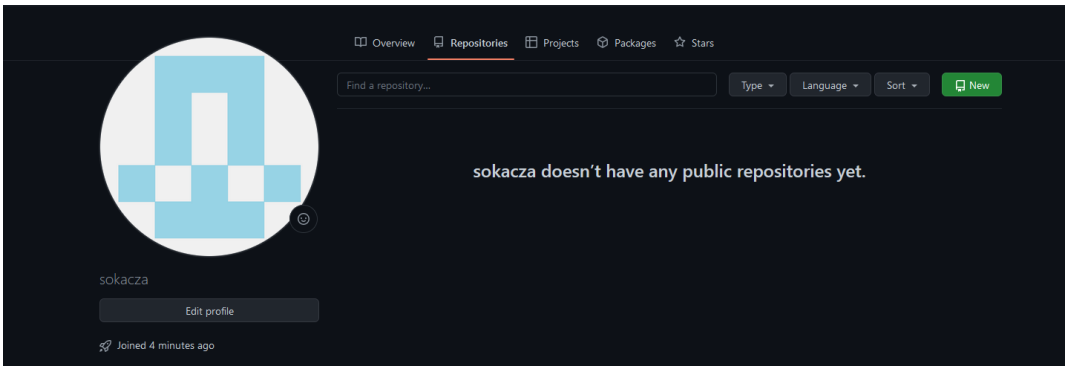


GitHub is **free with restrictions**, see their documentation for more information.

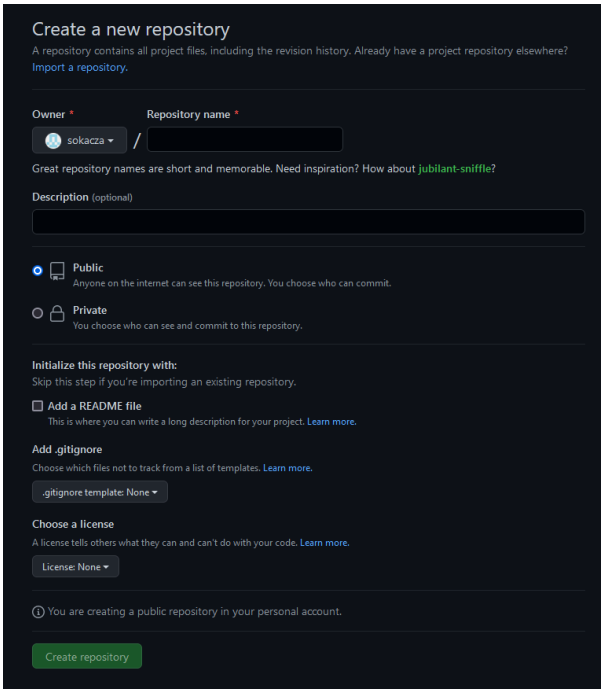
Next, we will be creating a remote repository. A remote repository is a software project that is stored in GitHub's servers. To do this, click on your account icon in the top right corner of the screen and select **Your repositories**.



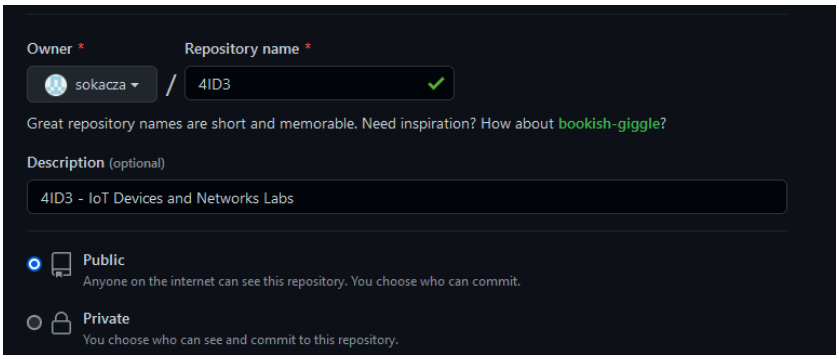
Press **New**.



Fill in the requested fields. Set your repository to **public** so anyone can view it without requesting access. Ensure that there are no spaces or uncommon characters in the repository name because this will make it difficult to search for.

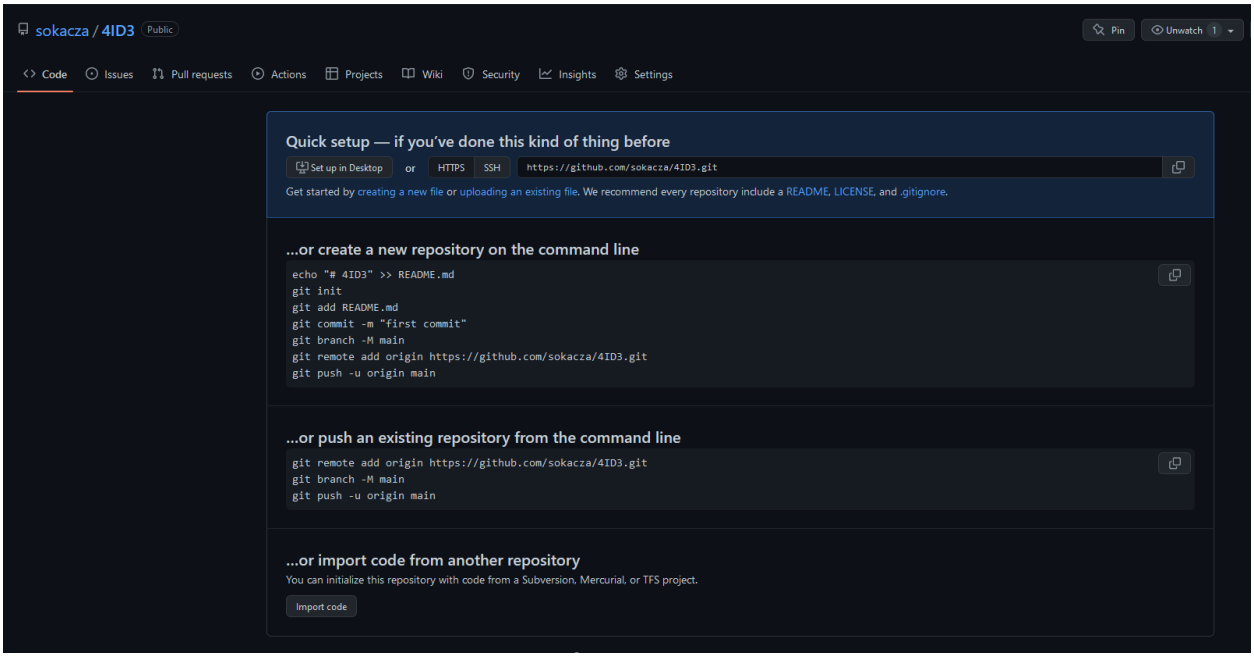


We can set up one repository for the entire course, and you can resync your code after completing each lab assignment.



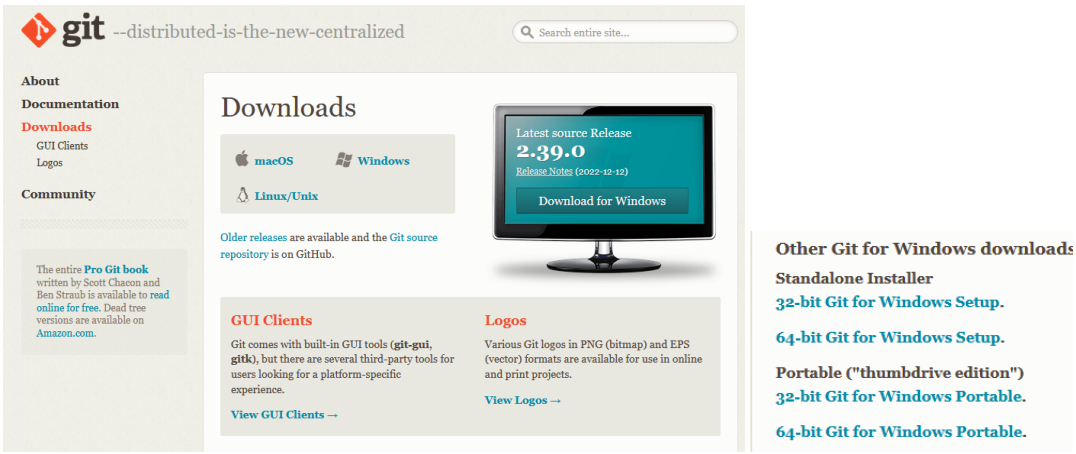
Press **Create Repository.**

GitHub will now generate a list of commands for further setup. **Do not exit the page** just yet.



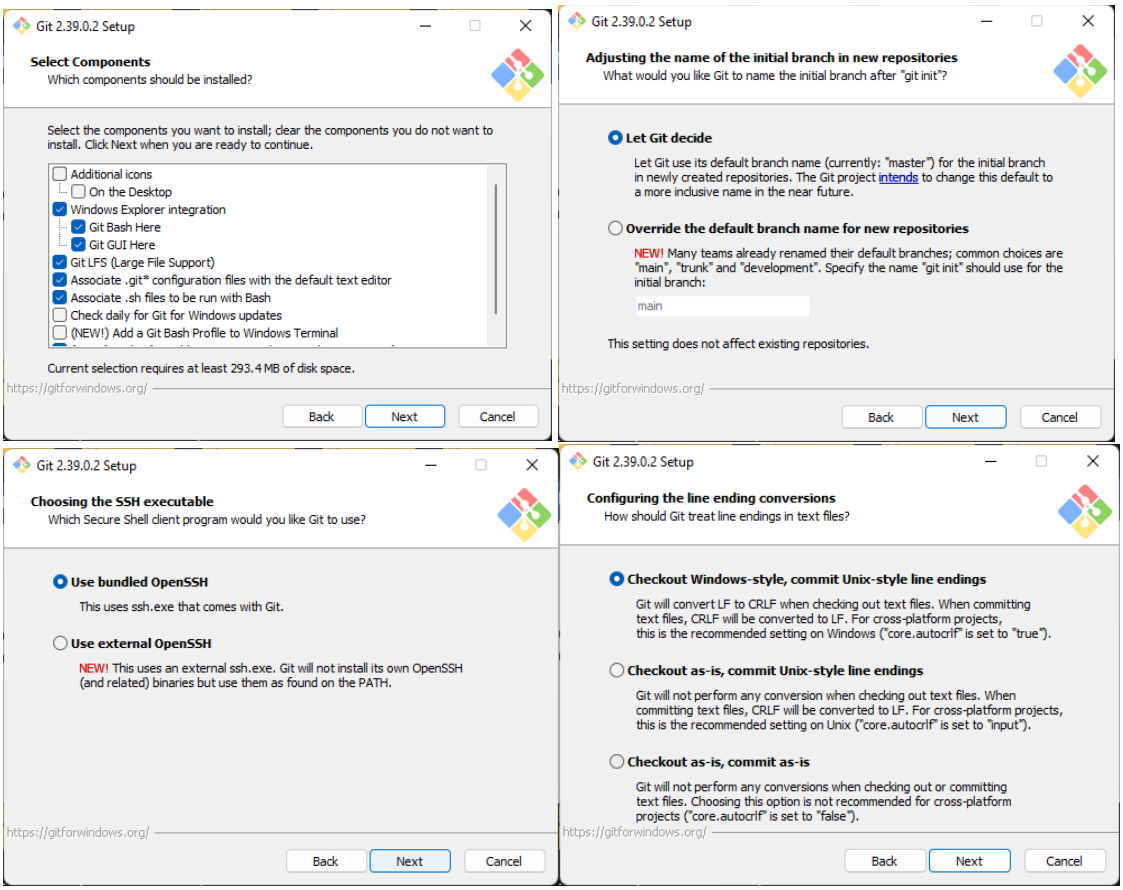
The next step is to download **Git**. Git Bash is a command line tool to create, download, and sync repositories. It can be downloaded at the following link:

https://git-scm.com/downloads



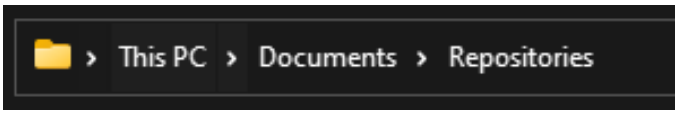
Download the 64 bit version for your operating system.

Leave the default install options.

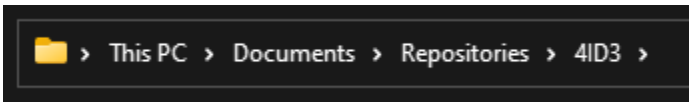


Once installed, we now need to create a local repository and connect it to our remote repository.

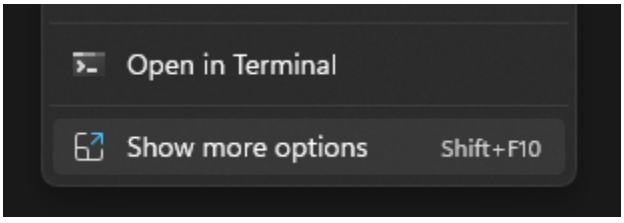
In your **Documents** folder, create a new folder called **Repositories**.



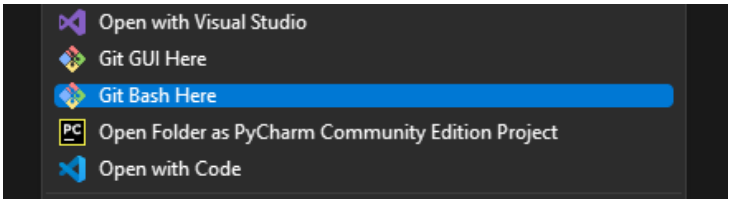
Within the **Repositories** folder, create another folder called **4ID3**.



If you are on Windows 11, right click on the folder and select **Show more options**.

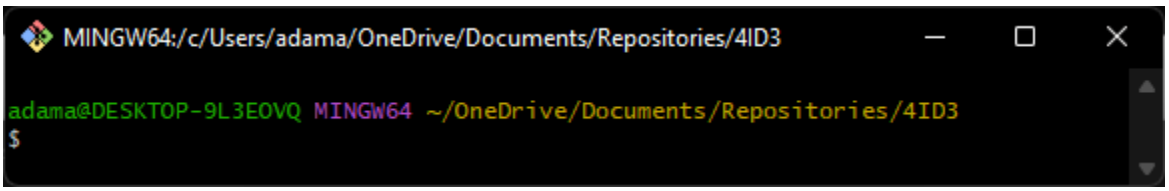


Select **Git Bash here**.



If this option is unavailable, a) Restart your PC, b) find the program in the Start Menu and navigate to this directory.

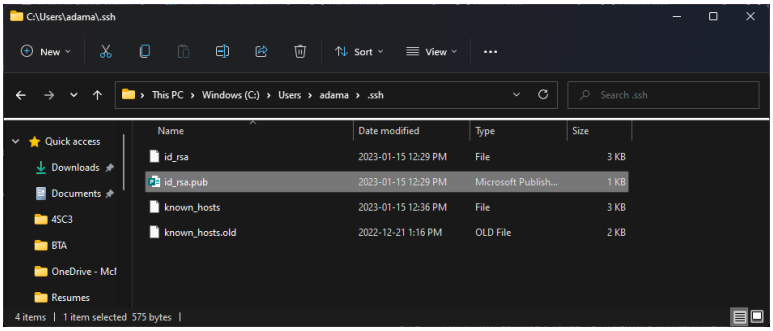
A terminal window should open.



Now, we'll be going back to those commands that GitHub gave us earlier and pasting them here one-by-one. But before that, we need to set up SSH so that the file transfer is encrypted.

Press **space** to not use a password.

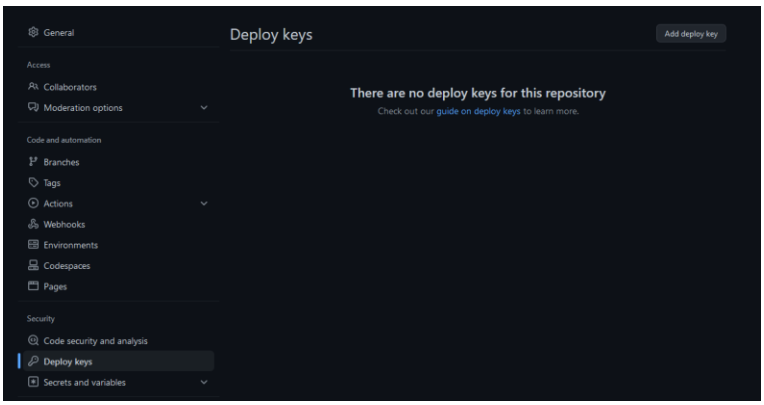
Navigate to the output directory in **file explorer**.



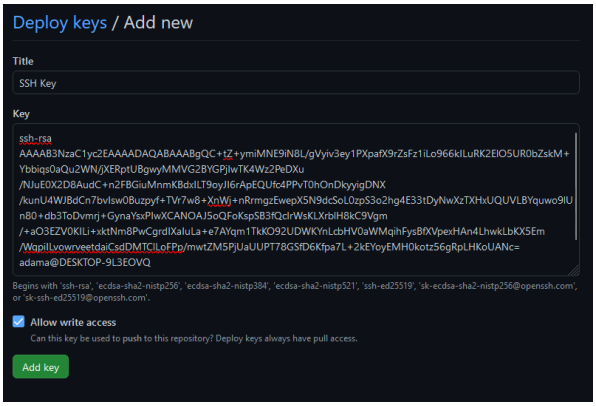
Open **is_rsa.pub** using **Notepad** and copy all the text.

Inside your GitHub repository, select **Settings > Deploy Keys**.

Press **Add deploy key**.

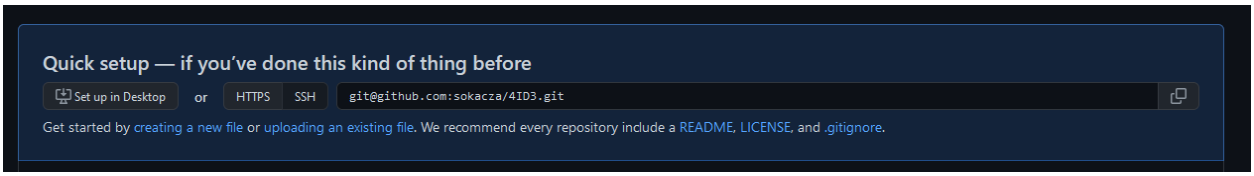


Name the key, **allow write access**, and press **Add key**.

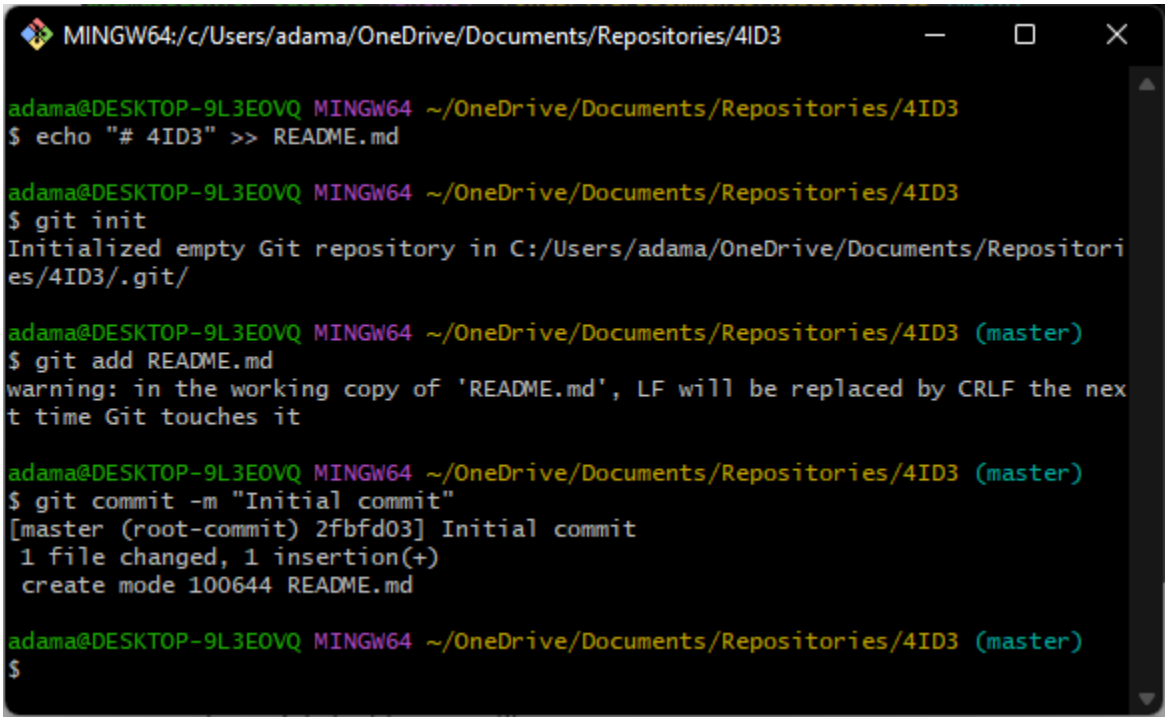


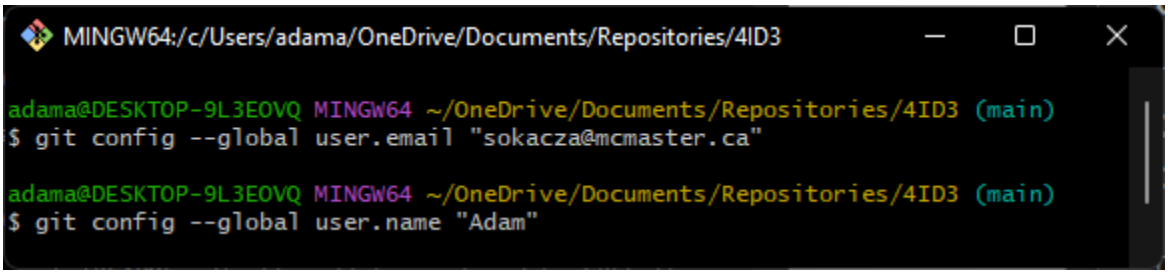
Once the key is added, navigate back to the **Code** tab.

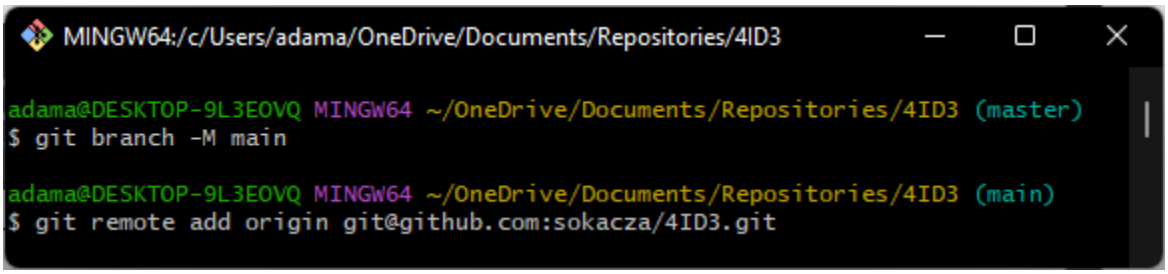
Press **SSH** in the instructions to generate the commands that use SSH.

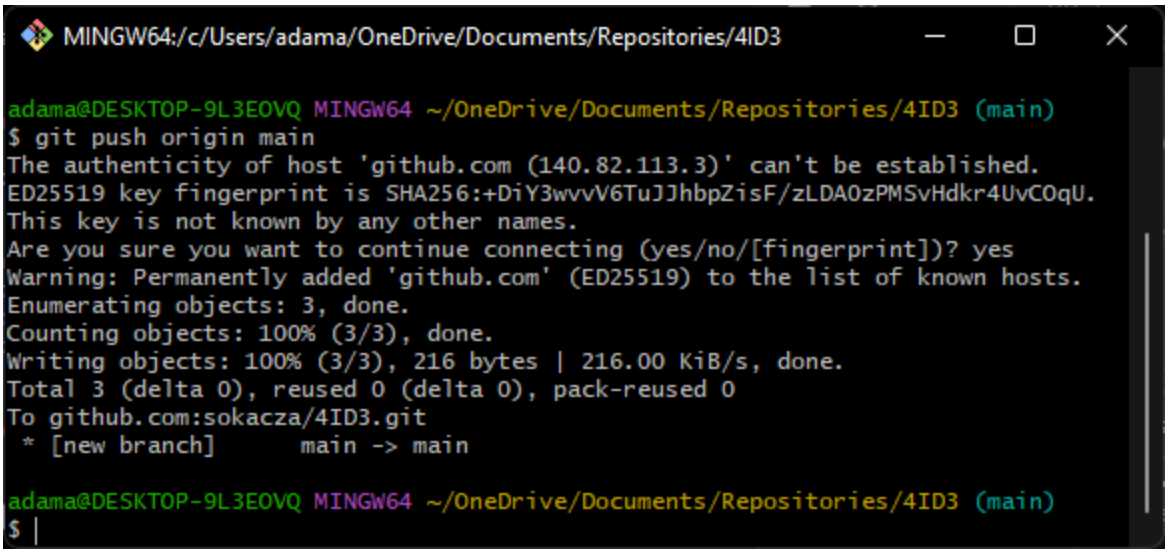


Copy and paste the listed commands into Git Bash.









If you get an error, delete all the files in the SSH key folder and regenerate the key. Upload the new key to GitHub.

echo - prints text to a new file

git init - initializes a local repository

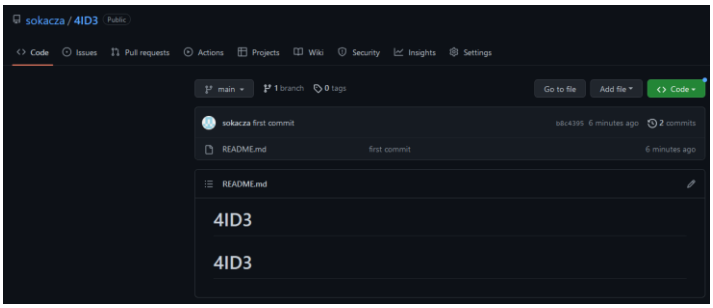
add - tells git that this file has been changed and needs to be synced with GitHub

branch - tells git which version of the software project you wish to sync to. If in doubt, use **main**

remote add - tells git which remote repository to sync to

push - syncs the changes

Now, you should see a file called README.md in your GitHub. While the setup of GitHub might be tedious, using it from here on out is straightforward.



Common problems:

1. GitHub does not recognize your PC

Follow the instructions printed to the terminal on how to add your email and name.

git config –global user.email “my@email.com”

git config –global user.name “my name”

2. Some refs failed to push

Regenerate the repo by doing the following:

git init

git add .

git commit -m “First commit”

git push origin main

3. sokacza/4ID3 not found

In this situation, you used my repo as the origin url instead of your own. Do the following:

git remote remove origin

git remote add your_remote_url

4. You cannot find git bash in your toolbar

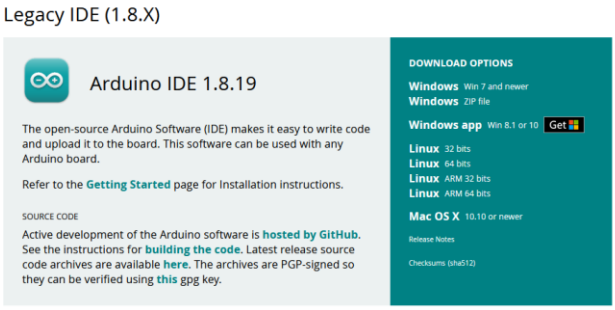
Ensure that git bash is installed. If it is, launch it from the start menu and use cd to navigate to the project directory.

Installing the Arduino IDE

The Arduino IDE is an integrated development environment created by the company Arduino Inc. An integrated development environment provides software developers the tools to efficiently edit code, manage libraries, and troubleshoot issues.

Navigate to the following website and download the latest version of Arduino IDE **version 1.XX.**

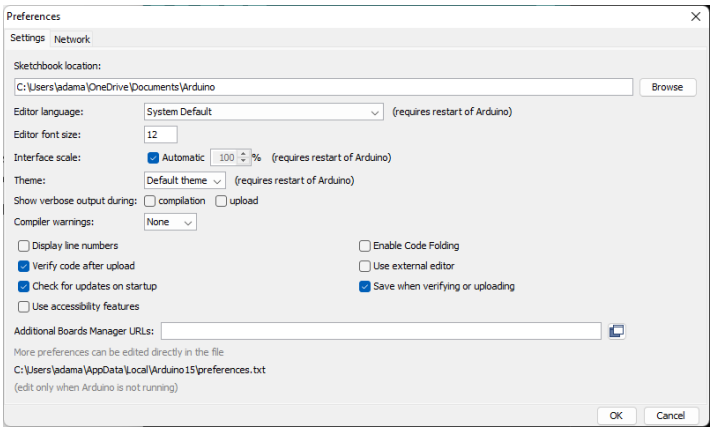
https://www.arduino.cc/en/software



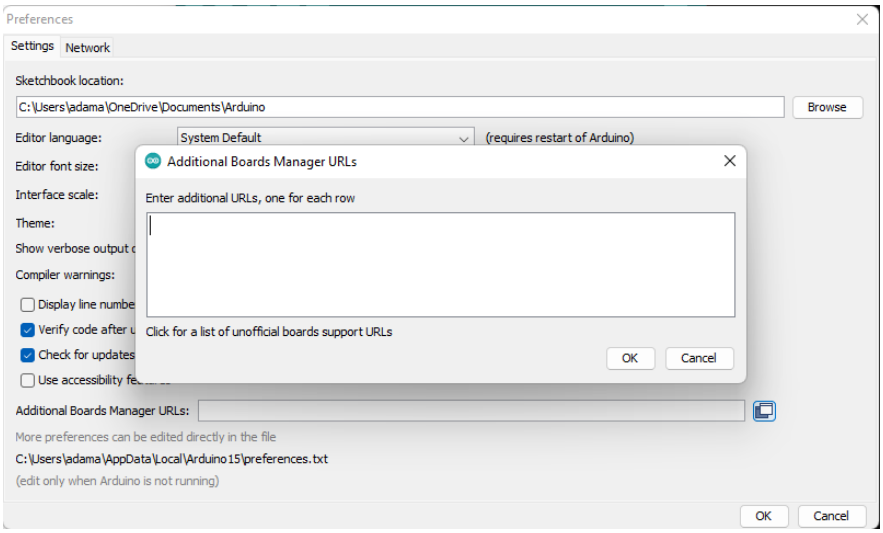
Navigate through the installer, selecting the default options.

The ESP32 and ESP8266 will be used frequently throughout this course. In order to compile for those boards and see them in the board manager, a custom URL must be added into the preferences.

Select **Preferences** and see the **Additional Board Manager URL's** field.



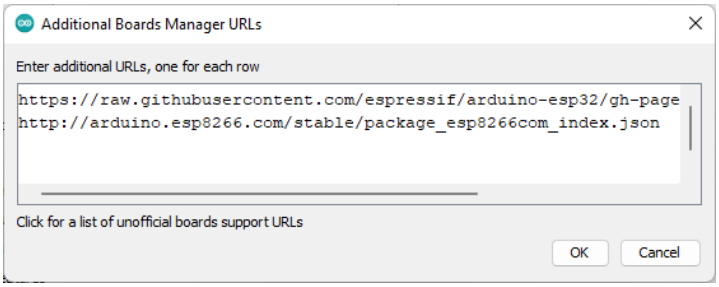
Press the button to the right of the field.



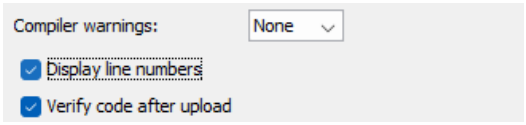
Paste the following **two** url's on separate lines:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

http://arduino.esp8266.com/stable/package_esp8266com_index.json

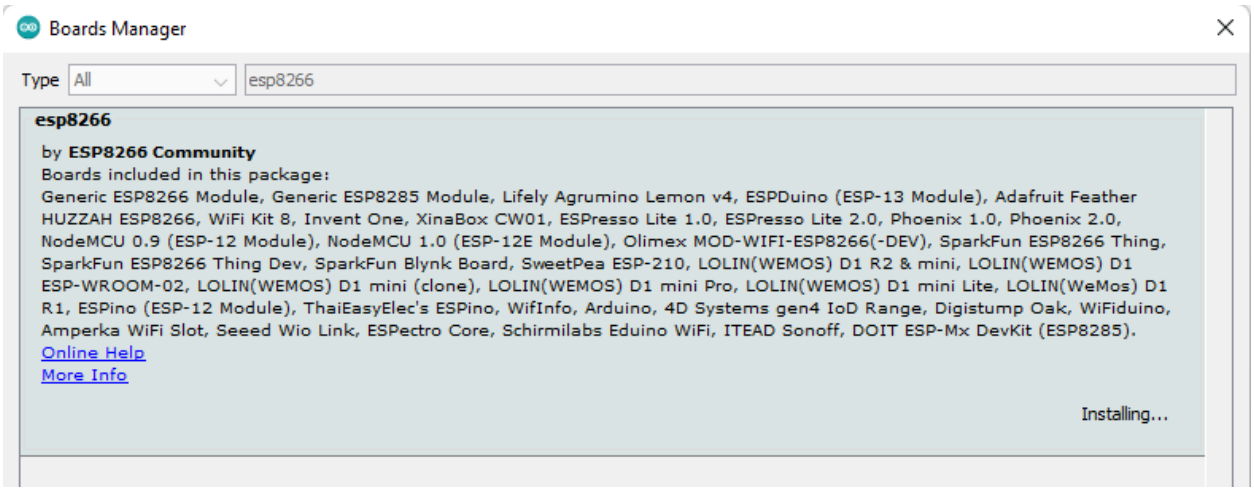


Press **OK**. Also, within preferences, check **Display line numbers**.



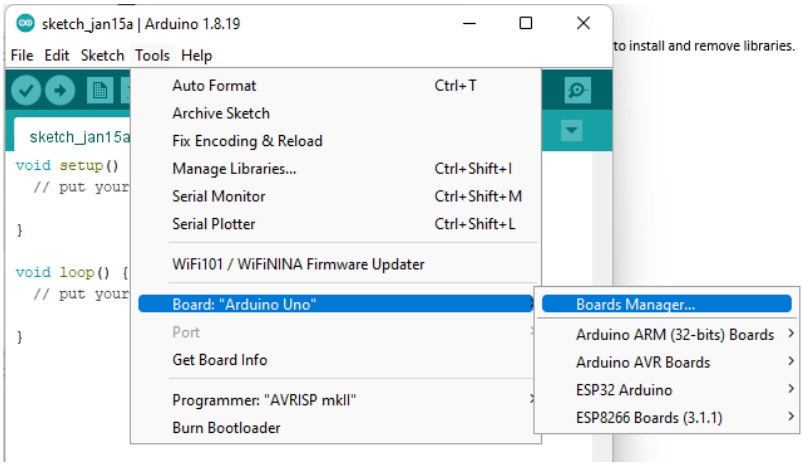
Press **OK** to exit.

Install the ESP8266 board by ESP8266 Community.



Lastly, install the ESP32 board by Espressif Systems.

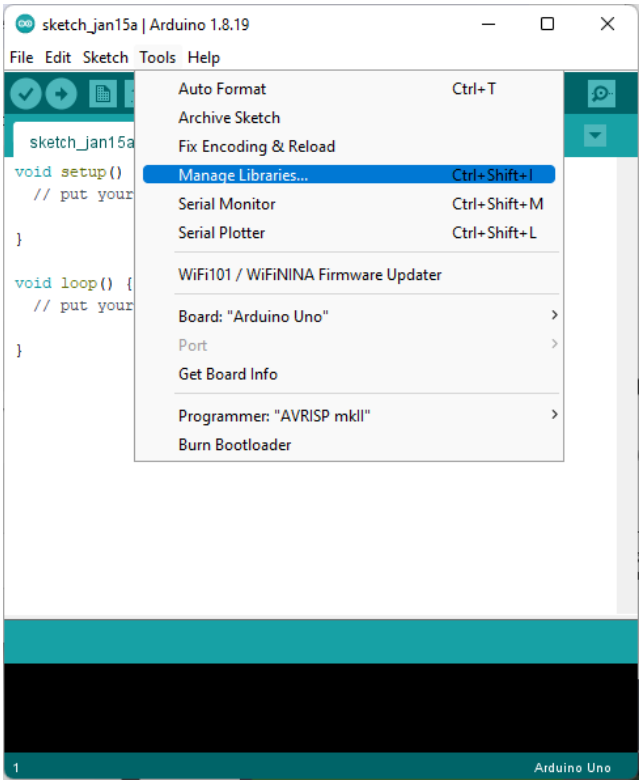
Ensure that you can see the boards available in your **Board Manager**.



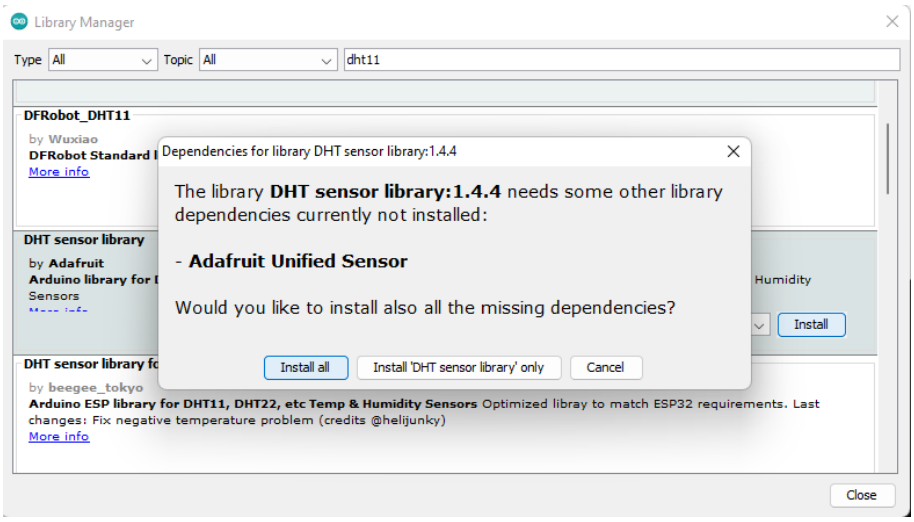
Installing Arduino Sensor Libraries

The Arduino IDE has a built-in library manager that can be used to install and remove libraries.

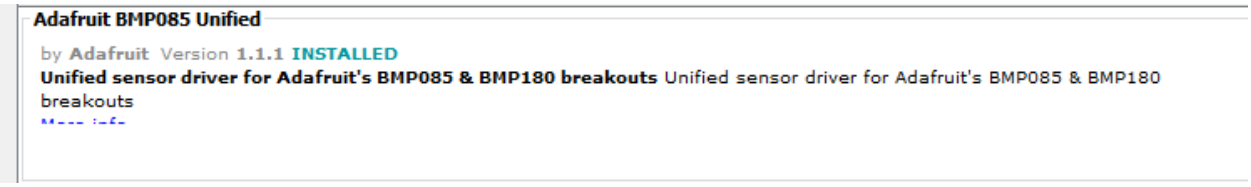
Navigate to **Tools > Manage Libraries**.



Search for the DHT sensor library by Adafruit and install the latest version. If prompted to install the unified library, choose **Install all.**

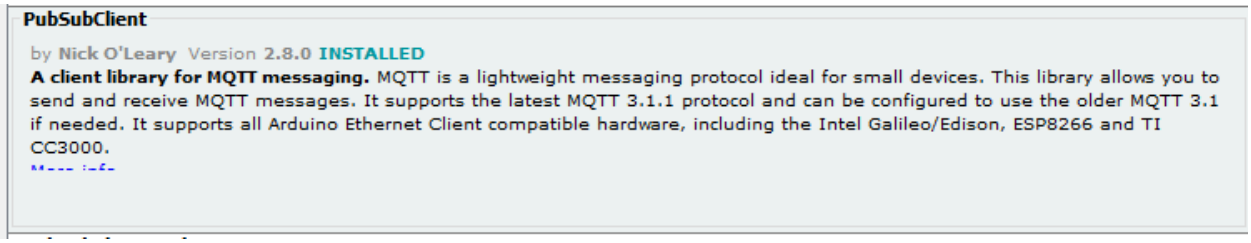


Next, install the BMP Unified sensor library by Adafruit.



Next, install the BH1750 library by Stefan Armborst.

Lastly, install the PubSubClient MQTT library by Nick O'Leary.



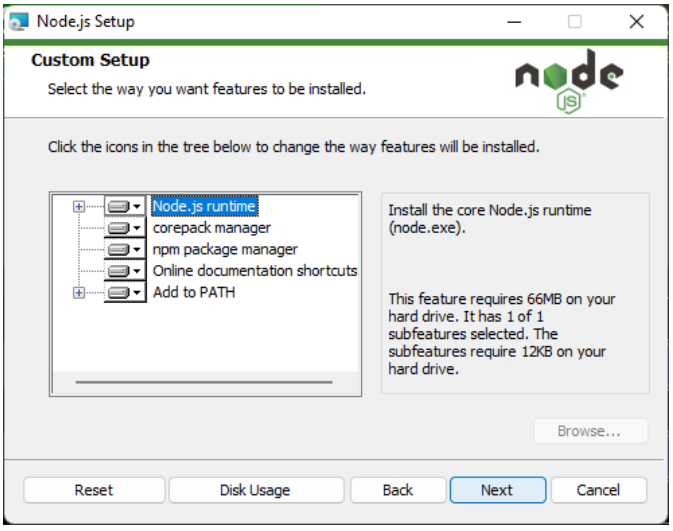
Installing NodeJS

NodeJS is a JavaScript runtime. This means that it allows JavaScript software to run outside of the web browser and have access to the filesystem and other computing resources. NodeJS is a prerequisite to installing NodeRED, which is a graphical data-driven programming environment that can be used to rapidly build and deploy IoT applications and dashboards.

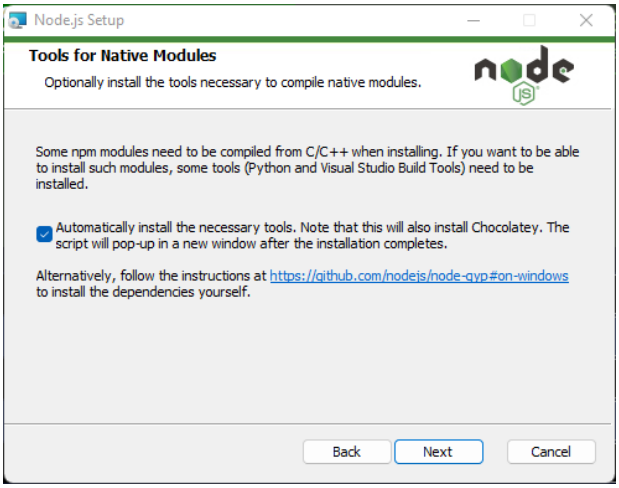
Navigate to the following url:

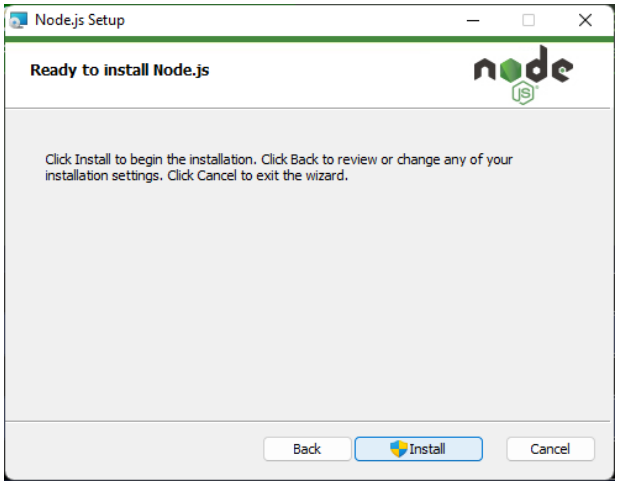
https://nodejs.org/en/download/

Select the correct installer for your operating system.

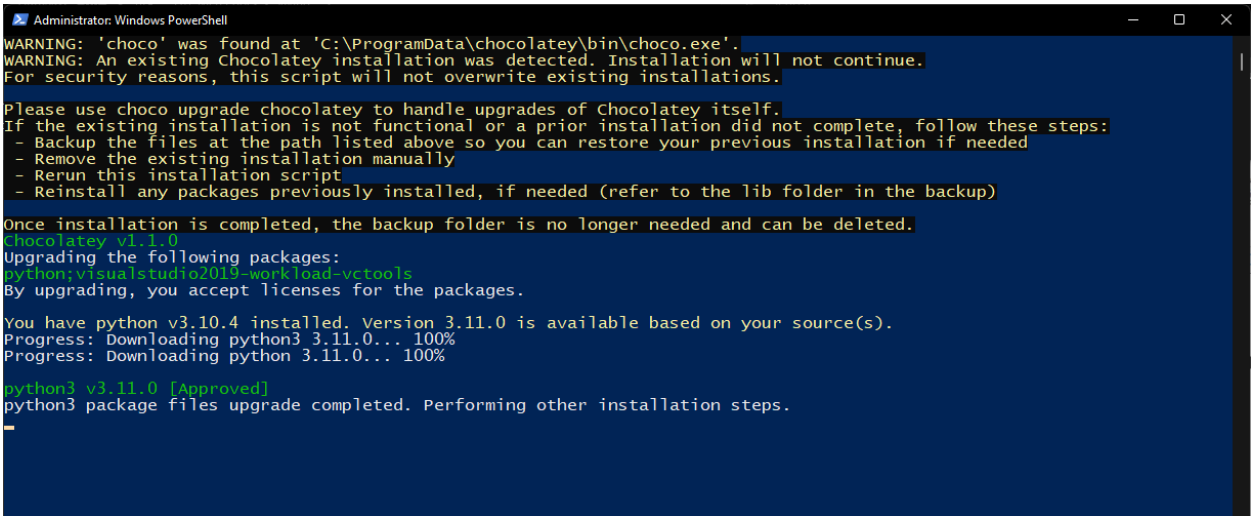


Have it automatically install the necessary tools and press **Install**.

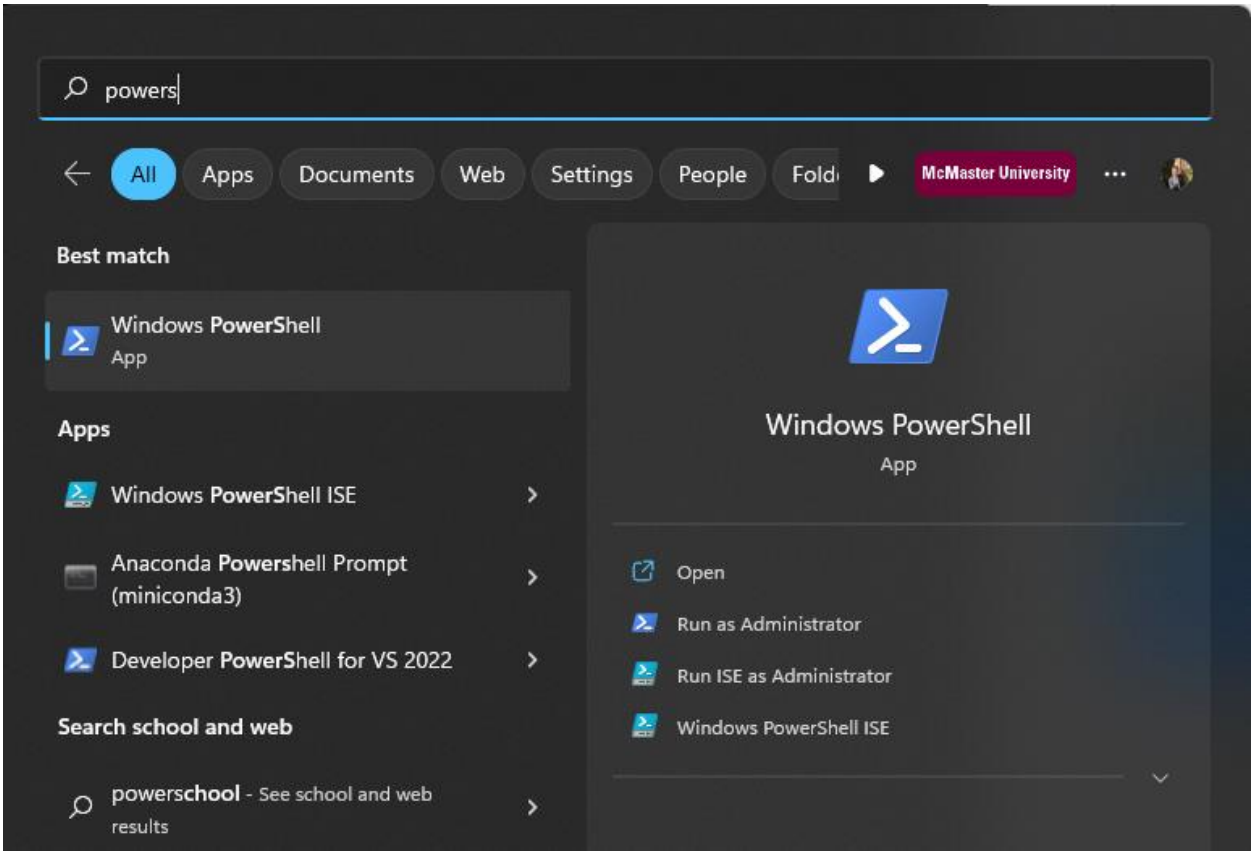




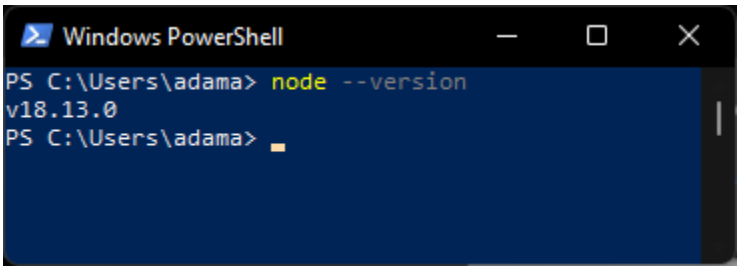
Allow the additional tools to install. This may take 2 minutes.



To verify that it has been installed correctly, using the **Start Button**, search for **Powershell**.



Run the node command with the version flag to echo out the current installed version of NodeJS.



node --version

Installing NodeRED

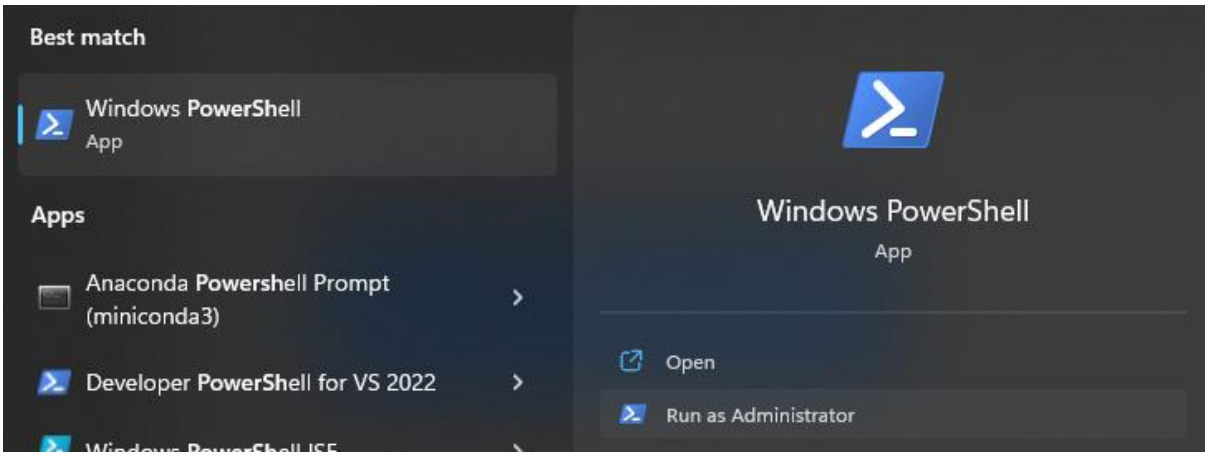
NodeRED will be installed by NPM. NPM is an acronym for Node Package Manager. When programming, sometimes different programs need to use to the same libraries or frameworks. A package manager ensures the same libraries are not installed twice, they remain up-to-date, and they have the correct dependencies installed for them to work properly. NPM will be used to install NodeRED.

The official install documentation can be found here:

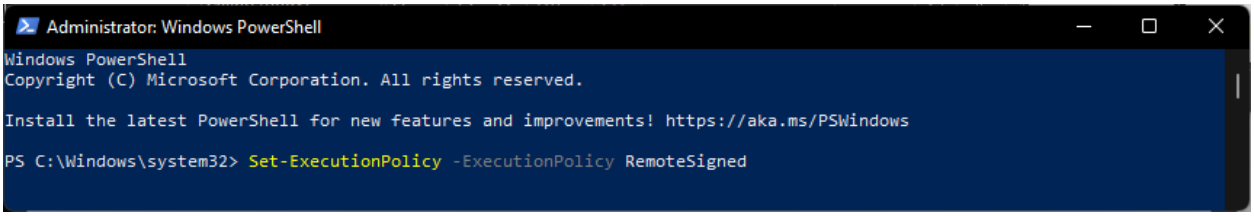
https://nodered.org/docs/getting-started/windows

Open **Powershell**.

Launch PowerShell as **Administrator.**

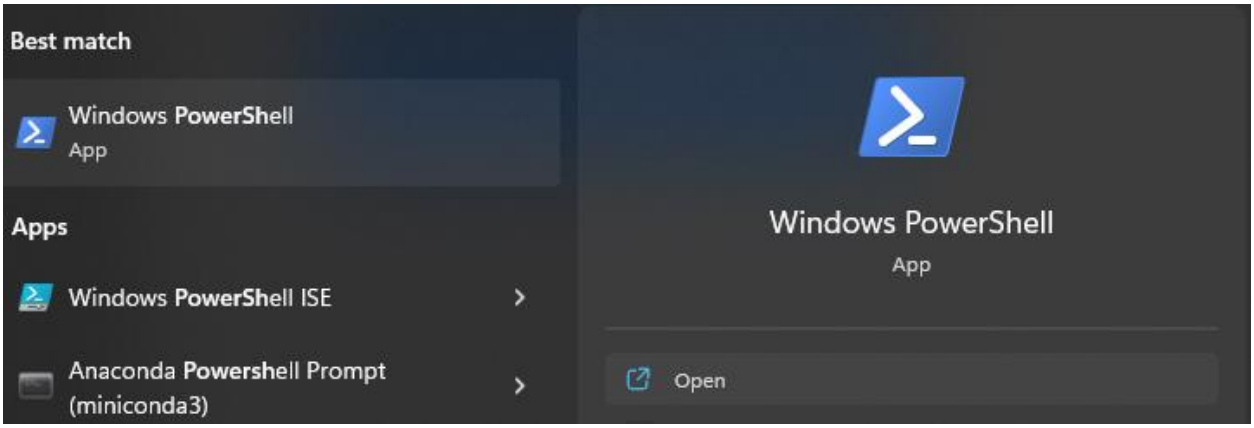


Use the following command to change the execution policy.

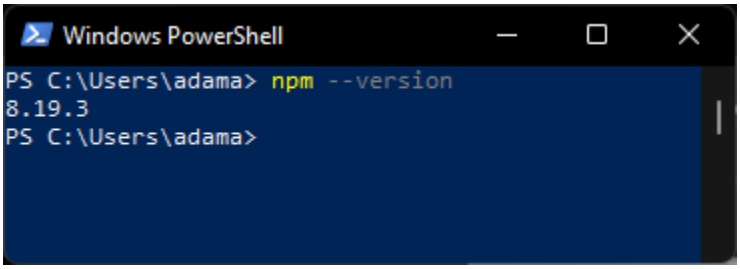


Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

Next, re-open Powershell as a normal user.

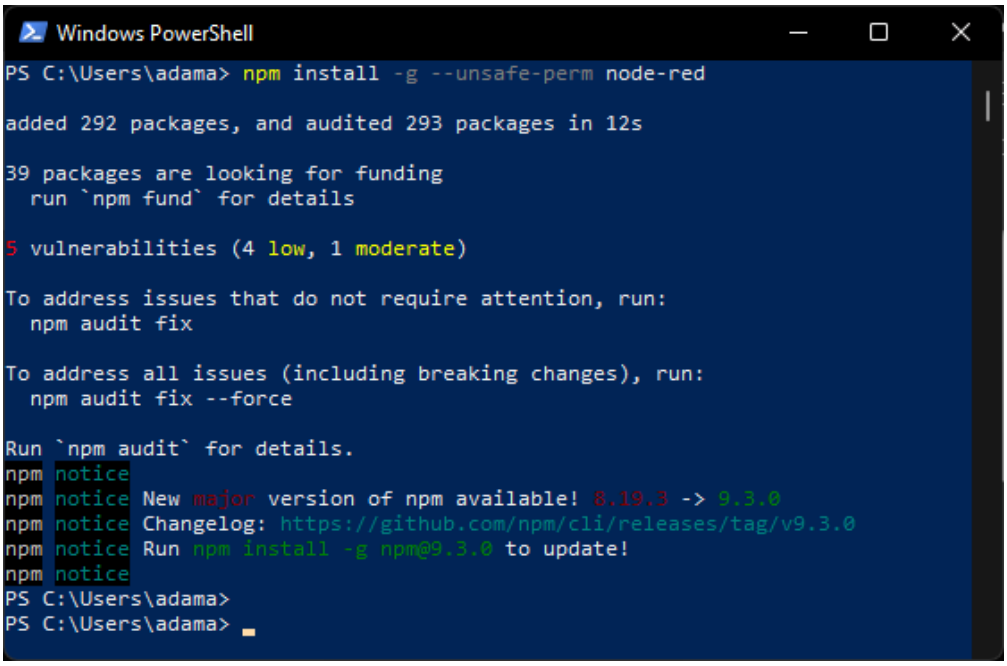


Check to ensure that NPM has been installed correctly.



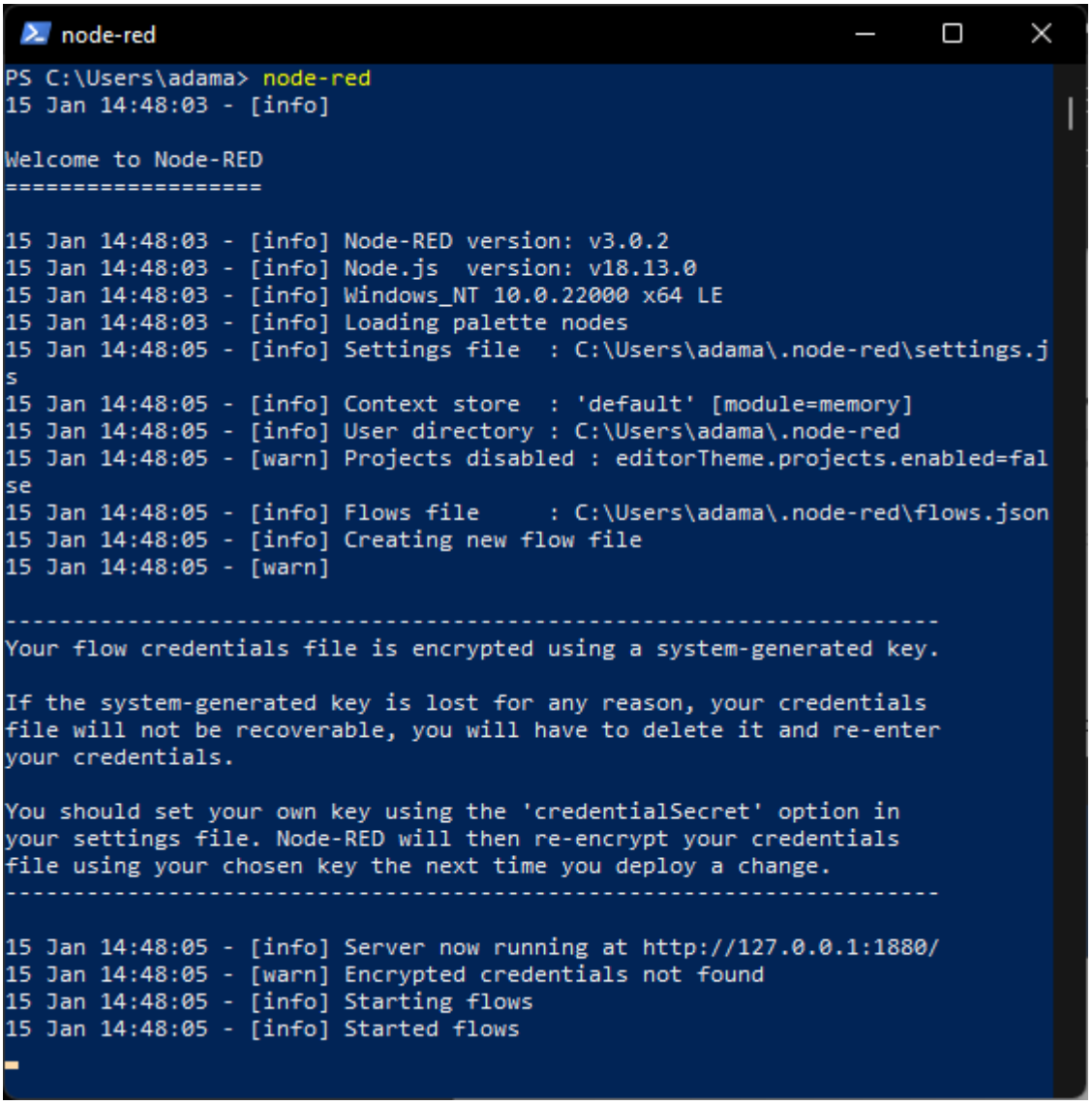
npm --version

Install NodeRED using the following command:



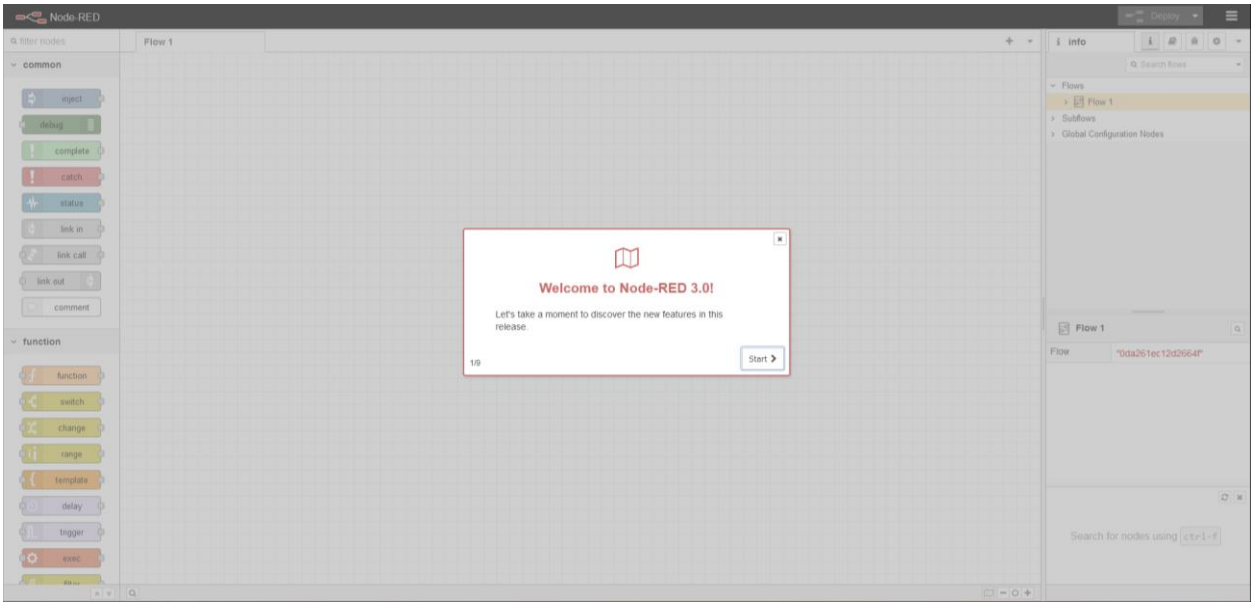
npm install -g --unsafe-perm node-red

To start NodeRED, use the following command:



Notice the line **Server now running at…**

Type that **IP address** into your **web browser**. Do not close this terminal window until you are done using NodeRED. Ensure that you see the **Welcome to Node-RED** screen.

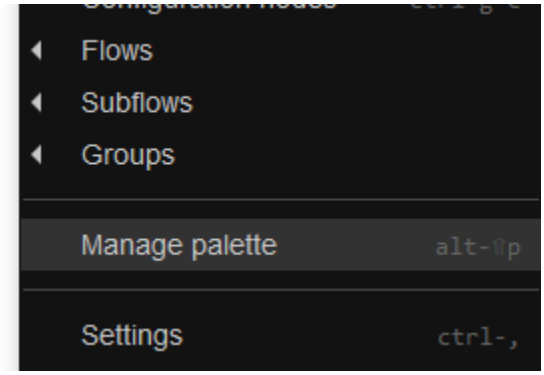


To safely close NodeRED, open the Powershell window and press **CTRL + C**.

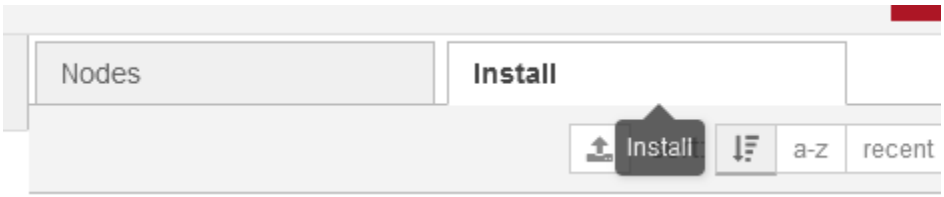
Installing NodeRED Dashboard

NodeRED Dashboard provides us a ready-to-use interface for visualizing our NodeRED applications.

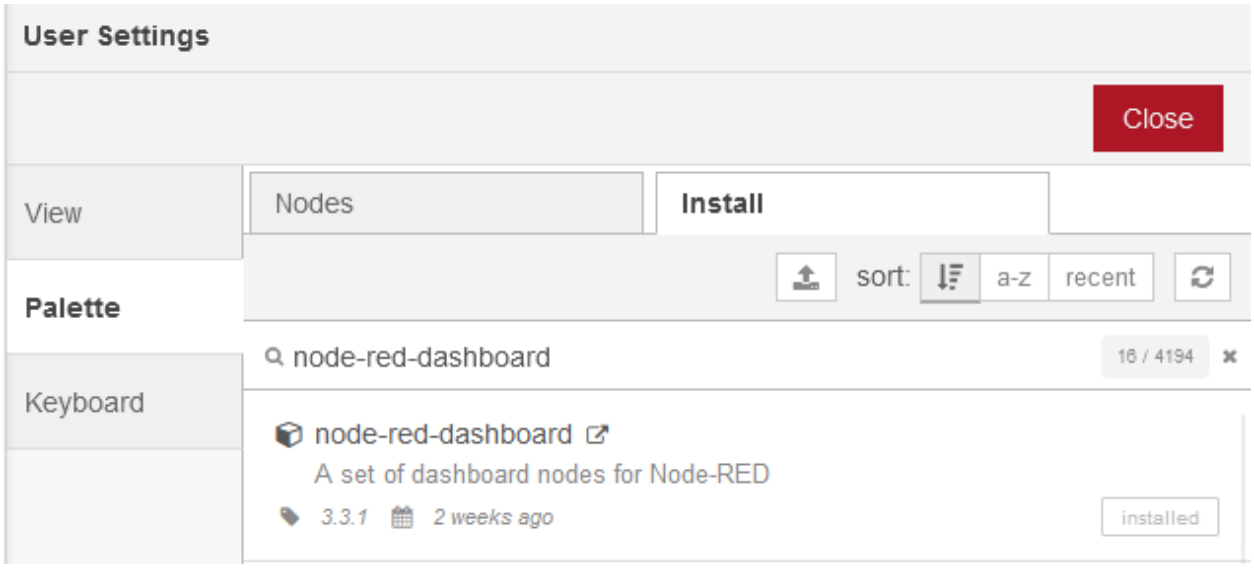
To add the **node-red-dashboard** add-on, navigate to **Manage palette** by pressing on the hamburger menu on the top right corner of the NodeRED interface.



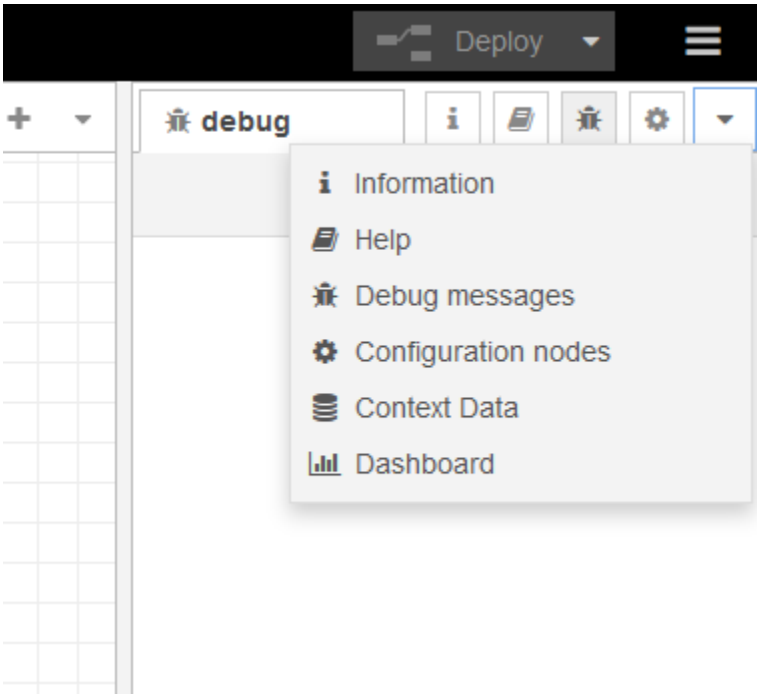
Click the **Install** tab.



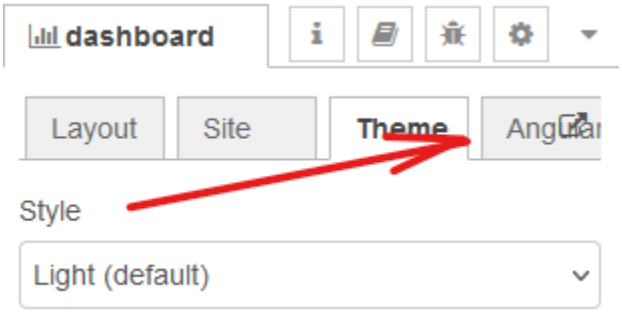
Filter for **node-red-dashboard** and press **Install**.

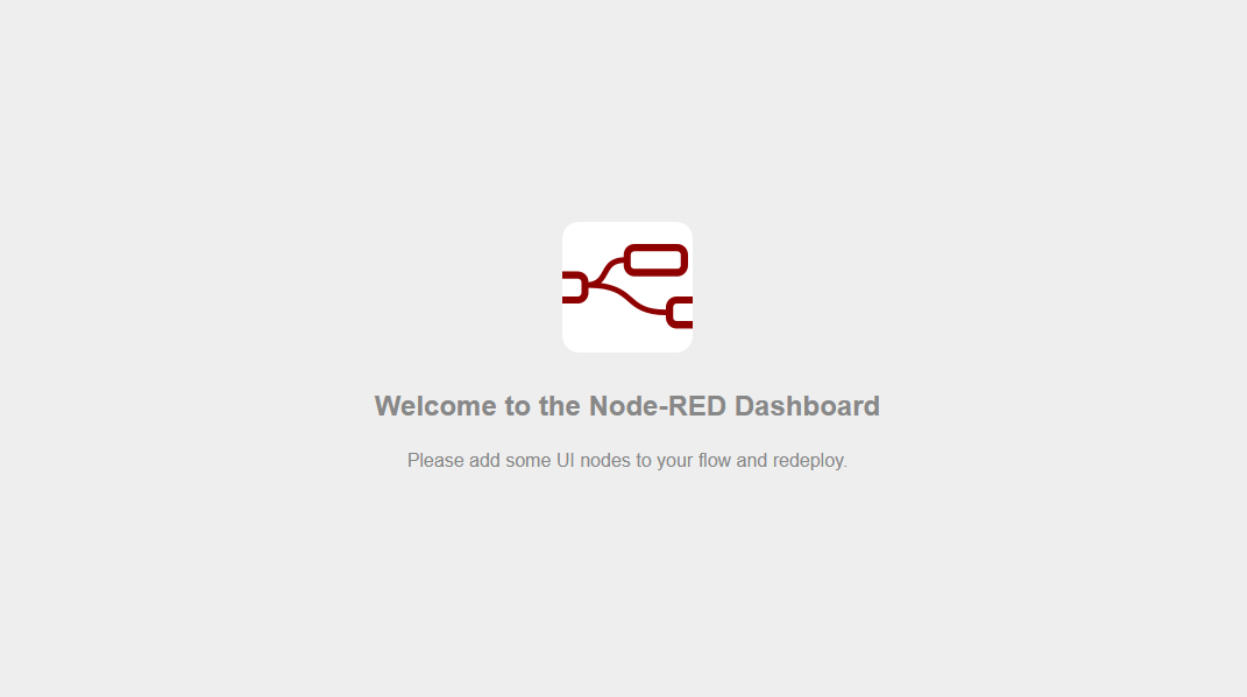


Using the **dropdown menu**, click on **Dashboard**.



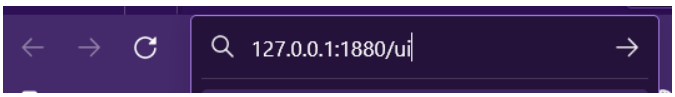
Press the hard to see button to open the dashboard.





If you want to check if it successfully installed, to get to the dashboard without pressing the **Open Dashboard Icon**, you can instead launch **node-red** from Powershell and in your web brower, use the following url:

127.0.0.1:1880/ui



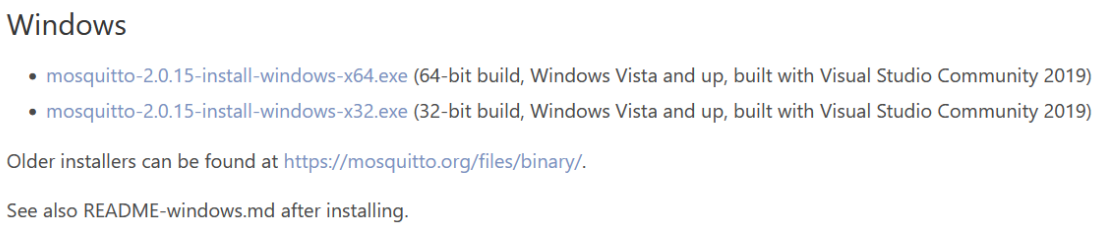
Mosquitto MQTT Broker Installation

Mosquitto is a free MQTT broker from the Eclipse foundation.

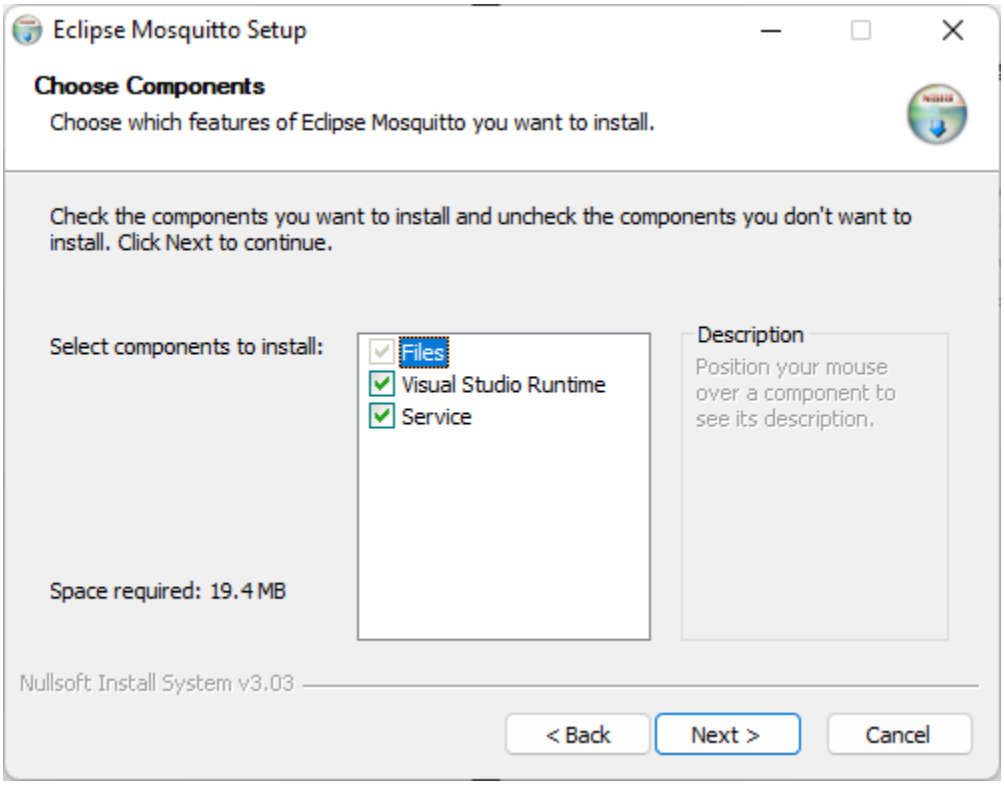
Navigate to the following downloads page:

https://mosquitto.org/download/

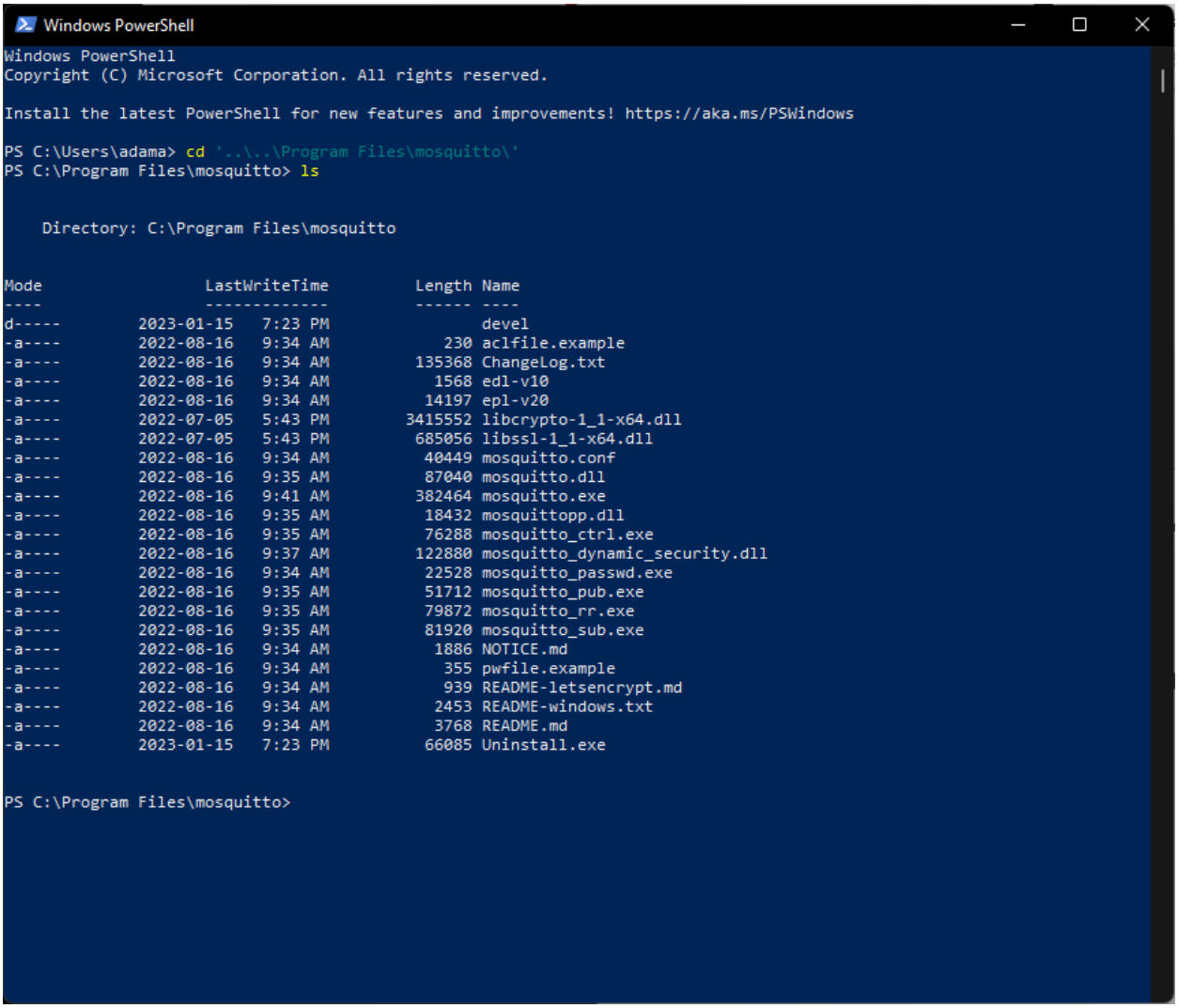
Select the version for your operating system.



Install using the default options.

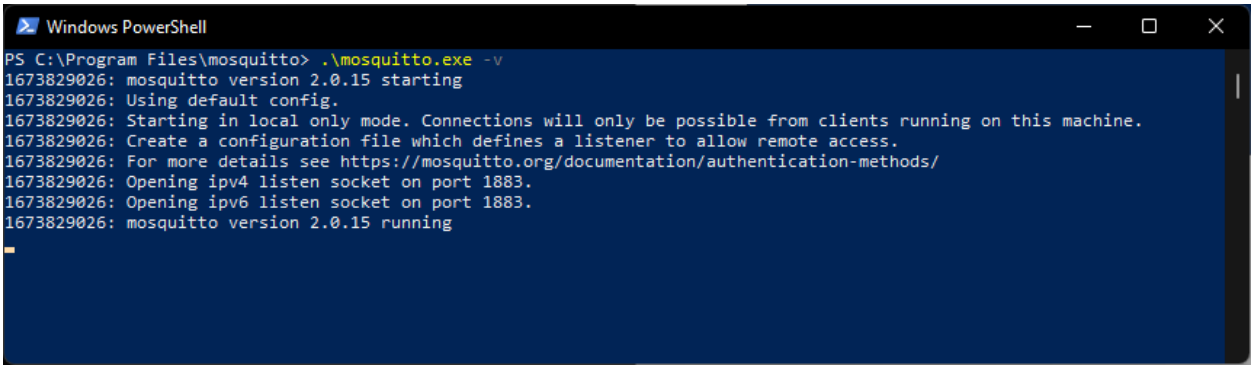


Once installed, open Powershell and navigate to the install directory.



cd 'C:\Program Files\mosquitto\'

Start the broker using the following command:



.\mosquitto.exe -v

If the server starts up without errors then the installation has been successful.

End