WSL (Windows-only):

Run a PowerShell terminal as Administrator. Enter the following commands:

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

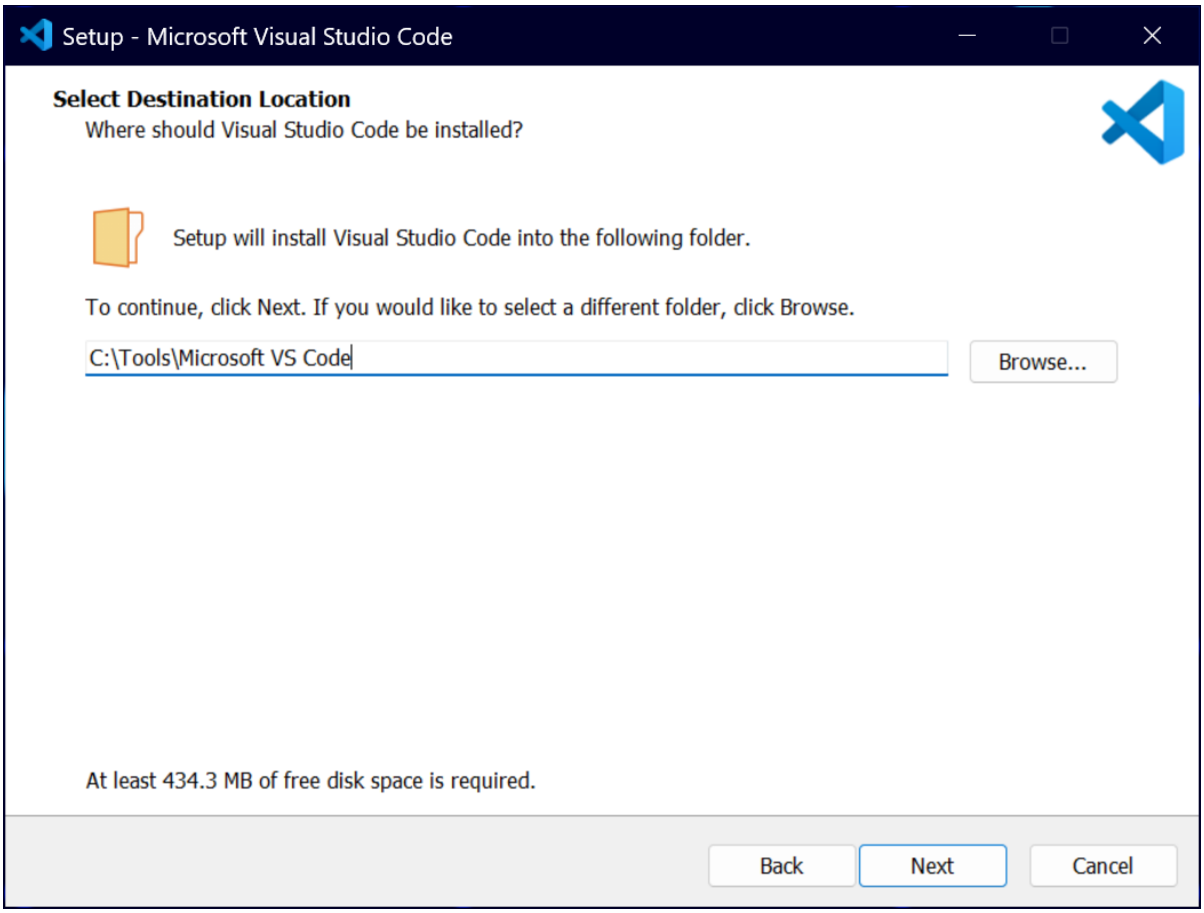
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

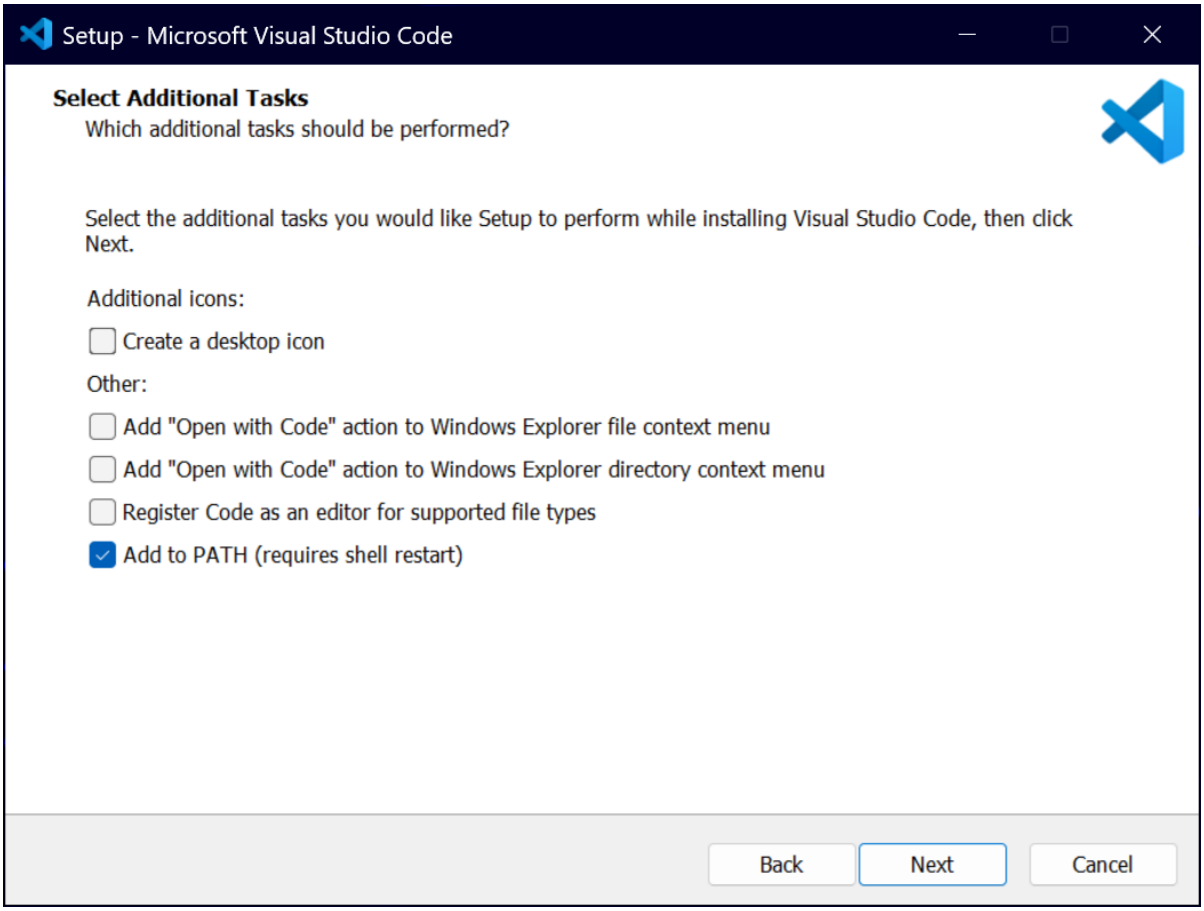
Reboot. Then re-open a PowerShell a terminal as Administrator. Run the following commands:

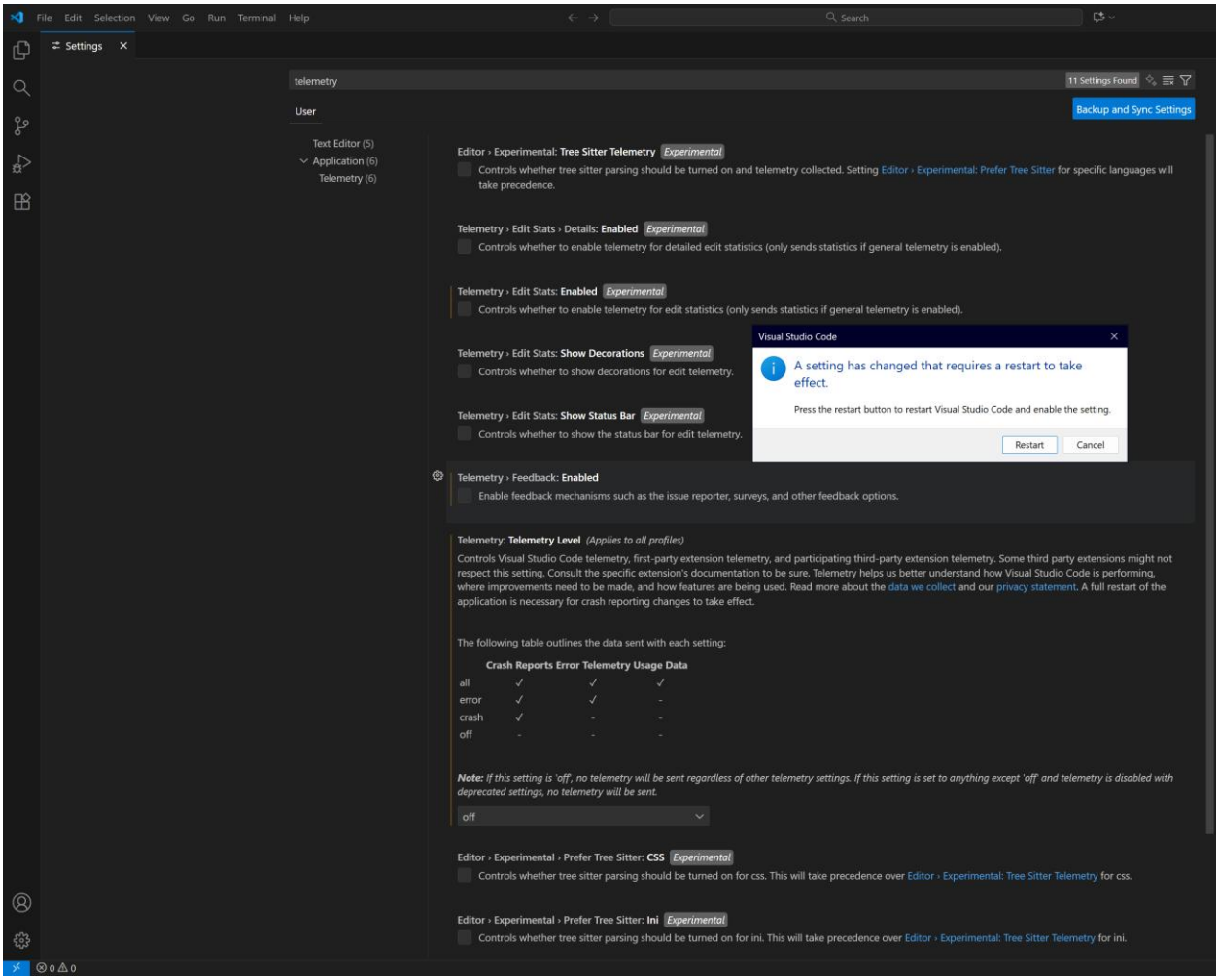
wsl --set-default-version 2

wsl --install Ubuntu-24.04

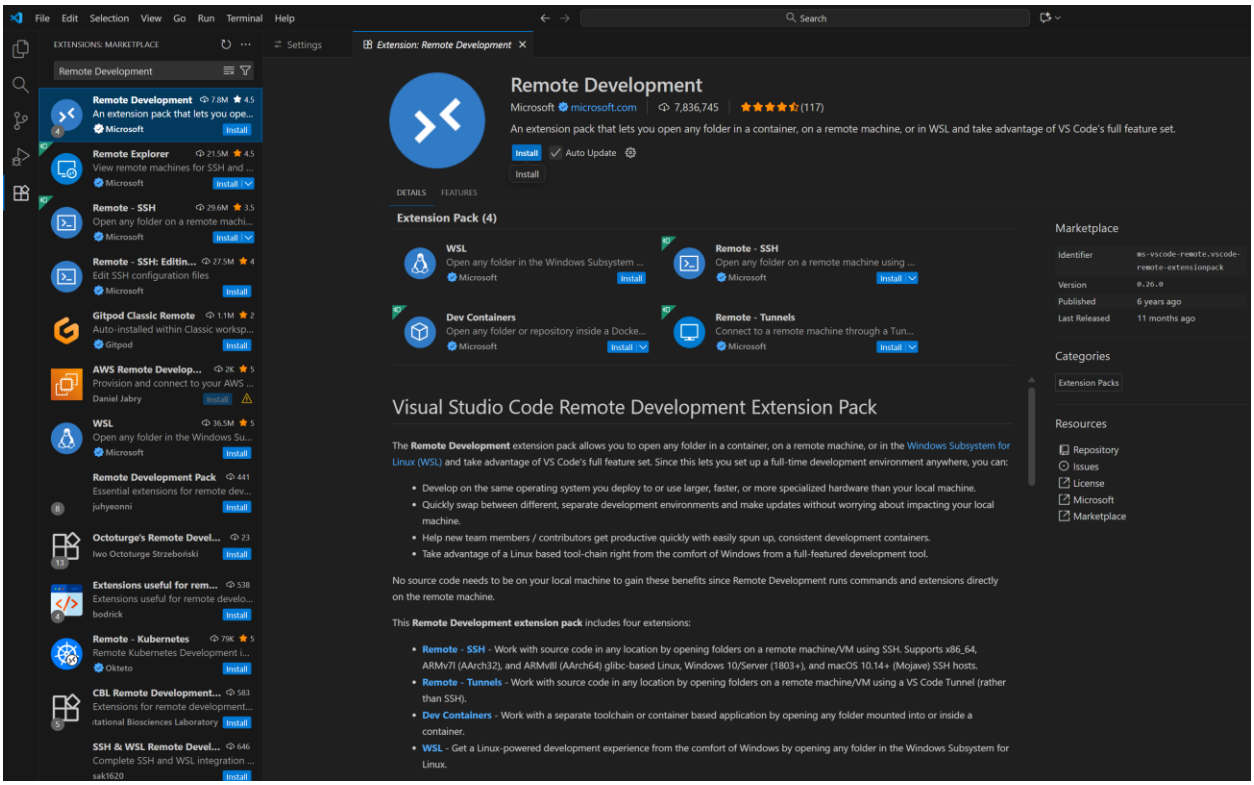
VSCODE:

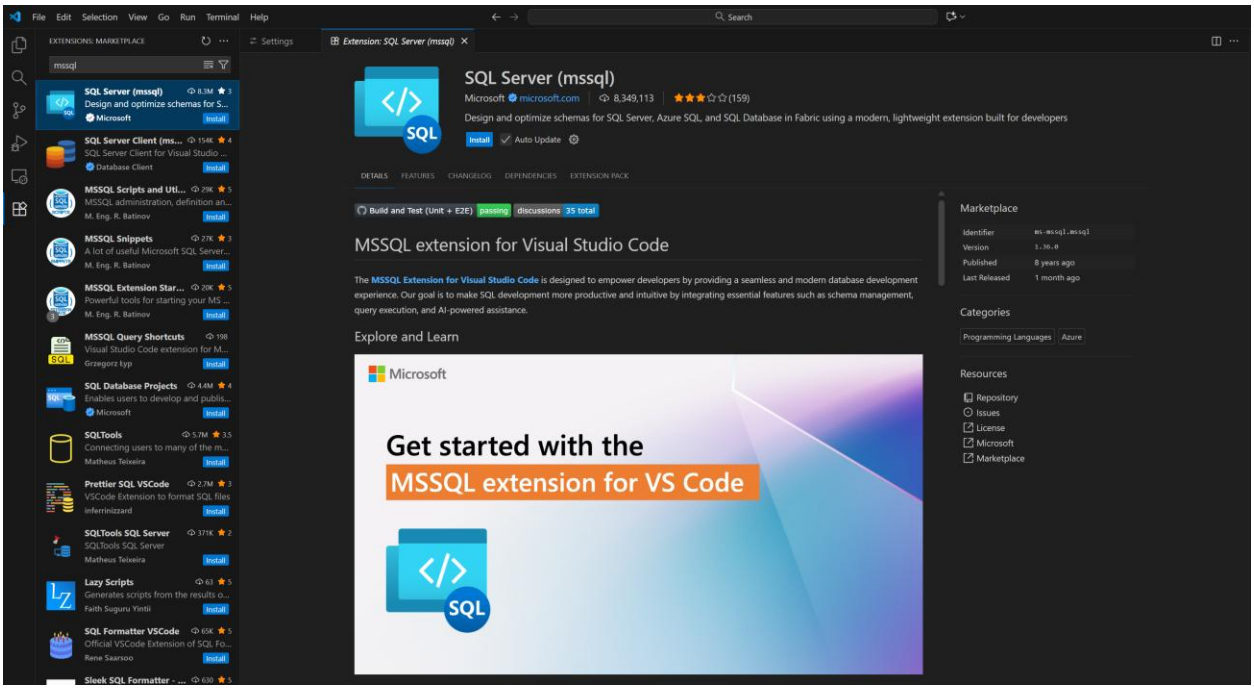

Setup - Microsoft Visual Studio Code
Select Destination Location
Where should Visual Studio Code be installed?
Setup will install Visual Studio Code into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Tools\Microsoft VS Code
Browse...
At least 434.3 MB of free disk space is required.
Back
Next
Cancel

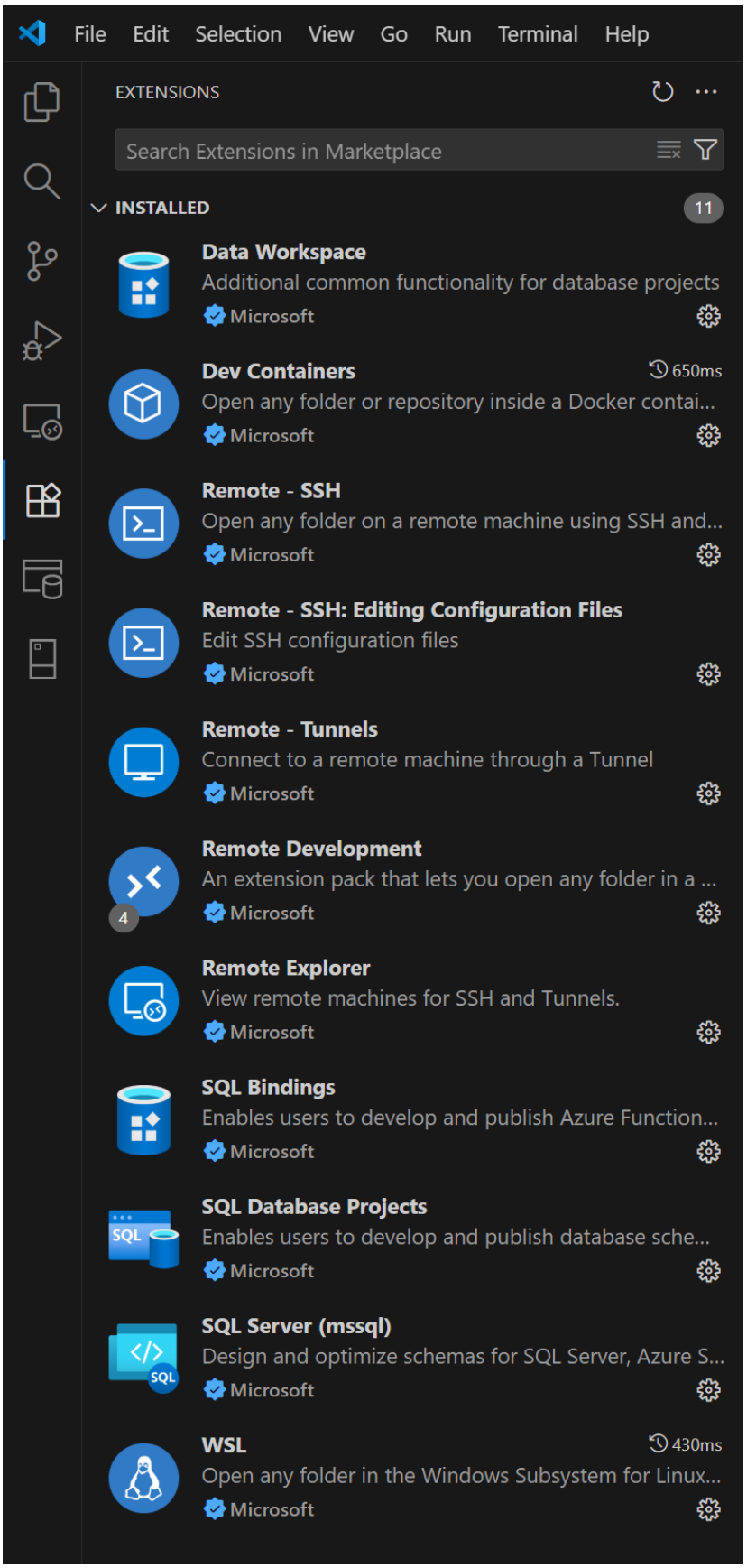

Setup - Microsoft Visual Studio Code
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.
Additional icons:
Create a desktop icon
Other:
Add "Open with Code" action to Windows Explorer file context menu
Add "Open with Code" action to Windows Explorer directory context menu
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
Back
Next
Cancel


File Edit Selection View Go Run Terminal Help
Search
Settings
telemetry
11 Settings Found
User
Backup and Sync Settings
Text Editor (5)
Application (6)
Telemetry (6)
Editor › Experimental: Tree Sitter Telemetry Experimental
Controls whether tree sitter parsing should be turned on and telemetry collected. Setting Editor › Experimental: Prefer Tree Sitter for specific languages will take precedence.
Telemetry › Edit Stats › Details: Enabled Experimental
Controls whether to enable telemetry for detailed edit statistics (only sends statistics if general telemetry is enabled).
Telemetry › Edit Stats: Enabled Experimental
Controls whether to enable telemetry for edit statistics (only sends statistics if general telemetry is enabled).
Telemetry › Edit Stats: Show Decorations Experimental
Controls whether to show decorations for edit telemetry.
Telemetry › Edit Stats: Show Status Bar Experimental
Controls whether to show the status bar for edit telemetry.
Visual Studio Code
A setting has changed that requires a restart to take effect.
Press the restart button to restart Visual Studio Code and enable the setting.
Restart
Cancel
Telemetry › Feedback: Enabled
Enable feedback mechanisms such as the issue reporter, surveys, and other feedback options.
Telemetry: Telemetry Level (Applies to all profiles)
Controls Visual Studio Code telemetry, first-party extension telemetry, and participating third-party extension telemetry. Some third party extensions might not respect this setting. Consult the specific extension's documentation to be sure. Telemetry helps us better understand how Visual Studio Code is performing, where improvements need to be made, and how features are being used. Read more about the data we collect and our privacy statement. A full restart of the application is necessary for crash reporting changes to take effect.
The following table outlines the data sent with each setting:
Crash Reports Error Telemetry Usage Data
all ✓ ✓ ✓
error ✓ ✓ -
crash ✓ - -
off - - -
Note: If this setting is 'off', no telemetry will be sent regardless of other telemetry settings. If this setting is set to anything except 'off' and telemetry is disabled with deprecated settings, no telemetry will be sent.
off
Editor › Experimental › Prefer Tree Sitter: CSS Experimental
Controls whether tree sitter parsing should be turned on for css. This will take precedence over Editor › Experimental: Tree Sitter Telemetry for css.
Editor › Experimental › Prefer Tree Sitter: Ini Experimental
Controls whether tree sitter parsing should be turned on for ini. This will take precedence over Editor › Experimental: Tree Sitter Telemetry for ini.
0 0

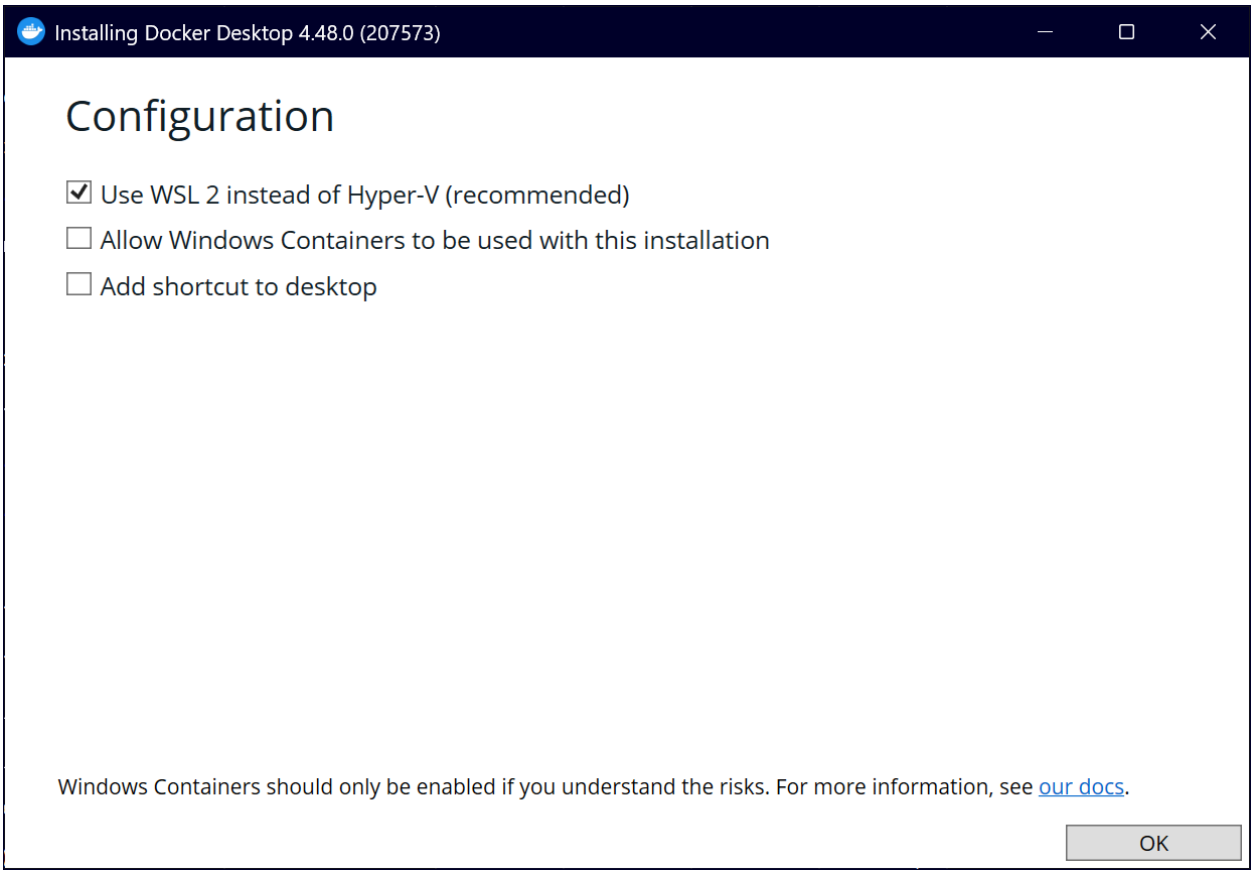
Windows-only:



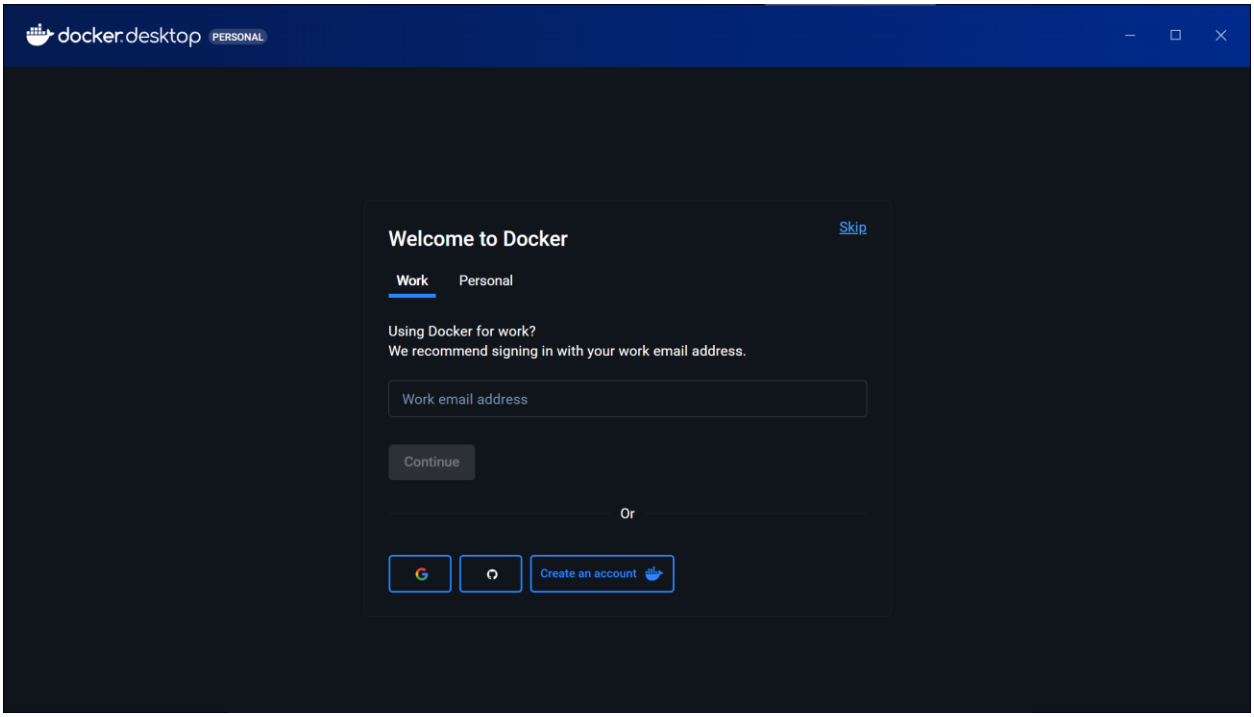



File Edit Selection View Go Run Terminal Help
EXTENSIONS
Search Extensions in Marketplace
INSTALLED 11
Data Workspace
Additional common functionality for database projects
Microsoft
Dev Containers 650ms
Open any folder or repository inside a Docker contai...
Microsoft
Remote - SSH
Open any folder on a remote machine using SSH and...
Microsoft
Remote - SSH: Editing Configuration Files
Edit SSH configuration files
Microsoft
Remote - Tunnels
Connect to a remote machine through a Tunnel
Microsoft
Remote Development
An extension pack that lets you open any folder in a ...
Microsoft
4
Remote Explorer
View remote machines for SSH and Tunnels.
Microsoft
SQL Bindings
Enables users to develop and publish Azure Function...
Microsoft
SQL Database Projects
Enables users to develop and publish database sche...
Microsoft
SQL Server (mssql)
Design and optimize schemas for SQL Server, Azure S...
Microsoft
WSL 430ms
Open any folder in the Windows Subsystem for Linux...
Microsoft

Docker Desktop:



Logout and then log in.

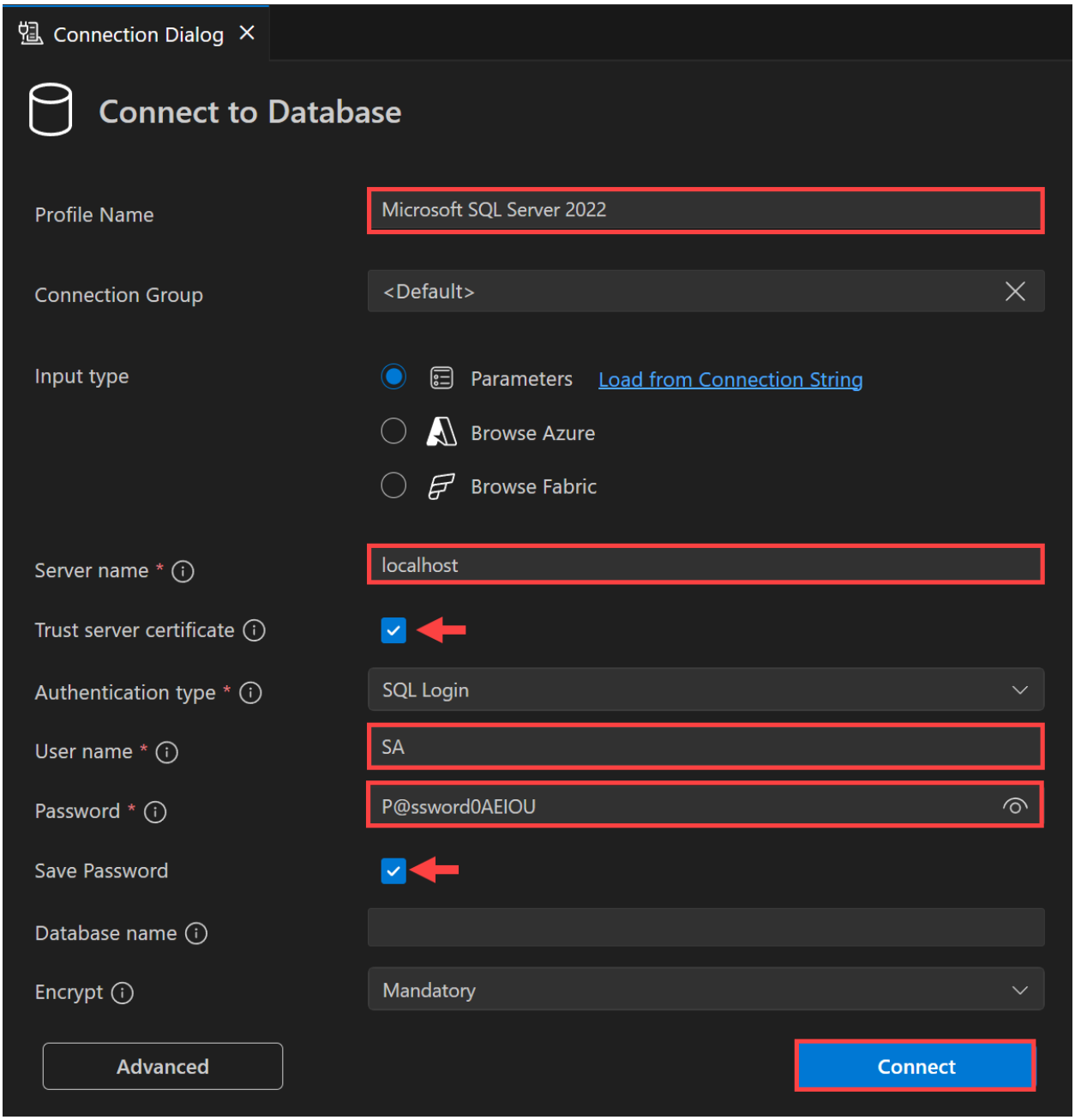


MSSQL:

In a (WSL) terminal, enter the following commands:

docker network create tc1004b

docker run -d --name mssql --network tc1004b -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=P@ssword0AEIOU" -p 1433:1433 mcr.microsoft.com/mssql/server:2022-latest




```
CREATE DATABASE tc1004b;
GO

USE tc1004b;
```

```
CREATE TABLE prj (
        id              INT             IDENTITY(1,1) PRIMARY KEY,
        timestamp       DATETIME        NOT NULL,
        sensor          VARCHAR(25)     NULL,
        value           REAL            NULL);

INSERT INTO prj (timestamp, sensor, value)
VALUES (GETDATE(), 'sensor', 0.0);
```

```
USE master;
GO
DROP DATABASE tc1004b;
GO
```

MOSQUITTO

In a (WSL) terminal, enter the following commands:

```
docker run -d --name mosquitto --network tc1004b -p 1883:1883 eclipse-mosquitto:latest
docker exec mosquitto /bin/sh -c 'cat << __EOF__ >> /mosquitto/config/mosquitto.conf
```

listener 1883

allow_anonymous true

__EOF__

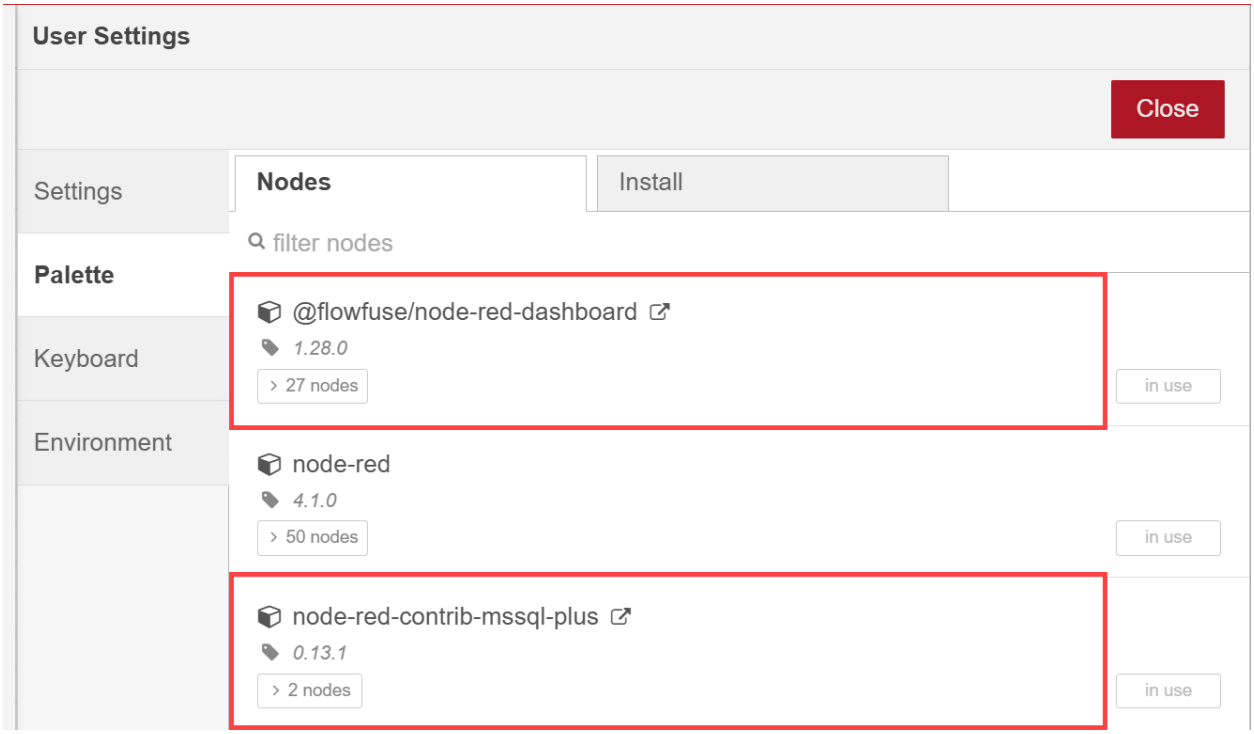
'

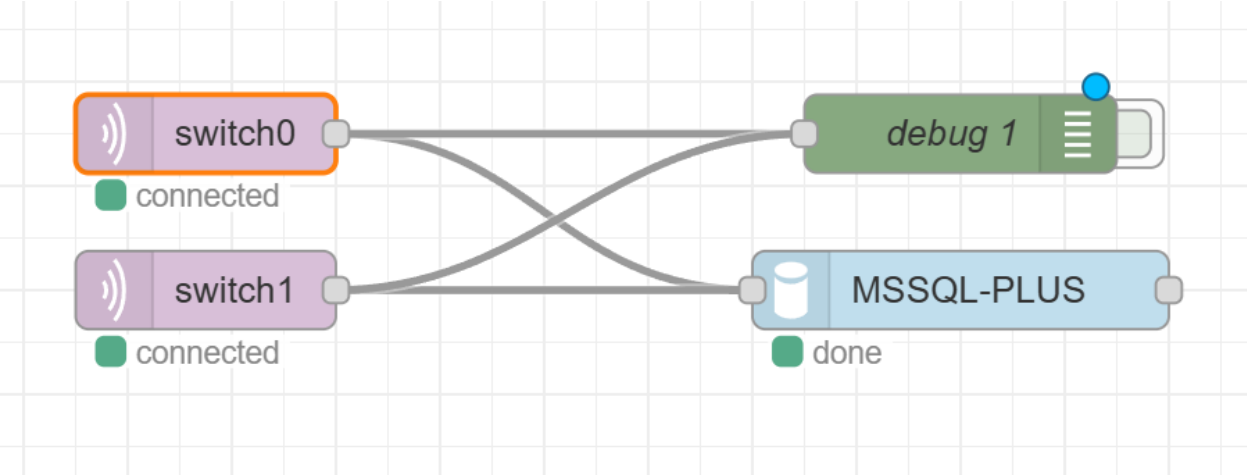
docker restart mosquitto

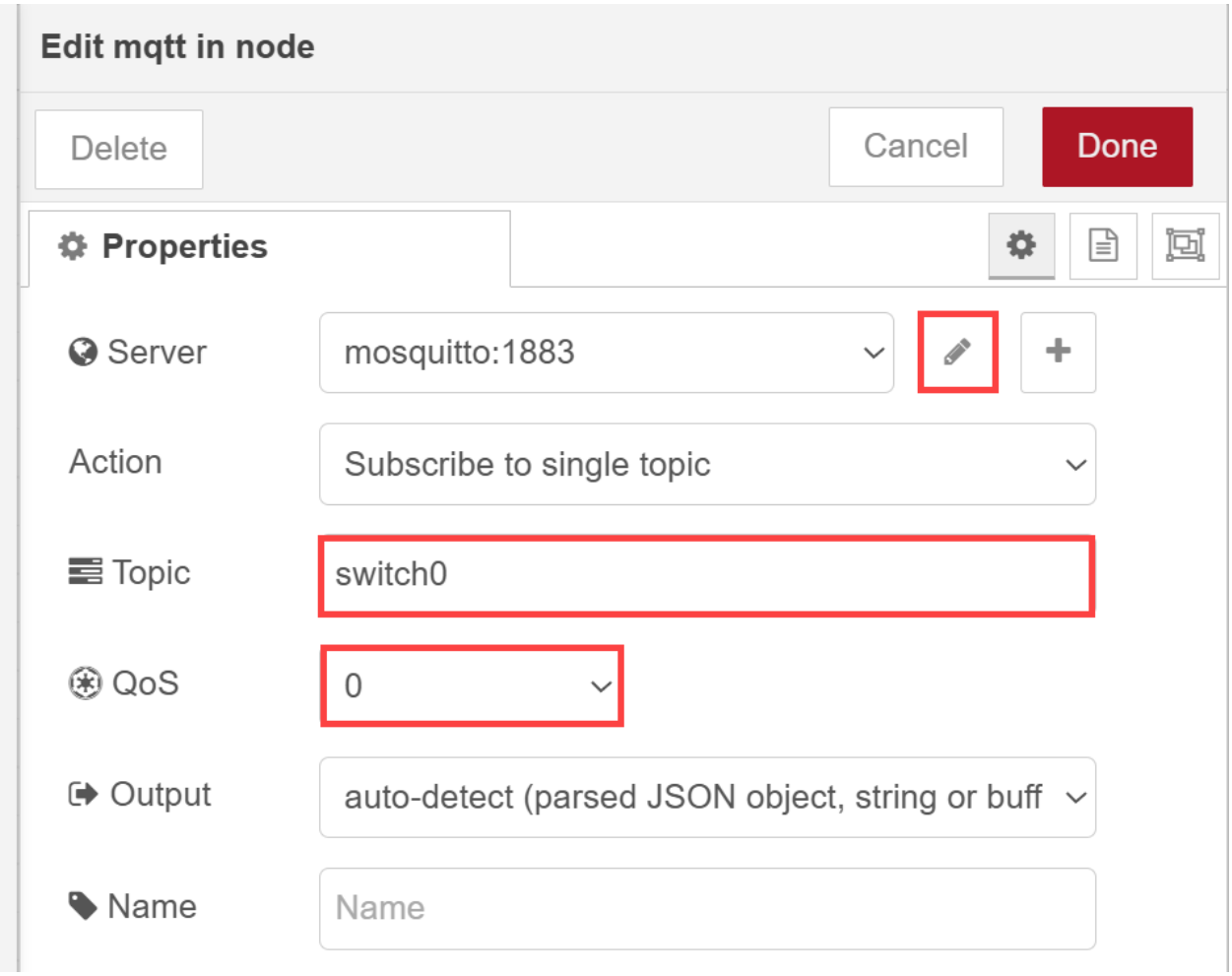
NODE-RED

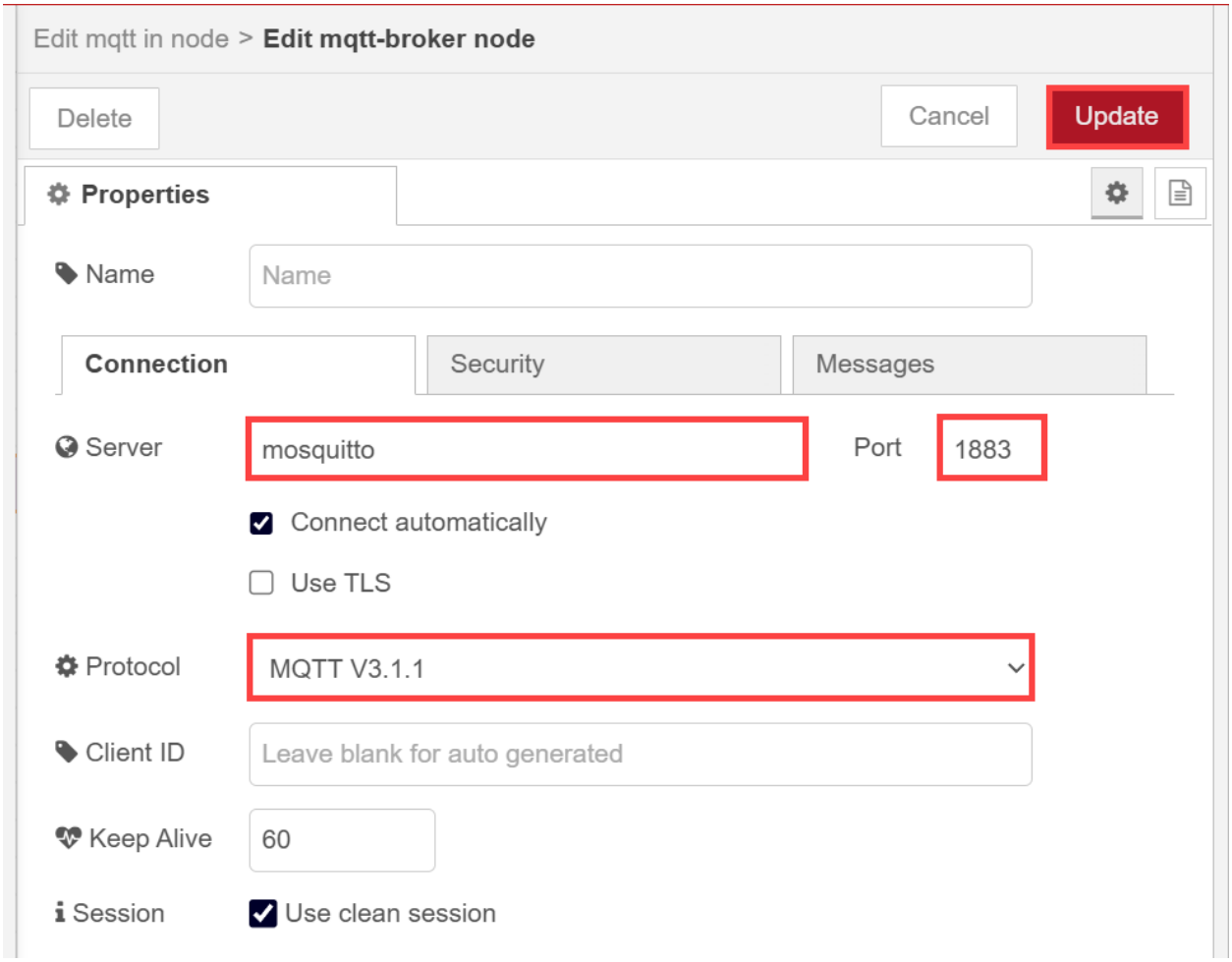
docker run -d --name nodered --network tc1004b -p 1880:1880 nodered/node-red

In a browser enter: http://localhost:1880/

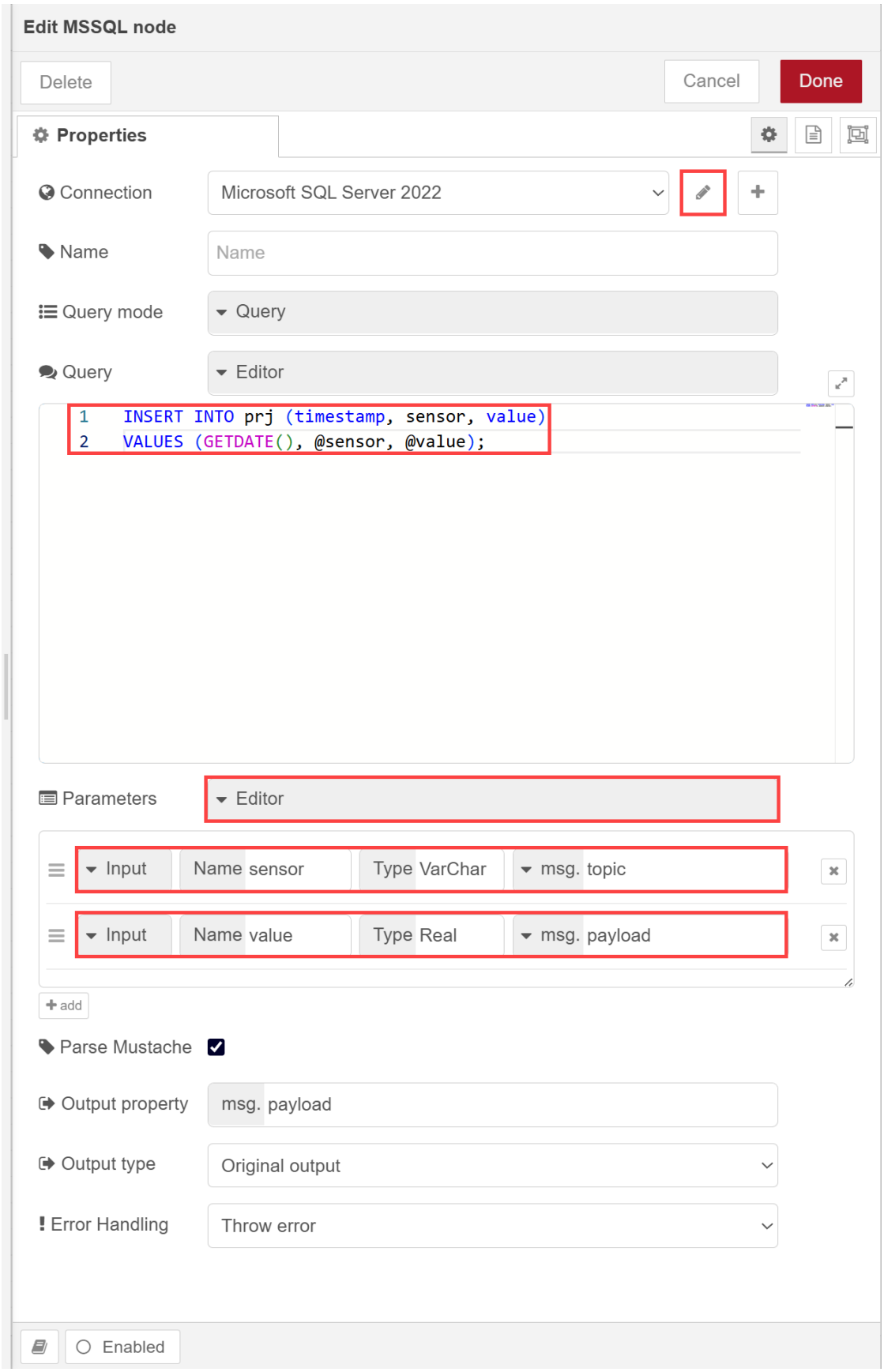

User Settings
Close
Settings
Palette
Keyboard
Environment
Nodes
Install
filter nodes
@flowfuse/node-red-dashboard
1.28.0
27 nodes
in use
node-red
4.1.0
50 nodes
in use
node-red-contrib-mssql-plus
0.13.1
2 nodes
in use

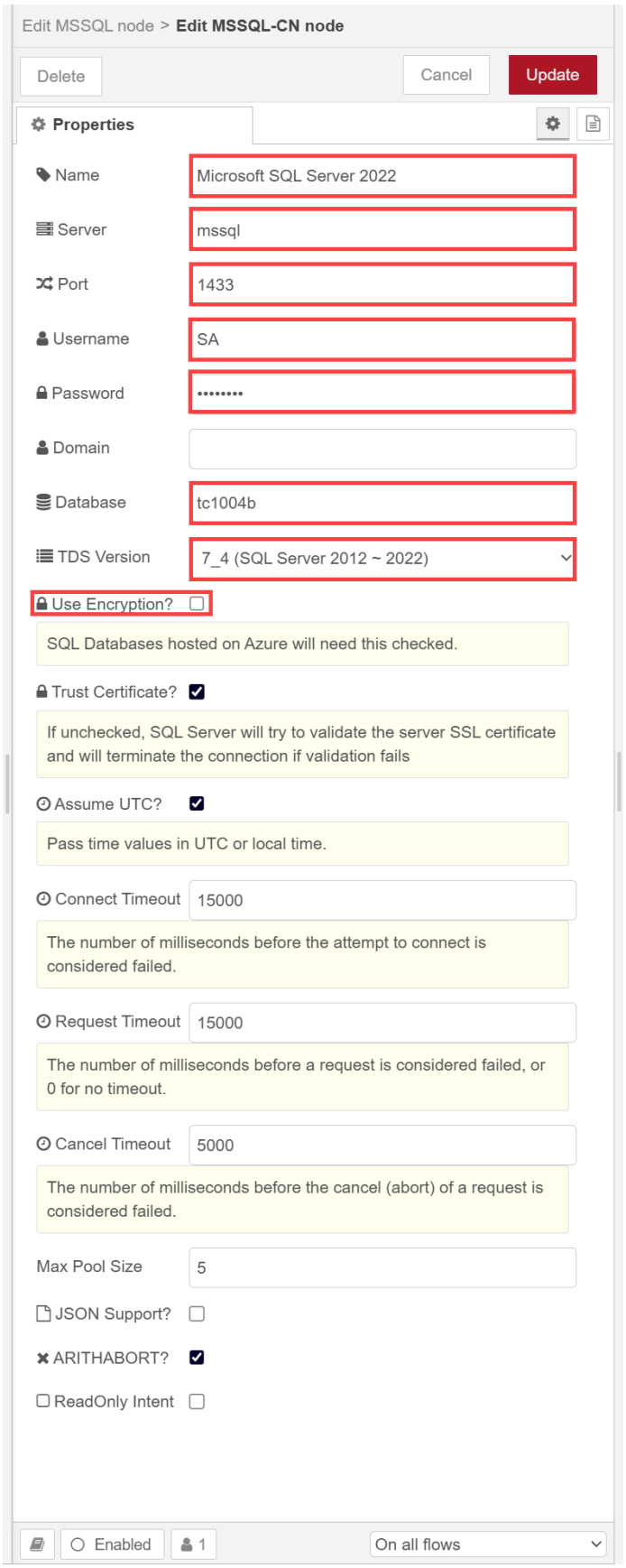

switch0
connected
switch1
connected
debug 1
MSSQL-PLUS
done


Edit mqtt in node
Delete
Cancel
Done
Properties
Server
mosquitto:1883
Action
Subscribe to single topic
Topic
switch0
QoS
0
Output
auto-detect (parsed JSON object, string or buff
Name
Name


Edit mqtt in node > Edit mqtt-broker node
Delete
Cancel
Update
Properties
Name
Name
Connection
Security
Messages
Server
mosquitto
Port
1883
Connect automatically
Use TLS
Protocol
MQTT V3.1.1
Client ID
Leave blank for auto generated
Keep Alive
60
Session
Use clean session

```
INSERT INTO prj (timestamp, sensor, value)
VALUES (GETDATE(), @sensor, @value);
```

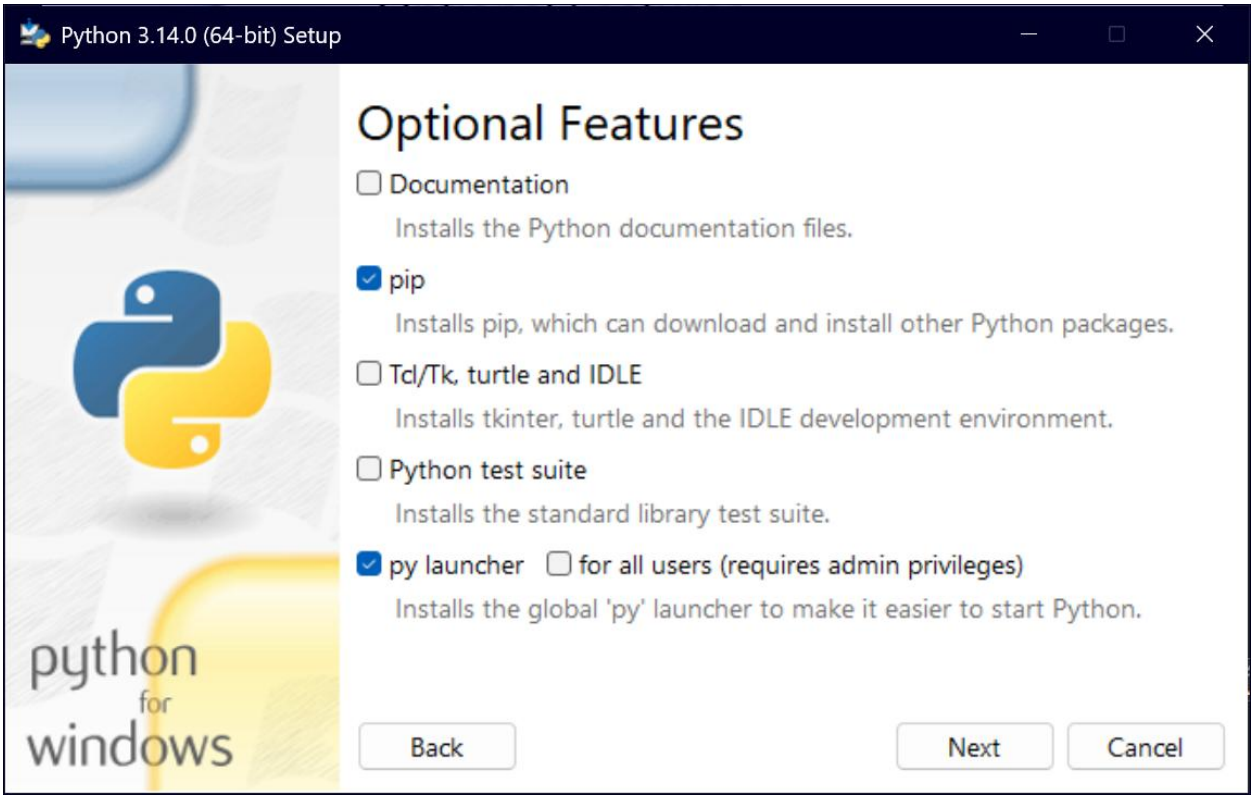

Edit MSSQL node
Delete
Cancel
Done
Properties
Connection
Microsoft SQL Server 2022
Name
Name
Query mode
Query
Query
Editor
1 INSERT INTO prj (timestamp, sensor, value)
2 VALUES (GETDATE(), @sensor, @value);
Parameters
Editor
Input
Name sensor
Type VarChar
msg. topic
Input
Name value
Type Real
msg. payload
add
Parse Mustache
Output property
msg. payload
Output type
Original output
Error Handling
Throw error
Enabled

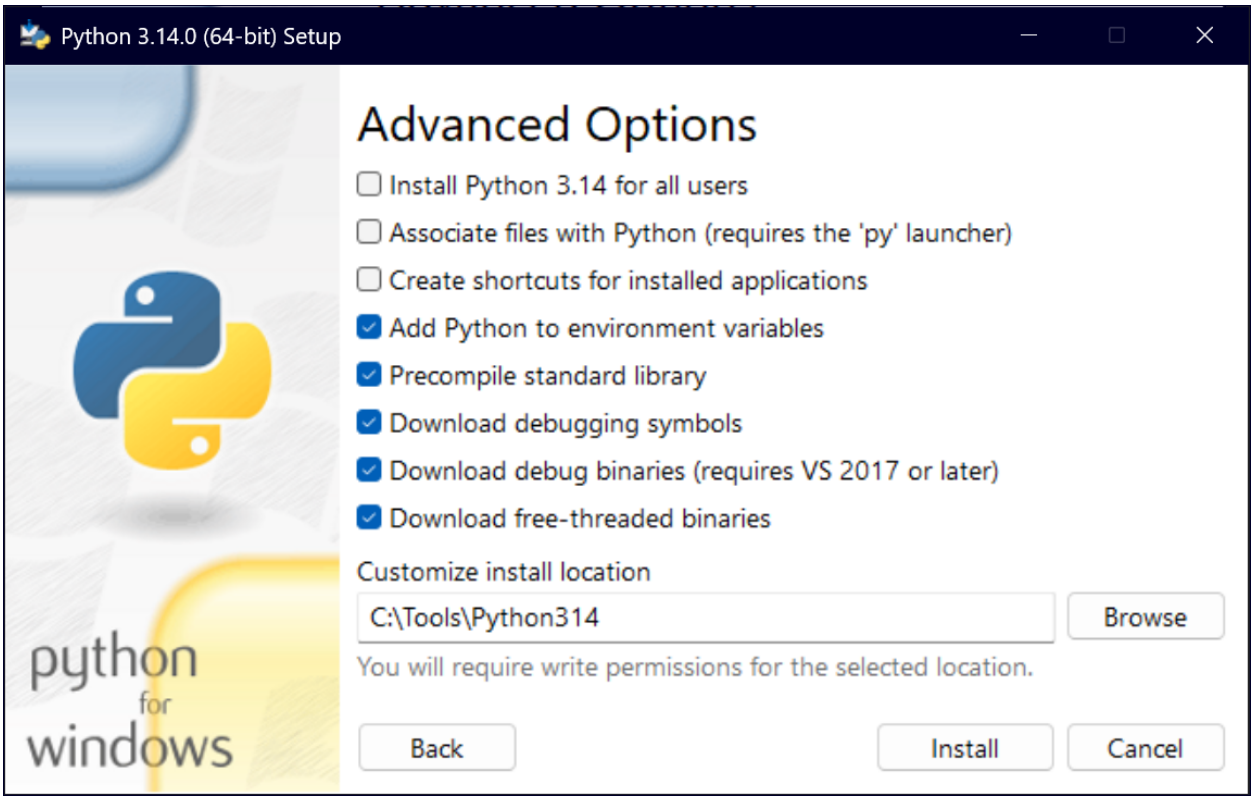

Edit MSSQL node > Edit MSSQL-CN node
Delete
Cancel
Update
Properties
Name
Microsoft SQL Server 2022
Server
mssql
Port
1433
Username
SA
Password
Domain
Database
tc1004b
TDS Version
7_4 (SQL Server 2012 ~ 2022)
Use Encryption?
SQL Databases hosted on Azure will need this checked.
Trust Certificate?
If unchecked, SQL Server will try to validate the server SSL certificate and will terminate the connection if validation fails
Assume UTC?
Pass time values in UTC or local time.
Connect Timeout
15000
The number of milliseconds before the attempt to connect is considered failed.
Request Timeout
15000
The number of milliseconds before a request is considered failed, or 0 for no timeout.
Cancel Timeout
5000
The number of milliseconds before the cancel (abort) of a request is considered failed.
Max Pool Size
5
JSON Support?
ARITHABORT?
ReadOnly Intent
Enabled
1
On all flows

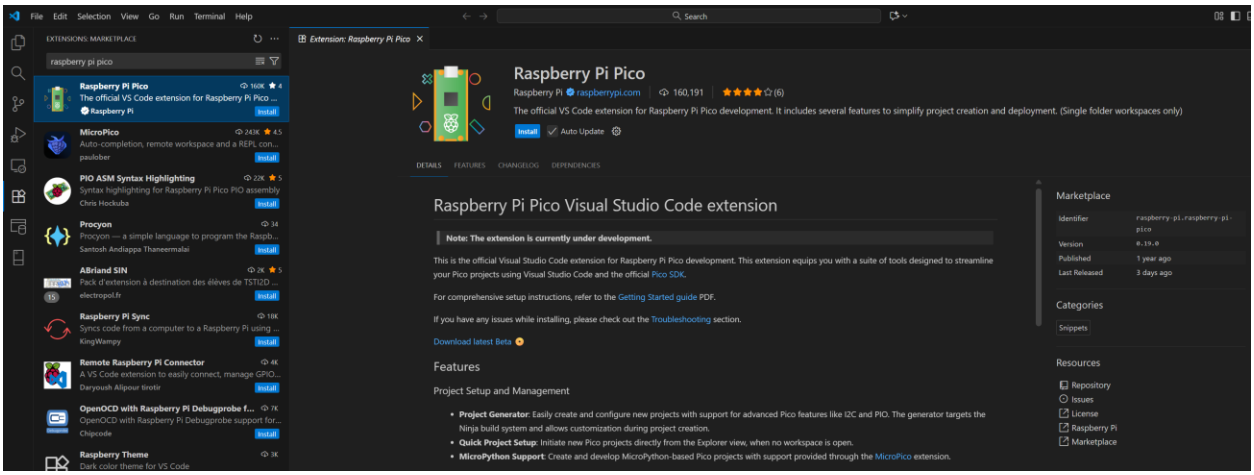
```
mosquitto_pub -t switch0 -m 3.1416


SELECT TOP (10) [id]
      ,[timestamp]
      ,[sensor]
      ,[value]
  FROM [tc1004b].[dbo].[prj]
  ORDER BY id DESC
```

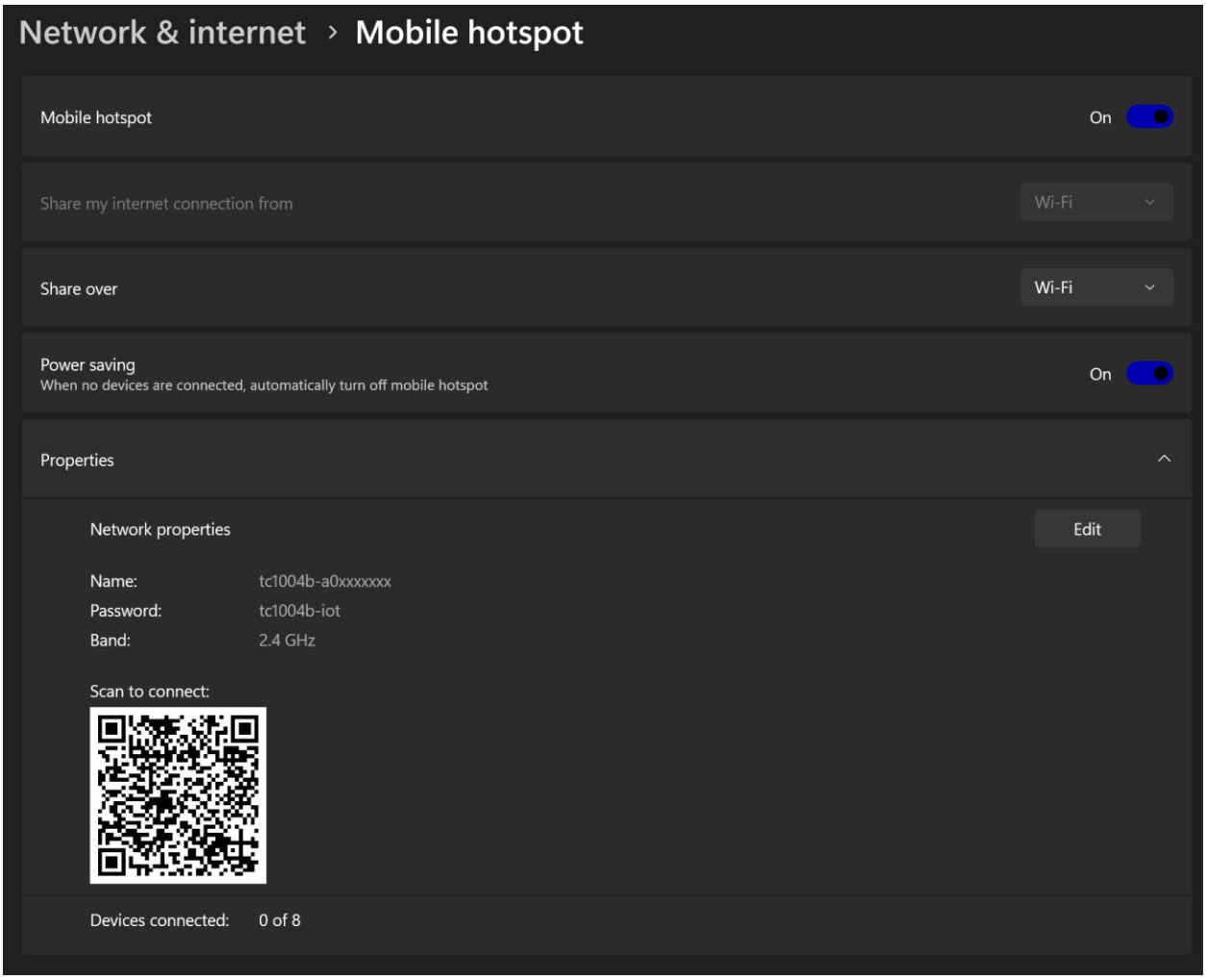
RPi-Pico W2:


Python 3.14.0 (64-bit) Setup
Install Python 3.14.0 (64-bit)
Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.
NOTE: This installer is being retired and will no longer be available after Python 3.15. More info
Install Now
C:\Tools\Python314
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python
for
windows
Use admin privileges when installing py.exe
Add python.exe to PATH
Cancel


Python 3.14.0 (64-bit) Setup
Optional Features
Documentation
Installs the Python documentation files.
pip
Installs pip, which can download and install other Python packages.
Tcl/Tk, turtle and IDLE
Installs tkinter, turtle and the IDLE development environment.
Python test suite
Installs the standard library test suite.
py launcher
for all users (requires admin privileges)
Installs the global 'py' launcher to make it easier to start Python.
python
for
windows
Back
Next
Cancel


Python 3.14.0 (64-bit) Setup
Advanced Options
Install Python 3.14 for all users
Associate files with Python (requires the 'py' launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2017 or later)
Download free-threaded binaries
Customize install location
C:\Tools\Python314
Browse
You will require write permissions for the selected location.
python
for
windows
Back
Install
Cancel


File Edit Selection View Go Run Terminal Help
EXTENSIONS: MARKETPLACE
raspberry pi pico
Raspberry Pi Pico
The official VS Code extension for Raspberry Pi Pico ...
Raspberry Pi
MicroPico
Auto-completion, remote workspace and a REPL con...
paulober
PIO ASM Syntax Highlighting
Syntax highlighting for Raspberry Pi Pico PIO assembly
Chris Hockuba
Procyon
Procyon — a simple language to program the Raspb...
Santosh Andiappa Thaneermalai
ABrland SIN
Pack d'extension à destination des élèves de TSTI2D ...
electropol.fr
Raspberry Pi Sync
Syncs code from a computer to a Raspberry Pi using ...
KingWampy
Remote Raspberry Pi Connector
A VS Code extension to easily connect, manage GPIO...
Daryoush Alipour tirotir
OpenOCD with Raspberry Pi Debugprobe f...
OpenOCD with Raspberry Pi Debugprobe support for...
Chipcode
Raspberry Theme
Dark color theme for VS Code
Extension: Raspberry Pi Pico
Raspberry Pi Pico
Raspberry Pi raspberrypi.com 160,191
The official VS Code extension for Raspberry Pi Pico development. It includes several features to simplify project creation and deployment. (Single folder workspaces only)
Install Auto Update
DETAILS FEATURES CHANGELOG DEPENDENCIES
Raspberry Pi Pico Visual Studio Code extension
Note: The extension is currently under development.
This is the official Visual Studio Code extension for Raspberry Pi Pico development. This extension equips you with a suite of tools designed to streamline your Pico projects using Visual Studio Code and the official Pico SDK.
For comprehensive setup instructions, refer to the Getting Started guide PDF.
If you have any issues while installing, please check out the Troubleshooting section.
Download latest Beta
Features
Project Setup and Management
Project Generator: Easily create and configure new projects with support for advanced Pico features like I2C and PIO. The generator targets the Ninja build system and allows customization during project creation.
Quick Project Setup: Initiate new Pico projects directly from the Explorer view, when no workspace is open.
MicroPython Support: Create and develop MicroPython-based Pico projects with support provided through the MicroPico extension.
Marketplace
Identifier raspberry-pi.raspberry-pi-pico
Version 0.19.0
Published 1 year ago
Last Released 3 days ago
Categories
Snippets
Resources
Repository
Issues
License
Raspberry Pi
Marketplace


Network & internet › Mobile hotspot
Mobile hotspot
On
Share my internet connection from
Wi-Fi
Share over
Wi-Fi
Power saving
When no devices are connected, automatically turn off mobile hotspot
On
Properties
Network properties
Edit
Name: tc1004b-a0xxxxxxx
Password: tc1004b-iot
Band: 2.4 GHz
Scan to connect:
Devices connected: 0 of 8

```
import network
from utime import sleep

ssid='TC1004B-A0XXXXXXX'
password='TC1004B-IOT'

wlan = network.WLAN(network.STA_IF)
wlan.active(True)

while not wlan.isconnected():
    wlan.connect(ssid, password)
    sleep(5)

print("Connection successful!")

import mip
mip.install('umqtt.simple')
```

```
from machine import Pin
from utime import sleep
import network
from umqtt.simple import MQTTClient

switch      = []
switch_input = []
switch.append(Pin(2, Pin.IN, Pin.PULL_UP))
switch_input.append(0)
switch.append(Pin(5, Pin.IN, Pin.PULL_UP))
switch_input.append(0)

pin = Pin("LED", Pin.OUT)

SSID     = 'TC1004B-A0XXXXXXX'
PASSWORD = 'TC1004B-IOT'
wlan     = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(SSID, PASSWORD)
i = 0
while not wlan.isconnected() and i < 5:
   print('Waiting for network connection...\n')
   sleep(2)
   i += 1

MQTT_SERVER      = "192.168.137.1"
```

```
CLIENT_ID        = "RPI-PICO-2W"
MQTT_BROKER_PORT  = 1883
TOPICS           = ['switch0', 'switch1']
print("Connecting to MQTT broker...")
client = MQTTClient(CLIENT_ID, MQTT_SERVER, MQTT_BROKER_PORT)
client.connect()

while True:
    try:
        pin.toggle()

        switch_input[0] = switch[0].value()
        switch_input[1] = switch[1].value()
        print(f'switch_input[0] = {switch_input[0]}')
        print(f'switch_input[1] = {switch_input[1]}')
        client.publish(TOPICS[0], f'{switch_input[0]}', qos=0)
        client.publish(TOPICS[1], f'{switch_input[1]}', qos=0)

        sleep(2) # sleep 1sec
    except KeyboardInterrupt:
        print("Execution terminated via <CTRL>+C.\n")
        break

pin.off()
print("Finished execution.\n")
```