

**[ DATA  
INTELLECT ]**

# **[ KDB-X Lightning Labs Set Up Instructions**

**SETTING UP VS CODE, KDB-X, AND CONNECTING TO KX NOTEBOOKS**

**JANUARY 2026**

This guide walks you through installing VS Code, KDB-X, and connecting a local q process to KX notebooks. Instructions apply to Windows, macOS, and Linux.

## 1 Install VS Code

Download and install Visual Studio Code for your platform:

<https://code.visualstudio.com/download>

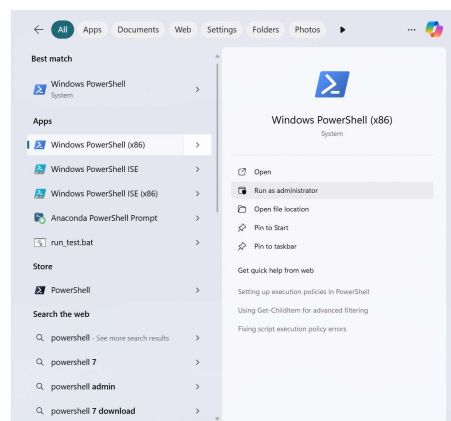
## 2 Platform-Specific Notes

### 2.1 Windows

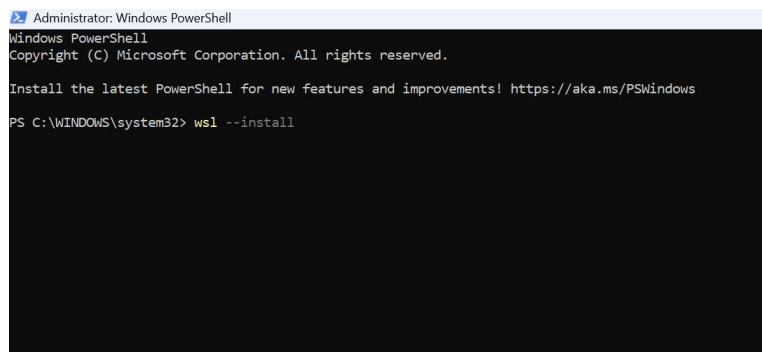
Install Windows Subsystem for Linux (WSL) if you plan to run a Linux environment.

Open PowerShell as Administrator and run:

`wsl --install`



*Image 1: Open PowerShell as Administrator*



*Image 2: Run 'wsl --install' in PowerShell*

If the x86 version of PowerShell does not recognise the command, use the non-x86 version.

**Note:** You may need to restart your laptop after installing and connecting to wsl.

## 2.2 macOS / Linux

Skip WSL and Remote Explorer steps. Open VS Code normally and continue.

## 3 Setup VS Code

The two necessary extensions are **Remote Explorer** and **wsl**.

To install, first search for 'Remote Explorer' and 'wsl' in the Extensions tab on the left sidebar.

Then on each click the install button.

### 3.1 Install unzip

1. Open terminal (Terminal --> new terminal)
2. sudo apt install unzip
3. type in password

This is necessary for installing kdb-x later.

### 3.2 Prepare Environment (Windows Only)

In VS Code, connect to the WSL environment:

1. Click Remote Explorer on the left sidebar.
2. Select WSL Targets from the dropdown options.
3. Choose your installed Linux distribution (default: Ubuntu)

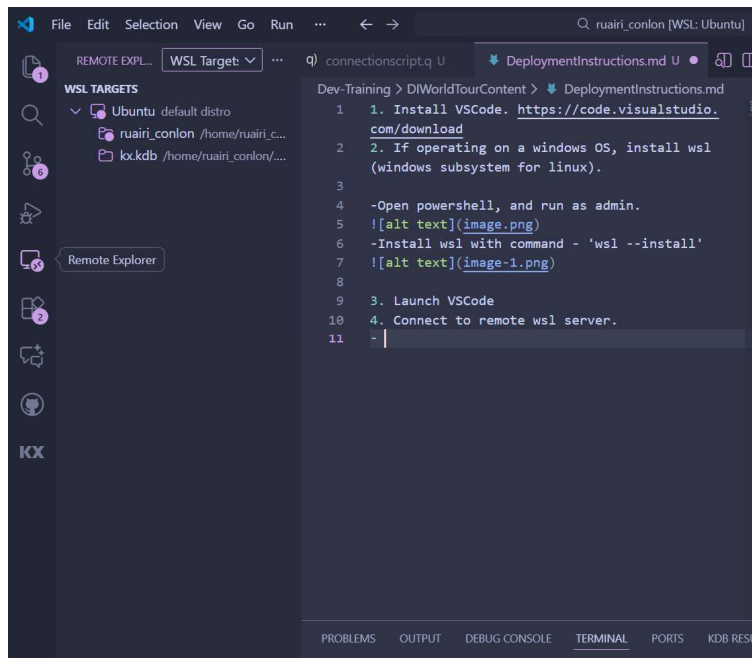


Image 3: Showing location of Remote Explorer tab in VSCode

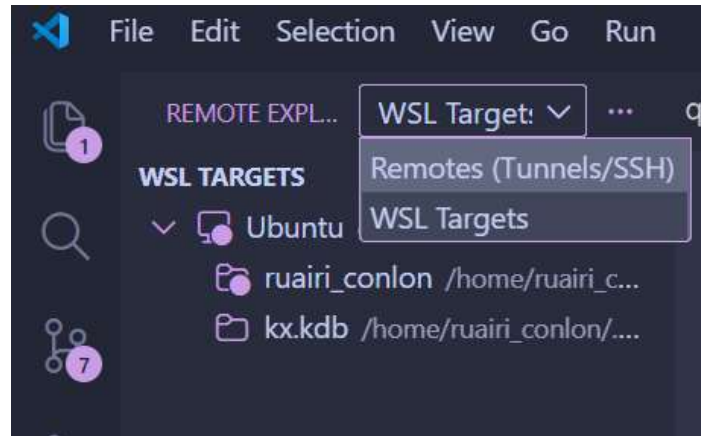


Image 4: Selecting WSL Targets when in the Remote Explorer tab

## 4 Set Up KDB-X

### KDB

Open Extensions (left sidebar) and search for kdb. When extension shows up, select it and click install.

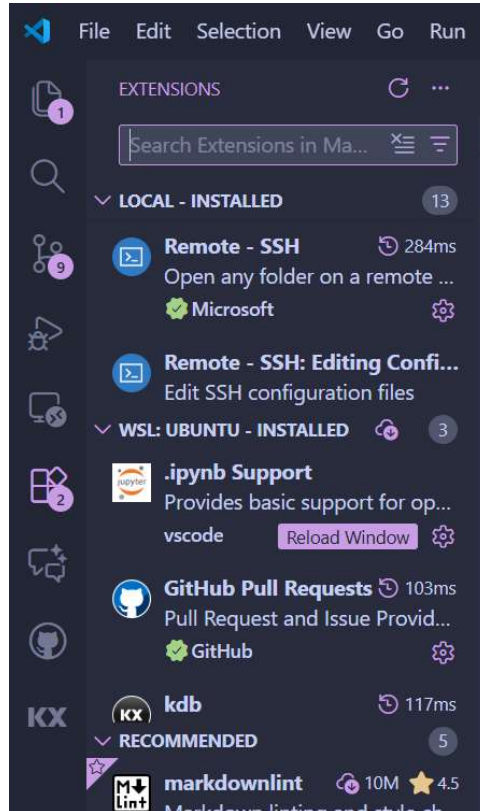


Image 5: Showing what the extensions tab looks like when opened



Image 6: kdb extension that should be installed

After installation, a welcome page will prompt you to install KDB-X.  
Click Install and Continue.

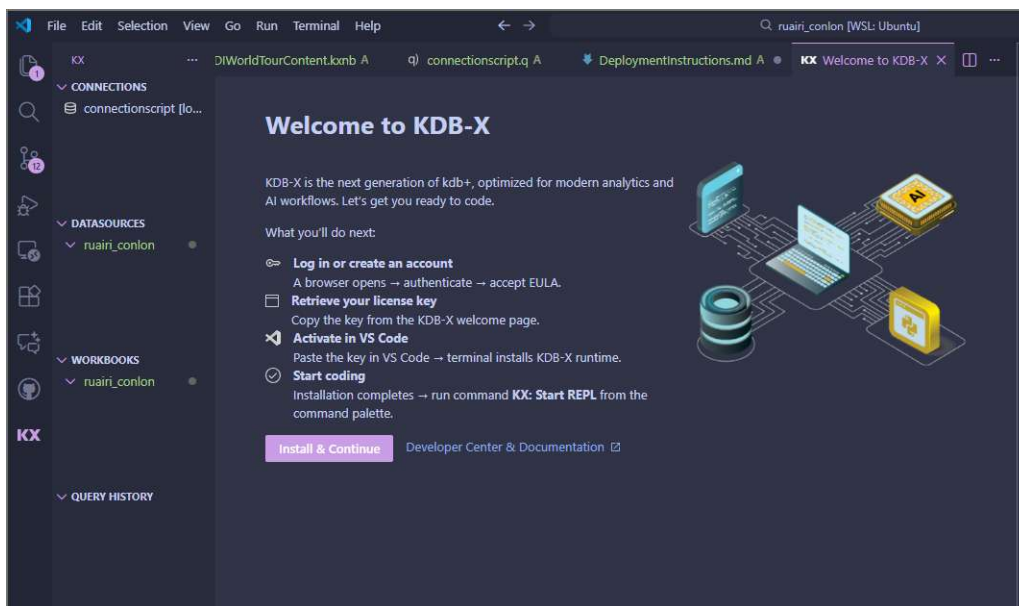


Image 7: KDB-X welcome page, click 'install & continue' button

A browser window will open prompting you to create a KX account if you don't have one.

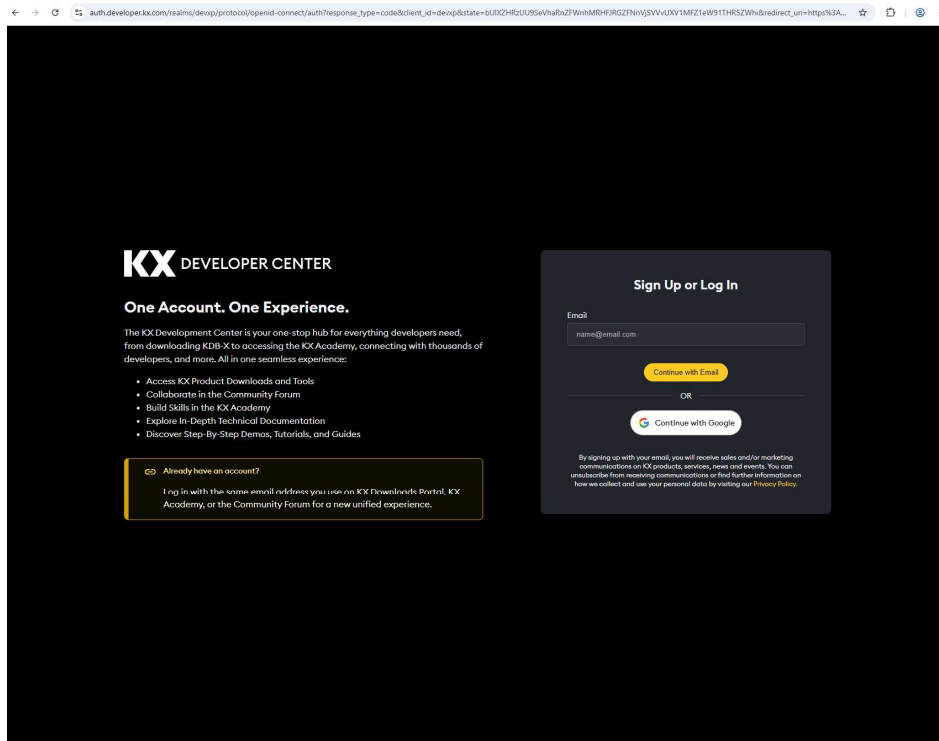


Image 8: Create KX account

Once signed in, you'll be directed to the KDB-X install page.

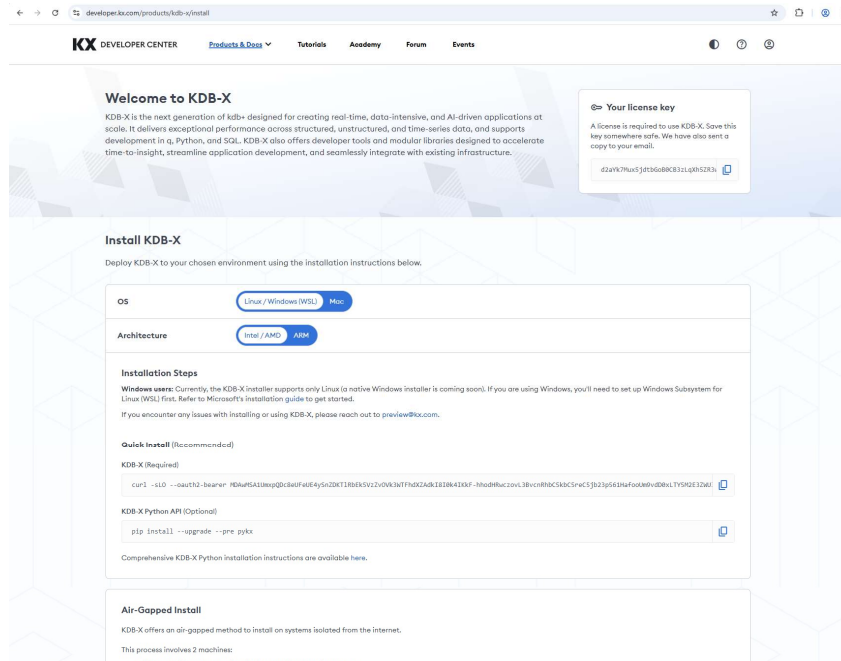


Image 9: KDB-X website, license key in the top right corner that to be copied

Copy the license key from the top right and return to VS Code, where you'll be prompted to enter it. After this, the KDB-X installation starts.

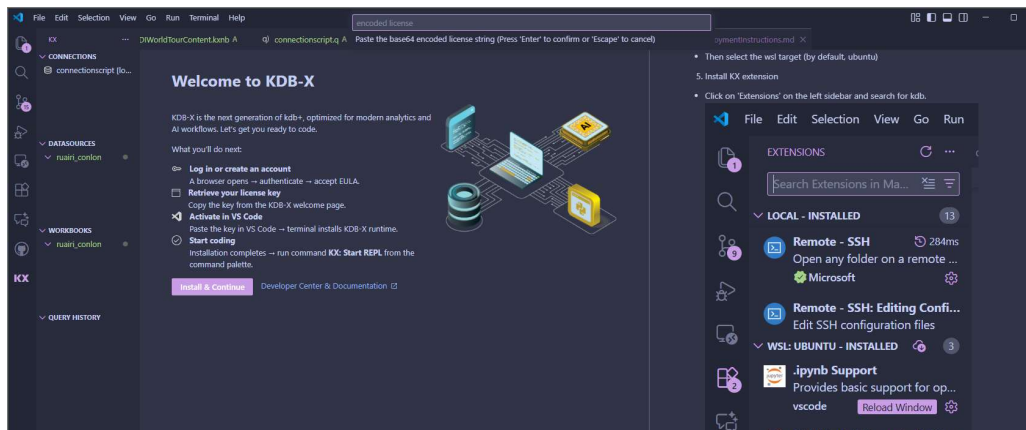


Image 10: Shows prompt near the top middle of the screen to paste license key

## 5 Unzip your files

Unzip your provided files into a your wsl home/user/ folder, which can be done using the usual windows method.

In VS Code, go to Explorer, click Open Folder, and navigate to the location of your files.

## 6 Start the Local q Process

Run the mockDataGen script from your terminal:

```
q /home/<username>/<path-to-repo>/Dev-Training/DIWorldTourContent/mockDataGen.q
```

Note 1: Replace paths with your actual local directory.

Note 2: If the terminal doesn't recognise the q command, simply replace q with ~/.kx/bin/q .

Note 3: To interact with the KX notebooks, the terminal in which mockDataGen.q is run needs to remain active. The script essentially provides a connection to a port that allows the code snippets to be run within the notebooks. This script should be run from the directory that it is in, ie. the Dev-Training/DIWorldTourContent/ directory. This will run on port 5067 by default.

Note 4: This script produces the mock database that will provide the notebooks with data to for examples throughout. This will appear as a directory called "hdb" containing three date partitions, each with a depth, quote and trade tables.

If for any reason you disconnect or want to start again after a break and the terminal goes down, then to reconnect do the following:

1. Reconnect to wsl
2. Open the content in the file explorer
3. Run "q hdb/ -p 5607" or just rerun the mockGenData.q script
4. Reconnect to the server in the KX tab

## 7 Connect KX Notebook to Local q

1. Open the KX extension in VS Code (left sidebar)
2. Click Add Connection

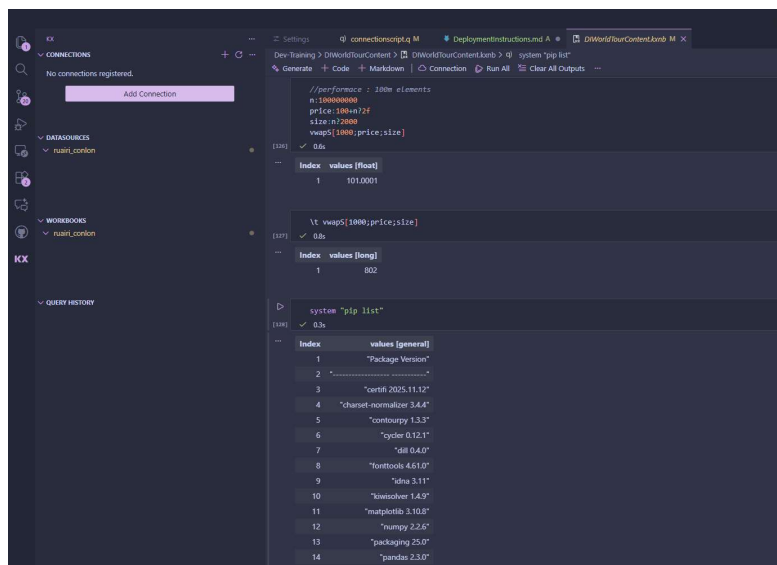


Image 11: Shows what the KX extension tab looks like and the 'add connection' button

3. Fill in the connection details:  
 Server Name: your choice (e.g., KXnotebook)  
 Connection Address: localhost  
 Port Number: 5067  
 Username / Password: leave blank (local q process)  
 Click **Create Connection**.

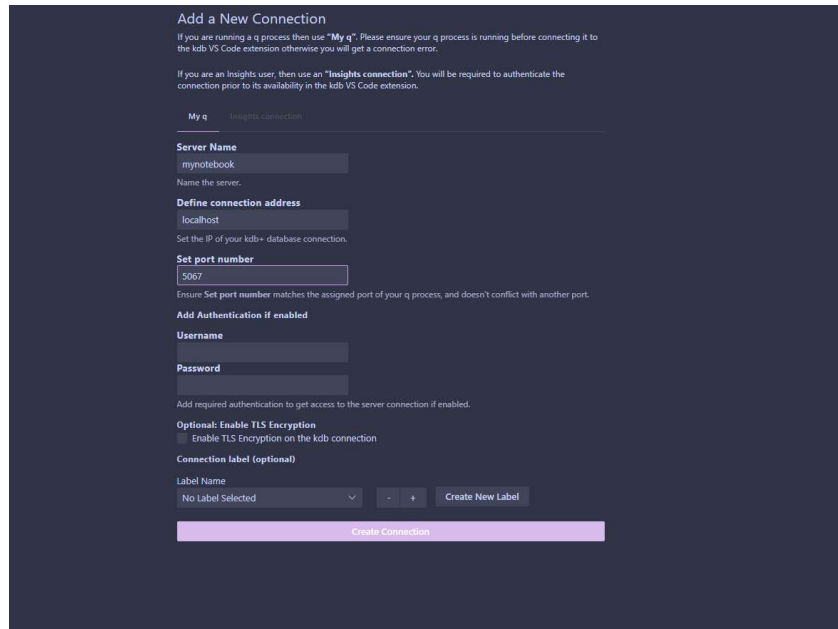


Image 12: Shows the new connection information filled in

4. Right-click your connection and select Connect Server.

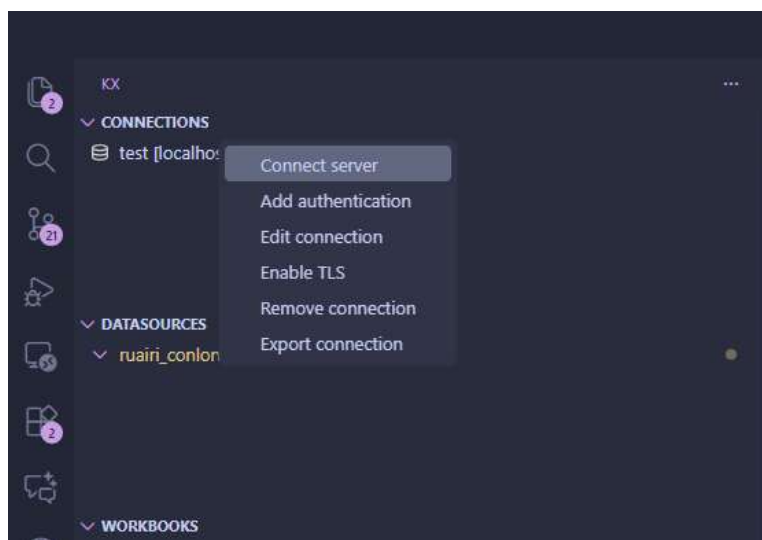


Image 13: Shows what will appear when right-clicking the new connection

You can now run code in the notebook against your local q process.

**[ CHALLENGE  
ACCEPTED.]**