MongoDB WSL - Setup Instructions:

Windows 10 (and above) and WSL

Preface

WSL is great for learning about Linux, and it is a useful tool to run lots of tools needed for development in a Windows environment, however it can be daunting for beginners to configure and is aimed at a more seasoned dev, while a Mac is definitely more user friendly for beginners. We will do our best to simplify some instructions, but errors do happen. If and when these errors do happen, note which step you are at so that your TA or Mentor can help you.

Step 1: Check Your Windows and WSL Version

Check your Windows version and WSL version. You should be working with Windows 10 (or 11) Pro or Home editions. If you are running anything older than Windows 10, it is time to upgrade. Microsoft provides Windows 10 for free, you won't have all the desktop customization options with the free version but all the tools which you will need for the course will work.

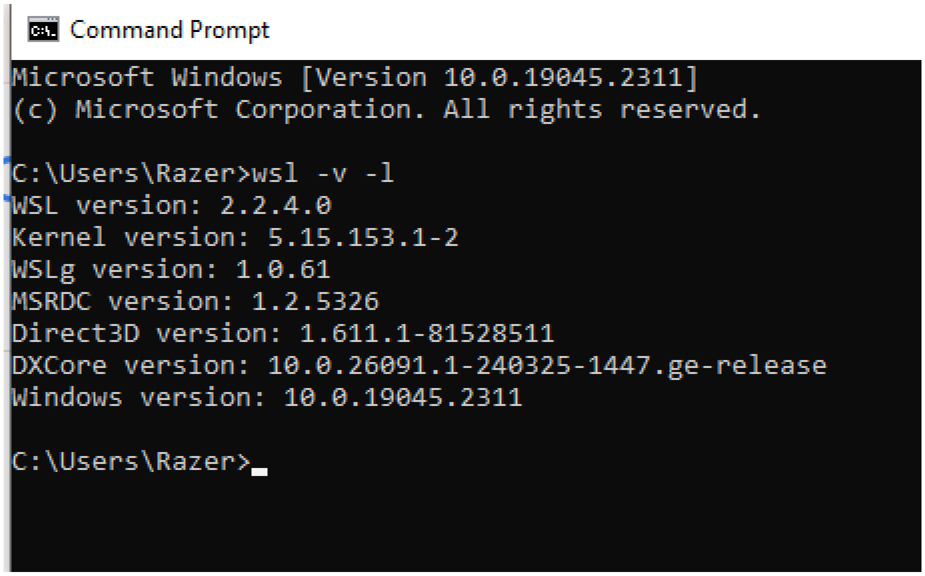
If you do not have WSL installed, written instructions can be found here, courtesy of Microsoft.

If WSL is already installed:

- Open up the Command Prompt
- Type in the following command:

```
wsl -v -l
```

You should get a similar output



```
Microsoft Windows [Version 10.0.19045.2311]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Razer>wsl -v -l
WSL version: 2.2.4.0
Kernel version: 5.15.153.1-2
WSLg version: 1.0.61
MSRDC version: 1.2.5326
Direct3D version: 1.611.1-81528511
DXCore version: 10.0.26091.1-240325-1447.ge-release
Windows version: 10.0.19045.2311

C:\Users\Razer>
```

Notice the WSL version. In this case it is 2.2.4.0. Anything over 2.0 will work well for us, if you have a lower version follow the following instructions to update from WSL1 to WSL2.

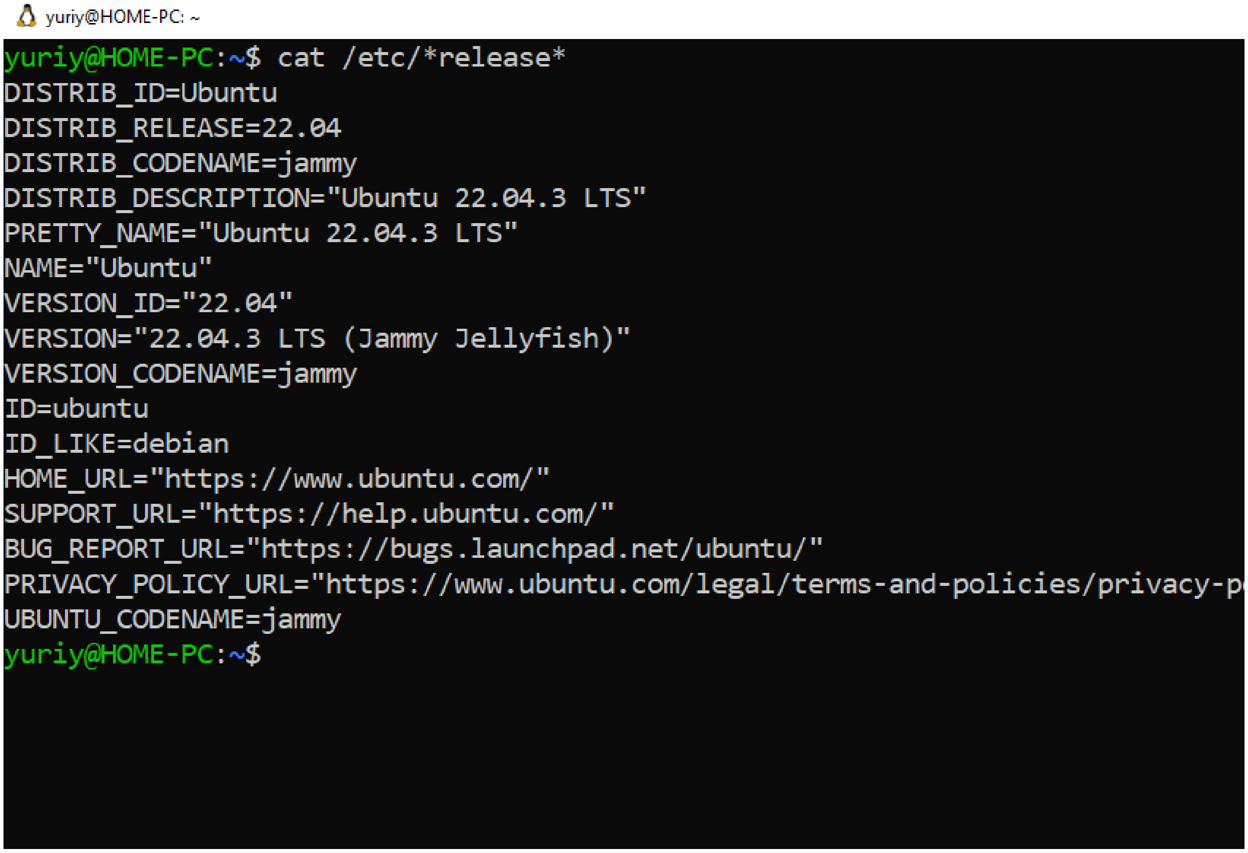
Step 2: Check your WSL distro (what Linux distro and version is installed)

While any version of WSL above 2.0 will do, the Ubuntu version is a bit more important here. Some of the tools which we will use are still working on the support for the LATEST Ubuntu release 24.04 (Noble Nombat). While older versions might not support the newer features. The Ubuntu version we are targeting is 22.04 LTS (Jammy Jellyfish)

- Open WSL (NOT the Command Prompt)
- Type in the following command:

```
cat /etc/*release*
```

You should get a similar output:



```
yuriy@HOME-PC:~$ cat /etc/*release*
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=22.04
DISTRIB_CODENAME=jammy
DISTRIB_DESCRIPTION="Ubuntu 22.04.3 LTS"
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.3 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-p
UBUNTU_CODENAME=jammy
yuriy@HOME-PC:~$
```

Notice a couple of keys here:

DISTRIB_RELEASE=22.04
NAME=”Ubuntu”
VERSION=”22.04.3 LTS (Jammy Jellyfish)”

If your versions match these ones here (22.04 is the key here), then you’re all set to go to Step 3. Otherwise let’s get your WSL2 Ubuntu distro version to be “just right” for all the tools which we use. This WON’T guarantee that you will have no errors going along, but at least they should be few and much more easily solvable.

Step 3: Set a Correct Ubuntu Distro Version

(Optional: If needed)

If your distro is NOT Ubuntu, or does not match “22.04.3 LTS (Jammy Jellyfish)”. Let’s make sure we install a correct version of the Ubuntu distro. There is also a custom video which you can follow along linked here.

- Open up your Windows Command prompt in Administrator mode.

- List all the available Linux distros with the following command:

```
wsl –list –online
```

(note: two dashes (--) before *list* and *online*)

- Select the correct Ubuntu distro to install. In our case its Ubuntu-22.04
- Run the following command:

```
wsl.exe –install -d Ubuntu-22.04
```

This should take some time to download the required files for Ubuntu-22.04. When you are prompted to choose a username, choose one. When you are prompted to choose a password. Choose and confirm a password.

Lastly we must ensure that the DEFAULT version is now Ubuntu-22.04. After the installation finishes, you may be taken into WSL shell by default. Exit and open up Windows terminal again.

- Run the following command:

```
wsl –set-default-version Ubuntu-22.04
```

You should see an asterisk (*) next to version 22.04. If that is correct, let’s now install MongoDB server on WSL.

Step 4: Install MongoDB in WSL2

Assuming we are running WSL2 and Ubuntu 22.04, we are ready to install MongoDB server. Open up your WSL shell and let's begin. For reference you can also check out the custom video here.

- First navigate to the root folder in the WSL2 shell. All commands will now be entered in WSL unless otherwise noted.

```
cd /
```

- Install GNUPG (GNU Privacy Guard)

```
sudo apt-get install gnupg
```

This should take a few seconds. If you have a prompt to confirm with a YES (y) or NO (n). Press Y. You may also be asked for your WSL2 password.

Next we must import something called GPGkeys. Without going too much into the details, it’s a security feature, and the devs of MongoDB are happy to oblige.

- Paste the following command into the WSL shell

```
curl -fsSL https://pgp.mongodb.com/server-6.0.asc | \
  sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg \
  --dearmor
```

It may look like nothing happened. But believe me something did. Unless you get an error, we are ready to move on to the next command.

- We will then create a list file. Again don’t worry about understanding what it does just yet, but copy and paste the following command into the WSL shell

```
echo "deb [ arch=amd64,arm64
signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg ]
https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 multiverse" | sudo
tee /etc/apt/sources.list.d/mongodb-org-6.0.list
```

If you get a prompt to confirm anything, just type Y for yes. Assuming no errors respond back at you , let's run some package updates.

- Type the following command next

```
sudo apt-get update
```

This may take anywhere from a few seconds to a few minutes depending on your system. This command just runs necessary updates. After that process finishes we are ready to install MongoDB packages.

- The following command will attempt MongoDB installation

```
sudo apt-get install -y mongodb-org
```

You may get a prompt to reenter your password. You may also get a prompt for Y or N to confirm that additional space will be used in your system (around 500-600mb).

This may take up to a few minutes depending on your system. If installation fails here, make sure to note (ideally screenshot) the error for your TAs.

If the installation was successful lets then set some settings:

- The following command will make a data directory if it did not create one already

```
sudo mkdir -p data/db
```

- Next command will set some permissions

```
sudo chown -R `id -un` data/db
```

Finally we are ready to test run MongoDB. If any errors have happened within these steps, please note the error and at which step it has failed.

- Test run MongoDB with the following command

```
mongod
```

You should see a BUNCH of output. As long as it is not throwing errors. You are fine. We can go to the next step and set up MongoDB Compass.