Installation Instructions: Docker, WSL, and Visual Studio Code

Milestone 1: Development Environment Setup

Introduction

This document provides detailed instructions for setting up Docker, WSL (Windows Subsystem for Linux), and Visual Studio Code on a Windows machine. These tools are essential for AI and data science projects developed in containers.

WSL Installation

1. Open PowerShell as Administrator

- Right-click on PowerShell and select "Run as administrator".

2. Install WSL

- Execute the following command

```
wsl --install
```

- Restart your machine after installation.

3. Verify WSL Installation

- Open PowerShell or Command Prompt as administrator.
- Check the WSL version using

```
wsl --version
```

Output:

```
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: 10.0.22631.3810
```

4. Check WSL Distributions

- To check the WSL mode and distributions, run

```
wsl.exe -l -v
```

Output:

```
NAME                   STATE           VERSION
* Ubuntu               Running         2
docker-desktop         Running         2
```

Installing Docker Desktop

1. Download Docker Desktop

- Visit the Docker Desktop download page.
- Download and run the Docker Desktop installer.

2. Configure Docker Desktop

- Start Docker Desktop from the Windows Start menu.
- Navigate to **Settings > Resources > WSL Integration**.
- Check "Enable integration with default WSL distro".
- Select **Apply & Restart**.

3. Verify Docker Installation

- Open any terminal or PowerShell.
- Check Docker version using:

```
docker --version
```

Output:

```
Docker version 26.1.4, build 5650f9b
```

Installing Visual Studio Code

1. Download Visual Studio Code

- Visit the Visual Studio Code download page.
- Download and run the Visual Studio Code installer for Windows.

2. Install Visual Studio Code

- Follow the installation prompts.
- During installation, ensure to select the following additional tasks:
    - 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)

Configuring Visual Studio Code for WSL and Docker

1. Install VS Code Extensions

- Open Visual Studio Code.
- Go to the Extensions view (`Ctrl+Shift+X`).
- Install the following extensions:
    - Remote - WSL
    - Remote - Containers
    - Docker

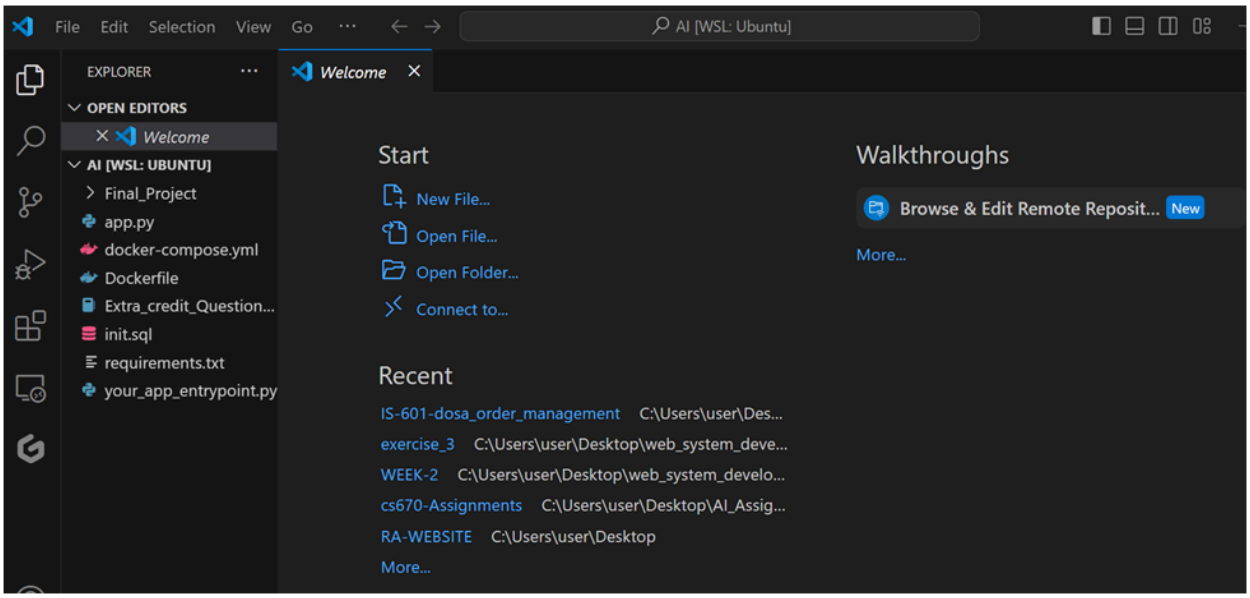
2. Open Project in VS Code from WSL

- Open WSL terminal (e.g., Ubuntu).
- Navigate to the project directory using WSL paths.

- Launch Visual Studio Code with the current directory using:

```
code .
```

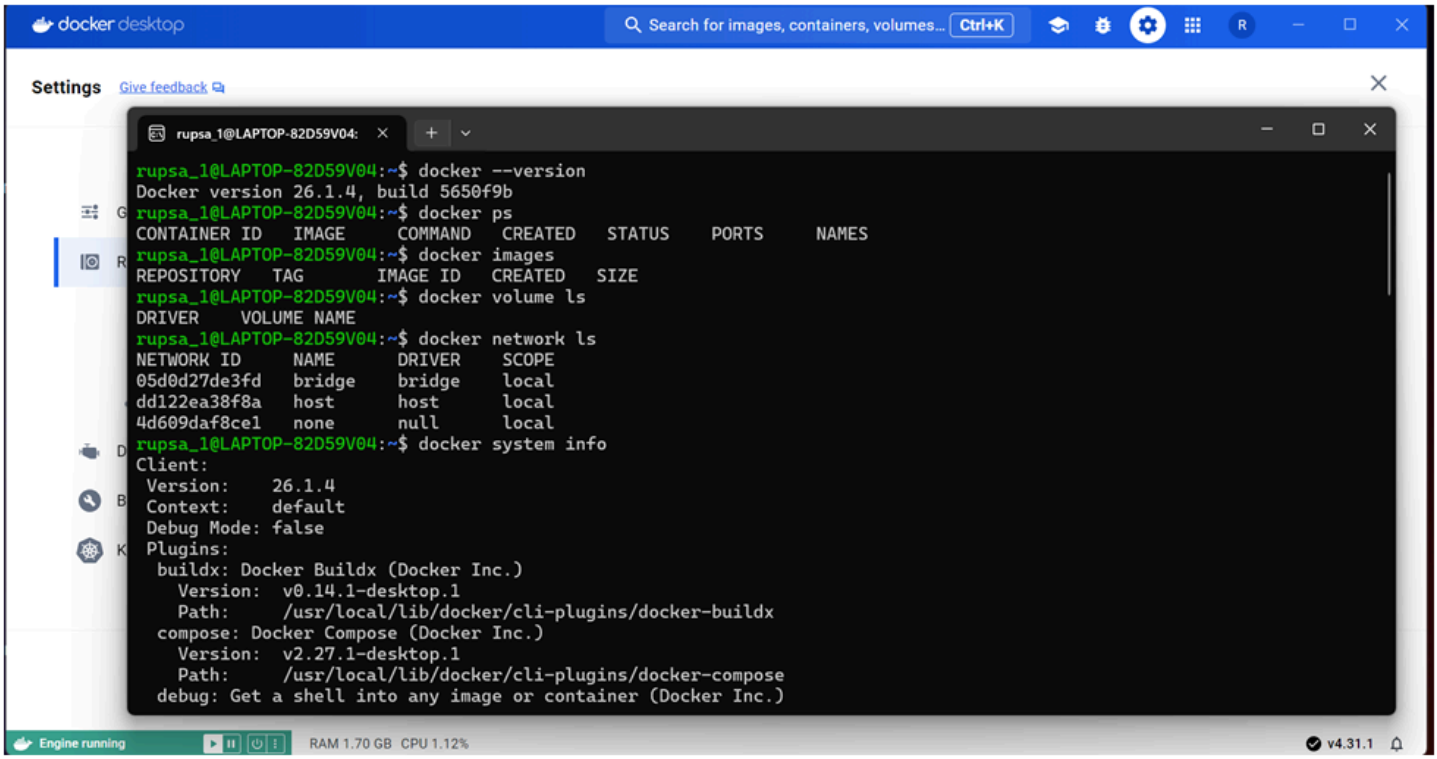
```
rupsa_1@LAPTOP-82D59V04:~$ cd /mnt/c/Users/user/Desktop/AI
rupsa_1@LAPTOP-82D59V04:/mnt/c/Users/user/Desktop/AI$ code .
Installing VS Code Server for Linux x64 (5437499feb04f7a586f677b155b039bc2b3669eb)
Downloading: 100%
Unpacking: 100%
Unpacked 1689 files and folders to /home/rupsa_1/.vscode-server/bin/5437499feb04f7a586f677b155b039bc2b3669eb.
Looking for compatibility check script at /home/rupsa_1/.vscode-server/bin/5437499feb04f7a586f677b155b039bc2b3669eb/bin/
helpers/check-requirements.sh
Running compatibility check script
Compatibility check successful (0)
rupsa_1@LAPTOP-82D59V04:/mnt/c/Users/user/Desktop/AI$
```



A screenshot of my docker Container terminal prompt:

- Verify Docker installation by running commands like:

- Ensure Docker Desktop is running on Windows machine for Docker commands to work.



Conclusion

By following these steps, I have a robust development environment set up with Docker, WSL, and Visual Studio Code, enabling me to effectively develop and debug AI and data science projects in containers.