# CS-634-Data-Mining

## Milestone 1: (Week 1, 10 points)

Set up a development environment in your laptop. Learn the basics of docker by watching the following video:

[![Watch the video](https://img.youtube.com/vi/pTFZFxd4hOI/0.jpg)](https://youtu.be/pTFZFxd4hOI)

If you are on Windows you will need to follow [these instructions](https://docs.docker.com/desktop/windows/wsl/) and install [Docker Desktop](https://www.docker.com/products/docker-desktop/) and [WSL2](https://learn.microsoft.com/en-us/windows/wsl/install).

Independent of your OS, you may want to use VS Code IDE if you have no IDE experience before.

Submit the github repository URL with a branch titled ‘milestone-1’ with the README.md file containing the installation instructions you followed and a screenshot of your docker container terminal prompt. Add as collaborator the TA.

---

# Install Docker Desktop for Windows with WSL2 backend

## Table of Contents

1. [References](#references)

2. [Prerequisites](#prerequisites)

3. [Create example "Getting Started" container using Visual Studio Code (VS Code)](#create-example-getting-started-container-using-visual-studio-code-vs-code)

4. [Create example "Hello World" docker container](#create-example-hello-world-docker-container)

## References

As provided in the above previous section detailing the Milestone 1 deliverables, the following references were used for Docker Desktop and WSL2 installation:

- [Docker Tutorial for Beginners (video)](https://youtu.be/pTFZFxd4hOI)

- [Docker Desktop Download](https://www.docker.com/products/docker-desktop/)

- [WSL Installation for Windows](https://learn.microsoft.com/en-us/windows/wsl/install)

- [Docker Desktop WSL2 backend on Windows](https://docs.docker.com/desktop/windows/wsl/)

I also used the following:

- [WSL2 Linux kernel Update](https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package)

- [Docker remote containers on WSL2](https://learn.microsoft.com/en-us/windows/wsl/tutorials/wsl-containers)

<br>

## Prerequisites

### Verify Windows version

Ensure that you have either:

- Windows 10 version 21H2 or higher

- Windows 11 version 21H2 or higher

This can be done by following the below steps:

1. Hold the `WINDOWS\_KEY + R` on your keyboard to open the `Run` window

2. Type `winver`

3. Press the `Enter` key

4. Check that your `Version` is indicated as previously stated such as in the below screenshot.

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/4d403101-c546-4be6-b26f-ac0953f11436)

### Install WSL and set to WSL2

If you don't already have WSL installed, then enter the following command into Windows Command Prompt (accessed by typing `cmd` in the `Run` window) or PowerShell (accessed by typing `powershell` in the `Run` window):

```

wsl --install

```

Now, ensure that you have WSL version 1.1.3.0 or above. To do this, enter the following command such as in the below screenshot:

```

wsl --version

```

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/1c0c4524-0006-4df2-8daf-36ea33875b49)

If WSL2 is not already set as the default version, configure the default with the following command:

```

wsl --set-default-version 2

```

### WSL2 Linux kernel update

Ensure that you have the latest WSL2 Linux kernel by navigating to the following link and downloading `wsl\_update\_x64.msi`: [Step 4 - Download the Linux kernel update package](https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package)

<br>

## Download, Installation, and Configuration of Docker Desktop

1. Navigate to [this link](https://www.docker.com/products/docker-desktop/) to install Docker Desktop

2. Within the installation launcher, ensure that `Enable WSL 2 Windows Features` is selected

3. Open up Windows Powershell and enter the following command to ensure your docker engine is running (you should have a window similar to below image):

```

docker version

```

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/0acd3a2f-0ce3-4203-ad0d-80600d78b856)

4. Launch the Docker Desktop application

5. Go to `Settings`

6. Under `General`, ensure that `Use the WSL 2 based engine` is selected.

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/acc52529-315b-47bd-bcf5-8ada34296d30)

8. Under `Resources`, go to `WSL integration` and ensure that `Enable integration with my default WSL distro` is selected.

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/dd9709a2-939a-4f56-8825-45036d0cb0f3)

<br>

## Create example "Getting Started" container using Visual Studio Code (VS Code)

I had created a "Getting Started" docker container using VS Code by following the steps as indicated in [this tutorial](https://learn.microsoft.com/en-us/visualstudio/docker/tutorials/docker-tutorial).

1. In VS Code, make sure that you've installed `Docker` under `Extensions`

2. Open a `New Terminal` under `Terminal` and enter the following command:

```

docker run -d -p 80:80 docker/getting-started

```

3. In Docker Desktop, click on `Containers` to see the "getting started" docker container you created in VS Code

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/3f48b780-2310-42d0-a610-0a0dfe2fef81)

4. Open a browser and go to `http://localhost/tutorial/`

<br>

## Create example "Hello World" docker container

I had created a "Hello World" docker container by following the steps as indicated in the [video](https://youtu.be/pTFZFxd4hOI) as provided in the beginning under the Milestone 1 deliverables.

1. Open Windows PowerShell and create a directory to place your "Hello World" docker container using the `mkdir` command

2. Go to the folder by using the `cd` command and then use the `code .` command to open the Visual Studio Code editor

3. In VS Code, create a new file and name it `app.js` and enter the follow line of code:

```js

console.log("Hello World!");

```

4. Create a new file called `Dockerfile` and enter the following lines of code:

```

FROM node:alpine

COPY . /app

WORKDIR /app

CMD node app.js

```

5. Back in PowerShell, enter the following command within the directory containing your docker file:

```

docker build -t hello-docker

```

6. Use the `docker image ls` command to check

7. Finally, run the following command:

```

docker run hello-docker

```

![image](https://github.com/GHcpv24/CS-634-Data-Mining/assets/106451112/9b21a91d-8e3e-4fde-9ad8-55fa361a98a5)