

Integrated Vision System App Installation via Docker on Windows System

Prepared by: Yap Jun Kang

Contents

1. Docker Installation	3
1.1. WSL Installation	3
1.2. Docker Desktop Installation	3
2. Enabling GPU support for the App.....	4
2.1. Windows Insider Program Registration	4
2.2. Setting Up GPU Support for Docker	6
3. Application Installation	7
3.1. Docker Pull	7

1. Docker Installation

1.1. WSL Installation

- 1) To run docker on Windows, WSL is needed. Follow this [link](#) for more detailed installation guide.
- 2) Run CMD as administrator and run the following command to install WSL.

```
wsl --install
```

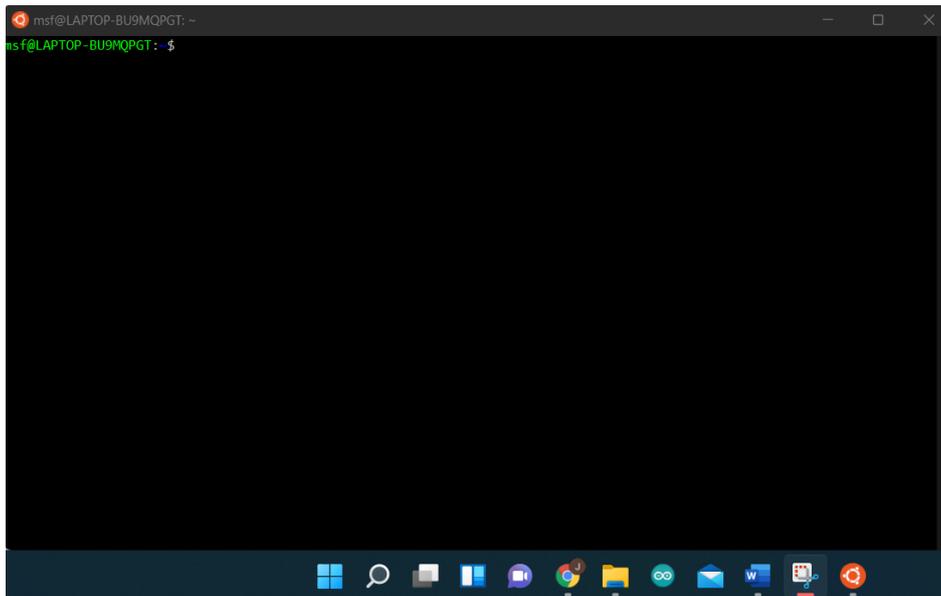
- 3) Once installed, after restarting the PC, you will be prompted to set a username and password for the Ubuntu virtual machine. Enter a username and password to setup the Ubuntu virtual machine.
- 4) If WSL have been installed before run the following command to ensure it is up to date.

```
wsl --update
```

- 5) Install the Ubuntu 20.04 using the following command. This is to install a WSL with distro.

```
wsl --install -d Ubuntu-20.04
```

- 6) Check the installation of Ubuntu by searching Ubuntu and running the application. If successful, a window like the following will show and prompt a username and password setup.



1.2. Docker Desktop Installation

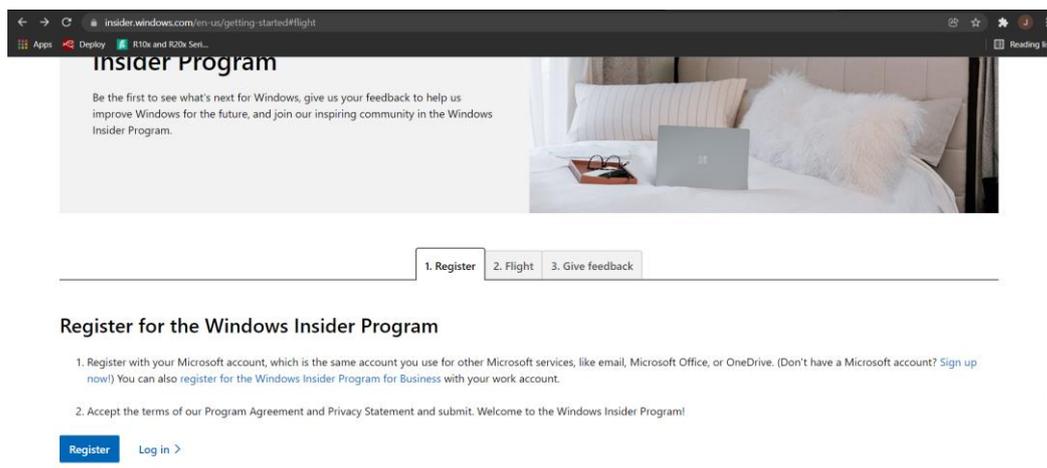
- 1) To install docker on Windows follow this [link](#).
- 2) Download the file and execute it once the download is done.
- 3) Once the installation is done, go to Start and search for the application to verify that installation is successful.



2. Enabling GPU support for the App

2.1. Windows Insider Program Registration

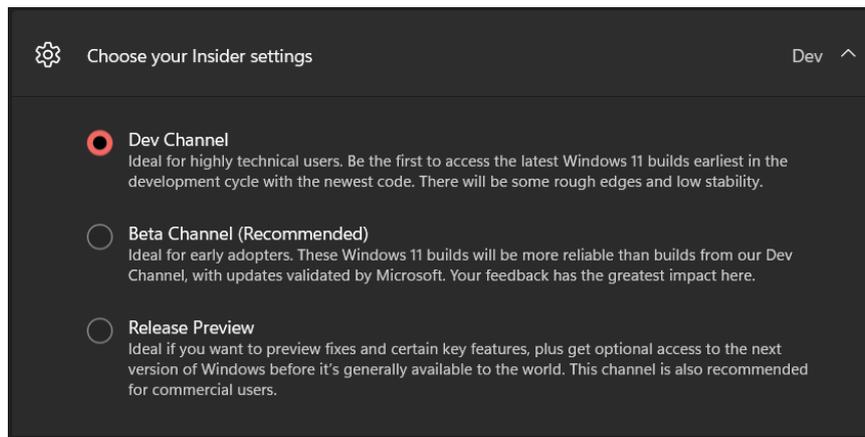
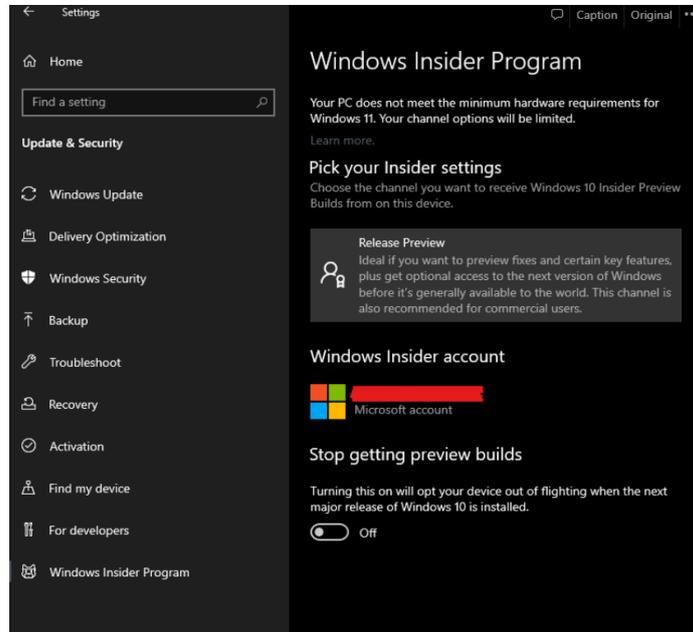
- 1) To enable GPU support for the app, register for the Windows Insider Program.
- 2) Open settings and search for **Windows Insider Program**.
- 3) If your system is running on Windows 11, the registration can be done directly through the settings.
- 4) Log in into a Microsoft account and click get started to start registering. Follow the prompted steps.
- 5) However, if using Windows 10, navigate to this [link](#) to start registering.



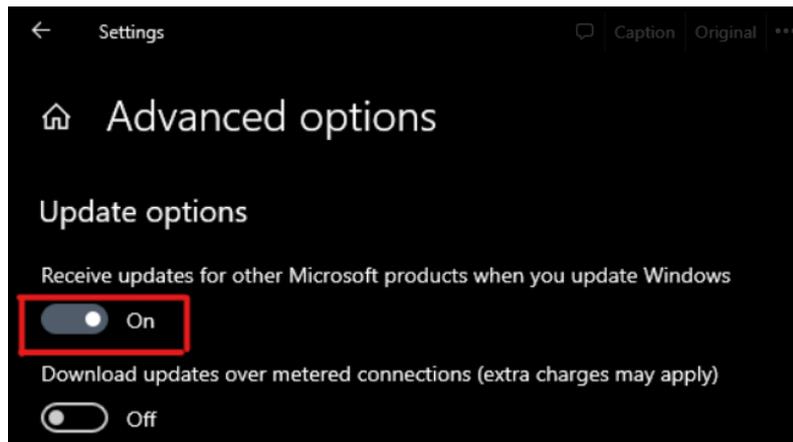
- 6) Start by logging in and then register. Once register, follow the prompt and start flight.
- 7) Once successfully registered, navigate to settings and in settings navigate to **Advanced Windows Update option**
- 8) Select Receive update for other Microsoft products.
- 9) Next ensure your PC is up to date before proceeding.

WARNING: The Windows Insider Program is very likely to force your PC to update to Windows 11

- 10) Navigate back to the **Windows Insider Program** to check if it is registered successfully.
- 11) If Windows Insider Program requires fixing, follow the prompts in the settings to fix the issue.
- 12) Choose whichever that is needed, but it is recommended to choose the **Release Preview** for the Windows Insider Program as the rest will force the PC to upgrade to an extremely unstable Windows version.



- 13) Next, go to settings and search for Advanced Windows Update Options and enable 'Receive updates for other Microsoft products'

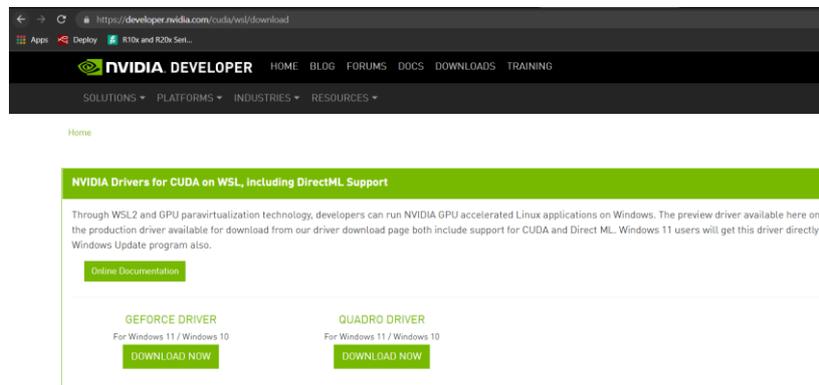


14) Finally, run CMD as administrator and run the following code to update WSL once more.

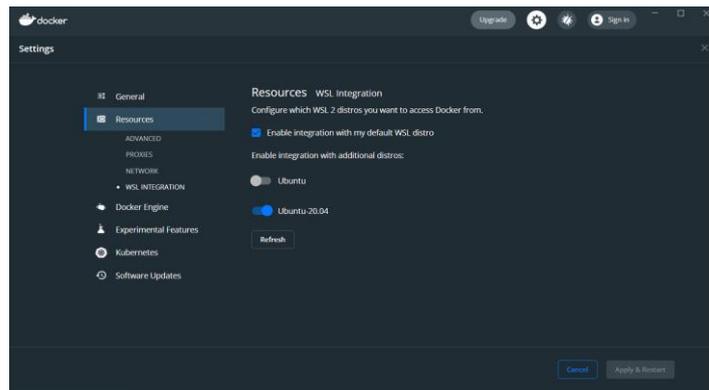
```
wsl -update
```

2.2. Setting Up GPU Support for Docker

- 1) After setting up Windows Insider Program, follow this [link](#) to install the cuda drivers, be sure to install the correct version for your GPU.
- 2) Download the file and then execute it. If the driver shows that the system is incompatible, be sure to update your Windows.



- 3) Finally, launch Docker Desktop and navigate to settings to enable GPU support for the application.
- 4) Select the desired Linux OS and then launch the distro terminal by searching the OS name in search bar.
- 5) Use the command `<nvidia-smi>` to check if GPU support is enabled.



```
msf@LAPTOP-BU9MQPGT:~$ nvidia-smi
Mon Jan 24 22:18:47 2022

+-----+
| NVIDIA-SMI 510.00      Driver Version: 510.06      CUDA Version: 11.6     |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A    Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|  Memory-Usage  GPU-Util  Compute M. |
|                                           MIG M.     |
+-----+-----+
| 0   NVIDIA GeForce ...  On          | 00000000:01:00.0 Off      |    N/A             N/A     |
| N/A   40C    P5     9W /  N/A     | 153MiB / 6144MiB |    N/A             Default |
|                                           N/A             N/A     |
+-----+-----+

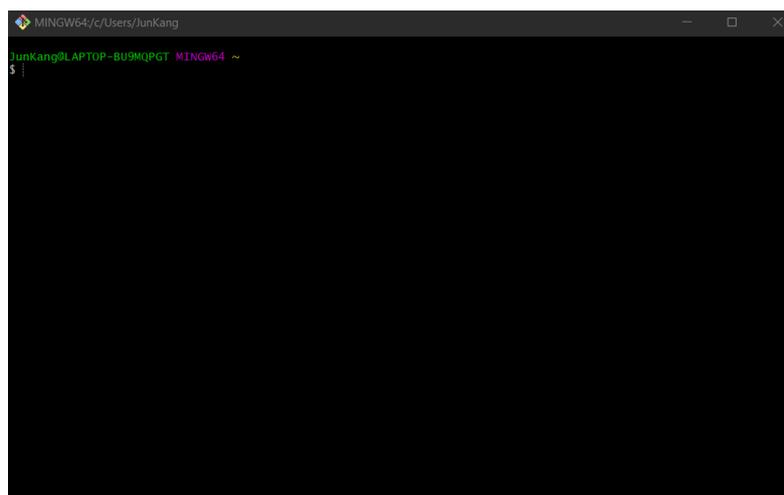
Processes:
+-----+-----+
| GPU   GI      CI           PID  Type   Process name          GPU Memory |
| ID   ID      ID             |              |           Usage      |
+-----+-----+
| No running processes found |
+-----+-----+

msf@LAPTOP-BU9MQPGT:~$
```

3. Application Installation

3.1. Docker Pull

- 1) To start the installation, first install Git on the PC using this [link](#).
- 2) Once installed, launch Git bash as administrator.

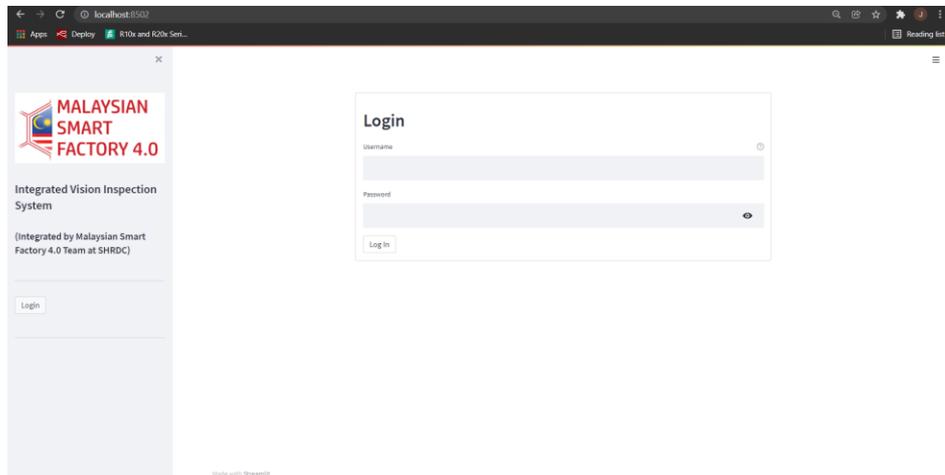


- 3) Run the following commands in Git Bash to setup the repository and install the application.

```
git clone https://github.com/msf4-0/Integrated-Vision-Inspection-System.git
cd Integrated-Vision-Inspection-System
```

docker-compose up -d (use **docker compose -f docker-compose.cpu.yml up -d** instead if gpu support is not enabled)

- 4) Next, launch the application by opening a browser and navigate to the link <localhost:5802>
- 5) The app will be launch if the installation is successful.
- 6) The default username and password will be <admin>



- 7) To stop the app, use the command <docker-compose down>