



Setting up Flywheel using Docker and WSL

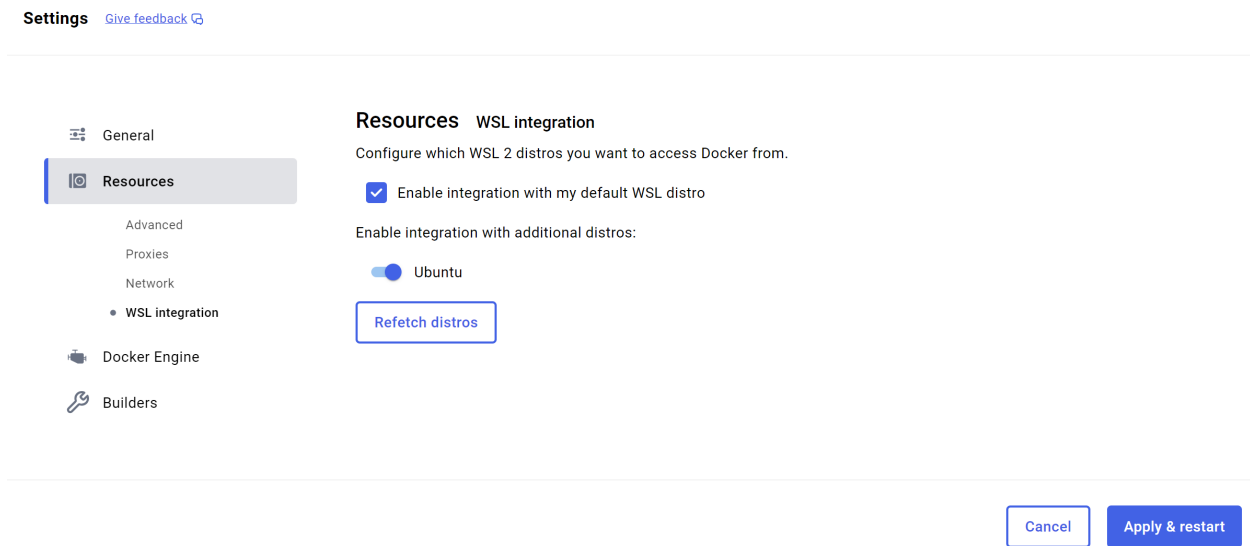
1. Download Docker Desktop for Windows:
<https://www.docker.com/products/docker-desktop/>
2. Open Docker Desktop and get your account set up. You can either link it to your Github, or your Google account.
3. Go to Settings > Resources > WSL Integration :

Settings [Give feedback](#)

The screenshot shows the Docker Desktop Settings interface. On the left is a sidebar with navigation options: General, Resources (selected), Advanced, Proxies, Network, WSL integration (indicated by a dot), Docker Engine, and Builders. The main panel is titled 'Resources WSL integration' and contains the text 'Configure which WSL 2 distros you want to access Docker from.' Below this is a checked checkbox labeled 'Enable integration with my default WSL distro'. A message states: 'You don't have any WSL 2 distros installed. Please convert a WSL 1 distro to WSL 2, or install a new distro and it will appear here.' There is a 'More info' link and a 'Refetch distros' button.

- If you don't see any WSL distros installed as shown in the screenshot, go to step 4. If you already have a WSL distro that Docker can access, skip to step 6.
4. Download Ubuntu from your Microsoft Store or the Ubuntu website
<https://ubuntu.com/download/desktop>
 5. Open Ubuntu and set up your UNIX profile (username and password)

6. Once set up correctly, you should now see this under Docker Settings > Resources > WSL integration. Toggle the Ubuntu distro below, and click **Apply and Restart**



7. Open PowerShell and run:

```
ws1 --set-default Ubuntu
```

This will ensure that Ubuntu is your **default** WSL distribution, then start wsl by simply running the command:

```
wsl
```

8. To have you set the folder structure to follow the UNITY team members' local folder structure, run the following commands from your WSL **command line**:

```
cd /home
mkdir unity
cd unity
mkdir fw-gears
```

You will be prompted for your password if you are running this command as **sudo** (needing administrative rights)*. Please input the password you just set up for your UNIX profile in Step 5.

*You might not need to use `sudo` if you already have the command line prompt open *as administrator*.

To clone a Flywheel gear, in this example, the **fw-SynthSeg-gear**, run the following:

```
cd fw-gears
sudo git clone https://github.com/UNITY-Physics/fw-SynthSeg-gear
```

You will now have your Flywheel gears in your directory `/home/unity`, the structure will resemble the below:

```
.
├── username
├── unity
│   └── Flywheel-docs
│       ├── README.md
│       ├── fw-gears
│       │   ├── fw-2pieR
│       │   │   ├── Dockerfile
│       │   │   ├── LICENSE
│       │   │   ├── README.md
│       │   │   ├── app
│       │   │   ├── __init__.py
│       │   │   ├── __pycache__
│       │   │   ├── __init__.cpython-36.pyc
│       │   │   ├── __init__.cpython-39.pyc
│       │   │   ├── ants_vbm.cpython-39.pyc
│       │   │   ├── beta_triplane_SRIImageReconViaANTs.c
│       │   │   ├── command_line.cpython-39.pyc
│       │   │   ├── gatherDemographics.cpython-39.pyc
│       │   │   └── generate_command.cpython-39.pyc
```

```

├── main.cpython-39.pyc
├── metadata.cpython-39.pyc
├── parser.cpython-36.pyc
├── parser.cpython-39.pyc
├── ants_vbm.py
├── calculateHeadCircumference.py
├── command_line.py
├── constants.py
├── context.py
├── gatherDemographics.py
├── generate_command.py
├── parser.py
├── templates
├── over2.nii.gz
├── over20Outline.nii.gz
├── under2.nii.gz
├── under20Outline.nii.gz
├── test
├── TEST_ants_vbm.py
├── __pycache__
├── TEST_ants_vbm.cpython-39.pyc
├── test_fsl.cpython-39.pyc
├── test_fsl.py
├── tmp.sh
├── docs
├── GitHubDocs.md
├── changelog.md
├── gearDev.md
├── release_notes.md
├── manifest.json
├── run.py
├── start.sh
├── utils
├── __init__.py
├── __pycache__
├── ROI.cpython-39.pyc

```

```

├── __init__.cpython-39.pyc
├── registration.cpython-39.pyc
├── dry_run.py
├── fly
├── __init__.py
├── set_performance_config.py
├── metadata.py
├── fw-ciso
│   ├── Dockerfile
│   ├── LICENSE
│   ├── README.md
│   ├── app
│   │   ├── __init__.py
│   │   ├── alternative.sh
│   │   ├── ciso-gear.sh
│   │   ├── command_line.py
│   │   ├── constants.py
│   │   ├── context.py
│   │   ├── findMatchedScans.py
│   │   ├── generate_command.py
│   │   └── parser.py
│   ├── docs
│   │   ├── GitHub Branches.md
│   │   ├── changelog.md
│   │   ├── gearDev.md
│   │   └── release_notes.md
│   ├── manifest.json
│   ├── pyproject.toml
│   ├── requirements-dev.txt
│   ├── requirements.txt
│   ├── run.py
│   └── utils
│       ├── __init__.py
│       ├── dry_run.py
│       ├── fly
│       └── __init__.py

```

```

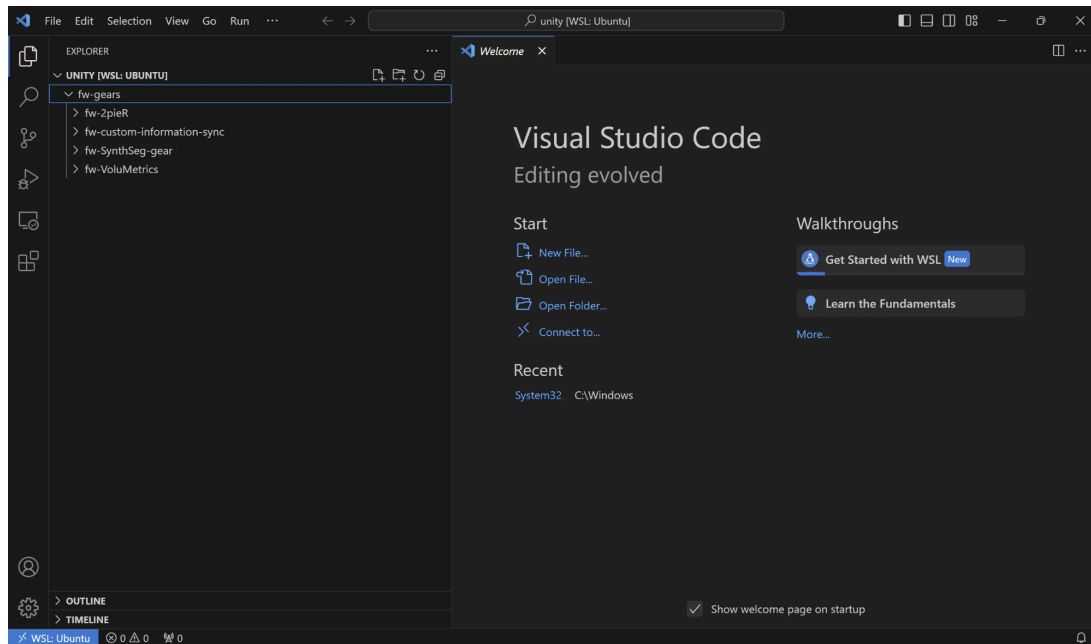
|   |   |   |   └─ set_performance_config.py
|   |   |   └─ metadata.py
|   |   └─ niftiHeader.py
|   └─ fw-phantomCuration
|       └─ Dockerfile
|       └─ LICENSE
|       └─ README.md
|       └─ app
|           └─ __init__.py
|           └─ context.py
|           └─ main.py
|       └─ docs
|           └─ Github commands for new repo.md
|           └─ gearDev.md
|           └─ release_notes.md
|       └─ manifest.json
|       └─ poetry.lock
|       └─ pyproject.toml
|       └─ run.py
└─ fw-sdk
    └─ README.md
    └─ pull-SynthSegResults.py
    └─ pull-mriqc.py
    └─ run-CISO.py
    └─ run-FreeSurfer.py
    └─ run-SynthSeg.py
└─ fw-upload
    └─ Getting setup on Flywheel (Mac).md
    └─ Getting setup on Flywheel (Mac).pdf
    └─ UNITY-data-naming-SOP.md
    └─ dicom_upload.sh

```

Now to start your dev work, you can launch Visual Studio code **from your fw-gears directory** (which should be your current directory) from your PowerShell:

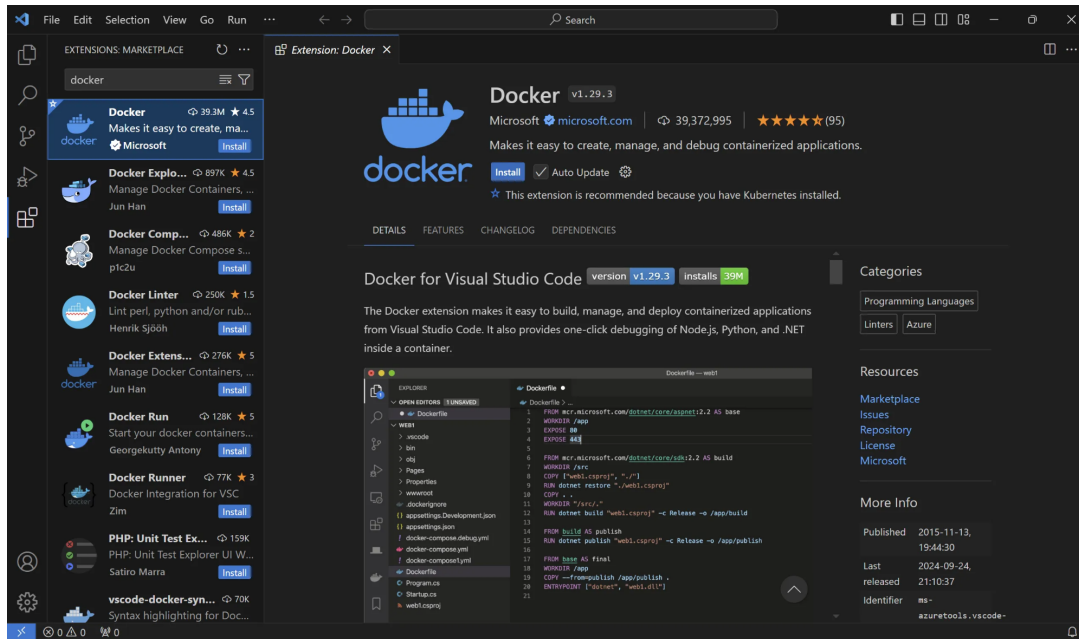
```
code .
```

You should see the VS code window below, using your fw-gears as your working directory:



Setting up Visual Studio Code:

1. Install the Docker extension in VS code



Do the same with the **Python** package, and **WSL** (if it's not already installed).

Set up Flywheel CLI:

From your VS code terminal, run the following:

```
curl https://storage.googleapis.com/flywheel-dist/fw-cli/stable,
```

Set up your Flywheel API key:

```
vim ~/.bash_profile
```

In the bash_profile file:

```
FW_CLI_API_KEY=YOURAPIKEY
```

Replace *YOURAPIKEY* with the key you generated on FW.

```
source ~/.bash_profile
fw-beta login --api-key=${FW_CLI_API_KEY}
```


You should now be logged in to your flywheel instance.

Create a python environment in your `/home/user/unity` directory, install the necessary requirements, and activate it:

```
python -m venv ./venv
source /venv/bin/activate
pip install -r requirements.txt
source ./venv/bin/activate
```