

# MACHINE LEARNING OPERATIONS



Presented by **Asst. Prof. Dr. Tuchsanai Ploysuwan**



# เครื่องมือที่ใช้

---

-  วัตถุประสงค์: เรียนรู้ MLOps บน Linux (Ubuntu) Environment
-  ความท้าทาย: นักศึกษาส่วนใหญ่ใช้ Windows เป็นหลัก
-  วิธีแก้ไข: ใช้ Docker Container เพื่อจำลองสภาพแวดล้อม Linux
-  Container Image: [tuchsanai/mlops\\_2568\\_2:latest](https://hub.docker.com/r/tuchsanai/mlops_2568_2)



# ทำไมต้อง Docker?



ความสะดวก: เรียกใช้ Linux บน Windows ได้ทันที



ความสม่ำเสมอ: สภาพแวดล้อมเดียวกันสำหรับทุกนักศึกษา



ลดปัญหา: ไม่ต้องติดตั้ง dependencies แยกบน Windows

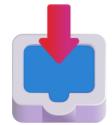


ความปลอดภัย: ไม่กระทบต่อระบบ Windows ของคุณ



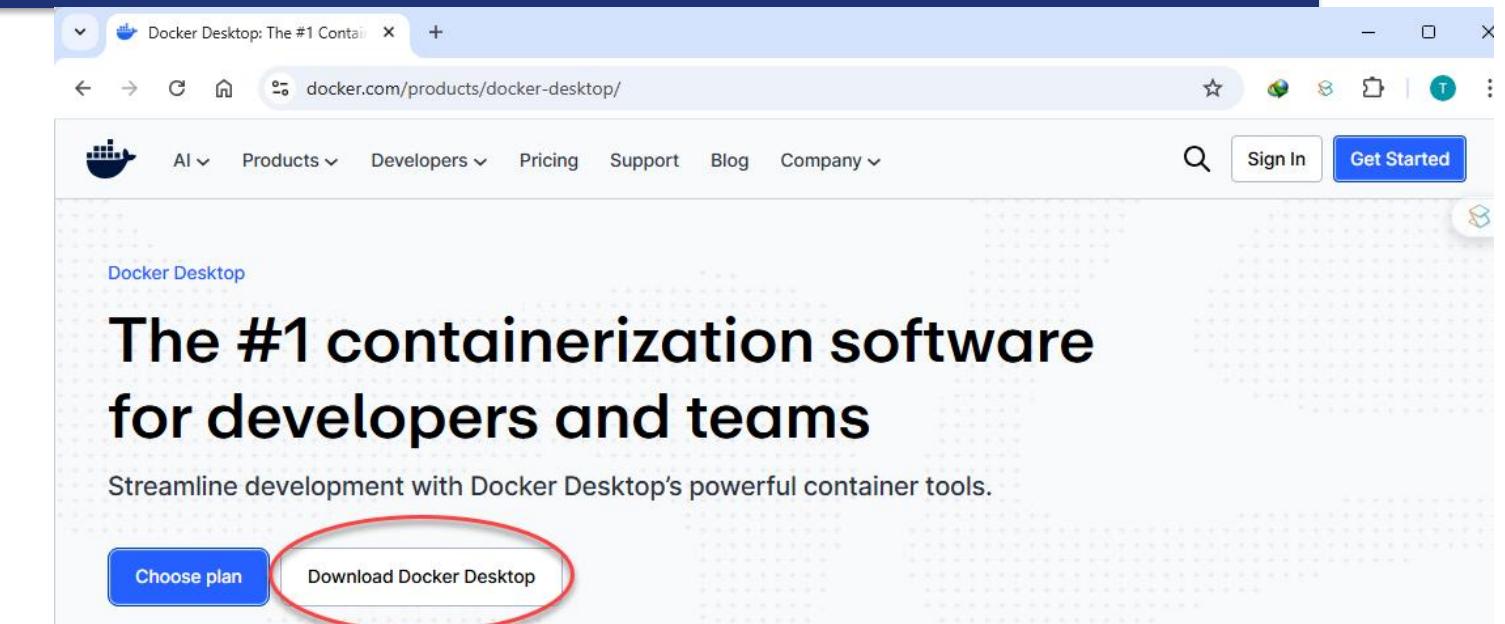
# ก่อนเริ่มใช้งาน

## 1 ติดตั้ง Docker Desktop

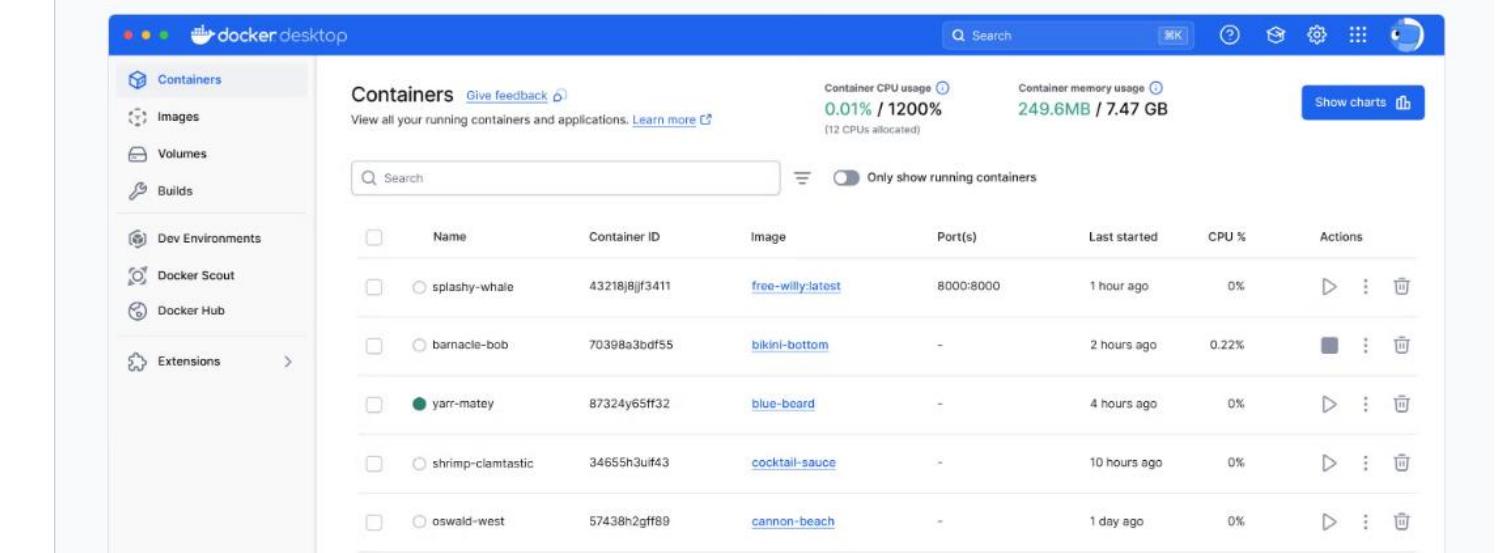


ดาวน์โหลดจาก:

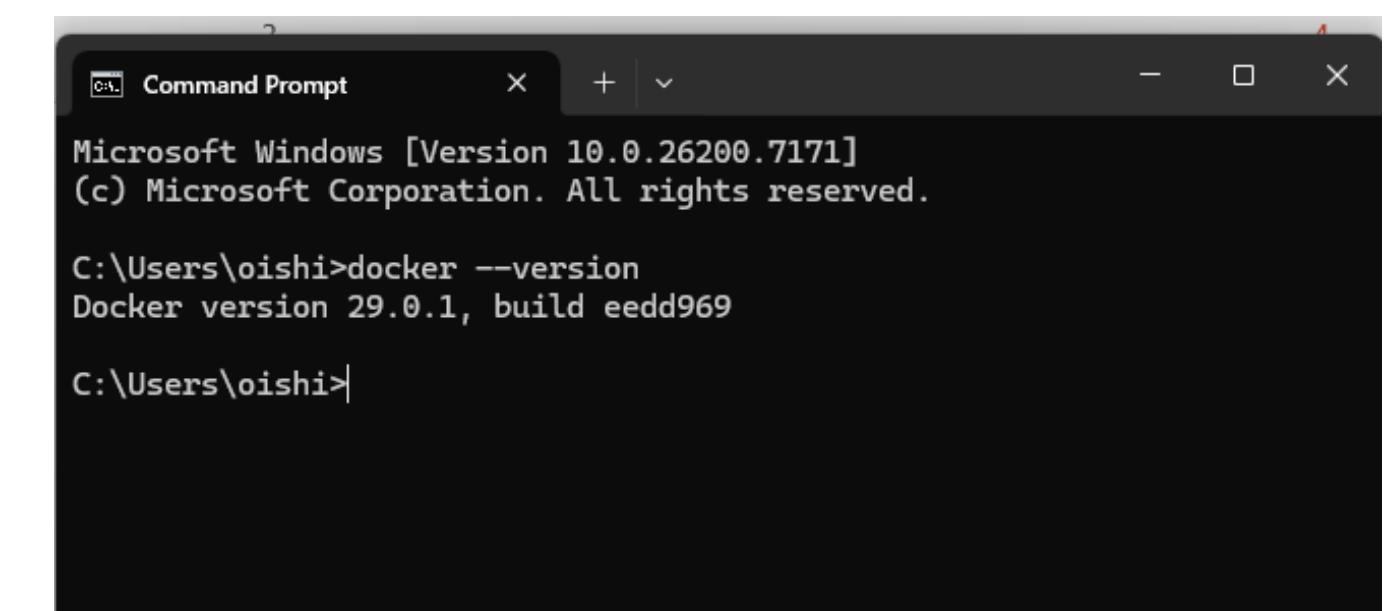
<https://www.docker.com/products/docker-desktop>



## 2 ติดตั้งเสร็จแล้วรีบูต Computer



## 3 เปิด Command Prompt หรือ PowerShell



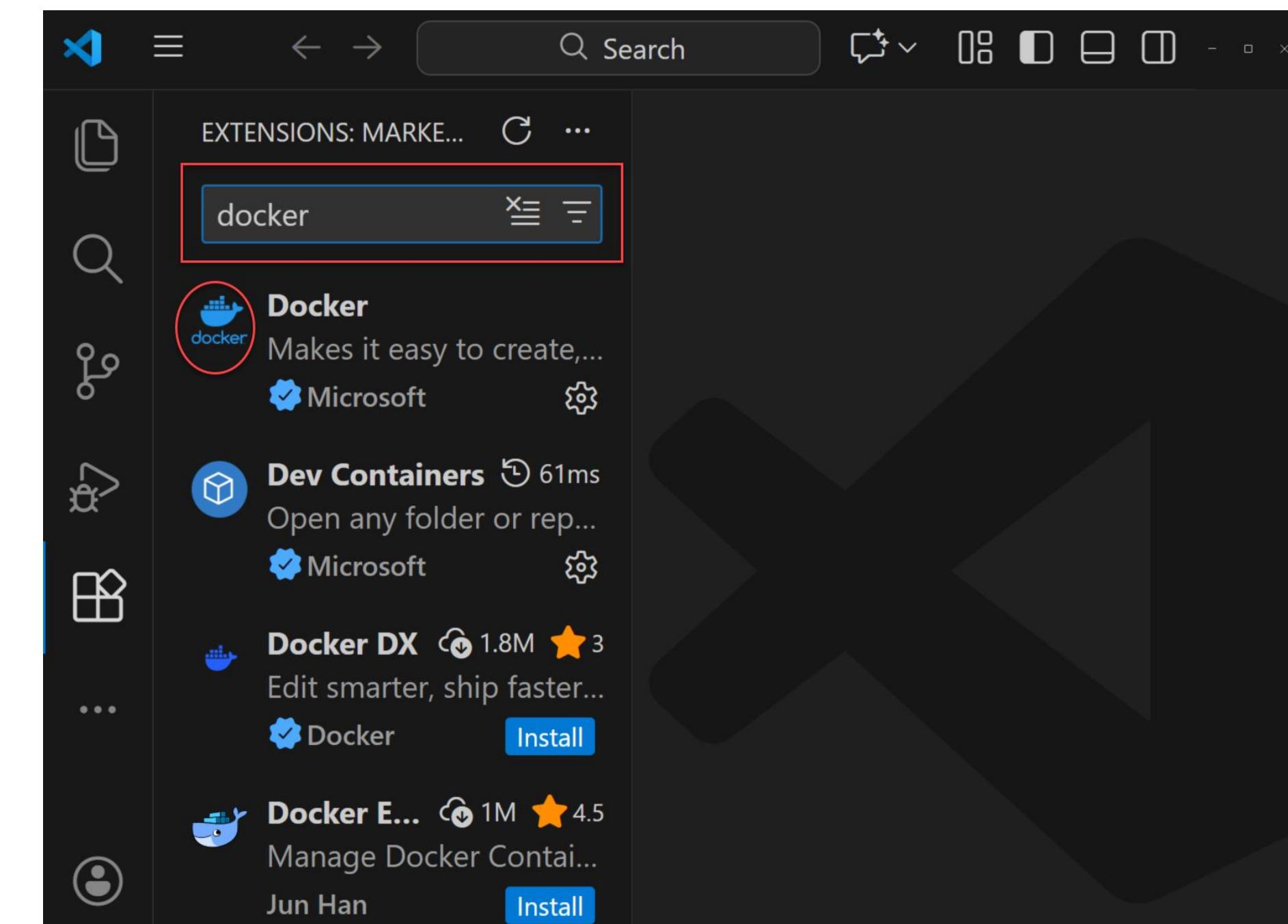
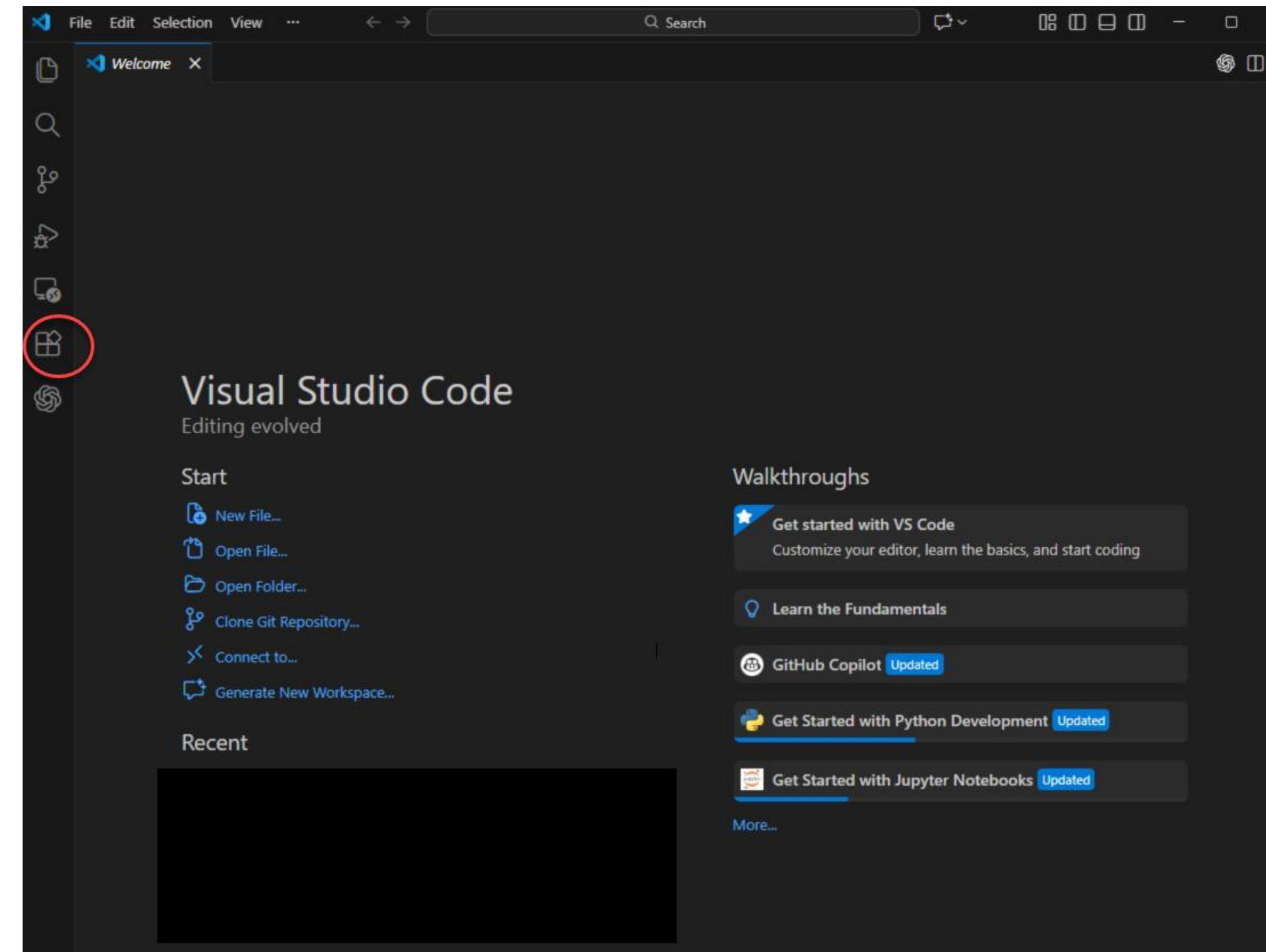
## 4 ทดสอบ Docker: พิมพ์ 'docker --version'

# docker desktop for windows

The screenshot shows the Docker Desktop for Windows application window. The title bar reads "docker desktop PRO". The left sidebar has a "Containers" section highlighted, along with other options like "Images", "Volumes", "Kubernetes", "Builds", "Models", "MCP Toolkit", "Docker Hub", "Docker Scout", and "Extensions". The main area is titled "Containers" and features a placeholder message: "Your running containers show up here. A container is an isolated environment for your code". Below this are two cards: "What is a container?" (5 mins) and "How do I run a container?". At the bottom, there's a link to "View more in the Learning center". The status bar at the bottom shows "Engine running", "RAM 1.55 GB CPU 0.03%", "Disk: 11.36 GB used (limit 1006.85 GB)", and "Terminal v4.52.0".

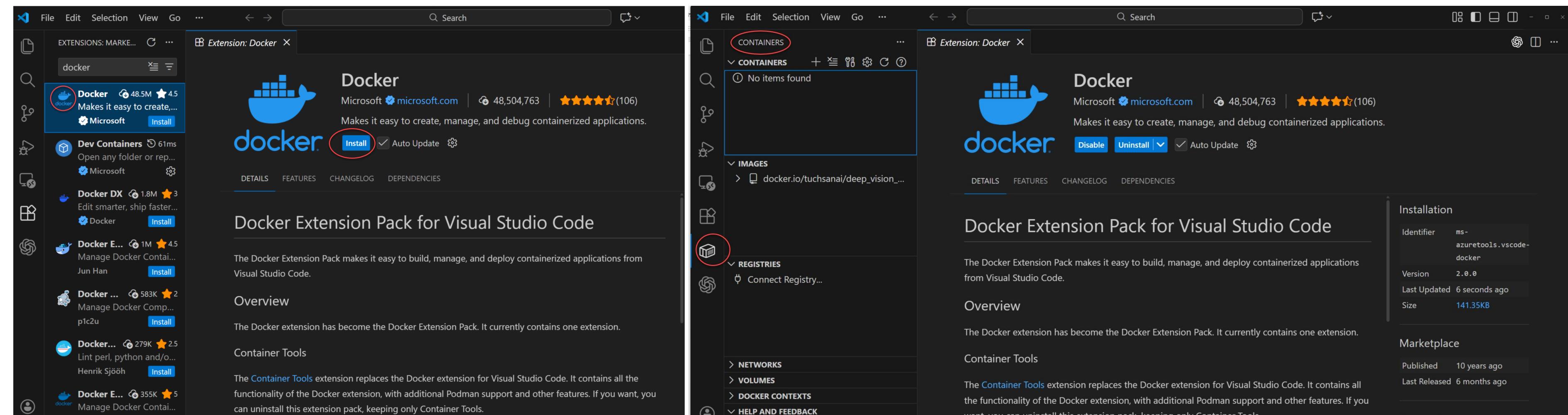


# การตั้งค่า Docker ใน VScode

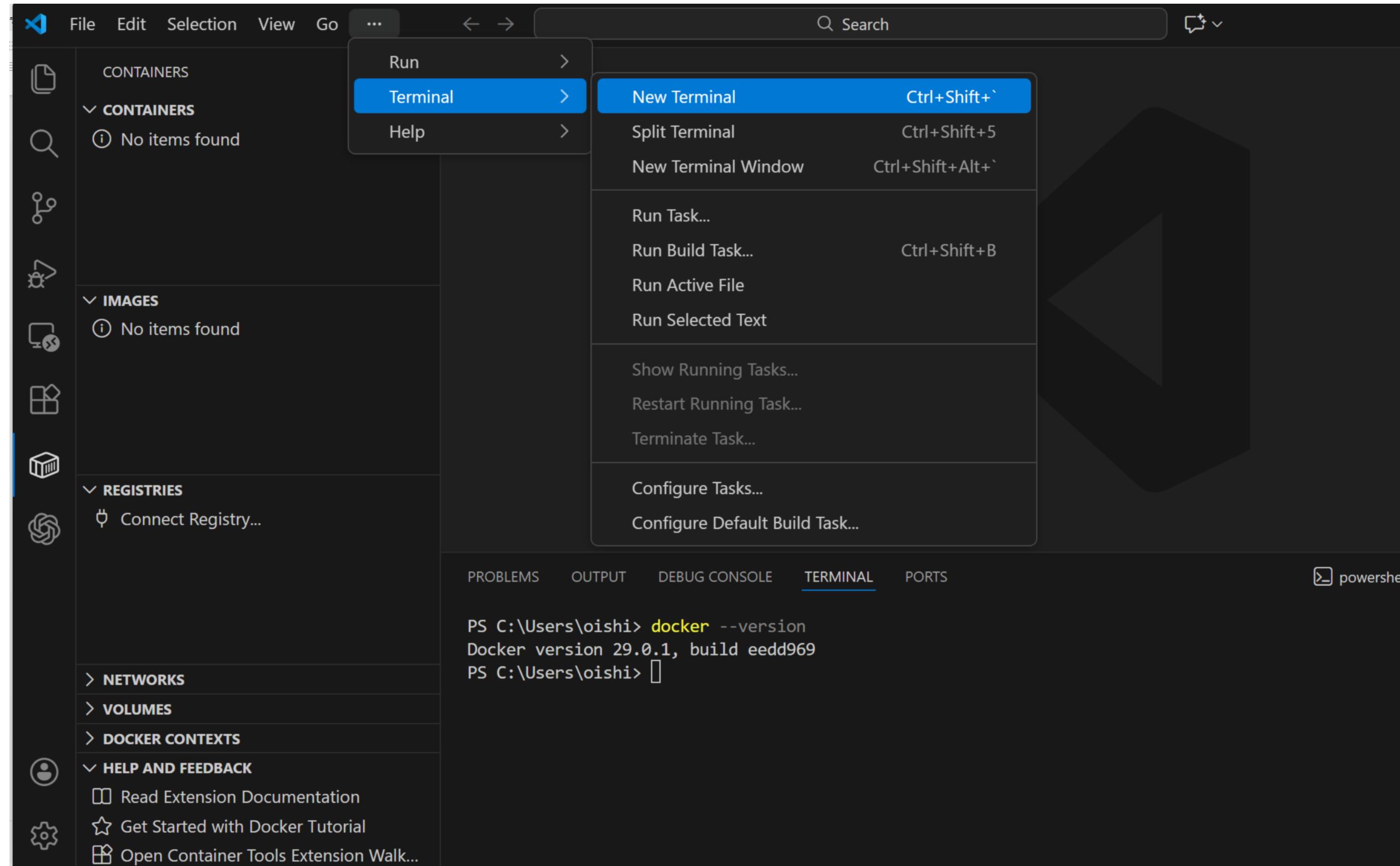


# หลงจากติดตั้งแล้วจะมี แบบ container ขึ้นมา

กด Install



# ทดสอบว่า Vscode สามารถเชื่อมต่อกับ Docker ได้

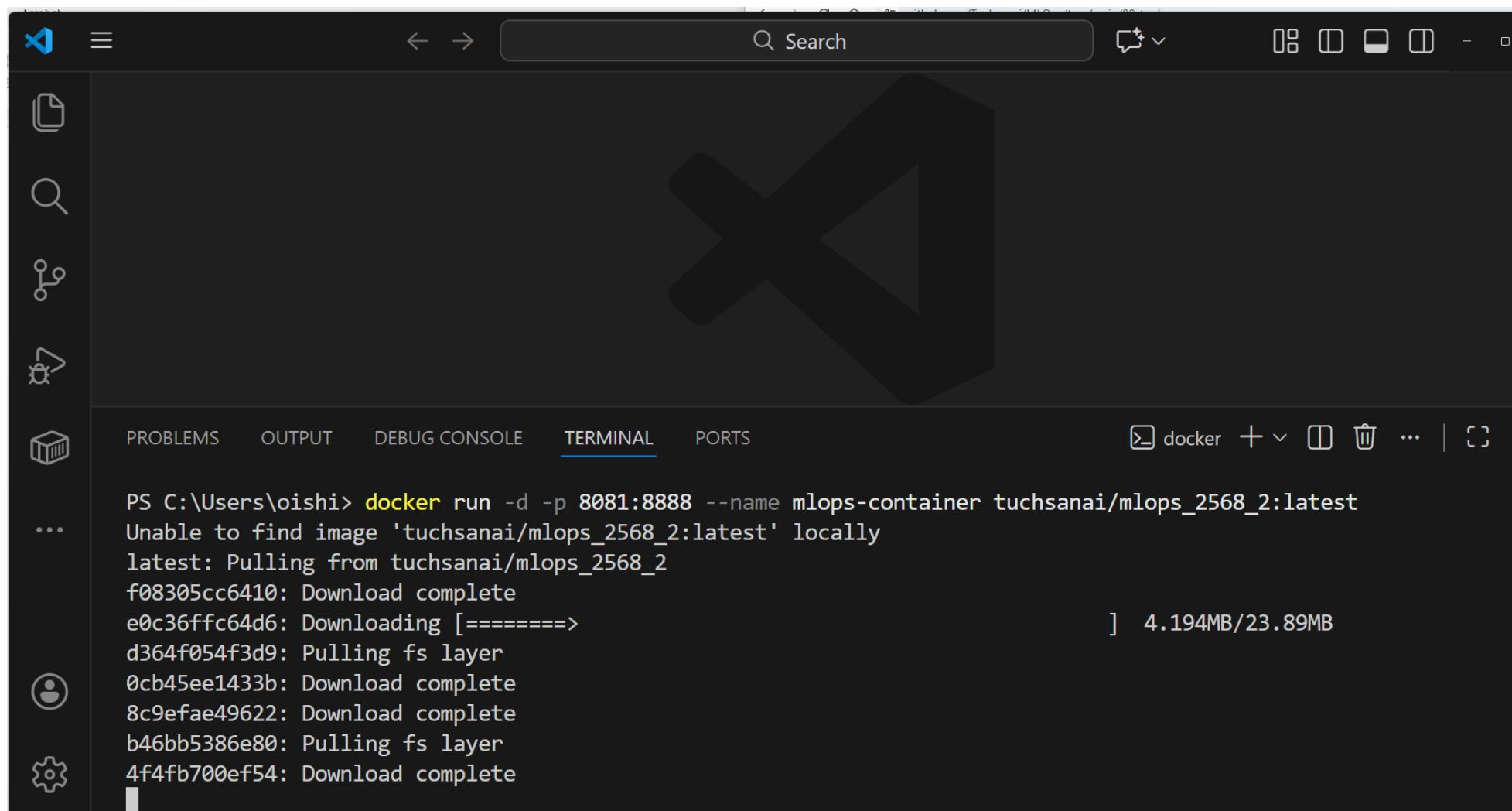


# Run container

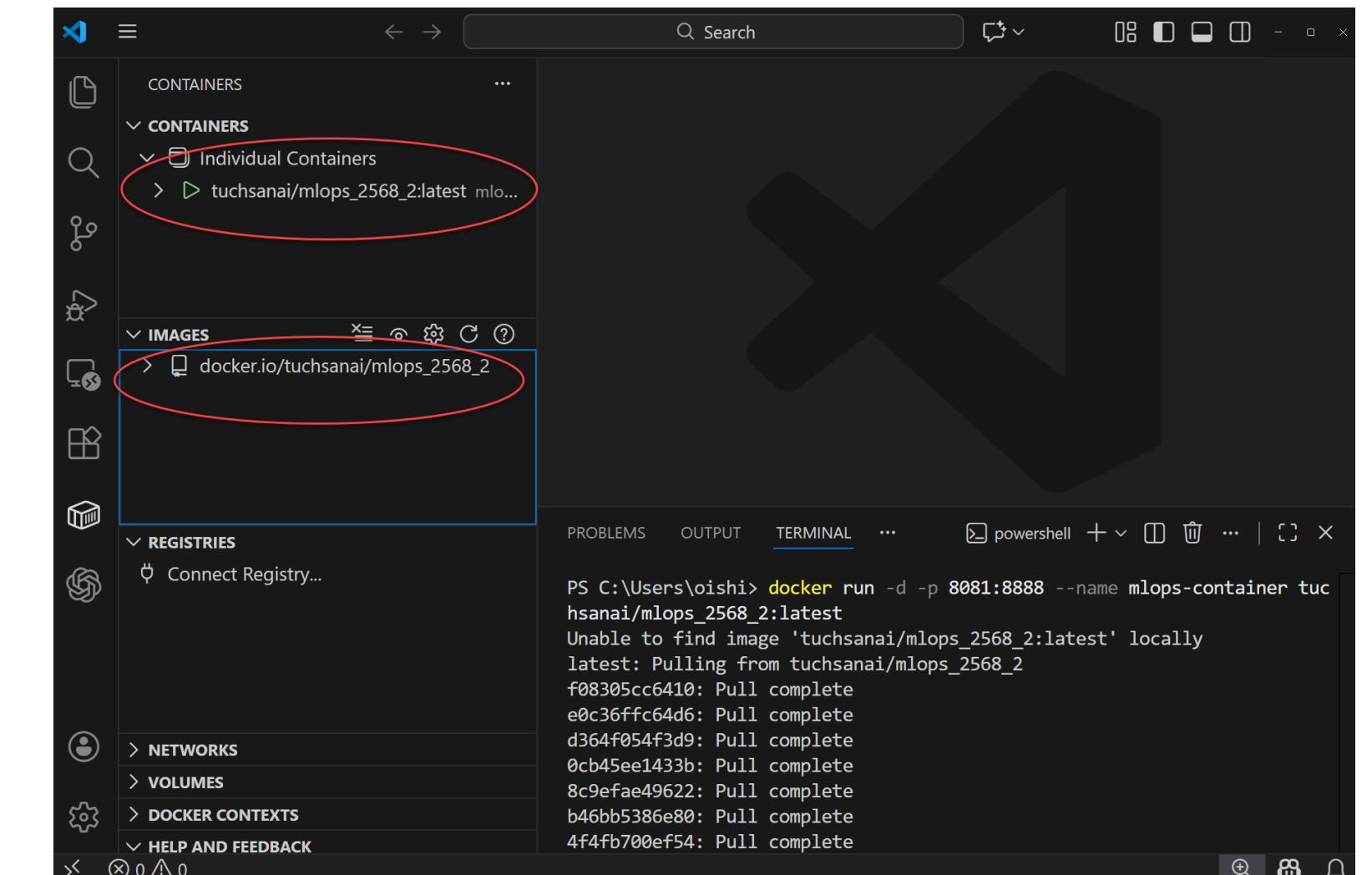
## Run the Container

```
docker run -d -p 8081:8888 --name mlops-container tuchsanai/mlops_2568_2:latest
```

- `-d` : Run in detached mode (background)
- `-p 8081:8888` : Map container port 8888 to local port 8081
- `--name mlops-container` : Assigns a custom container name
- Launches JupyterLab accessible via <http://localhost:8081>



```
PS C:\Users\oishi> docker run -d -p 8081:8888 --name mlops-container tuchsanai/mlops_2568_2:latest
Unable to find image 'tuchsanai/mlops_2568_2:latest' locally
latest: Pulling from tuchsanai/mlops_2568_2
f08305cc6410: Download complete
e0c36ffc64d6: Downloading [=====>] 4.194MB/23.89MB
d364f054f3d9: Pulling fs layer
0cb45ee1433b: Download complete
8c9efae49622: Download complete
b46bb5386e80: Pulling fs layer
4f4fb700ef54: Download complete
```



```
PS C:\Users\oishi> docker run -d -p 8081:8888 --name mlops-container tuchsanai/mlops_2568_2:latest
Unable to find image 'tuchsanai/mlops_2568_2:latest' locally
latest: Pulling from tuchsanai/mlops_2568_2
f08305cc6410: Pull complete
e0c36ffc64d6: Pull complete
d364f054f3d9: Pull complete
0cb45ee1433b: Pull complete
8c9efae49622: Pull complete
b46bb5386e80: Pull complete
4f4fb700ef54: Pull complete
```

# Container in Docker Desktop

```
# Base image: Ubuntu 25.04
FROM ubuntu:25.04

# Install dependencies and set up environment
RUN apt-get update && \
    apt-get install -y python3-pip python3-dev curl && \
    apt-get clean

# Install Python packages
RUN pip3 install --upgrade pip && \
    pip3 install \
        jupyterlab \
        ipykernel \
        pandas \
        scikit-learn \
        opencv-python-headless

# Set working directory
WORKDIR /home/student/workspace

# Launch Jupyter Lab with specific configuration
CMD ["jupyter", "lab", "--ip=0.0.0.0", "--port=8888", "--no-browser", "--NotebookApp.token='mlops'"]
```

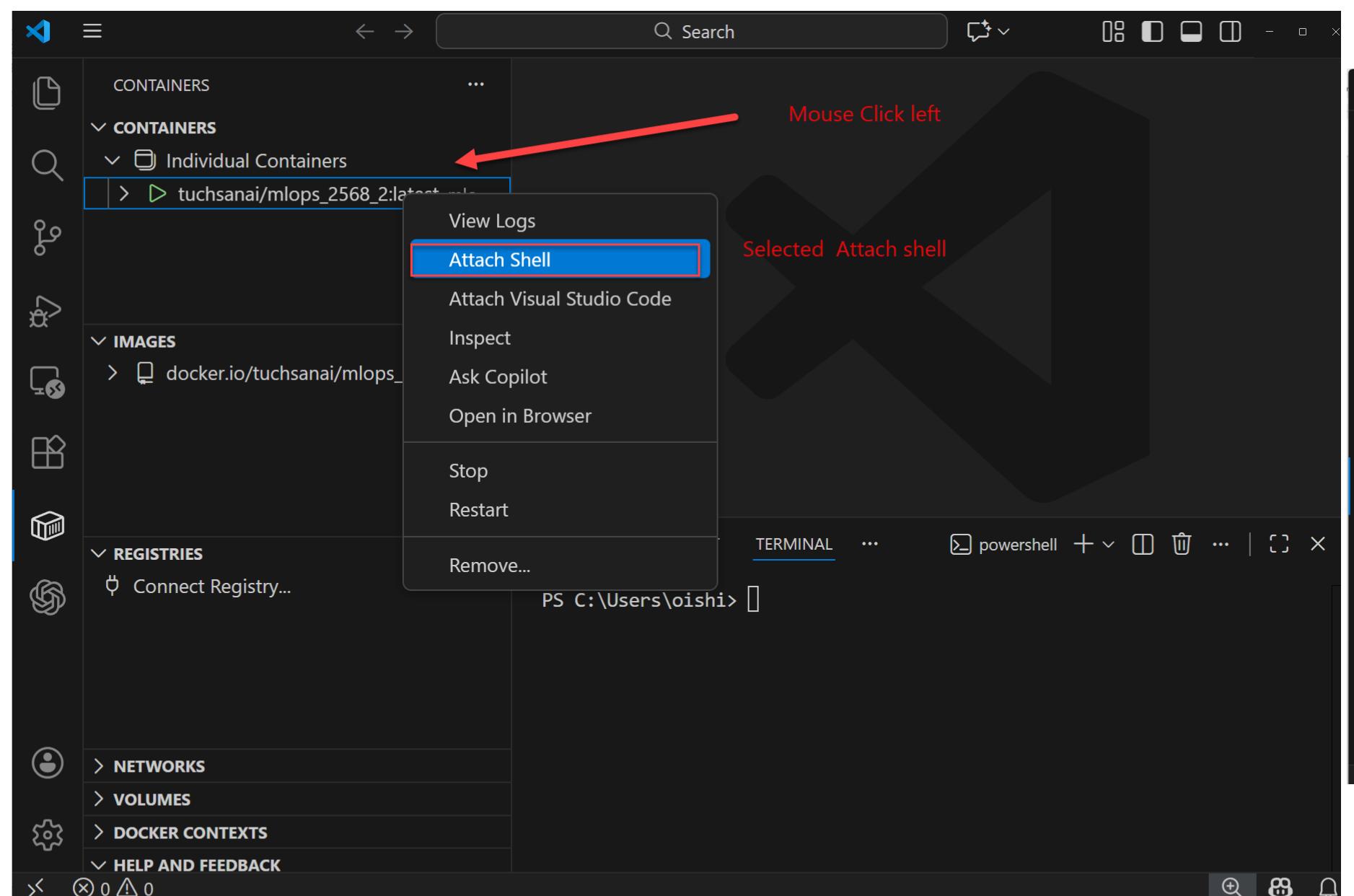
The screenshot shows the Docker Desktop interface. On the left, there is a sidebar with various options: Ask Gordon (BETA), Containers (selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' with a 'Give feedback' link. It displays container statistics: Container CPU usage (0.00% / 3200%) and Container memory usage (89.73MB / 60.14GB). A search bar and a 'Only show running containers' toggle are also present. A table lists the running container:

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
mlops-container	6f5650cf980a	tuchsanai/mlops_2568_2:latest	8081:8888	0%	3 minutes ago	[Run, Stop, Delete]

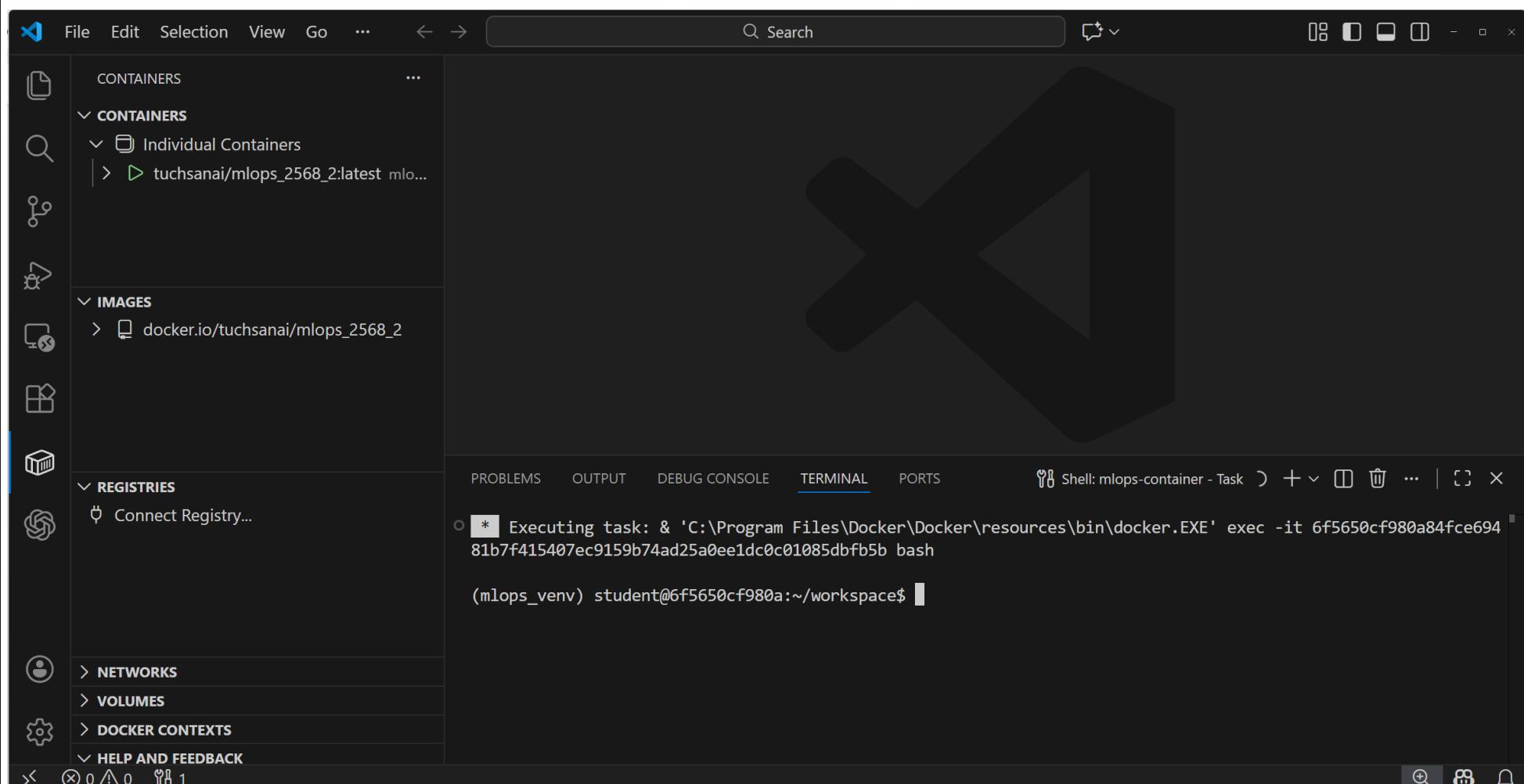
A red oval highlights the 'Image' column for the running container.

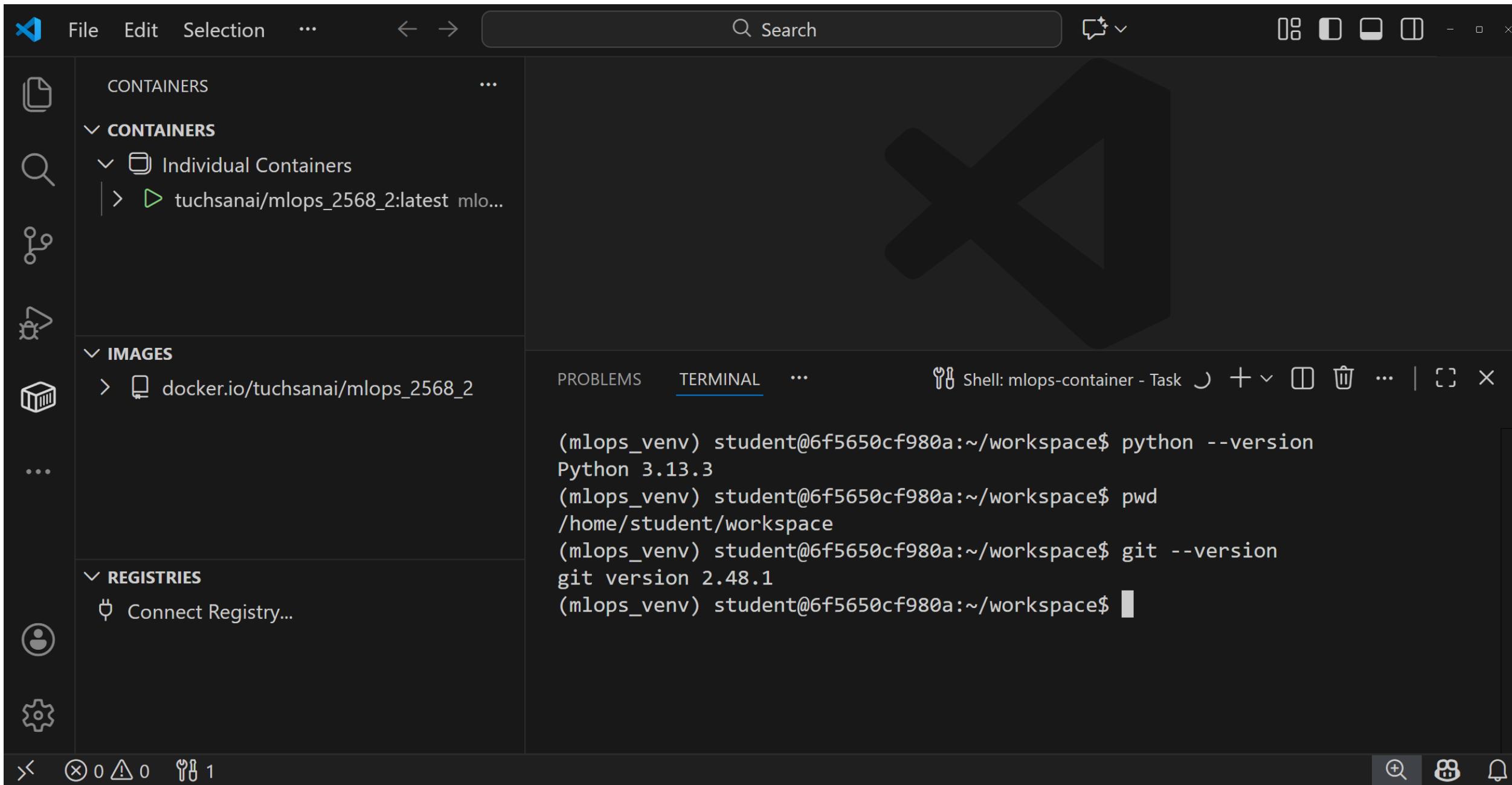
การ Login เข้าไปใน Container วิธีที่ 1

# Selected container



# Login ใน container เรียบร้อย





## . คำสั่งใน Terminal (ภายใน container)

### คำสั่ง

```
python --version
```

### ความหมาย

ตรวจสอบว่า Python ใน container เวอร์ชันอะไร 👉 Python 3.13.3

```
pwd
```

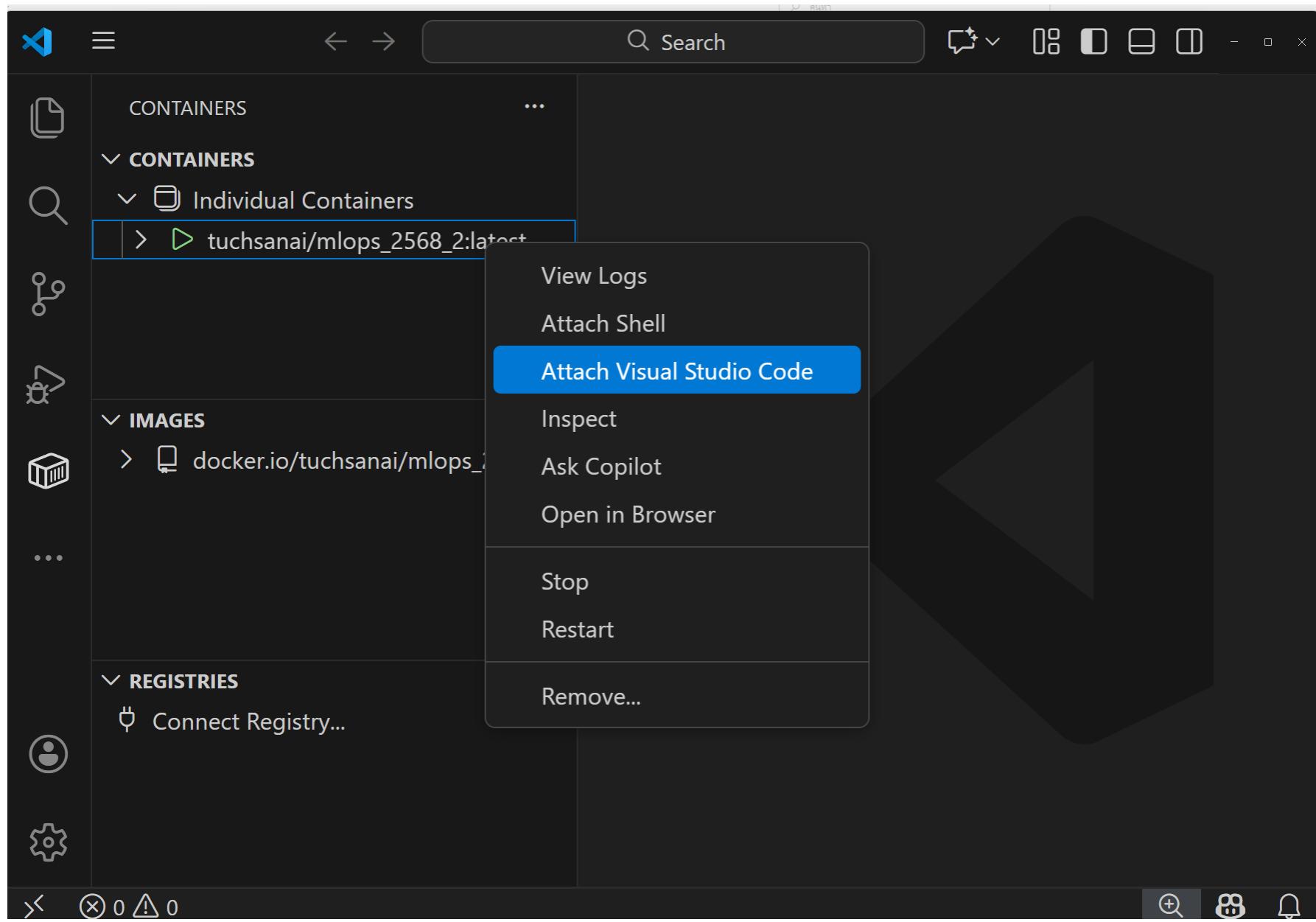
ดูตำแหน่งโฟลเดอร์ที่อยู่ตอนนี้ 👉 /home/student/workspace

```
git --version
```

ตรวจสอบ git เวอร์ชัน 👉 git version 2.48.1

การ Login เข้าไปใน Container วิธีที่ 2

# Selected container



# Login ใน container เรียบร้อย

