

# MACHINE LEARNING OPERATIONS



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

WEEK 1



# 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

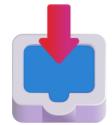


ความปลอดภัย: ไม่กระทบต่อระบบ Windows ของคุณ



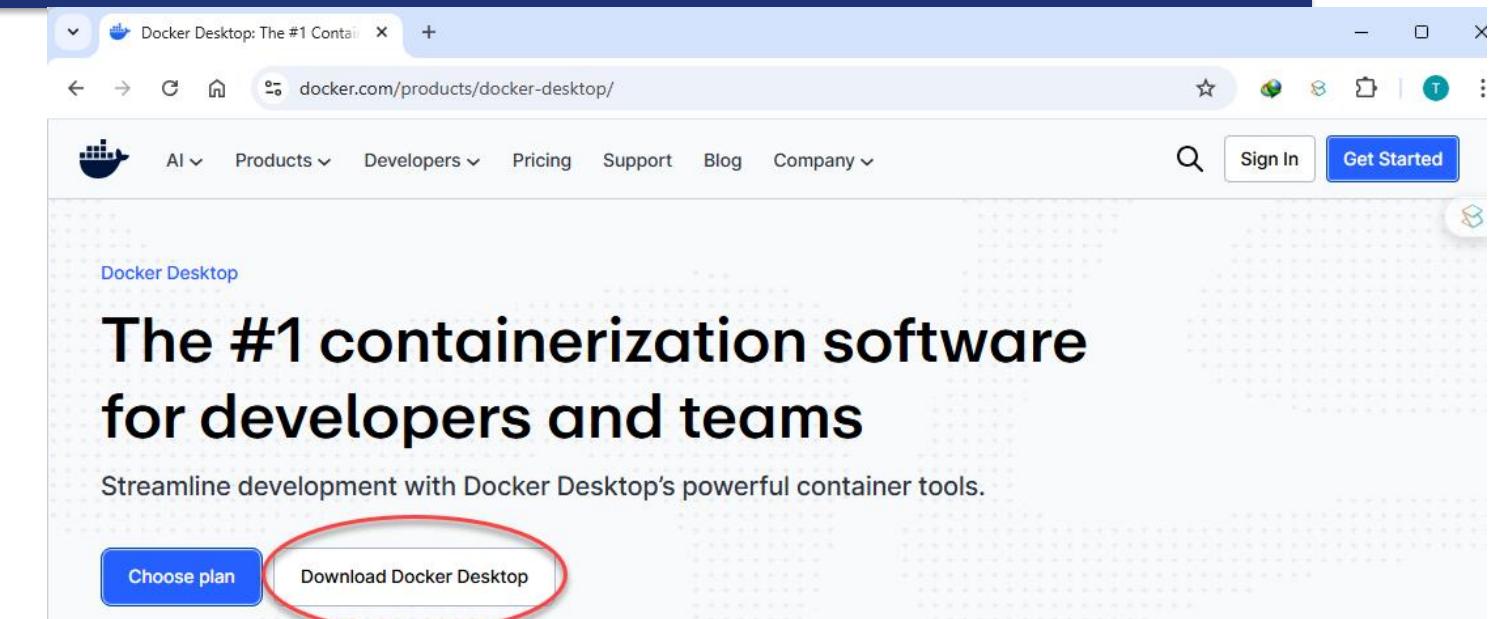
# ก่อนเริ่มใช้งาน

## 1 ติดตั้ง Docker Desktop

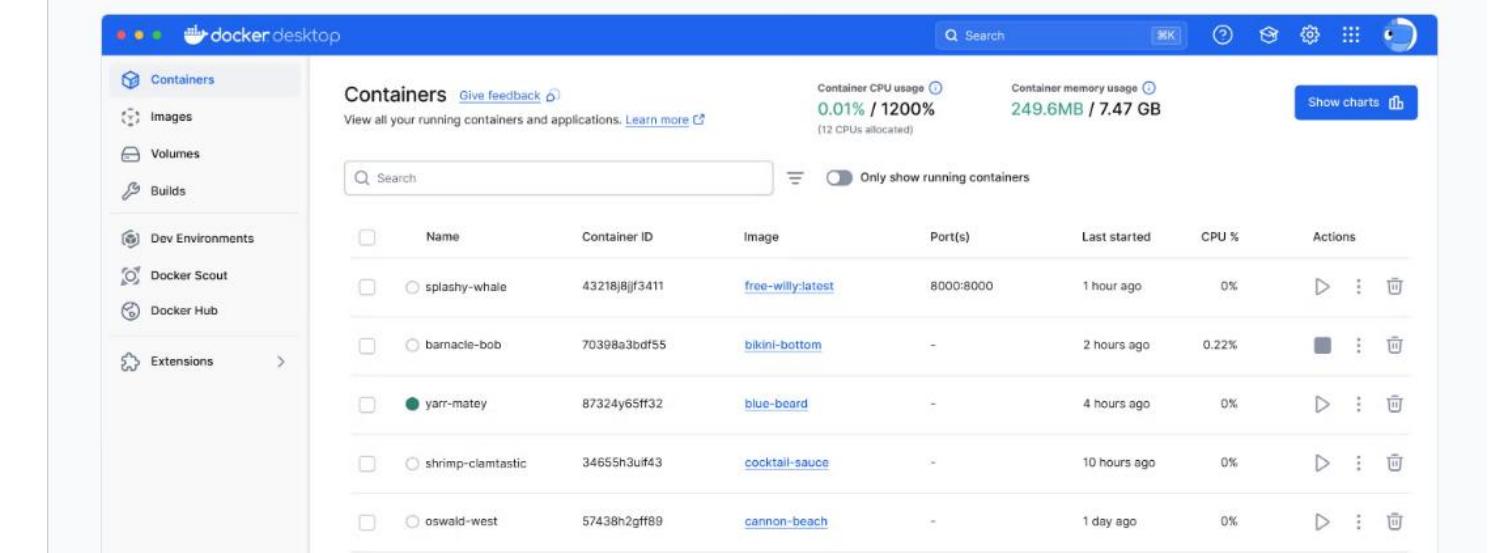


ดาวน์โหลดจาก:

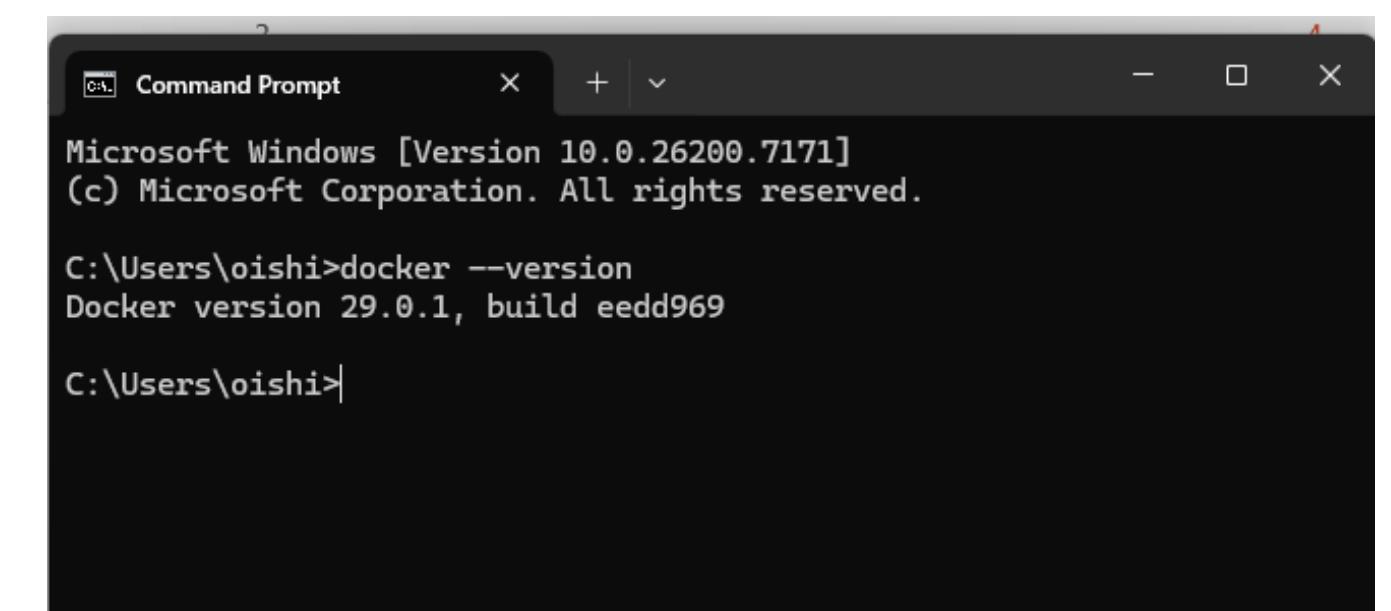
<https://www.docker.com/products/docker-desktop>



## 2 ติดตั้งเสร็จแล้วรีบูต Computer

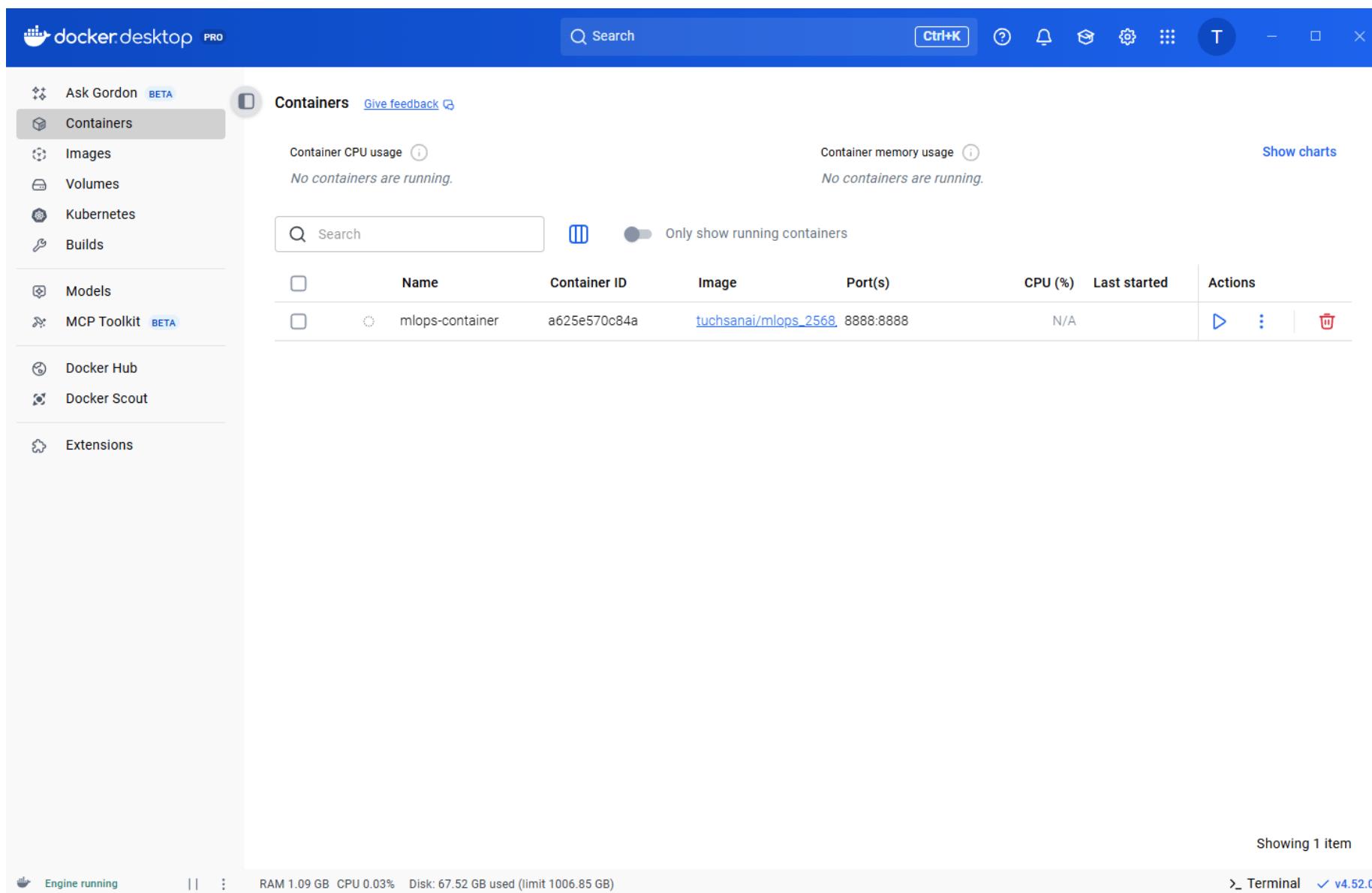


## 3 เปิด Command Prompt หรือ PowerShell



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

# docker desktop for windows



```

# Base image: Ubuntu 25.04
FROM ubuntu:25.04

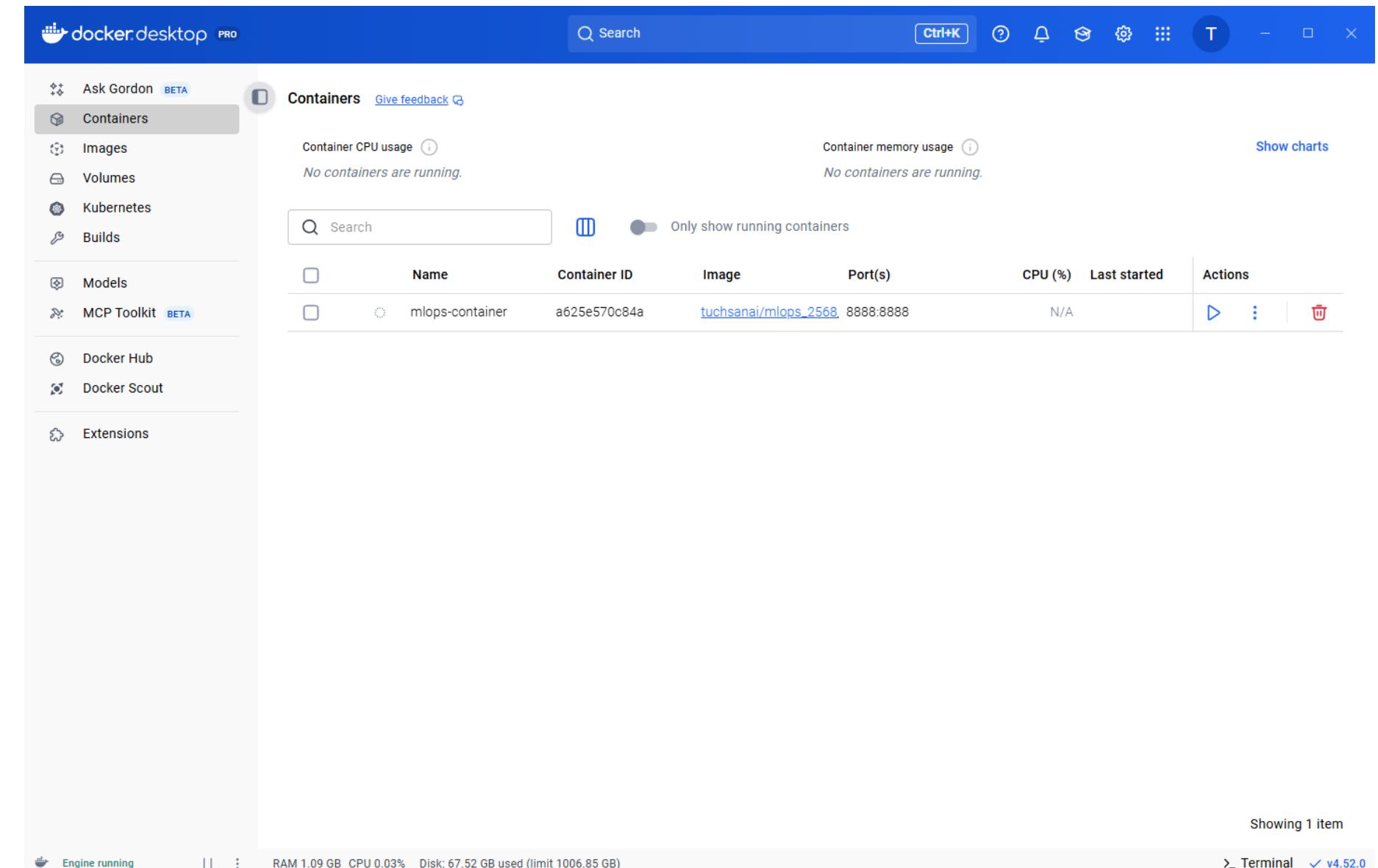
# Install dependencies and set up environment
RUN apt-get update && \
    apt-get install -y python3-pip python3-dev curl && \
    apt-get clean

# 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

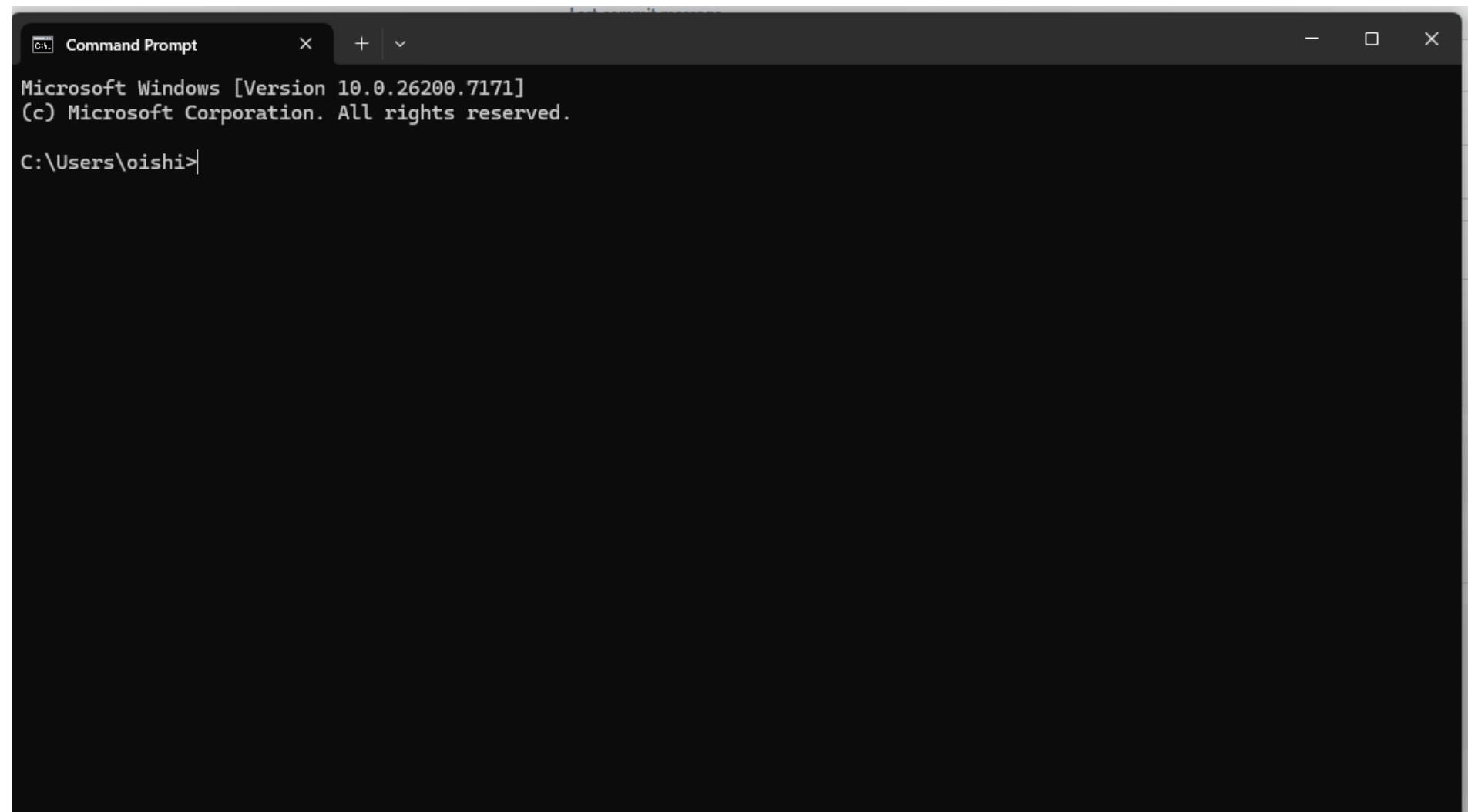
# Launch Jupyter Lab with specific configuration
CMD ["jupyter", "lab", "--ip=0.0.0.0", "--port=8888", "--no-browser", "--NotebookApp.token='mlops'"]

```



## For the Windows Command Prompt (CMD)

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

















# Docker Container Environment Setup



# Dockerfile (Container Recipe)

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

# Install dependencies
RUN apt-get update && \
    apt-get install -y python3-pip 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 JupyterLab
CMD ["jupyter", "lab", "--ip=0.0.0.0", "--port=8888",
      "--no-browser", "--NotebookApp.token='mlops'"]
```



# การรัน Container บน Windows (CMD)

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



คำอธิบาย:

-d = รันในแบบ Background  
-p 8888:8888 = เผด Port 8888  
--name = ตั้งชื่อ Container  
^ = Line continuation character (Windows)



# การรัน Container บน Windows (PowerShell)

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



คำอธิบาย:

-d = รันในแบบ Background  
-p 8888:8888 = เผด Port 8888  
--name = ตั้งชื่อ Container  
` = Line continuation character (PowerShell)



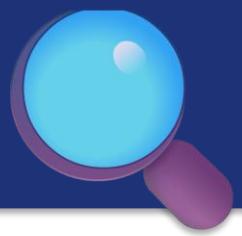
# การรัน Container บน Linux/Mac

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



คำอธิบาย:

-d = รันในแบบ Background  
-p 8888:8888 = 映射 Port 8888  
--name = ตั้งชื่อ Container  
\ = Line continuation character (Linux/Mac)



# ตรวจสอบว่า Container รันได้หรือไม่

**1** ดูรายการ Container ที่รันอยู่:

```
$ docker ps
```

**2** ดูทั้งหมด (รวมตัวที่ปิดแล้ว):

```
$ docker ps -a
```

**3** ดู Logs:

```
$ docker logs mlops-container
```



# เข้าใช้ JupyterLab

**1** เปิด Web Browser (Chrome, Firefox, Edge)

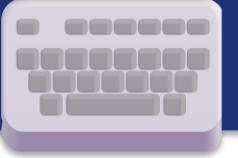
**2** ไปที่: <http://localhost:8888>

**3** ใส่ Password: mlops

**4** ตอนนี้คุณอยู่ใน Linux Environment บน Container!



ทำงานใน: /home/student/workspace



# Docker Commands ที่สำคัญ

เข้ม Container: docker start mlops-container

หยุด Container: docker stop mlops-container

ลบ Container: docker rm mlops-container

ดู Containers: docker ps -a

ดู Logs: docker logs mlops-container



# การแก้ไขปัญหา

❓ Container ไม่รัน?

→ ตรวจสอบว่า Docker Desktop เปิดอยู่

❓ Port 8888 ถูกใช้งาน?

→ เปลี่ยน Port: -p 8889:8888

❓ ลืมอยากหยุด Container?

→ docker stop mlops-container

❓ ต้องการหลาย Container?

→ เปลี่ยนชื่อและ Port ให้ต่างกัน



# Best Practices



สร้างโฟลเดอร์เพื่อ Mount:

ช่วยให้ไฟล์ยังคงอยู่หลังปิด Container



บันทึก Work: Save บ่อยๆ ใน JupyterLab



ทำความสะอาด: ลบ Container เก่าที่ไม่ใช้



Update Image: ดึง Image ใหม่เป็นครั้งคราว



# ขั้นตอนถัดไป

**1** ติดตั้ง Docker Desktop บน Windows

**2** รัน Container ตามคำสั่ง

**3** เข้า JupyterLab บน localhost:8888

**4** สร้าง Notebook ใหม่เพื่อเริ่มเรียน MLOps



ดำเนินปัญหา ติดต่อทีมสอน