**Assignment12: Docker**

**Basic Docker**1.เข้าใช้งาน play with docker [Play with Docker (play-with-docker.com)](https://labs.play-with-docker.com/) สามารถล็อกอินโดยใช้ google account หรือ github account

2.รันคำสั่ง   
$docker version

|  |
| --- |
|  |

3.รันคำสั่ง  
$docker run ubuntu echo Hello World

|  |
| --- |
|  |

อธิบายคำสั่ง

|  |
| --- |
|  |

4.รันคำสั่ง  
$docker ps -a

|  |
| --- |
|  |

อธิบายคำสั่ง

|  |
| --- |
|  |

5.รันคำสั่ง  
$docker start -a [container\_id] or [container\_name]

|  |
| --- |
|  |

อธิบายคำสั่ง

|  |
| --- |
|  |

6.รันคำสั่ง  
$docker run -it ubuntu

|  |
| --- |
|  |

$docker ps -a

|  |
| --- |
|  |

$docker start [container\_id]

|  |
| --- |
|  |

$docker exec -it [container\_id] bash

|  |
| --- |
|  |

$docker exec -it [container\_id] echo hello world

|  |
| --- |
|  |

$docker ps

|  |
| --- |
|  |

$docker stop [container\_id]  
$docker ps -a

|  |
| --- |
|  |

ความแตกต่างระหว่าง docker ps -a และ docker ps คือ

|  |
| --- |
|  |

7.ลองรันคำสั่ง  
$docker rm [container\_id]  
$docker ps -a

|  |
| --- |
|  |

8.ลองรันคำสั่ง  
$docker run -it ubuntu   
$docker ps -a

|  |
| --- |
|  |

$docker start [container\_id]  
$docker rm -f [container\_id]

|  |
| --- |
|  |

9.ลองรันคำสั่ง  
$docker run --name os -it ubuntu  
$docker ps -a

|  |
| --- |
|  |

$docker rm -f os  
$docker ps -a

|  |
| --- |
|  |

Option --name ไว้ทำอะไร

|  |
| --- |
|  |

10. รันคำสั่ง  
$docker image ls

|  |
| --- |
|  |

11. รันคำสั่ง   
$docker image rm ubuntu

|  |
| --- |
|  |

**Advanced Docker**  
1.ให้นิสิตสร้าง folder ขึ้นมาใหม่ดังชื่อว่า docker\_hello

2.จากนั้นให้สร้างไฟล์ hello.py และนำใส่ folder นี้เช่นกัน ขึ้นมาโดยใส่โค้ดดังนี้

import time

i=0

while True:

print(f'Hello [Your name] {i}')

i=i+1

time.sleep(1)

ซึ่งโค้ดนี้ให้นิสิตแทน [Your name] ด้วยชื่อของนิสิต การทำงานของโค้ดนี้จะทำการวนลูปแสดงข้อความ Hello [Your name] และมีตัวเลขนับการแสดงผล โดยจะแสดงและหยุดหนึ่งวิไปเรื่อยๆ

3.จากนั้นให้สร้างไฟล์ชื่อ Dockerfile อยู่ใน folder นี้เช่นกันโดยใส่รายละเอียดดังนี้

FROM python:3

WORKDIR /app

COPY hello.py .

CMD ["python", "hello.py"]

จาก Dockerfile ข้างต้นจะเป็นการลง python บรรทัดที่สองจะเป็นการกำหนด working directory เป็น /app และบรรทัดที่สามจะทำการก็อปปี้โค้ด hello.py ในโฟลเดอร์ docker\_hello ไปวางไว้ที่ current working directory ใน container ซึ่งก็คือ /app นั้นเอง และบรรทัดสุดท้ายคือคำสั่งที่ต้องทำเมื่อ container ถูกรัน

4.รันคำสั่งดังนี้เพื่อสร้าง image file ที่ชื่อ hello\_image

docker build -t hello\_image .

5.สร้างและรัน container โดยให้ชื่อว่า hello\_container จาก image file ดังกล่าวด้วยคำสั่ง

docker run -dt --name hello\_container hello\_image

-d จะเป็นการรันแบบ background

-t จะเป็นการสร้างท่อที่เชื่อมระหว่าง container และ terminal ให้สามารถรับ input และ output ได้

อ้างอิง [docker run | Docker Docs](https://docs.docker.com/reference/cli/docker/container/run/)

6.container ดังกล่าวจะรันแบบ background โดยจะแสดงข้อความ Hello [Your name] และมีเลขนับ โดยจะแสดงและหยุดหนึ่งวิไปเรื่อยๆ ถ้าอยากดูว่า container ดังกล่าวรันอะไรอยู่ในใช้คำสั่ง

docker logs -f hello\_container

ก็จะแสดงข้อความรันเรื่อยๆ ให้กด ctrl+c เพื่อออกจากการดู log

7.เราสามารถสั่ง stop container ได้ดังนี้

docker stop hello\_container

8.เราสามารถสั่งให้ start container ได้ดังนี้

docker start hello\_container

9.และตรวจสอบว่ามี container ไหนรันอยู่บ้างด้วยคำสั่ง docker ps -a

10.และสามารถลบ container ได้ด้วยคำสั่ง docker rm hello\_container หรือ สังให้หยุดและลบในคราวเดียวกันด้วยคำสั่ง docker rm -f hello\_container

11.เราสามารถดู image ทั้งในหมดในเครื่องได้ด้วยคำสั่ง docker images และทำการลบ image ที่ต้องการได้โดยใช้คำสั่ง docker rmi [IMAGE name or IMAGE id]

12.จากนั้นเราจะแชร์ image นี้ (hello\_image) ขึ้นไปยัง docker hub โดยทำดังนี้

a. สมัคร account กับ docker hub ก่อน [Docker Hub Container Image Library | App Containerization](https://hub.docker.com/)

b. จากนั้น login ให้เรียบร้อย คลิกปุ่ม Create Repository

c. สำหรับ repo name ให้ตั้งว่า hello\_image และให้เซ็ท Visibility เป็น public

d. กดปุ่ม Create

e. ใน terminal ให้พิมพ์ docker login -u YOUR-USER-NAME

​\*\*แทน YOUR-USER-NAME เป็นชื่อแอคเคาท์ user ของนิสิต

f. พิมพ์ docker tag hello\_image YOUR-USER-NAME/hello\_image

​\*\*แทน YOUR-USER-NAME เป็นชื่อแอคเคาท์ user ของนิสิต

g. จากนั้นทำการส่ง image นี้จากเครื่องเราไปที่ docker hub ด้วยคำสั่ง

docker push YOUR-USER-NAME/hello\_image

​\*\*แทน YOUR-USER-NAME เป็นชื่อแอคเคาท์ user ของนิสิต

h. เมื่ออัพเสร็จแล้วนิสิตสามารถทดสอบ image ได้ที่ [Play with Docker (play-with-docker.com)](https://labs.play-with-docker.com/) ให้กดปุ่ม login และเลือก docker และกด start

i. กดสร้าง +ADD NEW INSTANCE

j. จะมี terminal โผล่มาให้นิสิตรันด้วยคำสั่ง

docker run -t --name hello\_container YOUR-USER-NAME/hello\_image

​​\*\*แทน YOUR-USER-NAME เป็นชื่อแอคเคาท์ user ของนิสิต

คอนเทนเนอร์นี้ก็จะรันแสดงข้อความไปเรื่อยๆ ให้กด ctrl+c เพื่อออก

k. ให้นิสิตส่งลิงค์ของ public repository ของ docker hub มาส่งใน msteam

ตัวอย่างลิงค์ https://hub.docker.com/repository/docker/korawitorkphol/hello\_image/general

อ้างอิง [Overview of the get started guide | Docker Docs](https://docs.docker.com/get-started/)