

Lab Worksheet

ชื่อ-นามสกุล นายชลพัฒน์ ปิ่นมณี รหัสนักศึกษา 653380126-0 Section 1

Lab#8 – Software Deployment Using Docker

วัตถุประสงค์การเรียนรู้

1. ผู้เรียนสามารถอธิบายเกี่ยวกับ Software deployment ได้
2. ผู้เรียนสามารถสร้างและรัน Container จาก Docker image ได้
3. ผู้เรียนสามารถสร้าง Docker files และ Docker images ได้
4. ผู้เรียนสามารถนำซอฟต์แวร์ที่พัฒนาขึ้นให้สามารถรันบนสภาพแวดล้อมเดียวกันและทำงานร่วมกันกับสมาชิกในทีมพัฒนาซอฟต์แวร์ผ่าน Docker hub ได้
5. ผู้เรียนสามารถเริ่มต้นใช้งาน Jenkins เพื่อสร้าง Pipeline ในการ Deploy งานได้

Pre-requisite

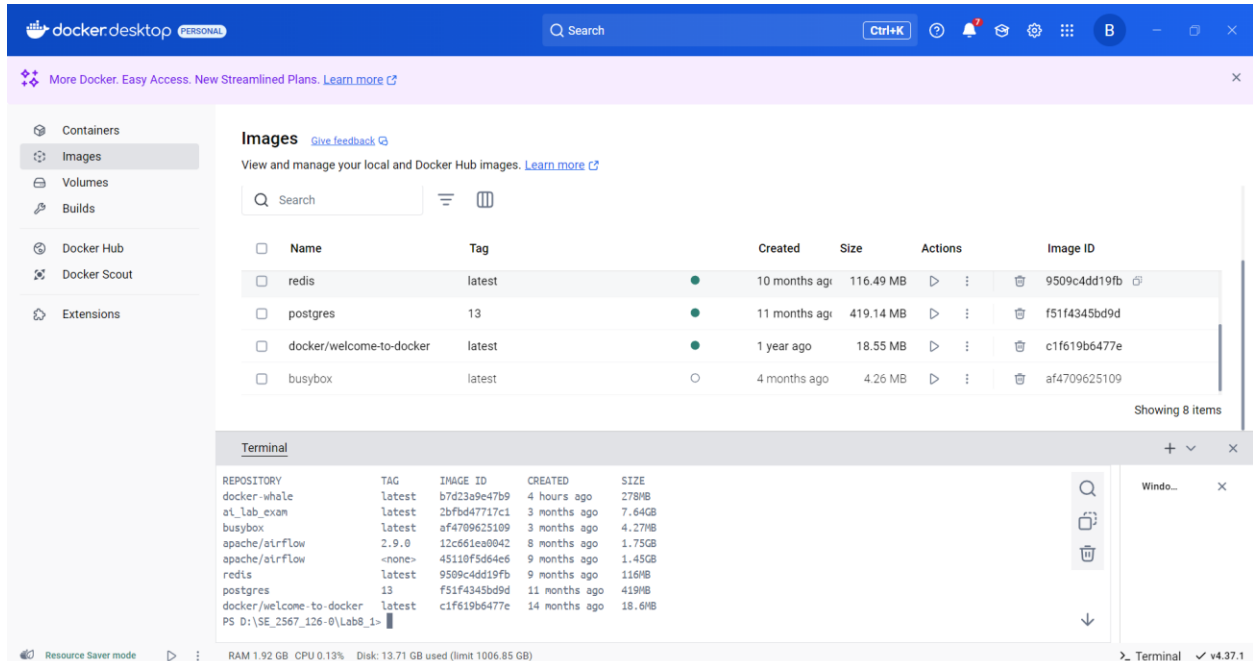
1. ติดตั้ง Docker desktop ลงบนเครื่องคอมพิวเตอร์ โดยดาวน์โหลดจาก <https://www.docker.com/get-started>
2. สร้าง Account บน Docker hub (<https://hub.docker.com/signup>)
3. กำหนดให้ \$ หมายถึง Command prompt และ <> หมายถึง ให้ป้อนค่าของพารามิเตอร์ที่กำหนด

แบบฝึกปฏิบัติที่ 8.1 Hello world - รัน Container จาก Docker image

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
1. เปิด Command line หรือ Terminal บน Docker Desktop จากนั้นสร้าง Directory ชื่อ Lab8_1
2. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_1 เพื่อใช้เป็น Working directory
3. ป้อนคำสั่ง \$ docker pull busybox หรือ \$ sudo docker pull busybox สำหรับกรณีที่ติดปัญหา Permission denied
(หมายเหตุ: BusyBox เป็น software suite ที่รองรับคำสั่งบางอย่างบน Unix - <https://busybox.net>)
4. ป้อนคำสั่ง \$ docker images

Lab Worksheet

[Check point#1] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ พร้อมกับตอบคำถามต่อไปนี้



(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

ตอบ ชื่อของ Repository ที่มีการเก็บ Images ของ Docker ไว้

(2) Tag ที่ใช้บ่งบอกถึงอะไร

ตอบ ใช้บ่งบอกถึงเวอร์ชันของ Images ใน repository นั้นๆ เช่น latest เป็นต้น

5. ป้อนคำสั่ง \$ docker run busybox

6. ป้อนคำสั่ง \$ docker run -it busybox sh

7. ป้อนคำสั่ง ls

8. ป้อนคำสั่ง ls -la

9. ป้อนคำสั่ง exit

10. ป้อนคำสั่ง \$ docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"

11. ป้อนคำสั่ง \$ docker ps -a

Lab Worksheet

[Check point#2] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ตั้งแต่ขั้นตอนที่ 6-12 พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows the Docker Desktop interface. On the left, the 'Containers' tab is selected. The main area displays a table of running containers:

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
cool_almeida	324ba2f92be6	busybox		N/A	5 minutes ago	[Stop] [Refresh] [Delete]
gifted_bardeen	ff9e8dcc7fbb	busybox		N/A	3 minutes ago	[Stop] [Refresh] [Delete]
agitated_solomon	7cebe244a97f	busybox		N/A	2 minutes ago	[Stop] [Refresh] [Delete]
demo_airflow	-	-	-	N/A	2 hours ago	[Stop] [Refresh] [Delete]

Below the table is a terminal window showing the following commands and output:

```
PS D:\SE_2567_126-0\Lab8_1> docker run busybox
PS D:\SE_2567_126-0\Lab8_1> docker run -it busybox sh
/ # ls
bin dev etc home lib lib64 proc root sys tmp usr var
/ # ls -la
total 48
drwxr-xr-x 1 root root 4096 Jan 22 08:37 .
drwxr-xr-x 1 root root 4096 Jan 22 08:37 ..
-rwxr-xr-x 1 root root 0 Jan 22 08:37 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
```

The bottom status bar indicates: Engine running, RAM 1.95 GB, CPU 0.00%, Disk: 13.71 GB used (limit 1006.85 GB).

The screenshot shows the Docker Desktop interface. On the left, the 'Containers' tab is selected. The main area displays a table of running containers, identical to the previous screenshot.

Below the table is a terminal window showing the following commands and output:

```
/ # exit
PS D:\SE_2567_126-0\Lab8_1> docker run busybox echo "Hello Chonlaphat Pinnunee from busybox"
Hello Chonlaphat Pinnunee from busybox
PS D:\SE_2567_126-0\Lab8_1> docker ps -a
```

The output of the 'docker ps -a' command is shown in a table:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7cebe244a97f	busybox	"echo 'Hello Chonlap..."	21 seconds ago	Exited (0) 20 seconds ago	
agitated_solomon	busybox	"sh"	2 minutes ago	Exited (0) About a minute ago	
ff9e8dcc7fbb	busybox				
gifted_bardeen	busybox				

The bottom status bar indicates: Engine running, RAM 1.96 GB, CPU 0.13%, Disk: 13.71 GB used (limit 1006.85 GB).

Lab Worksheet

- (1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

ตอบ -i: ทำให้สามารถเข้าไปภายในคอนเทนเนอร์และโต้ตอบได้ผ่านทาง terminal

-t: สร้าง pseudo-TTY (เทอร์มินัลเสมือน) ให้กับคอนเทนเนอร์ เพื่อให้สามารถใช้คำสั่งต่างๆ ภายใน terminal ได้

- (2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

ตอบ แสดงถึงสถานะของคอนเทนเนอร์ในขณะนั้น เช่น Exited (0) 20 seconds ago แสดงถึงคอนเทนเนอร์ได้หยุดทำงานแล้วโดยที่ไม่พบข้อผิดพลาด (exit code 0) และหยุดเมื่อ 20 วินาทีก่อนหน้านี้

12. ป้อนคำสั่ง \$ docker rm <container ID ที่ต้องการลบ>

[Check point#3] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 13

The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with navigation options: Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' and shows a table of running containers. Below the table is a terminal window showing the execution of the 'docker rm' command.

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
busy_ardinghelli	b267c5327749	docker-whale		N/A	4 hours ago	[Stop] [Refresh] [Delete]
welcome-to-docker	526b0a5f5c10	docker/welcome-to-docker	8088:80	N/A	9 months ago	[Stop] [Refresh] [Delete]
cool_almeida	324ba2f92be6	busybox		N/A	24 minutes ago	[Stop] [Refresh] [Delete]
gifted_bardeen	ff9e8dc77bb	busybox		N/A	22 minutes ago	[Stop] [Refresh] [Delete]

Showing 6 items

Terminal

```
demo_stirflow-stirflow-worker-1
13a5a168f489 apache/stirflow:2.9.0
demo_stirflow-stirflow-unit-1
e8ee4e83ff2d redis:latest
demo_stirflow-redis-1
526b0a5f5c10 docker/welcome-to-docker:latest
welcome-to-docker
PS D:\SE_2567_126-0\Lab8_1> docker rm c07a217aee897b64f7ecff241388e08b4334332435b92bce0db2df5e4b6aad11
c07a217aee897b64f7ecff241388e08b4334332435b92bce0db2df5e4b6aad11
PS D:\SE_2567_126-0\Lab8_1>
```

Engine running | RAM 1.98 GB CPU 3.17% Disk: 12.85 GB used (limit 1006.85 GB) | Terminal v4.37.1

แบบฝึกปฏิบัติที่ 8.2: สร้าง Docker file และ Docker image

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
2. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_2
3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_2 เพื่อใช้เป็น Working directory

Lab Worksheet

4. สร้าง Dockerfile.swp ไว้ใน Working directory

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ (Windows) บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

```
FROM busybox
```

```
CMD echo "Hi there. This is my first docker image."
```

```
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"
```

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

```
$ cat > Dockerfile << EOF
```

```
FROM busybox
```

```
CMD echo "Hi there. This is my first docker image."
```

```
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"
```

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

```
$ docker build -t <ชื่อ Image> .
```

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

[Check point#4] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 5 พร้อมกับตอบคำถามต่อไปนี้

Lab Worksheet

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Search

Name	Tag	Created	Size	Actions	Image ID
postgres	13	11 months ago	419.14 MB		f51f4345bd9d
docker/welcome-to-docker	latest	1 year ago	18.55 MB		c1f619b6477e
busybox	latest	4 months ago	4.26 MB		af4709625109
my_first_docker_image	latest	4 months ago	4.26 MB		4568b2ea847b

Showing 9 items

Terminal

```
View build details: docker-desktop://dashboard/build/desktop:linux/desktop:linux/aagp24uxu153klab99e6zap1m
PS D:\SE_2567_126-0\Lab8_2> docker build -t my_first_docker_image .
[+] Building 0.3s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 190B
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
```

Engine running | RAM 2.03 GB | CPU 0.13% | Disk: 12.85 GB used (limit 1006.85 GB) | Terminal v4.37.1

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
gifted_bardeen	ff9e8dcc7fbb	busybox		N/A	60 minutes ago	
agitated_solomon	7cebe244a97f	busybox		N/A	58 minutes ago	
tender_brown	762aa24d03ba	my_first_docker_image		N/A	22 seconds ago	
demo airflow	-	-	-	N/A	3 hours ago	

Showing 7 items

Terminal

```
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)

What's next:
View a summary of image vulnerabilities and recommendations --docker scout quickview
PS D:\SE_2567_126-0\Lab8_2> docker run my_first_docker_image
ขงจขจข 653380126-0 126
PS D:\SE_2567_126-0\Lab8_2> []
```

Engine running | RAM 2.04 GB | CPU 0.13% | Disk: 12.85 GB used (limit 1006.85 GB) | Terminal v4.37.1

(1) คำสั่งที่ใช้ในการ run คือ

ตอบ docker run my_first_docker_image

(2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

Lab Worksheet

ตอบ Option -t ในคำสั่ง docker build ใช้สำหรับกำหนดชื่อ image และ tag ให้กับ Docker image ที่สร้างขึ้น

แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
2. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_3
3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_3 เพื่อใช้เป็น Working directory
4. สร้าง Dockerfile.swp ไว้ใน Working directory

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

FROM busybox

CMD echo "Hi there. My work is done. You can run them from my Docker image."

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

\$ cat > Dockerfile << EOF

FROM busybox

CMD echo "Hi there. My work is done. You can run them from my Docker image."

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้
\$ docker build -t <username ที่ลงทะเบียนกับ Docker Hub>/lab8
6. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง
\$ docker run <username ที่ลงทะเบียนกับ Docker Hub>/lab8

Lab Worksheet

[Check point#5] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 5

The screenshot shows the Docker Desktop interface. The 'Images' tab is active, displaying a list of Docker images. Below the list, a terminal window is open, showing the output of a Docker build command.

Name	Tag	Created	Size	Actions	Image ID
docker/welcome-to-docker	latest	1 year ago	18.55 MB		c1f619b6477e
busybox	latest	4 months ago	4.26 MB		af4709625109
my_first_docker_image	latest	4 months ago	4.26 MB		4568b2ea847b
boat2356/lab8	latest	4 months ago	4.26 MB		5897c15aca10

Showing 10 items

```

Start a build
PS D:\SE_2567_126-0\Lab8_3> docker build -t boat2356/lab8 .
[+] Building 0.3s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
  
```

RAM 2.03 GB CPU 0.00% Disk: 12.85 GB used (limit 1006.85 GB)

The screenshot shows the Docker Desktop interface. The 'Containers' tab is active, displaying a list of running containers. Below the list, a terminal window is open, showing the output of a Docker run command.

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
agitated_solomon	70ebe244a97f	busybox		N/A	2 hours ago	
tender_brown	762aa24d03ba	my_first_docker_image		N/A	35 minutes ago	
optimistic_roentgen	07991df9814d	boat2356/lab8		N/A	1 minute ago	
demo airflow	-	-		N/A	3 hours ago	

Showing 8 items

```

Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SE_2567_126-0\Lab8_3> docker run boat2356/lab8
python3 main.py 653380126-0

D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
  
```

RAM 2.05 GB CPU 0.00% Disk: 12.85 GB used (limit 1006.85 GB)

7. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง
- ```
$ docker push <username ที่ลงทะเบียนกับ Docker Hub>/lab8
```



## Lab Worksheet

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

\$ docker login แล้วป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้คำสั่ง

\$ docker login -u <username> -p <password>

8. ไปที่ Docker Hub กด Tab ชื่อ Tags หรือไปที่ Repository ก็ได้

[Check point#6] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดง Repository ที่มี Docker image (<username>/lab8)

The screenshot shows the Docker Desktop application window. The top bar is blue with the Docker logo and 'PERSONAL' label. Below it is a search bar and a 'Ctrl+K' button. A purple banner at the top says 'More Docker. Easy Access. New Streamlined Plans. Learn more'. The main area is divided into a sidebar on the left with icons for Images, Containers, Volumes, and Settings. The 'Images' tab is selected, showing a list of images. Below the list is a terminal window showing the output of a 'docker build' command.

| Name                     | Tag    | Created      | Size     | Actions                          | Image ID     |
|--------------------------|--------|--------------|----------|----------------------------------|--------------|
| docker/welcome-to-docker | latest | 1 year ago   | 18.55 MB | [Pull] [Push] [Refresh] [Delete] | c1f619b6477e |
| busybox                  | latest | 4 months ago | 4.26 MB  | [Pull] [Push] [Refresh] [Delete] | af4709625109 |
| my_first_docker_image    | latest | 4 months ago | 4.26 MB  | [Pull] [Push] [Refresh] [Delete] | 4568b2ea847b |
| boat2356/lab8            | latest | 4 months ago | 4.26 MB  | [Pull] [Push] [Refresh] [Delete] | 5897c15aca10 |

Showing 10 items

```

PS D:\SE_2567_126-0\Lab8_3> docker build -t boat2356/lab8 .
[+] Building 0.3s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 202B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers

```

RAM 2.05 GB CPU 0.00% Disk: 12.85 GB used (limit 1006.85 GB)

## Lab Worksheet

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search  Only show running containers

|                          | Name                | Container ID | Image                 | Port(s) | CPU (%) | Last started   | Actions |
|--------------------------|---------------------|--------------|-----------------------|---------|---------|----------------|---------|
| <input type="checkbox"/> | agitated_solomon    | 7cebe244a97f | busybox               |         | N/A     | 2 hours ago    |         |
| <input type="checkbox"/> | tender_brown        | 762aa24d03ba | my_first_docker_image |         | N/A     | 44 minutes ago |         |
| <input type="checkbox"/> | optimistic_roentgen | 07991df9814d | boat2356/lab8         |         | N/A     | 10 minutes ago |         |
| <input type="checkbox"/> | demo airflow        | -            | -                     | -       | N/A     | 4 hours ago    |         |

Showing 8 items

**Terminal** + - x

```
(c) Microsoft Corporation. All rights reserved.

D:\SE_2567_126-0\Lab8_3>docker run boat2356/lab8
sha256:59654b796daad653380126-0

D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>docker push boat2356/lab8
Using default tag: latest
```

RAM 2.04 GB CPU 0.13% Disk: 12.85 GB used (limit 1006.85 GB)

Terminal v4.37.1

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search  Only show running containers

|                          | Name                | Container ID | Image                 | Port(s) | CPU (%) | Last started   | Actions |
|--------------------------|---------------------|--------------|-----------------------|---------|---------|----------------|---------|
| <input type="checkbox"/> | agitated_solomon    | 7cebe244a97f | busybox               |         | N/A     | 2 hours ago    |         |
| <input type="checkbox"/> | tender_brown        | 762aa24d03ba | my_first_docker_image |         | N/A     | 44 minutes ago |         |
| <input type="checkbox"/> | optimistic_roentgen | 07991df9814d | boat2356/lab8         |         | N/A     | 10 minutes ago |         |
| <input type="checkbox"/> | demo airflow        | -            | -                     | -       | N/A     | 4 hours ago    |         |

Showing 8 items

**Terminal** + - x

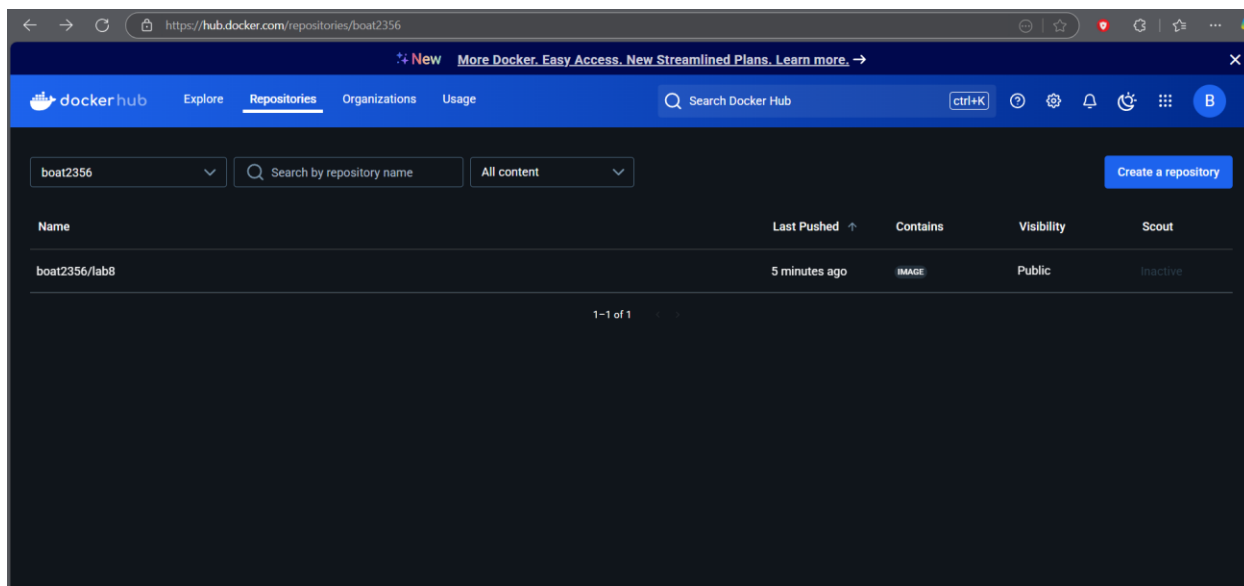
```
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>
D:\SE_2567_126-0\Lab8_3>docker push boat2356/lab8
Using default tag: latest
The push refers to repository [docker.io/boat2356/lab8]
59654b796daad: Mounted from library/busybox
latest: digest: sha256:ed209384bf19c9918bc849b1c91efa539034e462268f02a5da7aaf5121043dac size: 527

D:\SE_2567_126-0\Lab8_3>
```

RAM 2.04 GB CPU 0.25% Disk: 12.85 GB used (limit 1006.85 GB)

Terminal v4.37.1

## Lab Worksheet



#### แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_4
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก GitHub repository  
<https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง  
`$ git clone https://github.com/docker/getting-started.git`
3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

[Check point#7] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงที่อยู่ของ Source code ที่ Clone มาและเนื้อหาของไฟล์ package.json

## Lab Worksheet

```

1 {
2 "name": "101-app",
3 "version": "1.0.0",
4 "main": "index.js",
5 "license": "MIT",
6 "scripts": {
7 "prettify": "prettier -l --write \"**/*.js\"",
8 "test": "jest",
9 "dev": "nodemon src/index.js"
10 },
11 "dependencies": {
12 "101-app": "file:",
13 "express": "^4.18.2",
14 "mysql2": "^2.3.3",
15 "sqlite3": "^5.1.2",
16 "uuid": "^9.0.0",
17 "wait-port": "^1.0.4"
18 },
19 "resolutions": {
20 "ansi-regex": "5.0.1"
21 },
22 "prettier": {
23 "trailingComma": "all",
24 "tabWidth": 4,
25 "useTabs": false,

```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น

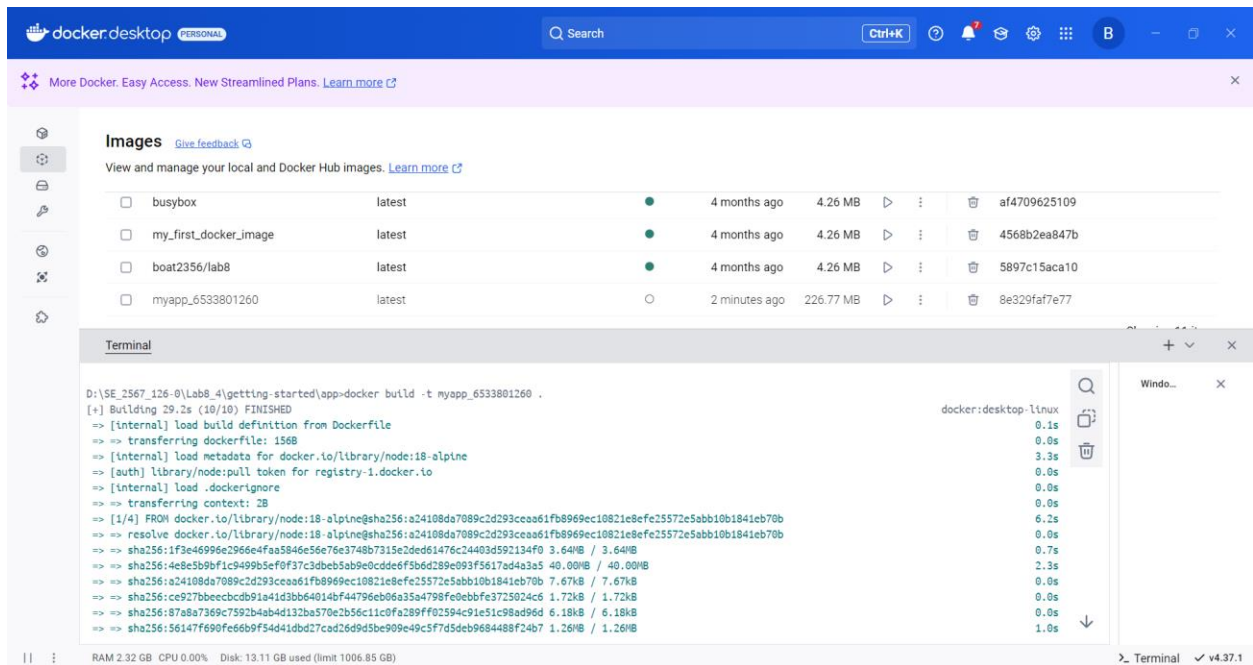
myapp\_รหัสสนศ. ไม่มีขีด

\$ docker build -t <myapp\_รหัสสนศ. ไม่มีขีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

## Lab Worksheet



6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

\$ docker run -dp 3000:3000 <myapp\_รหัสสนศ. ไม่มีขีด>

7. เปิด Browser ไปที่ URL = <http://localhost:3000>

[Check point#9] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

## Lab Worksheet

The screenshot shows the Docker Desktop interface. The top bar is blue with the Docker logo and 'PERSONAL' label. Below it, a purple banner says 'More Docker. Easy Access. New Streamlined Plans. Learn more'. The main area is divided into two sections: 'Containers' and 'Terminal'.

**Containers Section:**

View all your running containers and applications. [Learn more](#)

Search: [ ] Only show running containers

|                          | Name                | Container ID | Image                 | Port(s)   | CPU (%) | Last started   | Actions                |
|--------------------------|---------------------|--------------|-----------------------|-----------|---------|----------------|------------------------|
| <input type="checkbox"/> | tender_brown        | 762aa24d03ba | my_first_docker_image |           | 0%      | 1 hour ago     | [Play] [Stop] [Delete] |
| <input type="checkbox"/> | optimistic_roentgen | 07991df9814d | boat2356/lab8         |           | 0%      | 41 minutes ago | [Play] [Stop] [Delete] |
| <input type="checkbox"/> | suspicious_pare     | 8ad4667b8916 | myapp_6533801260      | 3000:3000 | 0%      | 50 seconds ago | [Play] [Stop] [Delete] |
| <input type="checkbox"/> | demo_aliflow        | -            | -                     | -         | 0%      | 4 hours ago    | [Play] [Stop] [Delete] |

Showing 9 items

**Terminal Section:**

View build details: `docker-desktop://dashboard/build/desktop:linux/desktop:linux/9e44t83h1ldn1svx1zt55gne9`

What's next:  
View a summary of image vulnerabilities and recommendations — [docker scout quickview](#)

```
D:\SE_2567_126-0\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801260
8ad4667b89165f15416c82a897b246d3f76d79bdec06f980c364bdd3b7ba60cb
D:\SE_2567_126-0\Lab8_4\getting-started\app>
```

RAM 2.37 GB CPU 0.13% Disk: 13.11 GB used (limit 1006.85 GB)

**Terminal Window:**

localhost:3000

New Item [ ] Add Item

No items yet! Add one above!

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้

a. เปิดไฟล์ `src/static/js/app.js` ด้วย Editor และแก้ไขบรรทัดที่ 56 จาก

## Lab Worksheet

<p className="text-center">No items yet! Add one above!</p> เป็น

<p className="text-center">There is no TODO item. Please add one to the list. By

ชื่อและนามสกุลของนักศึกษา</p>

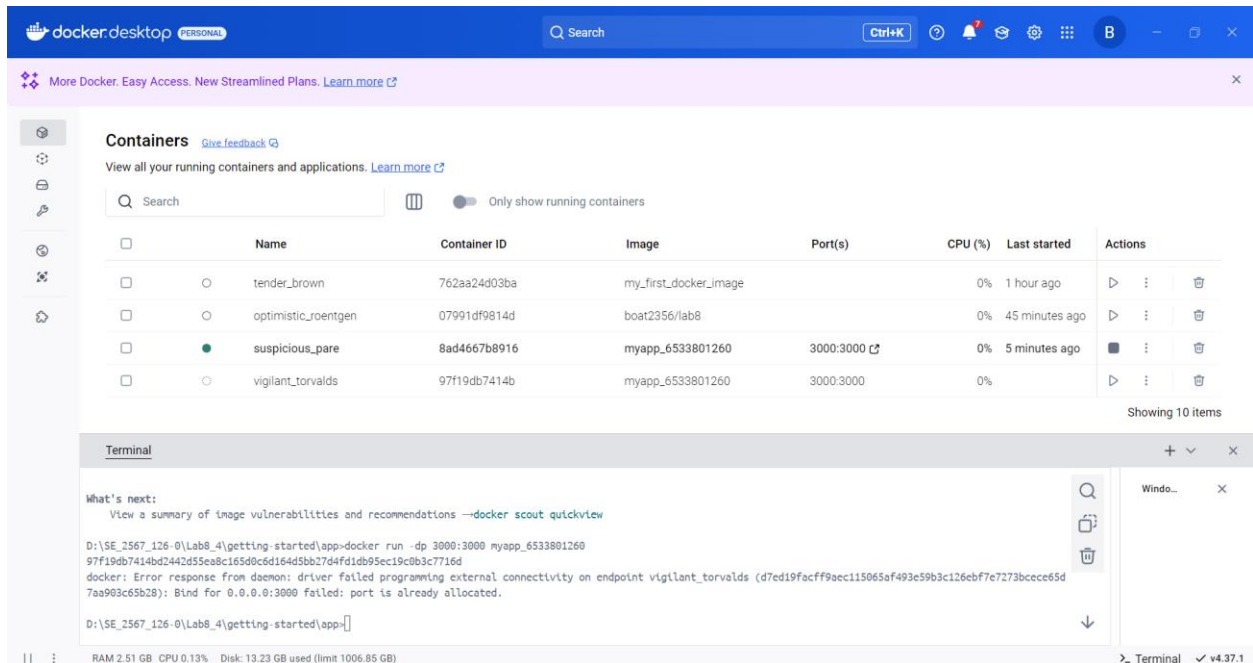
b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

[Check point#10] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้



(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

**ตอบ** Error นี้หมายถึง Docker ไม่สามารถผูกพอร์ต 3000 ของเครื่องโฮสต์ (host machine) กับคอนเทนเนอร์ได้ เพราะ พอร์ต 3000 บนเครื่องโฮสต์นั้นถูกใช้งานอยู่แล้ว จากโปรเซสหรือคอนเทนเนอร์อื่น ๆ ที่กำลังทำงานอยู่ สาเหตุที่เกิดขึ้นอาจจะเพราะมีคอนเทนเนอร์อื่นที่กำลังรันอยู่ใช้พอร์ต 3000 หรือไม่ก็ พอร์ต 3000 ถูกใช้งานโดยโปรแกรมอื่น

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

i. ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ

## Lab Worksheet

- ii. Copy หรือบันทึก Container ID ไว้
- iii. ใช้คำสั่ง `$ docker stop <Container ID ที่ต้องการจะลบ>` เพื่อหยุดการทำงานของ Container ดังกล่าว
- iv. ใช้คำสั่ง `$ docker rm <Container ID ที่ต้องการจะลบ>` เพื่อทำการลบ

## b. ผ่าน Docker desktop

- i. ไปที่หน้าต่าง Containers
- ii. เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- iii. ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

13. เปิด Browser ไปที่ URL = <http://localhost:3000>

[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้นบน Browser และ Dashboard ของ Docker desktop

The screenshot shows the Docker Desktop interface. At the top, there's a search bar and a 'Ctrl+K' shortcut. Below the header, there's a 'Containers' section with a search bar and a toggle for 'Only show running containers'. A table lists four containers:

| Name                | Container ID | Image                 | Port(s)   | CPU (%) | Last started   | Actions                |
|---------------------|--------------|-----------------------|-----------|---------|----------------|------------------------|
| tender_brown        | 762aa24d03ba | my_first_docker_image |           | 0%      | 1 hour ago     | [Play] [Stop] [Delete] |
| optimistic_roentgen | 07991df9814d | boat2356/lab8         |           | 0%      | 52 minutes ago | [Play] [Stop] [Delete] |
| quizzical_kilby     | 879ffce043e9 | myapp_6533801260      | 3000:3000 | 0%      | 2 minutes ago  | [Play] [Stop] [Delete] |
| devoid_alfred       | -            | -                     | -         | 0%      | 4 hours ago    | [Play] [Stop] [Delete] |

Below the table, there's a 'Terminal' section showing the command history:

```
D:\SE_2567_126-0\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801260
97f19db7414bd2442d55ea8c165d0c6d164d5bb27d4fd1db95ec19c0b3c7716d
docker: Error response from daemon: driver failed programming external connectivity on endpoint vigilant_torvalds (d7ed19facff9ec115065ef493e59b3c126ebf7e7273bcece65d7aa903c65b28): Bind for 0.0.0.0:3000 failed: port is already allocated.

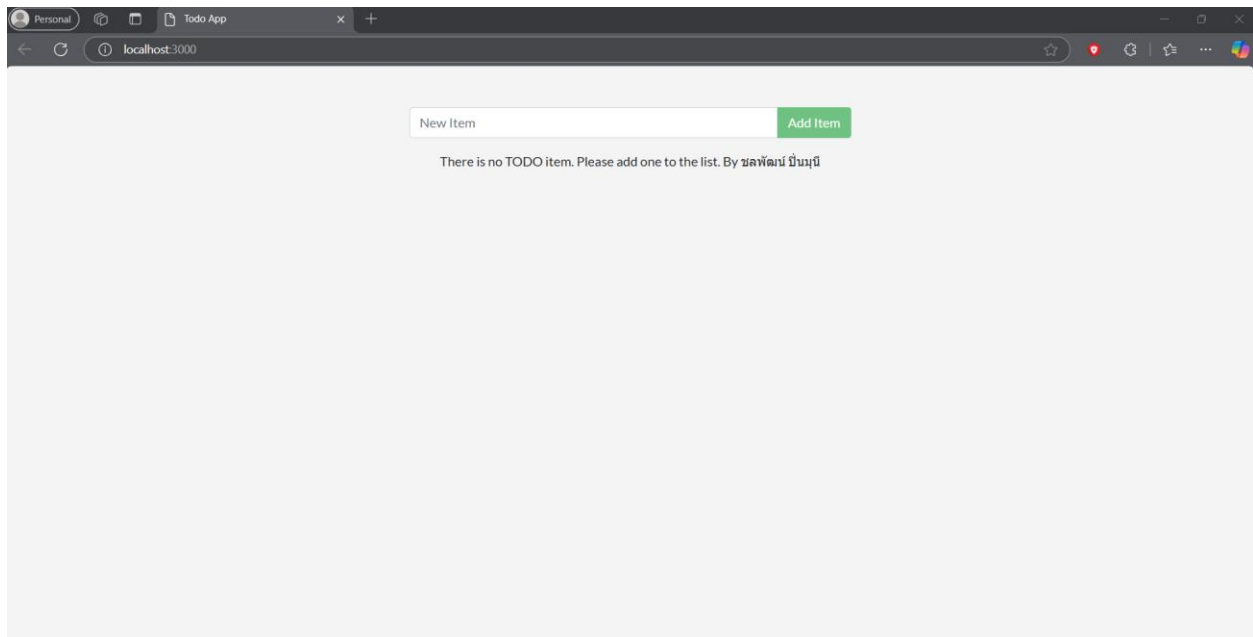
D:\SE_2567_126-0\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801260
879ffce043e91dc6e89e5386160c58eb159f2eaa78eff283a6899160e833d388

D:\SE_2567_126-0\Lab8_4\getting-started\app>
```

At the bottom, there's a status bar showing system resources: RAM 2.50 GB, CPU 0.13%, Disk 13.23 GB used (limit 1006.85 GB).



## Lab Worksheet



---

**แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins**

---

1. เปิด Command line หรือ Terminal บน Docker Desktop
2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต  

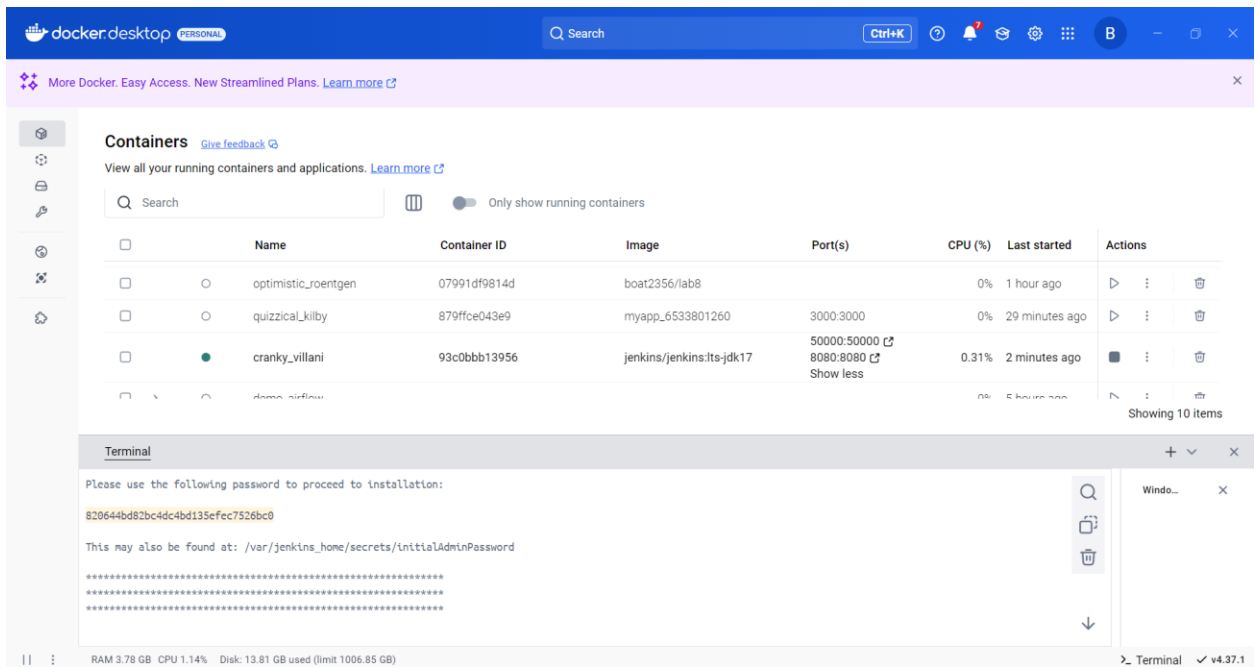
```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:its-jdk17
```

หรือ  

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v
jenkins_home:/var/jenkins_home jenkins/jenkins:its-jdk17
```
3. บันทึกการรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

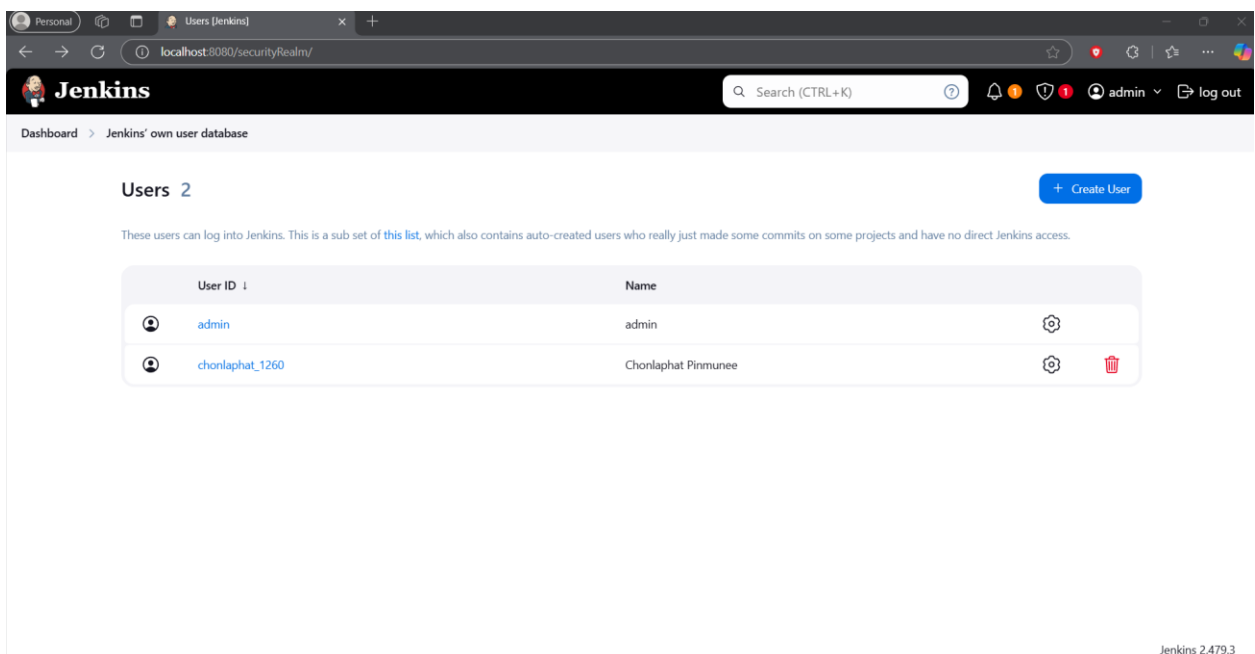
[Check point#12] Capture หน้าจอที่แสดงผล Admin password

## Lab Worksheet



4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062

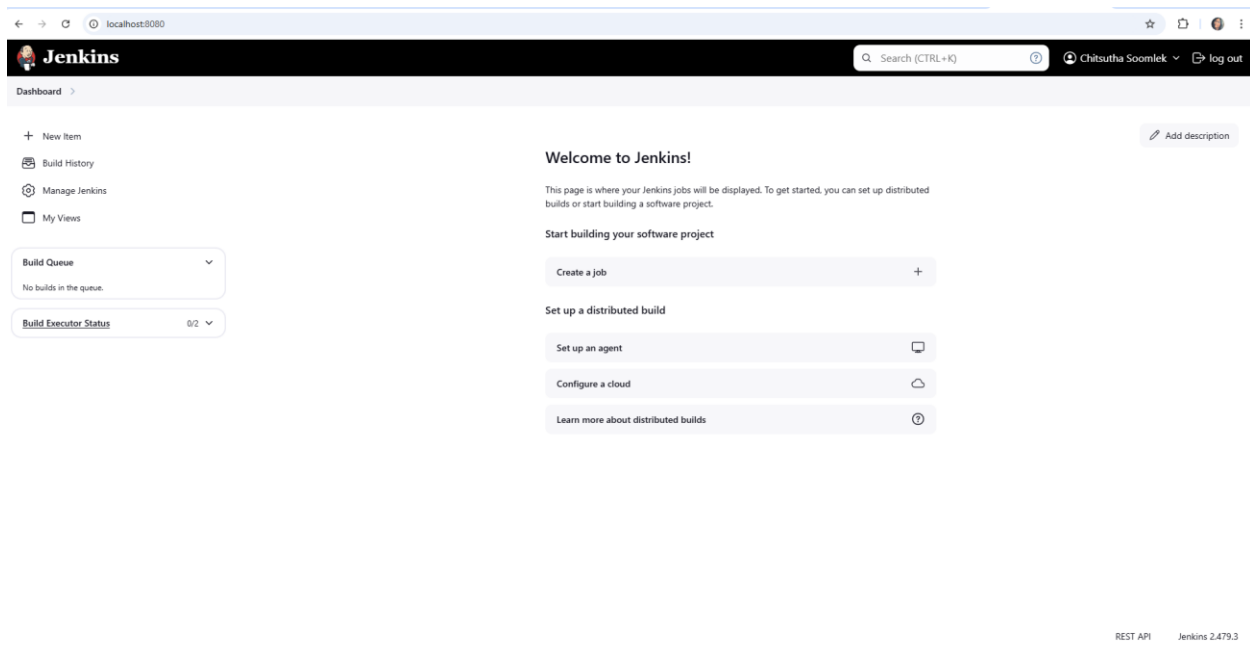
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



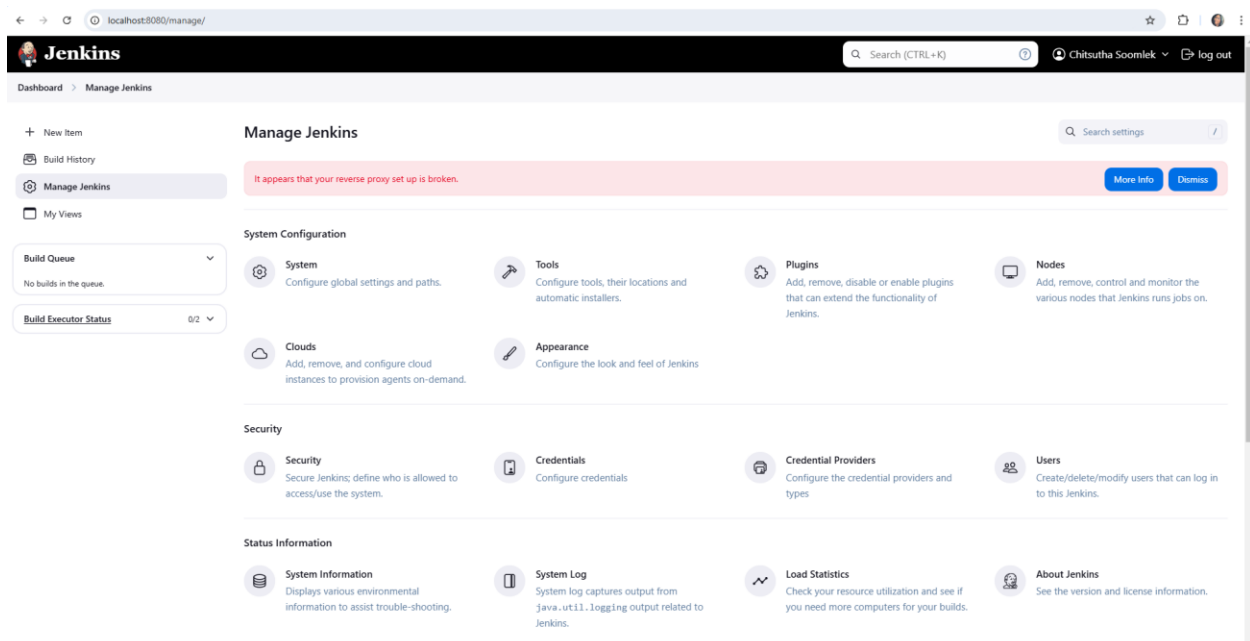
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

## Lab Worksheet

## 8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

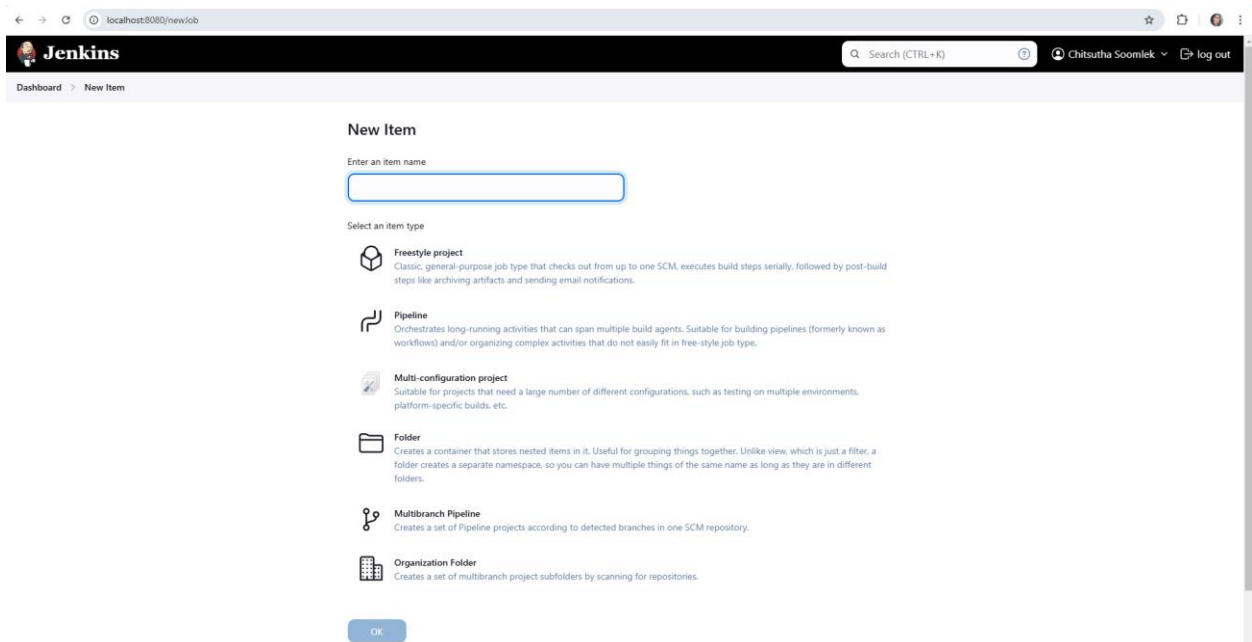


## Lab Worksheet

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



11. กลับไปที่หน้า Dashboard แล้วสร้าง Pipeline อย่างง่าย โดยกำหนด New item เป็น Freestyle project และตั้งชื่อเป็น UAT



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้นตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

Description: Lab 8.5

GitHub project: กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

## Lab Worksheet

**Build Trigger:** เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

**Build Steps:** เลือก Execute shell แล้วใส่คำสั่งในการรันไฟล์ .robot (หากไฟล์ไม่ได้อยู่ในหน้าแรกของ repository ให้ใส่ Path ไปถึงไฟล์ให้เรียบร้อยแล้ว)

**[Check point#14]** Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

(1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ

ตอบ

```
pip install --break-system-packages robotframework robotframework-seleniumlibrary
```

```
export PATH=$PATH:/var/jenkins_home/.local/bin
```

```
cd ./Lab\#7_653380126-0_sec1
```

```
chmod +x /var/jenkins_home/workspace/UAT/Lab#7_653380126-
```

```
0_sec1/ChromeForTesting/chromedriver.exe
```

```
robot form_tests/complete.robot
```

```
robot form_tests/incomplete.robot
```

**Post-build action:** เพิ่ม Publish Robot Framework test results ->

ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น

% ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น %

ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

13. กด Apply และ Save

14. สั่ง Build Now

**[Check point#15]** Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output

## Lab Worksheet

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

### Build Steps

**Execute shell** ?

Command

[See the list of available environment variables](#)

```

pip install --break-system-packages robotframework robotframework-seleniumlibrary
export PATH=$PATH:/var/jenkins_home/.local/bin
cd ./Lab/#7_653380126-0_sec1
chmod +x /var/jenkins_home/workspace/UAT/Lab/#7_653380126-0_sec1/ChromeForTesting/chromedriver.exe
robot form_tests/complete.robot
robot form_tests/incomplete.robot

```

Advanced ▾

Add build step ▾

### Post-build Actions

Save Apply

---

Dashboard > UAT > #22 > Console Output

Status

</> Changes

**Console Output**

Edit Build Information

Delete build '#22'

Timings

Git Build Data

Robot Results

Previous Build

### Console Output

Download Copy View as plain text

```

Started by user Chonlaphat Pimmunee
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/UAT/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Boat2356/Lab-7_Test-Automation_653380126-0_sec1 # timeout=10
Fetching upstream changes from https://github.com/Boat2356/Lab-7_Test-Automation_653380126-0_sec1
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/Boat2356/Lab-7_Test-Automation_653380126-0_sec1
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse origin/main^(commit) # timeout=10
Checking out Revision f0cc3bd7ee1a546e1beb226985b63f9339cf8d88 (origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f f0cc3bd7ee1a546e1beb226985b63f9339cf8d88 # timeout=10
Commit message: "Add Task Lab#7 to repository"
> git rev-list --no-walk f0cc3bd7ee1a546e1beb226985b63f9339cf8d88 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins8883960744778332810.sh
+ pip install --break-system-packages robotframework robotframework-seleniumlibrary
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: robotframework in /var/jenkins_home/.local/lib/python3.11/site-packages (7.2)
Requirement already satisfied: robotframework-seleniumlibrary in /var/jenkins_home/.local/lib/python3.11/site-packages (6.7.0)
Requirement already satisfied: selenium>4.3.0 in /var/jenkins_home/.local/lib/python3.11/site-packages (from robotframework-seleniumlibrary)

```

## Lab Worksheet

```

Dashboard > UAT > #22 > Console Output

Requirement already satisfied: selenium in /var/jenkins_home/.local/lib/python3.11/site-packages (from trio==0.17->selenium==4.3.0->robotframework-seleniumlibrary) (2.4.0)
Requirement already satisfied: idna in /var/jenkins_home/.local/lib/python3.11/site-packages (from trio==0.17->selenium==4.3.0->robotframework-seleniumlibrary) (3.10)
Requirement already satisfied: outcome in /var/jenkins_home/.local/lib/python3.11/site-packages (from trio==0.17->selenium==4.3.0->robotframework-seleniumlibrary) (1.3.0.post0)
Requirement already satisfied: sniffio==1.3.0 in /var/jenkins_home/.local/lib/python3.11/site-packages (from trio==0.17->selenium==4.3.0->robotframework-seleniumlibrary) (1.3.1)
Requirement already satisfied: wsproto==0.14 in /var/jenkins_home/.local/lib/python3.11/site-packages (from trio==0.17->selenium==4.3.0->robotframework-seleniumlibrary) (1.2.0)
Requirement already satisfied: pysocks==1.5.7, <2.0, >=1.5.6 in /var/jenkins_home/.local/lib/python3.11/site-packages (from urllib3[socks]<3, >=1.26->selenium==4.3.0->robotframework-seleniumlibrary) (1.7.1)
Requirement already satisfied: h11<1, >=0.9.0 in /var/jenkins_home/.local/lib/python3.11/site-packages (from wsproto==0.14->trio-websocket==0.9->selenium==4.3.0->robotframework-seleniumlibrary) (0.14.0)
+ export PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/jenkins_home/.local/bin
+ cd ./Lab#7_653380126-0_sec1
+ chmod +x /var/jenkins_home/workspace/UAT/Lab#7_653380126-0_sec1/ChromeForTesting/chromedriver.exe
+ robot form_tests/complete.robot

=====
Complete
=====
Open Form | FAIL |
WebDriverException: Message: Service /var/jenkins_home/workspace/UAT/Lab#7_653380126-0_sec1/ChromeForTesting/chromedriver.exe unexpectedly
exited. Status code was: 1

Record Success | FAIL |
No browser is open.

Complete | FAIL |
2 tests, 0 passed, 2 failed

```

```

Dashboard > UAT > #22 > Console Output

exited. Status code was: 1

Record Success | FAIL |
No browser is open.

Complete | FAIL |
2 tests, 0 passed, 2 failed
=====
Output: /var/jenkins_home/workspace/UAT/Lab#7_653380126-0_sec1/output.xml
Log: /var/jenkins_home/workspace/UAT/Lab#7_653380126-0_sec1/log.html
Report: /var/jenkins_home/workspace/UAT/Lab#7_653380126-0_sec1/report.html
Build step 'Execute shell' marked build as failure
Robot results publisher started...
INFO: Checking test criticality is deprecated and will be dropped in a future release!
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE

REST API Jenkins 2.479.3

```