

**Lab Worksheet**

ชื่อ-นามสกุล นางสาวสิริยากร เอกศิริ รหัสนักศึกษา 643020650-9 Section 2

**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. ผู้เรียนสามารถนำซอฟต์แวร์ที่พัฒนาขึ้นให้สามารถรันบนสภาพแวดล้อมเดียวกันและทำงานร่วมกันกับสมาร์ทโฟนที่ติดต่อผ่าน 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

**[Check point#1]** Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้พร้อมกับตอบคำถามต่อไปนี้

## Lab Worksheet

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr ~ % mkdir Lab8_1
mkdir: Lab8_1: File exists
nnoeysr@MacBook-Pro-nnoeysr ~ % cd Lab8_1
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
559c60843878: Download complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
busybox        latest    a5d0ce49aa80   4 months ago   6.02MB
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 %
```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

**ชื่อของอิมเมจ (Image Name) ที่ถูกดึงมาจาก Docker Hub หรือสร้างขึ้นเอง เช่น busybox**

(2) Tag ที่ใช้บ่งบอกถึงอะไร

**Tag ใช้บ่งบอกเวอร์ชันของอิมเมจ เช่น latest หมายถึงเวอร์ชันล่าสุดที่ถูกดึงมา หากไม่มีการระบุ เวอร์ชันตอนดึงอิมเมจ Docker จะใช้ latest เป็นค่าเริ่มต้น**

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

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

## Lab Worksheet

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	○ jolly_sanderson	bc2fc3a9690a	busybox		N/A	5 minutes ago	▷ ⚙️ 🗑️
□	○ ecstatic_jang	37942c8db2cc	busybox		N/A	5 minutes ago	▷ ⚙️ 🗑️
□	○ tender_spence	e1e37035a990	busybox		N/A	3 minutes ago	▷ ⚙️ 🗑️

Terminal	
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker run busybox	x
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker run -it busybox sh	zsh
/ # ls	🔍
bin home proc tmp	🗂️
dev lib root usr	🗑️
etc lib64 sys var	⬇️
/ # ls -la	⬇️
total 48	
drwxr-xr-x 1 root root 4096 Jan 28 07:48 .	
drwxr-xr-x 1 root root 4096 Jan 28 07:48 ..	
-rwxr-xr-x 1 root root 0 Jan 28 07:48 .dockerenv	

Terminal	
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin	🔍
drwxr-xr-x 5 root root 360 Jan 28 07:48 dev	🗂️
drwxr-xr-x 1 root root 4096 Jan 28 07:48 etc	🗑️
drwxr-xr-x 2 nobody nobody 4096 Sep 26 21:31 home	⬇️
drwxr-xr-x 2 root root 4096 Sep 26 21:31 lib	⬇️
lrwxrwxrwx 1 root root 3 Sep 26 21:31 lib64 -> lib	⬇️
dr-xr-xr-x 219 root root 0 Jan 28 07:48 proc	⬇️
drwx----- 1 root root 4096 Jan 28 07:48 root	⬇️
dr-xr-xr-x 11 root root 0 Jan 28 07:48 sys	⬇️
drwxrwxrwt 2 root root 4096 Sep 26 21:31 tmp	⬇️
drwxr-xr-x 4 root root 4096 Sep 26 21:31 usr	⬇️
drwxr-xr-x 4 root root 4096 Sep 26 21:31 var	⬇️
/ # exit	docker run busybox echo "Hello =====usybox==== Hell

Terminal						
/ # exit	docker run busybox echo "Hello =====usybox==== Hell					
o =====<<<81>>>===== from busybox" from busybox"	🔍					
Hello ลิริยากร เอกธิริ from busybox	🗂️					
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker ps -a	🗑️					
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e1e37035a990	busybox	"echo 'Hello ลิริยากร..."	About a minute ago	Exited (0) About a minute ago		tender
tender_spence						
37942c8db2cc	busybox	"sh"	3 minutes ago	Exited (0) 2 minutes ago		ecstat
ecstatic_jang						
bc2fc3a9690a	busybox	"sh"	3 minutes ago	Exited (0) 3 minutes ago		jolly_sanderson
jolly_sanderson						

Terminal						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e1e37035a990	busybox	"echo 'Hello ลิริยากร..."	17 minutes ago	Exited (0) 5 minutes ago		tender
_spence						
37942c8db2cc	busybox	"sh"	19 minutes ago	Exited (0) 7 minutes ago		ecstat
ic_jang						
bc2fc3a9690a	busybox	"sh"	19 minutes ago	Exited (0) 8 minutes ago		jolly_
jolly_sanderson						
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % 37942c8db2cc						
zsh: command not found: 37942c8db2cc						
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker rm 37942c8db2cc						
37942c8db2cc						
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 %						
RAM 0.60 GB CPU 0.00% Disk: 1.01 GB used (limit 1006.85 GB)						iPhone M

## Lab Worksheet

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสั้นๆ เช่น

Option -it ทำให้ค่อนเทนเนอร์รันในโหมดโต้ตอบ (Interactive Mode) :

- -i (interactive) → เปิดใช้งานอินพุตแบบโต้ตอบ (stdin) ทำให้สามารถป้อนคำสั่งในค่อนเทนเนอร์ได้
- -t (tty) → จำลอง terminal (TTY) เพื่อให้สามารถพิมพ์คำสั่งได้เหมือนอยู่ใน shell ปกติ

เมื่อนำ -it ไปใช้กับ docker run busybox sh

จะทำให้สามารถเข้าสู่ shell (sh) ของค่อนเทนเนอร์ และสามารถพิมพ์คำสั่งภายในค่อนเทนเนอร์ได้

- ถ้าไม่ใช้ -it ค่อนเทนเนอร์จะรันเสร็จแล้วปิดตัวลงทันที

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a และดูถึงข้อมูลอะไร

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
NAMES				
1a2b3c4d5e6f	busybox	"echo Hello..."	5 minutes ago	Exited (0) 2 minutes ago
hopeful_mestorf				
7g8h9i0j1k2l	busybox	"sh"	10 minutes ago	Exited (0) 8 minutes ago
sleepy_goldberg				
แสดงสถานะของค่อนเทนเนอร์ เช่น กำลังทำงาน (Up), หยุดทำงาน (Exited), หรือถูกสร้างแต่ยังไม่รัน (Created)				

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

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

## Lab Worksheet

Container CPU usage ⓘ  
No containers are running.

Container memory usage ⓘ  
No containers are running.

Show charts

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> jolly_sanderson	bc2fc3a9690a	busybox		N/A	28 minutes ago	<span>▷</span> <span>⋮</span> <span>🗑️</span>
<input type="checkbox"/>	<input type="radio"/> tender_spence	e1e37035a990	busybox		N/A	26 minutes ago	<span>▷</span> <span>⋮</span> <span>🗑️</span>

**Terminal**

```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e1e37035a990 busybox "echo 'Hello จิ้งจก.'" 17 minutes ago Exited (0) 5 minutes ago
37942c8db2cc busybox "sh" 19 minutes ago Exited (0) 7 minutes ago
lc_jang bc2fc3a9690a busybox "sh" 19 minutes ago Exited (0) 8 minutes ago
jolly_sanderson
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % 37942c8db2cc
zsh: command not found: 37942c8db2cc
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 % docker rm 37942c8db2cc
37942c8db2cc
nnoeysr@MacBook-Pro-nnoeysr Lab8_1 %

RAM 0.59 GB CPU 0.00% Disk: 1.01 GB used (limit 1006.85 GB)

```

zsh Terminal v4.37.2

## 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
4. สร้าง Dockerfile.swp ไว้ใน Working directory  
สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ (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

## Lab Worksheet

\$ 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

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr ~ % mkdir Lab8_2
mkdir: Lab8_2: File exists
nnoeysr@MacBook-Pro-nnoeysr ~ % cd Lab8_2
CMD echo "Hi there. This is my first docker image.">Lab8_2 %
CMD echo "there is my first image in ."
EOF
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % ls -l
total 8
-rw-r--r-- 1 nnoeysr staff 249 Jan 28 15:29 Dockerfile
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % docker build -t my-first-image .
[+] Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/0)
docker:desktop-li[+] Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/1) docker:desktop-li[+] Bu
ilding 0.2s (3/4) docker:desktop-li[+] Building 0.3s (3/4) docker:desktop-li[+] Building 0.5s (3/4) docke
r:desktop-li[+] Building 0.6s (3/4) docker:desktop-li[+] Building 0.8s (3/4) docker:desktop-li[+] Building
0.9s (3/4) docker:desktop-li[+] Building 1.1s (3/4) docker:desktop-li[+] Building 1.2s (3/4) docker:des
ktop-li[+] Building 1.3s (3/5) docker:desktop-li[+] Building 1.4s (4/5) docker:desktop-li[+] Building 1.6s
```

Terminal

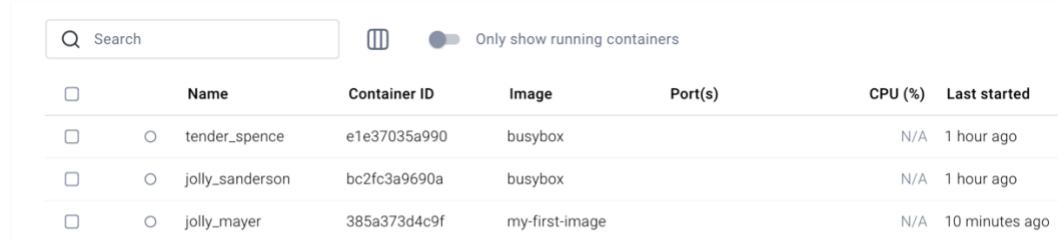
```
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % nano Dockerfile
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % docker build -t my-first-image .

[+] Building 0.0s (0/1) docker:desktop-linux[+] Building 0.1s (5/5) FINISHED docker:desktop-linux
=> [internal] load build definition 0.0s
=> => transferring dockerfile 224B 0.0s
=> [internal] load metadata for docker 0.0s
=> [internal] load .dockerrcignore 0.0s
=> => transferring context: 2B 0.0s
=> CACHED [1/1] FROM docker.io/library/buster 0.0s
=> => resolve docker.io/library/buster 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => exporting manifest sha256:46 0.0s
=> => exporting config sha256:f2a1 0.0s
=> => exporting attestation manifest 0.0s
=> => exporting manifest list sha2 0.0s
=> => naming to docker.io/library/buster 0.0s
=> => unpacking to docker.io/library/buster 0.0s
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % docker run my-first-image
```

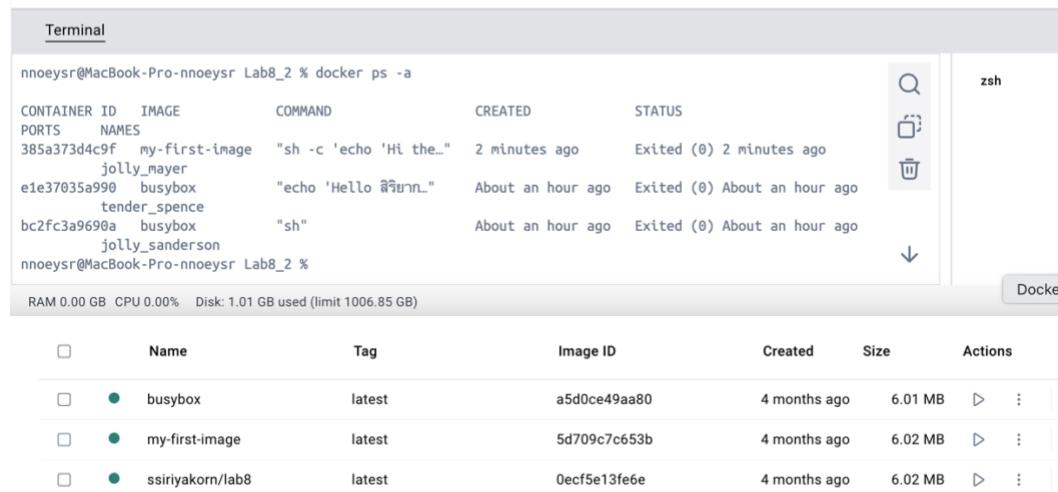
## Lab Worksheet

```
-->-- unpacking to docker . / lib / v . v . v
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % docker run my-first-image
```

Hi there. This is my first docker image.  
 Name: siriyakorn eaksiri ID: 643020650-9 Nickname: noey  
 nnoeysr@MacBook-Pro-nnoeysr Lab8\_2 %



<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	tender_spence	e1e37035a990	busybox		N/A	1 hour ago
<input type="checkbox"/>	jolly_sanderson	bc2fc3a9690a	busybox		N/A	1 hour ago
<input type="checkbox"/>	jolly_mayer	385a373d4c9f	my-first-image		N/A	10 minutes ago



Terminal

```
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
385a373d4c9f my-first-image "sh -c 'echo 'Hi the..." 2 minutes ago Exited (0) 2 minutes ago
jolly_mayer
e1e37035a990 busybox "echo 'Hello วิจิยา...' About an hour ago Exited (0) About an hour ago
tender_spence
bc2fc3a9690a busybox "sh" About an hour ago Exited (0) About an hour ago
jolly_sanderson
nnoeysr@MacBook-Pro-nnoeysr Lab8_2 %

RAM 0.00 GB CPU 0.00% Disk: 1.01 GB used (limit 1006.85 GB)
```

Docker

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	busybox	latest	a5d0ce49aa80	4 months ago	6.01 MB	  
<input type="checkbox"/>	my-first-image	latest	5d709c7c653b	4 months ago	6.02 MB	  
<input type="checkbox"/>	ssiriyakorn/lab8	latest	0ecf5e13fe6e	4 months ago	6.02 MB	  

(1) คำสั่งที่ใช้ในการ run คือ

คำสั่ง docker run ใช้เพื่อรัน Docker container จาก Docker image ที่ได้สร้างขึ้นในที่นี้ my-first-image คือชื่อของ Docker image ที่ได้ทำการ build ขึ้นมา

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

## Lab Worksheet

OPTION -T ในคำสั่ง DOCKER BUILD ใช้สำหรับการ ตั้งชื่อ (TAG) ให้กับ DOCKER IMAGE ที่สร้างขึ้นจาก DOCKERFILE. OPTION -T ช่วยให้การอ้างอิงและการใช้ DOCKER IMAGE ง่ายขึ้นในการเรียกใช้และจัดการกับหลายๆ IMAGE ที่อาจมีอยู่ในเครื่อง

## 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
- สำหรับเครื่องที่ใช้ระบบปฏิบัติการ Mac OS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้
- ```
FROM busybox
CMD echo "Hi there. My work is done. You can run them from my Docker image."
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"
```

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ Mac OS 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
```

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้
 

```
$ docker build -t <username> ที่ลงทะเบียนกับ Docker Hub>/lab8
```
5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง
 

```
$ docker run <username> ที่ลงทะเบียนกับ Docker Hub>/lab8
```

## Lab Worksheet

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

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr ~ % mkdir Lab8_3
nnoeysr@MacBook-Pro-nnoeysr ~ % cd Lab8_3
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % cat > Dockerfile << EOF
FROM busybox
CMD echo "Hi there. My work is done. You can run them from my Docker image."
CMD echo "siriayakorn eaksiri 643020650-9"
EOF
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % cat Dockerfile
FROM busybox
CMD echo "Hi there. My work is done. You can run them from my Docker image."
CMD echo "siriayakorn eaksiri 643020650-9"
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 %
```

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % docker build -t ssiriayakorn/lab8 .
[+] Building 0.0s (0/0) docker:desktop-l[+] Building 0.0s (0/1) docker:desktop-l[+] Building 0.1s (5/5) FINISHED docker:desktop-linux
=> [internal] load build definition 0.0s
=> => transferring dockerfile 225B 0.0s
=> [internal] load metadata for d 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> CACHED [1/1] FROM docker.io/library/b 0.0s
=> => resolve docker.io/library/b 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => exporting manifest sha256:4 0.0s
=> => exporting config sha256:8ec 0.0s
=> => exporting attestation manif 0.0s
=> => exporting manifest list sha 0.0s
=> => naming to docker.io/ssiriya 0.0s
=> => unpacking to docker.io/ssir 0.0s
=> => exporting manifest list sha 0.0s
=> => naming to docker.io/ssiriya 0.0s
=> => unpacking to docker.io/ssir 0.0s
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % docker run ssiriayakorn/lab8
Hi there. My work is done. You can run them from my Docker image.
siriayakorn eaksiri 643020650-9
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % docker run ssiriayakorn/lab8
```

6. ทำการ Push ตัว Docker image ไปเว็บน Docker Hub โดยการใช้คำสั่ง

`$ docker push <username> ที่ลงทะเบียนกับ Docker Hub>/lab8`

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

## Lab Worksheet

\$ docker login และป้อน Username และ Password ตามที่ระบุใน Command prompt

หรือใช้คำสั่ง

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

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

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

```
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 % docker push ssiriyakorn/lab8
Using default tag: latest
The push refers to repository [docker.io/ssiriyakorn/lab8]
559c60843878: Mounted from library/busybox
20fb8146a9bb: Pushed
Latest: digest: sha256:0ecf5e13fe6e3b932ba387b08c0e9da75eb95e8ba4ac16a52ced05dda0642838 size: 855
nnoeysr@MacBook-Pro-nnoeysr Lab8_3 %
```

[ssiriyakorn/lab8](#)

Last pushed 9 minutes ago

Add a description INCOMPLETE

Add a category INCOMPLETE

### Tags

This repository contains 1 tag(s).

| Tag    | OS | Type  | Pulled        | Pushed        |
|--------|----|-------|---------------|---------------|
| latest |    | Image | 9 minutes ago | 9 minutes ago |

[See all](#)

### 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 ซอฟต์แวร์ของเว็บแอปพลิเคชันจาก GitHub repository

<https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง

\$ git clone https://github.com/docker/getting-started.git

## Lab Worksheet

3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพับไฟล์ package.json ให้ใช้ Text editor  
ในการเปิดอ่าน

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

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr ~ % mkdir Lab8_4
nnoeysr@MacBook-Pro-nnoeysr ~ % cd Lab8_4
nnoeysr@MacBook-Pro-nnoeysr Lab8_4 % git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 980, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
Receiving objects: 26% (255/980), 1.07Receiving objects: 27% (265/980), 1.07Receiving objects: 28% (275/980), 1.07Receiving objects: 29% (285/980), 1.07Receiving objects: 30% (294/980), 1.07Receiving objects: 30% (302/980), 1.07Receiving objects: 31% (304/980), 2.27Receiving objects: 32% (314/980), 2.27Receiving objects: 33% (324/980), 2.27Receiving objects: 34% (334/980), 2.27Receiving objects: 35% (343/980), 2.27Receiving objects: 36% (353/980), 2.27Receiving objects: 37% (363/980), 2.27Receiving objects: 38% (373/980), 2.27Receiving objects: 39% (383/980), 2.27Receiving objects: 40% (392/980), 2.27Receiving objects: 41% (402/980), 2.27Receiving objects: 42% (412/980), 2.27Receiving objects: 43% (422/980), 2.27Receiving objects: 44% (432/980), 3.39Receiving objects: 45% (441/980), 3.39Receiving objects: 45% (443/980), 3.39Receiving objects: 46% (451/980), 3.39Receiving objects: 47% (461/980), 3.39Receiving objects: 48% (471/980), 3.39Receiving objects: 49% (481/980), 3.39Receiving objects: 50% (490/980), 3.39Receiving objects: 51% (500/980), 3.39Receiving objects: 52% (510/980), 3.39Receiving objects: 53% (520/980), 3.39Receiving objects: 54% (530/980), 3.39Receiving objects: 55% (540/980), 3.39Receiving objects: 56% (550/980), 3.39Receiving objects: 57% (560/980), 3.39Receiving objects: 58% (570/980), 3.39Receiving objects: 59% (580/980), 3.39Receiving objects: 60% (590/980), 3.39Receiving objects: 61% (600/980), 3.39Receiving objects: 62% (610/980), 3.39Receiving objects: 63% (620/980), 3.39Receiving objects: 64% (630/980), 3.39Receiving objects: 65% (640/980), 3.39Receiving objects: 66% (650/980), 3.39Receiving objects: 67% (660/980), 3.39Receiving objects: 68% (670/980), 3.39Receiving objects: 69% (680/980), 3.39Receiving objects: 70% (690/980), 3.39Receiving objects: 71% (700/980), 3.39Receiving objects: 72% (710/980), 3.39Receiving objects: 73% (720/980), 3.39Receiving objects: 74% (730/980), 3.39Receiving objects: 75% (740/980), 3.39Receiving objects: 76% (750/980), 3.39Receiving objects: 77% (760/980), 3.39Receiving objects: 78% (770/980), 3.39Receiving objects: 79% (780/980), 3.39Receiving objects: 80% (790/980), 3.39Receiving objects: 81% (800/980), 3.39Receiving objects: 82% (810/980), 3.39Receiving objects: 83% (820/980), 3.39Receiving objects: 84% (830/980), 3.39Receiving objects: 85% (840/980), 3.39Receiving objects: 86% (850/980), 3.39Receiving objects: 87% (860/980), 3.39Receiving objects: 88% (870/980), 3.39Receiving objects: 89% (880/980), 3.39Receiving objects: 90% (890/980), 3.39Receiving objects: 91% (900/980), 3.39Receiving objects: 92% (910/980), 3.39Receiving objects: 93% (920/980), 3.39Receiving objects: 94% (930/980), 3.39Receiving objects: 95% (940/980), 3.39Receiving objects: 96% (950/980), 3.39Receiving objects: 97% (960/980), 3.39Receiving objects: 98% (970/980), 3.39Receiving objects: 99% (980/980), 4.60remote: Total 980 (delta 5), reused 1 (delta 1), pack-reused 971 (from 2)
Receiving objects: 100% (980/980), 4.60Receiving objects: 100% (980/980), 5.28 MiB | 2.28 MiB/s, done.
Resolving deltas: 100% (523/523), done.
nnoeysr@MacBook-Pro-nnoeysr Lab8_4 % cd getting-started
nnoeysr@MacBook-Pro-nnoeysr getting-started % ls -l app
total 296
-rw-r--r--@ 1 nnoeysr staff      645 Jan 29 18:23 package.json
drwxr-xr-x@ 4 nnoeysr staff      128 Jan 29 18:23 spec
drwxr-xr-x@ 6 nnoeysr staff      192 Jan 29 18:23 src
-rw-r--r--@ 1 nnoeysr staff  147266 Jan 29 18:23 yarn.lock
nnoeysr@MacBook-Pro-nnoeysr getting-started % nano app/package.json
nnoeysr@MacBook-Pro-nnoeysr getting-started %
```

Terminal

```
.60Receiving objects: 85% (833/980), 4.60Receiving objects: 86% (843/980), 4.60Receiving objects: 87% (853/980), 4.60Receiving objects: 88% (863/980), 4.60Receiving objects: 89% (873/980), 4.60Receiving objects: 90% (882/980), 4.60Receiving objects: 91% (892/980), 4.60Receiving objects: 92% (902/980), 4.60Receiving objects: 93% (912/980), 4.60Receiving objects: 94% (922/980), 4.60Receiving objects: 95% (931/980), 4.60Receiving objects: 96% (941/980), 4.60Receiving objects: 97% (951/980), 4.60Receiving objects: 98% (961/980), 4.60Receiving objects: 99% (971/980), 4.60remote: Total 980 (delta 5), reused 1 (delta 1), pack-reused 971 (from 2)
Receiving objects: 100% (980/980), 4.60Receiving objects: 100% (980/980), 5.28 MiB | 2.28 MiB/s, done.
Resolving deltas: 100% (523/523), done.
nnoeysr@MacBook-Pro-nnoeysr Lab8_4 % cd getting-started
nnoeysr@MacBook-Pro-nnoeysr getting-started % ls -l app
total 296
-rw-r--r--@ 1 nnoeysr staff      645 Jan 29 18:23 package.json
drwxr-xr-x@ 4 nnoeysr staff      128 Jan 29 18:23 spec
drwxr-xr-x@ 6 nnoeysr staff      192 Jan 29 18:23 src
-rw-r--r--@ 1 nnoeysr staff  147266 Jan 29 18:23 yarn.lock
nnoeysr@MacBook-Pro-nnoeysr getting-started % nano app/package.json
nnoeysr@MacBook-Pro-nnoeysr getting-started %
```

## Lab Worksheet

```
{
  "name": "101-app",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "prettify": "prettier -l --write \"**/*.js\"",
    "test": "jest",
    "dev": "nodemon src/index.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "mysql2": "^2.3.3",
    "sqlite3": "5.1.2",
    "uuid": "^9.0.0",
    "wait-port": "^1.0.4"
  }
}
```

File: app/package.json

Terminal menu icons: Search, Copy, Delete.

Keyboard shortcuts:

- ^G Get Help
- ^X Exit
- ^O WriteOut
- ^J Justify
- ^R Read File
- ^W Where is
- ^Y Prev Pg
- ^V Next Pg
- ^K Cut Text
- ^U UnCut Text
- ^C Cur Pos
- ^T To Spell

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไว้ในไฟล์

```
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\_รหัสนศ. ไม่มีจีด

\$ docker build -t <myapp\_รหัสนศ. ไม่มีจีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

## Lab Worksheet

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr getting-started %
nnoeysr@MacBook-Pro-nnoeysr getting-started % cd getting-started/app

cd: no such file or directory: getting-started/app
nnoeysr@MacBook-Pro-nnoeysr getting-started % touch Dockerfile

nnoeysr@MacBook-Pro-nnoeysr getting-started % nano Dockerfile
```

Terminal

**UW PICO 5.09** File: Dockerfile

```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "./src/index.js"]
EXPOSE 3000
```

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr app % docker build -t myapp_6430206509 .

[+] Building 10.1s (8/9) docker:desktop-linu[+] Building 10.2s (8/9) docker:desktop-linu[+] Building 10.4s
(8/9) docker:desktop-linu[+] Building 10.5s (8/9) docker:desktop-linu[+] Building 10.7s (8/9) docker:des
ktop-linu[+] Building 10.8s (8/9) docker:desktop-linu[+] Building 11.0s (8/9) docker:desktop-linu[+] Build
ing 11.1s (8/9) docker:desktop-linu[+] Building 11.3s (8/9) docker:desktop-linu[+] Building 11.4s (8/9) d
ocker:desktop-linu[+] Building 11.6s (8/9) docker:desktop-linu[+] Building 11.7s (8/9) docker:desktop-linu
[+] Building 11.9s (8/9) docker:desktop-linu[+] Building 12.0s (8/9) docker:desktop-linu[+] Building 12.2s
(8/9) docker:desktop-linu[+] Building 12.3s (8/9) docker:desktop-linu[+] Building 12.5s (8/9) docker:des
ktop-linu[+] Building 12.6s (8/9) docker:desktop-linu[+] Building 12.8s (8/9) docker:desktop-linu[+] Build
ing 12.9s (8/9) docker:desktop-linu[+] Building 13.1s (8/9) docker:desktop-linu[+] Building 13.2s (8/9) d
ocker:desktop-linu[+] Building 13.4s (8/9) docker:desktop-linu[+] Building 13.5s (8/9) docker:desktop-linu
[+] Building 13.7s (8/9) docker:desktop-linu[+] Building 13.8s (8/9) docker:desktop-linu[+] Building 14.0s
(8/9) docker:desktop-linu[+] Building 14.1s (8/9) docker:desktop-linu[+] Building 14.3s (8/9) docker:des
ktop-linu[+] Building 14.4s (8/9) docker:desktop-linu[+] Building 14.6s (8/9) docker:desktop-linu[+] Build
ing 14.7s (8/9) docker:desktop-linu[+] Building 14.9s (8/9) docker:desktop-linu[+] Building 15.0s (8/9) d
ocker:desktop-linu[+] Building 15.2s (8/9) docker:desktop-linu[+] Building 15.3s (8/9) docker:desktop-linu
[+] Building 15.5s (8/9) docker:desktop-linu[+] Building 15.6s (8/9) docker:desktop-linu[+] Building 15.8s
(8/9) docker:desktop-linu[+] Building 15.9s (8/9) docker:desktop-linu[+] Building 16.1s (8/9) docker:des
ktop-linu[+] Building 16.2s (8/9) docker:desktop-linu[+] Building 16.4s (8/9) docker:desktop-linu[+] Build
ing 16.5s (8/9) docker:desktop-linu[+] Building 16.7s (8/9) docker:desktop-linu[+] Building 16.8s (8/9) d
```

RAM 1.26 GB CPU 0.12% Disk: 1.63 GB used (limit 1006.85 GB)

## Lab Worksheet

The screenshot shows two terminal panes within the Docker Desktop interface.

**Top Terminal:**

```
[+] Building 29.7s (9/10) docker:desktop-lin[+] Building 29.9s (9/10) docker:desktop-lin[+] Building 30.0s (9/10) docker:desktop-lin[+] Building 30.2s (9/10) docker:desktop-lin[+] Building 30.2s (9/10) docker:de sktop-lin[+] Building 30.3s (10/10) FINISHED docker:desktop-linuxUN yarn install --producti 23.7s
=> [internal] load build definition f 0.0s
=> => transferring dockerfile: 189B 0.0s
=> [internal] load metadata for docke 3.4s
=> [auth] library/node:pull token for 0.0s
=> [internal] load .dockignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/node: 0.0s
=> => resolve docker.io/library/node: 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 4.46kB 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 0.0s
=> [4/4] RUN yarn install --producti 23.7s
=> exporting to image 3.0s
=> => exporting layers 2.2s
=> => exporting manifest sha256:bf31c 0.0s
=> => exporting config sha256:dab2d1e 0.0s
=> => exporting attestation manifest 0.0s
```

RAM 1.26 GB CPU 1.25% Disk: 1.63 GB used (limit 1006.85 GB)

**Bottom Terminal:**

```
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 0.0s
=> [4/4] RUN yarn install --producti 23.7s
=> exporting to image 3.0s
=> => exporting layers 2.2s
=> => exporting manifest sha256:bf31c 0.0s
=> => exporting config sha256:dab2d1e 0.0s
=> => exporting attestation manifest 0.0s
=> => exporting manifest list sha256: 0.0s
=> => naming to docker.io/library/mya 0.0s
=> => unpacking to docker.io/library/ 0.8s
```

View build details: [docker-desktop://dashboard/build/desktop-linux/desktop-linux/yv9r8wepnv5vb3nw6ii6ywzkz](https://dashboard/build/desktop-linux/desktop-linux/yv9r8wepnv5vb3nw6ii6ywzkz)

| REPOSITORY       | TAG    | IMAGE ID     | CREATED        | SIZE   |
|------------------|--------|--------------|----------------|--------|
| myapp_6430206509 | latest | e1e148b24a06 | 11 seconds ago | 340MB  |
| my-first-image   | latest | 5d709c7c653b | 4 months ago   | 6.02MB |
| ssiriyakorn/lab8 | latest | 0ecf5e13fe6e | 4 months ago   | 6.02MB |
| busybox          | latest | a5d0ce49aa80 | 4 months ago   | 6.02MB |

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

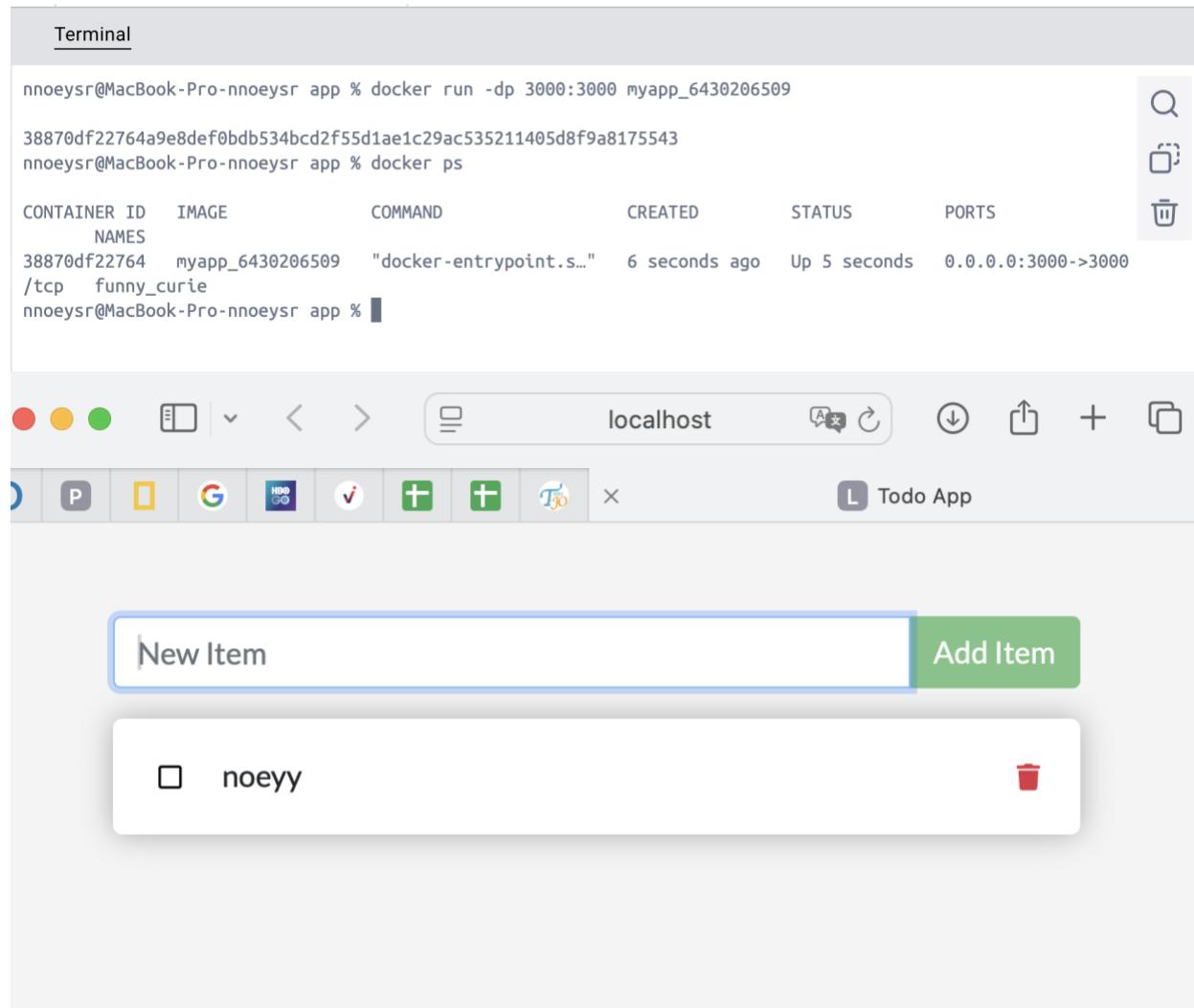
\$ docker run -dp 3000:3000 <myapp\_รหัสศ.ไม่มีชีด>

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

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

## Lab Worksheet

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

<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

b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

## Lab Worksheet

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

[Check point#10] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows a terminal window with two tabs: "Terminal" and "File: src/static/js/app.js". The "Terminal" tab contains the following command sequence:

```
nnoeysr@MacBook-Pro-nnoeysr Lab8_4 % cd getting-started/app
nnoeysr@MacBook-Pro-nnoeysr app % nano src/static/js/app.js
nnoeysr@MacBook-Pro-nnoeysr app % docker build -t myapp_6430206509 .

[+] Building 0.0s (0/0) docker:desktop-l[+] Building 0.0s (0/0) docker:desktop-l[+] Building 0.0s (0/0) docker:desktop-l[+] Building 0.0s (0/0) docker:desktop-l[+] Building 0.0s (0/1) docker:desktop-l[+] Building 0.2s (1/2) docker:desktop-l[+] Building 0.3s (1/2) docker:desktop-l[+] Building 0.5s (1/2) docker:deskt op-l[+] Building 0.6s (1/2) docker:desktop-l[+] Building 0.8s (1/2) docker:desktop-l[+] Building 0.9s (1/2 ) docker:desktop-l[+] Building 1.1s (1/2) docker:desktop-l[+] Building 1.1s (1/3) docker:desktop-l[+] Bu ilding 1.2s (2/3) docker:desktop-l[+] Building 1.4s (2/3) docker:desktop-l[+] Building 1.5s (2/3) docker:deskt op-l[+] Building 1.7s (2/3) docker:desktop-l[+] Building 1.8s (2/3) docker:desktop-l[+] Building 2.0s (2/3) docker:desktop-l[+] Building 2.1s (2/3) docker:desktop-l[+] Building 2.3s (2/3) docker:desktop-l[+] Building 2.4s (2/3) docker:desktop-l[+] Building 2.5s (3/3) docker:desktop-l[+] Building 2.7s (8/9) dock er:desktop-l[+] Building 2.9s (8/9) docker:desktop-l[+] Building 3.0s (8/9) docker:desktop-l[+] Building 3 .2s (8/9) docker:desktop-l[+] Building 3.3s (8/9) docker:desktop-l[+] Building 3.5s (8/9) docker:desktop-l[+] Building 3.6s (8/9) docker:desktop-l[+] Building 3.8s (8/9) docker:desktop-l[+] Building 3.9s (8/9) docker:desktop-l[+] Building 4.1s (8/9) docker:desktop-l[+] Building 4.2s (8/9) docker:desktop-l[+] Buildi ng 4.4s (8/9) docker:desktop-l[+] Building 4.5s (8/9) docker:desktop-l[+] Building 4.7s (8/9) docker:deskt op-l[+] Building 4.8s (8/9) docker:desktop-l[+] Building 5.0s (8/9) docker:desktop-l[+] Building 5.1s (8/ 9) docker:desktop-l[+] Building 5.3s (8/9) docker:desktop-l[+] Building 5.4s (8/9) docker:desktop-l[+] Bu
```

## Lab Worksheet

The terminal window shows the following Docker build logs:

```
=> exporting to image      3.0s
=> => exporting layers    2.2s
=> => exporting manifest sha256:0 0.0s
=> => exporting config sha256:880 0.0s
=> => exporting attestation manif 0.0s
=> => exporting manifest list sha 0.0s
=> => naming to docker.io/library 0.0s
=> => unpacking to docker.io/libr 0.8s
```

View build details: [docker-desktop://dashboard/build/desktop-linux/desktop-linux/i6bpab0nvfto3dhfuxwdzj8ba](https://docker-desktop://dashboard/build/desktop-linux/desktop-linux/i6bpab0nvfto3dhfuxwdzj8ba)

nnoeysr@MacBook-Pro-nnoeysr app % docker rm -f \$(docker ps -aq)

38870df22764  
e90987b3c2b1  
385a373d4c9f  
e1e37035a990  
bc2fc3a9690a

nnoeysr@MacBook-Pro-nnoeysr app % docker run -dp 3000:3000 myapp\_6430206509

de416c6910607798d395ac6e90092721254a19aac86b29772f8e7e3eefc571a0

The browser window shows a Todo App interface with the URL [localhost](http://localhost). The interface includes a header bar with icons for Home, Search, Refresh, Back, Forward, and Tabs, and a tab labeled "Todo App". Below the header is a search bar with placeholder "New Item" and a green "Add Item" button. The main content area displays the message: "There is no TODO item. Please add one to the list. By siriayakorn eaksiri".

(1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร

ไม่มี Error

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีเดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

- ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
- Copy หรือบันทึก Container ID ไว้
- ใช้คำสั่ง \$ docker stop <Container ID> ที่ต้องการจะลบ> เพื่อหยุดการทำงานของ Container ดังกล่าว

## Lab Worksheet

iv. ใช้คำสั่ง \$ docker rm <Container ID> ที่ต้องการจะลบ เพื่อทำการลบ

b. ผ่าน Docker desktop

- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในແຄວของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกรัง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

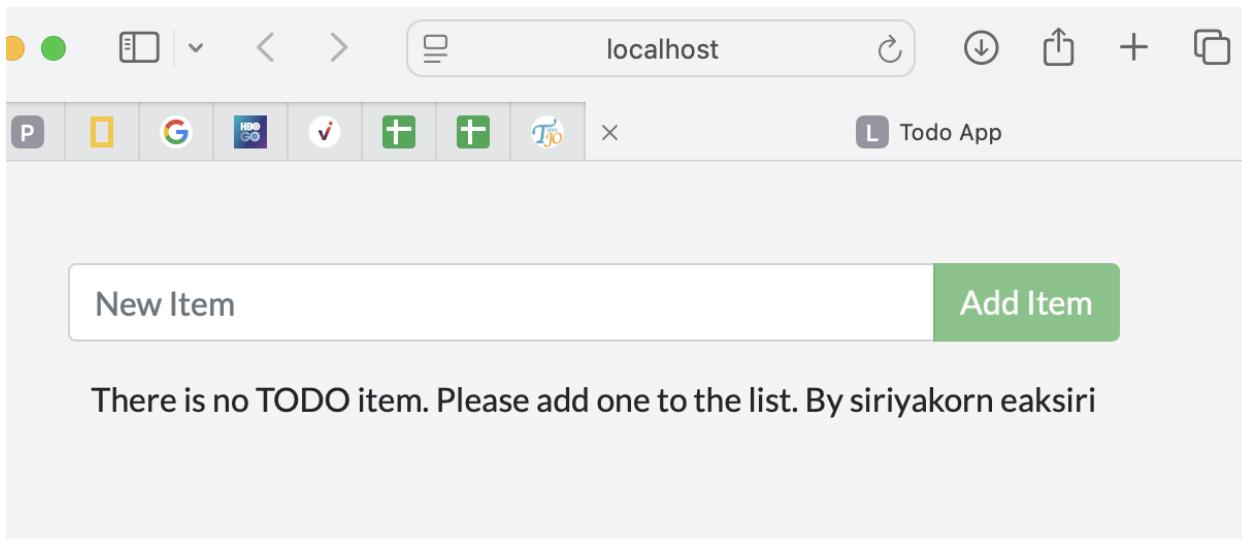
[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

The screenshot shows the Docker Desktop interface. At the top, it displays CPU and memory usage statistics. Below that is a search bar and a filter option to 'Only show running containers'. A prominent modal dialog box is centered, asking 'Delete container?' and stating that the 'cranky\_clarke' container is selected for deletion. It also notes that any anonymous volumes associated with these containers are also deleted. Two buttons are visible: 'Cancel' and 'Delete forever'. In the background, a table lists running containers. One container, 'dreamy\_roentgen', is highlighted with a green dot. Its details are shown: Name: dreamy\_roentgen, Container ID: ec0da8f6a2c6, Image: myapp\_64302065, Port(s): 3000:3000. The last started time is 52 seconds ago.

| Name            | Container ID | Image          | Port(s)   | CPU (%) | Last started   | Actions        |
|-----------------|--------------|----------------|-----------|---------|----------------|----------------|
| dreamy_roentgen | ec0da8f6a2c6 | myapp_64302065 | 3000:3000 | 0%      | 52 seconds ago | <span>⋮</span> |

The screenshot shows the Docker Desktop terminal window. The user has run the command `docker run -dp 3000:3000 myapp_6430206509`. The terminal output shows the container ID `ec0da8f6a2c6` and the user attempting to access the application at `http://localhost:3000`. The response from the browser indicates that there is no such file or directory at that path. The terminal window title is 'Terminal' and the tab name is 'zsh'.

## 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:lts-jdk17
```

หรือ

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v
```

```
jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17
```

3. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

## Lab Worksheet

**[Check point#12] Capture หน้าจอที่แสดงผล Admin password**

| <input type="checkbox"/> | Name            | Container ID | Image                   | Port(s)     | CPU (%) | Last started   | Action |
|--------------------------|-----------------|--------------|-------------------------|-------------|---------|----------------|--------|
| <input type="checkbox"/> | dreamy_roentgen | ec0da8f6a2c6 | myapp_6430206509        | 3000:3000   | 0%      | 29 minutes ago |        |
| <input type="checkbox"/> | thirsty_mclaren | 1733e5c996ab | jenkins/jenkins:lts-jdk | 50000:50000 | 0.36%   | 54 seconds ago |        |

Terminal

```
nnoeysr@MacBook-Pro-nnoeysr ~ % docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17

Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-01-29 12:51:37.356+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2025-01-29 12:51:37.406+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-01-29 12:51:37.442+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.16; built: 2024-12-09T21:02:54.535Z; git: c3f88baafb4e393f23204dc14dc57b042e84debc7; jvm 17.0.13+11
2025-01-29 12:51:37.657+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2025-01-29 12:51:37.686+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-01-29 12:51:37.914+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2025-01-29 12:51:37.986+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@682c1e93[Jenkins v2.479.3,,b=file:///var/jenkins_home/war,,a=AVAILABLE,h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@27068a50{STARTED}]
2025-01-29 12:51:37.996+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@2b9b7f1f{HTTP/1.1, {http/1.1}}{0.0.0.0:8080}
2025-01-29 12:51:38.004+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@b9b00e0{STARTING}[12.0.16,sto=0] @965ms
```

Terminal

```
2025-01-29 12:51:38.719+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-29 12:51:38.761+0000 [id=43] INFO jenkins.install.SetupWizard#init:
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
dde1799a622b433ba0f7f362420cfec8

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****
*****
*****
```

```
2025-01-29 12:52:13.219+0000 [id=43] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 12:52:13.265+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
```

4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดбраузอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3

## Lab Worksheet

6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062  
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....



The screenshot shows the Jenkins dashboard for the user 'siriayakorn eaksiri'. The top navigation bar includes the Jenkins logo and the user's name. Below the navigation bar, there are several menu items: 'Status' (selected), 'Builds', 'My Views', and 'Account'. To the right of the menu, the user's profile picture and name 'siriayakorn eaksiri' are displayed, along with the text 'Jenkins User ID: siriayakorn\_6509'.

7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>  
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

## Lab Worksheet

This screenshot shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The main header displays the Jenkins logo and the user 'Chitsutha Soomlek'. The dashboard features a 'Welcome to Jenkins!' message with instructions to start building software projects. It includes sections for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. On the left sidebar, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. A 'Build Queue' section indicates 'No builds in the queue.' and a 'Build Executor Status' section shows '0/2'. The bottom right corner of the dashboard shows 'REST API' and 'Jenkins 2.479.3'.

### 9. เลือก Manage Jenkins และไปที่เมนู Plugins

This screenshot shows the 'Manage Jenkins' page at [localhost:8080/manage/](http://localhost:8080/manage/). The 'Manage Jenkins' tab is active. The page is divided into several sections: 'System Configuration' (with links to System, Tools, Plugins, Clouds, Appearance), 'Security' (with links to Security, Credentials, Credential Providers, Nodes), 'Status Information' (with links to System Information, System Log, Load Statistics), and 'About Jenkins' (with a link to About Jenkins). A prominent message at the top states 'It appears that your reverse proxy set up is broken.' with 'More Info' and 'Dismiss' buttons. The bottom right corner shows 'Jenkins 2.479.3'.

**Lab Worksheet**

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม

11. กลับไปที่หน้า Dashboard และสร้าง 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 และใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

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

## Lab Worksheet

**Build Steps:** เลือก Execute shell และใส่คำสั่งในการรันไฟล์ .robot (หากไฟล์ไม่ได้อยู่ในหน้าแรกของ repository ให้ใส่ Path ไปถึงไฟล์ให้เรียบร้อยด้วย)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Description

Lab 8.5

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

Project url [?](#)

https://github.com/kku-computer-science/configuration-management-ssiriyakorn.git/

[Advanced](#) ▾

## Lab Worksheet

**Source Code Management** None Git [?](#)**Build Triggers** Trigger builds remotely (e.g., from scripts) [?](#) Build after other projects are built [?](#) Build periodically [?](#)Schedule [?](#)

H/15 \* \* \* \*

Would last have run at Wednesday, January 29, 2025 at 3:05:31 PM Coordinated Universal Time.  
2025 at 3:20:31 PM Coordinated Universal Time.

## Lab Worksheet

## Build Steps



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

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

## Post-build Actions

≡ Publish Robot Framework test results ?

Directory of Robot output

Path to directory containing robot xml and html files (relative to build workspace)

results/

Advanced ▾    Edited

Thresholds for build result ?

%  
20.0

%  
80.0

DEPRECATED! THIS FLAG DOES NOTHING! - Use thresholds for critical tests only

Include skipped tests in total count for thresholds

Save Apply Terminal

## ✖️ UAT

Lab 8.5



**Latest Robot Results:**

No results available yet.

## Permalinks

- [Last build \(#16\), 54 sec ago](#)
- [Last failed build \(#16\), 54 sec ago](#)
- [Last unsuccessful build \(#16\), 54 sec ago](#)
- [Last completed build \(#16\), 54 sec ago](#)

Error