

**Lab Worksheet**

ชื่อ-นามสกุล นาย เกรียงไกร ประเสริฐ รหัสนักศึกษา663380616-4 Section4

**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

```
krlangkrai@MacBook-Air--KRIANGKRAI:~ - % mkdir Lab8_1
krlangkrai@MacBook-Air--KRIANGKRAI:~ - % cd Lab8_1
krlangkrai@MacBook-Air--KRIANGKRAI:~/Lab8_1 % docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
b85757a5ca1a: Pull complete
Digest: sha256:e226d3308e590dbe282443c87e57365c96055228f0fe7f40731b5d84d37a06839
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview busybox
krlangkrai@MacBook-Air--KRIANGKRAI:~/Lab8_1 % docker images
```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
busybox:latest	cd9176cd36f9	4.11MB	0B	
docker/desktop-kubernetes:kubernetes-v1.34.1-cni-v1.7.1-critools-v1.33.0-cri-dockerd-v0.3.28-1-debian	0f1fc82826ab	383MB	0B	
docker/desktop-storage-provisioner:v2.0	c827a58faeb8	39.8MB	0B	
docker/desktop-vpnkit-controller:dc331cb2285be9ccdf97c84a9cfecaf44a1af6e	3750fce169f	35MB	0B	
docker/disk-usage-extension:0.2.9	83bc5b217a10	2.62MB	0B	
Fedora:latest	e108db52fd27	193MB	0B	U
Hello-World:latest	ca9995c726f0	5.2kB	0B	
Jenkins/jenkins:latest	7e39a0c49851	505MB	0B	U
nginx:latest	984dc4c1bec9	180MB	0B	U
registry.k8s.io/coredns/coredns:v1.12.1	138784d87c9c	72.1MB	0B	
registry.k8s.io/etcfd:v3.5.16-0	7fc9d4aa817a	142MB	0B	
registry.k8s.io/etcfd:v3.6.4-0	a1894772a478	205MB	0B	
registry.k8s.io/kube-apiserver:v1.32.2	6417e1437b6d	93.9MB	0B	
registry.k8s.io/kube-apiserver:v1.34.1	43911e833d64	83.7MB	0B	
registry.k8s.io/kube-controller-manager:v1.32.2	3c9285acf2df	87.2MB	0B	
registry.k8s.io/kube-controller-manager:v1.34.1	7eb2c6ff0c5a	71.5MB	0B	
registry.k8s.io/kube-proxy:v1.34.1	05baa95fs142	74.7MB	0B	
registry.k8s.io/kube-scheduler:v1.34.1	b5f57ec0d986	50.5MB	0B	
registry.k8s.io/pause:3.10	afbd61768ce38	514kB	0B	
registry.k8s.io/pause:3.10.1	d7b100cd9a77	514kB	0B	
ubuntu:latest	d90f11c975cd	101MB	0B	U

krlangkrai@MacBook-Air--KRIANGKRAI:~/Lab8\_1 %

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร \_\_\_\_\_ คือ busy box \_\_\_\_\_

(2) Tag ที่เข้าบ่งบอกถึงอะไร \_\_\_\_\_ คือ tag latest \_\_\_\_\_

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

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 % docker run busybox
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 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 Feb  2 07:06 .
drwxr-xr-x  1 root  root  4096 Feb  2 07:06 ..
-rw-rxr-x  1 root  root     0 Feb  2 07:06 .dockerenv
drwxr-xr-x  2 root  root 12288 Sep 26 2024 bin
drwxr-xr-x  5 root  root  360 Feb  2 07:06 dev
drwxr-xr-x  1 root  root  4096 Feb  2 07:06 etc
drwxr-xr-x  2 nobody nobody  4096 Sep 26 2024 home
drwxr-xr-x  2 root  root  4096 Sep 26 2024 lib
lrwxrwxrwx  1 root  root   3 Sep 26 2024 lib64 -> lib
dr-xr-xr-x 215 root  root   0 Feb  2 07:06 proc
drwx----- 1 root  root  4096 Feb  2 07:07 root
dr-xr-xr-x 11 root  root   0 Feb  2 07:06 sys
drwxrwxrwt  2 root  root  4096 Sep 26 2024 tmp
drwxr-xr-x  4 root  root  4096 Sep 26 2024 usr
drwxr-xr-x  4 root  root  4096 Sep 26 2024 var
/ # exit
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 % docker run busybox echo "Hello นาย เกierge ไก่ ประเสริฐ from busybox"
Hello นาย เกierge ไก่ ประเสริฐ from busybox
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a6de720dc959 busybox "echo 'Hello นาย เกierge...'" 11 seconds ago Exited (0) 11 seconds ago reverent_fermat
385aa190b722 busybox "sh" About a minute ago Exited (0) 35 seconds ago objective_nash
f4176117123a busybox "sh" About a minute ago Exited (0) About a minute ago tender_vaughan
1ea1df2cc26b fedora "/bin/bash" 4 days ago Created relaxed_lamarr
5c0e58bce5f8 ubuntu "bash" 4 days ago Created practical_euler
578fb841834b5 jenkins/jenkins:latest "/usr/bin/tini -- /..." 4 days ago Exited (143) 4 days ago test
17ab1982e388 nginx "/docker-entrypoint..." 4 days ago Exited (0) 4 days ago nifty_dewdney
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 %
```

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

ทำให้เราสามารถเข้าไปพิมพ์คำสั่งต่างๆ ภายใน Container ได้เหมือนใช้คอมอีกเครื่อง \_\_\_\_\_

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

สถานะปัจจุบันของ Container เช่น Up (กำลังทำงานอยู่) หรือ Exited (ทำงานเสร็จแล้วและปิดตัวลงแล้ว) \_\_\_\_\_

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

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

## Lab Worksheet

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 % docker rm a6de720dc959
a6de720dc959
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_1 % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
385aa190b722 busybox "sh" 4 minutes ago Exited (0) 4 minutes ago objective_nash
f4176117123a busybox "sh" 5 minutes ago Exited (0) 5 minutes ago tender_vaughan
1ead1f2cc26b fedora "/bin/bash" 4 days ago Created relaxed_lamarr
5c0e58bce5f8 ubuntu "bash" 4 days ago Created practical_euler
578f841834b5 jenkins/jenkins:latest "/usr/bin/tini -- /u..." 4 days ago Exited (143) 4 days ago test
17ab1982e388 nginx "/docker-entrypoint..." 4 days ago Exited (0) 4 days ago nifty_dewdney
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_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
4. สร้าง Dockerfile.swp ไว้ใน Working directory

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ส (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

## Lab Worksheet

\$ docker build -t <ชื่อ Image> .

- เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```
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
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to Image
=> exporting layers
=> writing image sha256:effc1e0cff7a77b79d69614a7c543940ddb308f7f443bd6512f7c4a4595220459
=> naming to docker.io/library/my-bio-image

3 warnings found (use docker --debug to expand):
- 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)
- MultipleCMDInstructionsNotAllowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)

View build details: docker://dashboard/build/desktop-linux/desktop-linux/mesabboufsstsgf5dSabw1Sc

What's next:
View a summary of image vulnerabilities and recommendations -> docker scan quickview
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker run my-bio-image
微妙 ทำให้ไม่เข้าใจ 653388016-4 aper
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker run my-bio-image
微妙 ทำให้ไม่เข้าใจ 653388016-4 aper
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % nano Dockerfile
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker run my-bio-image
微妙 ทำให้ไม่เข้าใจ 653388016-4 aper
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker run my-bio-image
微妙 ทำให้ไม่เข้าใจ 653388016-4 aper
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker build --no-cache -t my-bio-image .
[+] Building 0.0s (5/5) FINISHED
=> [Internal] load build definition from Dockerfile
=> => transferring Dockerfile: 195B
WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
=> [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
=> writing image sha256:5627afa97fcdaaf03189cc2eef0b64af3871f2681a679b31fd3c29d7c0d21a7f
=> naming to docker.io/library/my-bio-image

1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)

View build details: docker://desktop-linux/desktop-linux/ds0qbw12yylnhb0etco

What's next:
View a summary of image vulnerabilities and recommendations -> docker scan quickview
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 % docker run my-bio-image
微妙 ทำให้ไม่เข้าใจ 653388016-4 aper
kringkra@MacBook-Air-KRIANGKRAINEW-1195 Lab8_2 %
```

(1) คำสั่งที่ใช้ในการ run คือ

docker run my-bio-image

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

ใช้คำหรับตั้งชื่อ (Tag) ให้กับ Image ที่เราสร้างขึ้น เพื่อให้เรียกใช้งานได้ง่ายในภายหลัง (ถ้าไม่ใส่ Docker จะสุ่ม ID ยาวๆ มาให้ ซึ่งจำกาก)

### แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

- เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
- เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_3
- ย้ายตำแหน่งปัจจุบันไปที่ Lab8\_3 เพื่อใช้เป็น Working directory
- สร้าง Dockerfile.swp ไว้ใน Working directory

## Lab Worksheet

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ 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

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

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

## Lab Worksheet

```
kriangkrai@MacBook-Air-xRIANGKRAInew-1195 Lab8_3 % docker build -t kriangkrai54/Lab8 .
[+] Building 0.0s (5/5) FINISHED
=> [Internal] load build definition from Dockerfile
=> => transferring dockerfile: 21B
==> 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
=> => transferring context: 2B
=> CAHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:efba74f097a0d1d2d3a72c36e01c28e551e236afe26a175e1dd1c0bcd93050d
=> => naming to docker.io/kriangkrai54/lab8

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- 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)

View build details: docker-desktop://dashboard/build/desktop/linux/desktop-linux/h7ycjrrvp8dvx9tt9lpjwlm

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
kriangkrai@MacBook-Air-xRIANGKRAInew-1195 Lab8_3 % docker run kriangkrai54/Lab8
Hello there. My work is done. You can run them from my Docker image.
มา เก็บ.ไฟร์ เบอร์ 653388616-4
kriangkrai@MacBook-Air-xRIANGKRAInew-1195 Lab8_3 % docker build -t kriangkrai54/Lab8 .
[+] Building 0.0s (5/5) FINISHED
=> [Internal] load build definition from Dockerfile
=> => transferring dockerfile: 21B
==> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
=> [Internal] load metadata for docker.io/library/busybox:latest
=> [Internal] load .dockerignore
=> => transferring context: 2B
=> CAHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:1cfbfdf6af47974ed38cd69da21e24f93bec4bd630299dc0ddac4668dedcbfffd8
=> => naming to docker.io/kriangkrai54/lab8

1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)

View build details: docker-desktop://dashboard/build/desktop/linux/desktop-linux/yjqcc699ylcv6db5527hvvcn

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
kriangkrai@MacBook-Air-xRIANGKRAInew-1195 Lab8_3 % docker run kriangkrai54/Lab8
Hello there. My work is done. You can run them from my Docker image.
มา เก็บ.ไฟร์ เบอร์ 653388616-4
kriangkrai@MacBook-Air-xRIANGKRAInew-1195 Lab8_3 %
```

### 6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

\$ 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

## Lab Worksheet

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

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_3 % docker run kriangkrai54/lab8
```

```
Hi there. My work is done. You can run them from my Docker image.
```

```
นาย เกี้ยงไกร ประเสริฐ 663380616-4
```

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_3 % docker login
```

```
Authenticating with existing credentials... [Username: kriangkrai54]
```

*i Info → To login with a different account, run 'docker logout' followed by 'docker login'*

```
Login Succeeded
```

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_3 % docker push kriangkrai54/lab8
```

```
Using default tag: latest
```

```
The push refers to repository [docker.io/kriangkrai54/lab8]
```

```
43f051b5fedf: Mounted from library/busybox
```

```
latest: digest: sha256:08e35cb73c405bd6a4c674b1553a7ea1e076826b0ffe4a89aa69de279093c559 size: 527
```

```
kriangkrai@MacBook-Air--KRIANGKRAImeow-1195 Lab8_3 %
```

#### แบบฝึกปฏิบัติที่ 8.4: การ Build และ Update แอปพลิเคชันจาก Container image

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_4
2. ทำการ Clone ซอฟต์แวร์ Docker ของเว็บแอปพลิเคชันจาก 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

```
UN PICO 5.09                                         File: package.json

{
  "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": "v2.3.3",
    "sqlite3": "5.1.2",
    "uuid": "9.0.0",
    "walt-port": "v1.0.4"
  },
  "resolutions": {
    "ansi-regex": "5.0.1"
  },
  "prettier": {
    "trailingComma": "all",
    "tabWidth": 4,
    "useTabs": false,
    "semi": true,
    "singleQuote": true
  },
  "devDependencies": {
    "jest": "29.3.1",
    "nodemon": "2.0.20",
    "prettier": "2.7.1"
  }
}

kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 987, done.
remote: Counting objects: 100% (274/274), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 987 (delta 247), reused 230 (delta 230), pack-reused 713 (from 1)
Receiving objects: 100% (987/987), 5.28 MB | 9.88 MB/s, done.
Resolving deltas: 100% (539/539), done.
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ ls
getting-started
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ cd getting-started
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ ls
Dockerfile           LICENSE          README.md        app           build.sh      docker-compose.yml   docs          mkdocs.yml       requirements.txt
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ cd app
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ ls
package.json         spec            SRC             yarn.lock
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ app % nano package.json
kriangkrai@MacBook-Air--KRIANGKRAI:~/Desktop/lab8_4$ app %
```

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.

ศ. ไม่มีชื่อ

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

## Lab Worksheet

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

```
krLangkrai@MacBook-Air:~/KRIANGKRAINEW-1195 app % touch Dockerfile
krLangkrai@MacBook-Air:~/KRIANGKRAINEW-1195 app % nano Dockerfile
krLangkrai@MacBook-Air:~/KRIANGKRAINEW-1195 app % docker build -t myapp_6633806164 .
[+] Building 22.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 15B
=> [internal] load metadata for docker.io/library/node:18-alpine
[auth] https://registry-1.docker.io/: Get https://registry-1.docker.io/v2/_catalog?query=%7B%22name%22%3A%22node%22%7D: dial tcp: lookup https://registry-1.docker.io/ on 127.0.0.5:53: no such host
=> [internal] load index for docker.io/library/node:18-alpine
=> [internal] transfer context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:8d6421d6634c428f3ebc498332f249011d11b945588d0a35cb9bc4b8ca09d9e
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d6634c428f3ebc498332f249011d11b945588d0a35cb9bc4b8ca09d9e
=> => sha256:8d6421d6634c428f3ebc498332f249011d11b945588d0a35cb9bc4b8ca09d9e 7.67kB / 7.67kB
=> => sha256:c2281c62c44adff52ea71a6056c6b649634bd5a99d52ade5457fc9c29ba037 1.20kB / 1.20kB
=> => sha256:c5914b9dd279df05969336e5906c63c3cd2da5aae71e863e19e96963997fc1c17 6.20kB / 6.20kB
=> => sha256:6e7f1e15690e2fabf2332d3a3b7444954116de0002aa64419658b066cf81 3.99kB / 3.99kB
=> => sha256:084c815451acbc9606edbd47992922be57121df01ecc5b15b28c52c52dbaf19a 39.66MB / 39.66MB
=> => sha256:0fbfa36aa9dce514f6ada0a1fb7f1b7875da0d23310f0cb08ae1ecfa992f587f6 1.20kB / 1.20kB
=> => sha256:02b8a4e9f3412827f177bc6c020812249532a8425d0c1858e9d71b4dc4015f031 443B / 443B
=> => extracting sha256:02b8a4e9f3412827f177bc6c020812249532a8425d0c1858e9d71b4dc4015f031
=> => extracting sha256:0fbfa36aa9dce514f6ada0a1fb7f1b7875da0d23310f0cb08ae1ecfa992f587f6
=> => extracting sha256:0fbfa36aa9dce514f6ada0a1fb7f1b7875da0d23310f0cb08ae1ecfa992f587f6
=> => extracting sha256:0fbfa36aa9dce514f6ada0a1fb7f1b7875da0d23310f0cb08ae1ecfa992f587f6
=> => extracting sha256:0fbfa36aa9dce514f6ada0a1fb7f1b7875da0d23310f0cb08ae1ecfa992f587f6
=> [internal] load build context
=> => transferring context: 4.59MB
=> [2/4] WORKDIR /app
=> [3/4] COPY .
=> [4/4] RUN yarn install --production
=> exporting to image
=> exporting layers
=> writing image sha256:e9fce17328ef4b21feef7097d5377657b3660c405db777fc7d5b30a03fc4bb70
=> naming to docker.io/library/myapp_6633806164
```

View build details: docker-desktop://dashboard/build/decktop-linux/decktop-linux/nb4y5559xh6hnp9rpptddos

What's next:

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

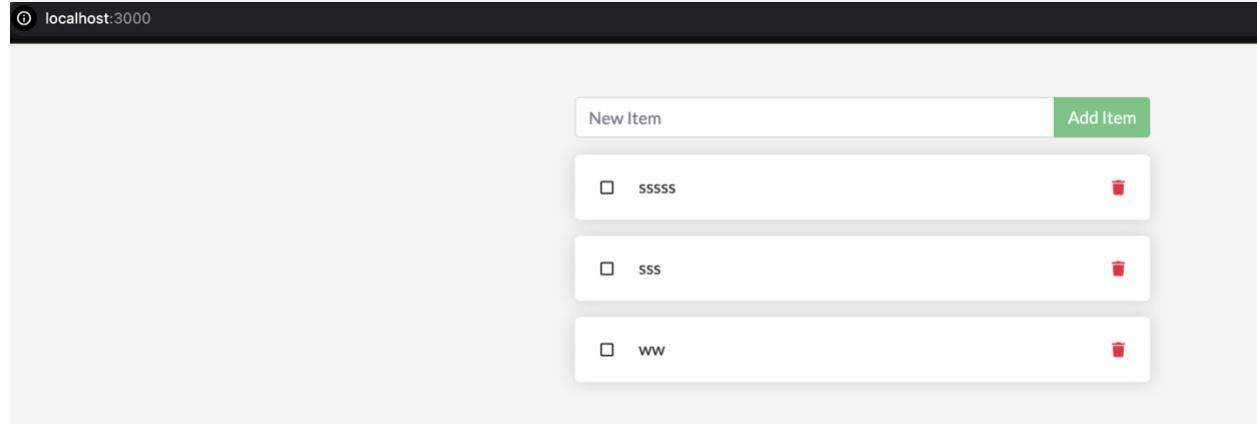
7. เปิด Browser ไปที่ URL = <http://localhost:3000>

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

# Lab Worksheet

[View build details](#): docker-desktop://dashboard/build/desktop-linux/desktop-linux/mb4y5559x6h6rp9rnptddkxs

```
What's next:  
  View a summary of Image vulnerabilities and recommendations → docker scout quickview  
krlangkra@akBob0:~/lr-KRIANGRAIMnew-1195 app % docker run -dp 3000:3000 myapp_6633886164  
e6ace42cbe31b38251cd7c9e8ac1d3ca6d435217521aa1aae2219687b  
krlangkra@akBob0:~/lr-KRIANGRAIMnew-1195 app %
```



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>

<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

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

## Lab Worksheet

[Check point#10] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้

```

krLangkrai@MacBook-Air-KRIANGKRAInew-1195 app % cd src
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 src % ls
  index.js  persistence  routes  static
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 src % cd static
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 static % ls
  css  index.html
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 static % cd js
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 js % ls
  app.js  babel.mln.js  react-bootstrap.js  react-dom.production.min.js  react.production.min.js
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 js % nano app.js
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 % docker build -t myapp_6633806164 .
[+] Building 0.0s (1/1) FINISHED
=> [internal] load build definition from dockerfile
=> => transferred dockerfile: 2B
Error: failed to build: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory

View build details: docker:desktop://dashboard/build/desktop-linux/desktop-1linux/zhi/fpg8hzduzrue5jog95
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 js % cd ..
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 static % cd ..
zsh: command not found: cd..
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 static % cd ..
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 app % docker build -t myapp_6633806164 .
[+] Building 12.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferred dockerfile: 150B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] resolve node@sha256:fcacbf5ae4527eb7244ea86853c879817d25d591cbd149f6ea955127c2c7f672
=> [internal] load dockerfile
=> [internal] transfer context: 28
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d11894558b0a35cb9bc4b8ca09d9
=> [internal] load context
=> [internal] transfer context: 26-538B
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> exporting layers
=> writing image sha256:fcacbf5ae4527eb7244ea86853c879817d25d591cbd149f6ea955127c2c7f672
=> naming to docker.io/library/myapp_6633806164

View build details: docker:desktop://dashboard/build/desktop-1linux/desktop-1linux/nyevatausdwaavhuarjlyj588

What's next:
  View a summary of image vulnerabilities and recommendations - docker scout quickview
krLangkrai@MacBook-Air-KRIANGKRAInew-1195 app % docker run -dp 3000:3000 myapp_6633806164
74572953fa7a07c64dd1ed400bb585a99686865baaa1fc093801920d5b6e46724
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint hopeful_snyder (c4b41874b2e51b609bab77786f9d46b78fee66f9f7e1a310e98aca9933254cdd): Bind for 0.0.0.0:3000 failed: port is already allocated

Run 'docker run -help' for more information

```

(1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร

เกิดจาก Port Conflict (พอร์ตชนกัน) เพราะ Container ตัวเดียวกันที่ต้องการใช้ Port 3000 ไว้อよฉะ เราไม่สามารถเปิด Container ใหม่ที่ใช้ Port 3000 ซ้ำกันพร้อมกันได้

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

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
- ใช้คำสั่ง \$ docker rm <Container ID> ที่ต้องการจะลบ เพื่อทำการลบ

b. ผ่าน Docker desktop

- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในແຕວของ Container ที่ต้องการจะลบ

## Lab Worksheet

## 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) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

```
localhost:3000

New Item Add Item

There is no TODO item. Please add one to the list. By นาย เกรียงไกร ประเสริฐ

Run 'docker run --help' for more information
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
74572953fa07 myapp_6633806164 "docker-entrypoint.s..." 2 minutes ago Created
e6ace2a2cbc3 e9fce17328e "docker-entrypoint.s..." 2 hours ago Up 2 hours
4223464bdf5c krtangkrai54/lab8 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
18b9f9462958 efba74ff997a9 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
1b026680e7cd my-blo-image "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
eab89b963435 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
8a38624d8013 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
e766a14f387b 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
48e5985ba43b 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
4e0e9fbefc8 29e4a32cc2dd "sh" 2 hours ago Exited (0) 2 hours ago
ae1299aa01f8 29e4a32cc2dd "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (2) 2 hours ago
aa50653e5651 29e4a32cc2dd "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (2) 2 hours ago
385aa190b722 busybox "sh" 3 hours ago Exited (0) 2 hours ago
f4176117123a busybox "sh" 3 hours ago Exited (0) 3 hours ago
1ea10f2cc26b fedora "/bin/bash" 4 days ago Created
5c0e58bce5f8 ubuntu "bash" 4 days ago Created
578fb41834b5 Jenkins/jenkins:latest "/usr/bin/tini -- /u..." 4 days ago Exited (143) 4 days ago
17ab1982e388 nginx "/docker-entrypoint..." 5 days ago Exited (0) 5 days ago
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app % docker stop e6ace2a2cbc3
e6ace2a2cbc3
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
74572953fa07 myapp_6633806164 "docker-entrypoint.s..." 4 minutes ago Created
e6ace2a2cbc3 e9fce17328e "docker-entrypoint.s..." 2 hours ago Exited (0) 3 seconds ago
4223464bdf5c krtangkrai54/lab8 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
18b9f9462958 efba74ff997a9 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
1b026680e7cd my-blo-image "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
eab89b963435 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
8a38624d8013 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
e766a14f387b 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
48e5985ba43b 4ffc1e0cf7a7 "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (0) 2 hours ago
4e0e9fbefc8 29e4a32cc2dd "sh" 2 hours ago Exited (0) 2 hours ago
ae1299aa01f8 29e4a32cc2dd "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (2) 2 hours ago
aa50653e5651 29e4a32cc2dd "/bin/sh -c 'echo \"\u20ac\"" 2 hours ago Exited (2) 2 hours ago
385aa190b722 busybox "sh" 3 hours ago Exited (0) 3 hours ago
f4176117123a busybox "sh" 3 hours ago Exited (0) 3 hours ago
1ea10f2cc26b fedora "/bin/bash" 4 days ago Created
5c0e58bce5f8 ubuntu "bash" 4 days ago Created
578fb41834b5 Jenkins/jenkins:latest "/usr/bin/tini -- /u..." 4 days ago Exited (143) 4 days ago
17ab1982e388 nginx "/docker-entrypoint..." 5 days ago Exited (0) 5 days ago
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app % docker rm e6ace2a2cbc3
e6ace2a2cbc3
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app % docker run -dp 3000:3000 myapp_6633806164
3d995ff5179c932a00f9a1029690c992970704407cecff3f2e5d2bfee05fac
krLangkrai@MacBook-Air--KRIANGKRAI:meow-1195 app %
```

## Lab Worksheet

## แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

- สร้าง Dockerfile เพื่อสร้าง Jenkins และ Environment ที่เหมาะสมกับการรัน Robot framework ใน Container

[Check point#12] ส่ง Dockerfile ที่ได้คำสั่งที่เกี่ยวข้องไว้

```

UW PICO 5.09                               File: Dockerfile
# เริ่มจาก Base image ของ Jenkins ตัวล่าสุด
FROM jenkins/jenkins:lts-jdk17

#  설정สิทธิ์เป็น root เพื่อติดตั้งโปรแกรมเสริม
USER root

# อัปเดตและติดตั้ง Python, pip และ Chromium (Browser สำหรับ Test)
RUN apt-get update && apt-get install -y \
    python3 \
    python3-pip \
    python3-venv \
    chromium \
    chromium-driver

# สร้าง Virtual Environment เพื่อป้องกันปัญหา Permission และ Library ตัดต่อ
ENV VIRTUAL_ENV=/opt/venv
RUN python3 -m venv $VIRTUAL_ENV
ENV PATH="$VIRTUAL_ENV/bin:$PATH"

# ติดตั้ง Robot Framework และ SeleniumLibrary
RUN pip install robotframework robotframework-seleniumlibrary

#  설정สิทธิ์กลับมาเป็น jenkins user ตามปกติ
USER jenkins

```

- เปิด Command line หรือ Terminal บน Docker Desktop
- Build image ที่สร้างขึ้นในข้อที่ 1 พร้อมกับตั้งชื่อของ image เป็น jenkins-robot-local  
\$ docker build -t jenkins-robot-local .
- รัน container โดยผูกพอร์ตให้เรียบร้อย เช่น

```

$ docker run -d \
  --name jenkins-robot \
  -p 8080:8080 -p 50000:50000 \
  -v jenkins_home:/var/jenkins_home \
  -v /var/run/docker.sock:/var/run/docker.sock \
  jenkins-robot-local

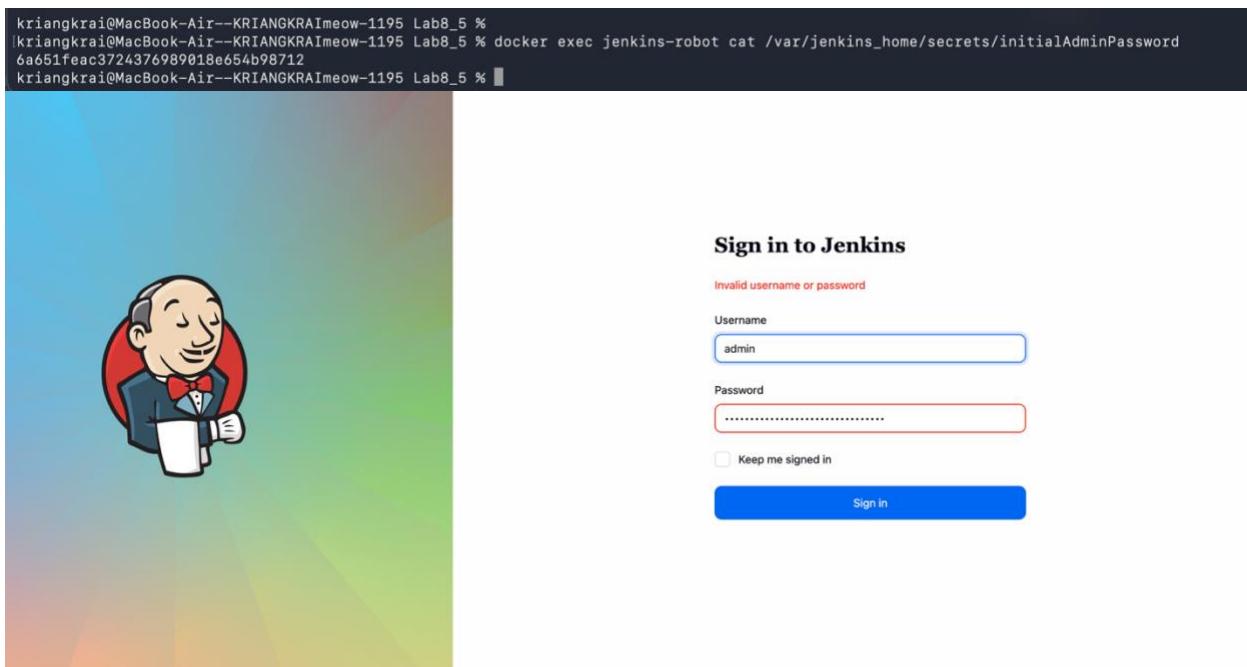
```

หรือ

## Lab Worksheet

```
$ docker run -it \
--name jenkins-debug \
-p 8080:8080 \
-v /var/run/docker.sock:/var/run/docker.sock \
jenkins-robot-local
```

[Check point#13] Capture หน้าจอที่แสดงผล Admin password



5. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก
6. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดбраузอร์ และป้อนที่อยู่เป็น <http://localhost:8080>
7. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
8. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062

[Check point#14] Capture หน้าจอที่แสดงผลการตั้งค่า

The screenshot shows the Jenkins management interface for users. It lists two users: "admin" and "kriangkrai\_6164". Each user entry includes a profile icon, the user ID, the name, and two small icons for editing and deleting the user.

User ID	Name	Action Icons
admin	admin	
kriangkrai_6164	kriangkrai_6164	

## Lab Worksheet

### 9. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

### 10. เลือก Manage Jenkins และไปที่เมนู Plugins

## Lab Worksheet

11. ไปที่เมนู Plugins > Available plugins และเลือกติดตั้ง Robot Framework เพิ่มเติม

Install	Name	Released	Health
<input type="checkbox"/>	Robot Framework 6.2.0 <small>Build Reports</small>	3 mo 15 days ago	<span style="color: green;">91</span>

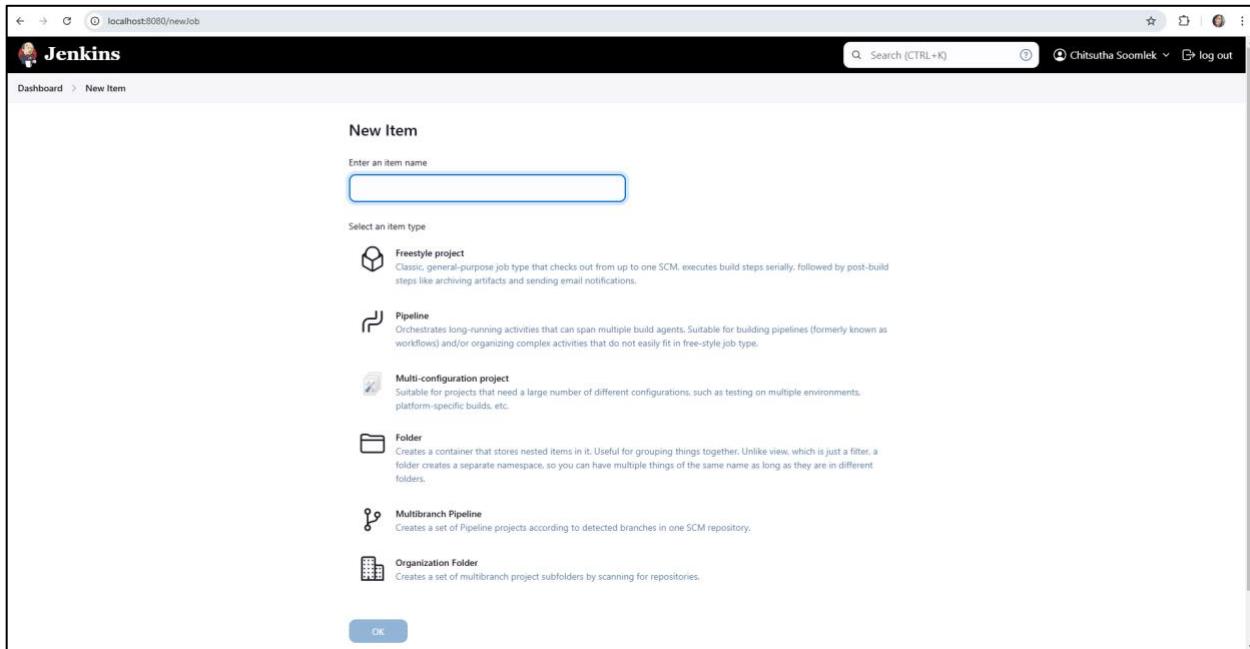
12. เมื่อติดตั้งสำเร็จจะพบกับรายการ Plugins ทั้งหมดที่ถูกติดตั้ง ถ้าติดตั้งสำเร็จให้เลือก “Restart Jenkins...” และกด Go back to the top page

13. สร้างไฟล์ Jenkinsfile ไม่มีนามสกุล เพื่อ execute คำสั่งต่าง ๆ กับ built-in agent และเอาไฟล์ดังกล่าวเก็บไว้ที่ root ของ GitHub Repository ของนักศึกษา

14. สร้าง folder ชื่อ tests/ บน GitHub Repository ของนักศึกษา และสร้างไฟล์ Lab8.robot และนำไฟล์ไปไว้ใน folder ที่สร้าง

15. กลับไปที่หน้า Dashboard ของ Jenkins และสร้าง Pipeline อย่างง่าย โดยกำหนด New item เป็น Freestyle project และตั้งชื่อเป็น UAT

## Lab Worksheet



16. ตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

**Description:** Lab 8.5

**GitHub project:** กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 14)

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

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

## Lab Worksheet

[Check point#15] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows the Jenkins 'Configure' screen for a job named 'UAT'. The 'Build Steps' section contains one 'Execute shell' step with the following command:

```
echo "Running Robot Framework..."  
# สำหรับเมื่อเริ่มทดสอบ  
md5dir -p results  
# นำ robot ที่มีอยู่ใน folder tests  
robot -d results tests/Lab8.robot
```

The 'Post-build Actions' section is currently empty. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

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

\_\_\_\_\_ robot -d results tests/Lab8.robot (robot คือคำสั่งเรียกโปรแกรม, -d results คือบอกให้เก็บไฟล์ report ไว้ในโฟลเดอร์ results, tests/Lab8.robot คือไฟล์ที่ต้องการทดสอบ) \_\_\_\_\_

**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)

17. กด Apply และ Save

18. สร้าง Build Now

## Lab Worksheet

## [Check point#16] Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output

Jenkins / UAT

Status: UAT

Changes: Lab 8.5

Workspace: [Configure](#)

Build Now: [Build Now](#)

Delete Project: [Delete Project](#)

Robot Results: [Robot Results](#)

Favorite: [Favorite](#)

Open Blue Ocean: [Open Blue Ocean](#)

Test Results Analyzer: [Test Results Analyzer](#)

Builds: #76 01:50, #75 01:45, #74 01:38, #72 01:35, #71 01:27

**Latest Robot Results:**

Total	Failed	Passed	Skipped	Pass %
All tests	1	0	1	100.0

- [Browse results](#)
- [Open report.html](#)
- [Open log.html](#)

**Permalinks:**

- Last build (#76), 5 min 46 sec ago
- Last stable build (#76), 5 min 46 sec ago
- Last successful build (#76), 5 min 46 sec ago
- Last failed build (#72), 20 min ago
- Last unsuccessful build (#72), 20 min ago
- Last completed build (#76), 5 min 46 sec ago

**Robot Framework Tests Trend (all tests)**

Number of test cases: 1  
Build: #76  
Legend: Skipped (Yellow), Passed (Green), Failed (Red)

Zoom to changes  Show only failed  all  Max builds [Show bigger image](#)

**Console Output:**

```

Started by timer
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/kriangkraili/jenkins-robot-lab.git # timeout=10
Fetching upstream changes from https://github.com/kriangkraili/jenkins-robot-lab.git
> git --version # timeout=10
> git --version # 'git version 2.47.3'
> git fetch --tags --force --progress -- https://github.com/kriangkraili/jenkins-robot-lab.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision fdd53d0156c70f844b84f37cecac855acbc78d7f (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f fdd53d0156c70f844b84f37cecac855acbc78d7f # timeout=10
Commit message: "Remove title assertion and clean Robot test"
> git rev-list --no-walk fdd53d0156c70f844b84f37cecac855acbc78d7f # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins16258025260001471454.sh
+ mkdir -p results
+ robot -d results tests/lab8.robot
=====
Lab8
=====
Open KKU Computing Website | PASS |
Lab8 | PASS |
1 test, 1 passed, 0 failed
=====
Output: /var/jenkins_home/workspace/UAT/results/output.xml
Log: /var/jenkins_home/workspace/UAT/results/log.html
Report: /var/jenkins_home/workspace/UAT/results/report.html
Robot results publisher started...
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: SUCCESS

```