

Lab Worksheet

ชื่อ-นามสกุล พิพัฒน์ มิตรเจริญรัตน์ รหัสนักศึกษา 653380208-8 Section 4

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 เอกำลัง
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

```
C:\Users\computer>mkdir Lab8_1
C:\Users\computer>cd Lab8_1

C:\Users\computer\Lab8_1>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

C:\Users\computer\Lab8_1>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
aiexam          latest        4b70f5809a07   3 months ago   7.64GB
busybox          latest        af4709625109   3 months ago   4.27MB
```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร aiexam , busybox
 (2) Tag ที่ใช้บ่งบอกถึงอะไร บ่งบอกถึง เวอร์ชัน ของ Docker Image ที่ใช้งานทั้ง aiexam และ busybox มี Tag เป็น latest ซึ่งหมายถึงเป็นเวอร์ชันล่าสุดที่ดึงมาจาก Docker Hub

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

```

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

C:\Users\computer>mkdir Lab8_1

C:\Users\computer>cd Lab8_1

C:\Users\computer\Lab8_1>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

C:\Users\computer\Lab8_1>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
aiexam          latest       4b70f5809a07   3 months ago   7.64GB
busybox         latest       af4709625109   3 months ago   4.27MB

C:\Users\computer\Lab8_1>docker run busybox
C:\Users\computer\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 23 04:21 .
drwxr-xr-x  1 root  root    4096 Jan 23 04:21 ..
-rw-r--r--  1 root  root     0 Jan 23 04:21 .dockerenv
drwxr-xr-x  2 root  root  12288 Sep 26 21:31 bin
drwxr-xr-x  5 root  root    360 Jan 23 04:21 dev
drwxr-xr-x  1 root  root    4096 Jan 23 04:21 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
drwxr-xr-x  391 root  root     0 Jan 23 04:21 proc
drwxr-xr-x  1 root  root    4096 Jan 23 04:21 root
dr-xr-xr-x  11 root  root     0 Jan 23 04:21 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

C:\Users\computer\Lab8_1>docker run busybox echo "Hello พี่ กูรู นี่ คระเจ๊ กูตี้ นะ from busybox"
"Hello พี่ กูรู นี่ คระเจ๊ กูตี้ นะ from busybox"

C:\Users\computer\Lab8_1>docker ps -a
CONTAINER ID        IMAGE           COMMAND          CREATED          STATUS          PORTS          NAMES
e81610da25bd        busybox         "echo \"Hello พี่ กูรู...\""  10 seconds ago   Exited (0) 9 seconds ago
b039e58b8737        busybox         "sh"            About a minute ago   Exited (0) About a minute ago
231d97c63362        busybox         "sh"            2 minutes ago    Exited (0) 2 minutes ago
61321e4fd7ef        aiexam:latest   "jupyter lab --ip=0..."  3 months ago    Exited (255) 15 minutes ago   0.0.0.0:8000->8000/tcp
naughty_swanson

C:\Users\computer\Lab8_1|

```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสั้นๆ

option -it มีความหมายดังนี้:

- i (interactive): ทำให้สามารถรับ Input จากผู้ใช้งานผ่าน Terminal ได้
- t (TTY): เปิดใช้งาน Terminal แบบจำลอง (pseudo-terminal) สำหรับใช้งานใน Container

ดังนั้น docker run -it ทำให้เราสามารถใช้งาน **Interactive mode** เพื่อโต้ตอบกับ Container โดยตรง

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a และถึงข้อมูลอะไร

- Container นั้นกำลัง รันอยู่

Lab Worksheet

- Container นั้น หยุดทำงาน

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

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

```
C:\Users\computer\Lab8_1>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e81610da25bd busybox "echo "Hello ทิ พ รุส..." 10 seconds ago Exited (0) 9 seconds ago
b039e58b8737 busybox "sh" About a minute ago Exited (0) About a minute ago
231d97c63362 busybox "sh" 2 minutes ago Exited (0) 2 minutes ago
61321e4fd7ef aiexam:latest "jupyter lab --ip=0..." 3 months ago Exited (255) 15 minutes ago 0.0.0.0:8000->8000/tcp naughty_swanson

C:\Users\computer\Lab8_1>docker rm b039e58b8737
b039e58b8737

C:\Users\computer\Lab8_1>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e81610da25bd busybox "echo "Hello ทิ พ รุส..." 10 minutes ago Exited (0) 10 minutes ago
231d97c63362 busybox "sh" 12 minutes ago Exited (0) 12 minutes ago
61321e4fd7ef aiexam:latest "jupyter lab --ip=0..." 3 months ago Exited (255) 25 minutes ago 0.0.0.0:8000->8000/tcp naughty_swanson
```

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

```
$ touch Dockerfile
แล้วใช้ Text Editor ในการใส่เนื้อหาแทน
```

Lab Worksheet

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) และแสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 5 พร้อมกับตอบคำถามต่อไปนี้

```
Command Prompt - CMD ect  X + ▾
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\computer>mkdir Lab8_2

C:\Users\computer>cd Lab8_2

C:\Users\computer\Lab8_2>FROM busybox
'FROM' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\computer\Lab8_2>CMD echo "Hi there. This is my first docker image."
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\computer\Lab8_2>CMD echo "พิ พ ฐ ร ษ ม ิ ตร า จ ร ิ ญ ช ร ์ ต น ់ 653380208-8 ឲបន ិ ស"
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\computer\Lab8_2>dir
 Volume in drive C is Win11GPT-hpgen11
 Volume Serial Number is 3AF9-E750

 Directory of C:\Users\computer\Lab8_2

01/23/2025  11:54 AM    <DIR>          .
01/23/2025  11:37 AM    <DIR>          ..
01/23/2025  11:54 AM           174 Dockerfile.txt
                           1 File(s)        174 bytes
                           2 Dir(s)   609,413,345,280 bytes free
```

Lab Worksheet

```
C:\Users\computer\Lab8_2>ren Dockerfile.txt Dockerfile

C:\Users\computer\Lab8_2>dir
Volume in drive C is Win11GPT-hpgen11
Volume Serial Number is 3AF9-E750

Directory of C:\Users\computer\Lab8_2

01/23/2025  12:01 PM      <DIR>          .
01/23/2025  11:37 AM      <DIR>          ..
01/23/2025  11:54 AM           174 Dockerfile
              1 File(s)       174 bytes
              2 Dir(s)  609,413,672,960 bytes free

C:\Users\computer\Lab8_2>docker build -t my-first-image .
[+] Building 0.0s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 213B
=> [internal] load .dockignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [1/1] FROM docker.io/library/busybox
=> exporting to image
=> => exporting layers
=> => writing image sha256:a7146a20321271b8873191b41edeffb4257cb1c81601a630e903b365d3a76390
=> => naming to docker.io/library/my-first-image

C:\Users\computer\Lab8_2>docker run my-first-image
พิ พ ขึ้น  มิ ตร เจริ  กษั ตโน  653380208-8  โภน  ส

C:\Users\computer\Lab8_2>
```

- (1) คำสั่งที่ใช้ในการ run คือ docker run my-first-image
- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

-t ย่อมาจาก tag ในคำสั่ง docker build

- ใช้กำหนดชื่อและแท็ก (tag) ให้กับ Docker Image ที่สร้างขึ้น
- รูปแบบทั่วไป: docker build -t <ชื่อ Image>:<tag> .
 - หากไม่ระบุ <tag> จะใช้ค่าเริ่มต้นเป็น latest

ผลที่ได้จากการใช้ -t:

1. ทำให้เราสามารถระบุชื่อ (name) และเวอร์ชัน (tag) ของ Image ได้ เช่น my-first-image:1.0
2. ช่วยให้จัดการและอ้างอิง Image ได้ง่ายขึ้นในคำสั่ง docker run, docker push, docker pull และคำสั่งอื่น ๆ

ตัวอย่าง: docker build -t my-first-image:1.0 .

- ชื่อของ Image คือ my-first-image
- Tag ของ Image คือ 1.0

Lab Worksheet

เมื่อสร้าง Image เสร็จ เราสามารถใช้คำสั่ง docker images เพื่อตรวจสอบชื่อและ Tag ของ Image ที่สร้างขึ้นได้

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

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

```

Command Prompt x + v
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\computer>mkdir Lab8_3

C:\Users\computer>cd Lab8_3

C:\Users\computer\Lab8_3>docker build -t pipatsorn/lab8 .

[+] Building 0.0s (2/2) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2B
=> [internal] load .dockerignore
=> => transferring context: 2B
0.0s
0.0s
0.0s
0.0s
ERROR: failed to solve: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount3083460511/Dockerfile: no such file or directory

C:\Users\computer\Lab8_3>ren Dockerfile.txt Dockerfile

C:\Users\computer\Lab8_3>dir
Volume in drive C is Win11GPT-hpgen11
Volume Serial Number is 3AF9-E750

Directory of C:\Users\computer\Lab8_3

01/23/2025 12:15 PM <DIR> .
01/23/2025 12:09 PM <DIR> ..
01/23/2025 12:12 PM 181 Dockerfile
1 File(s) 181 bytes
2 Dir(s) 609,391,620,096 bytes free

C:\Users\computer\Lab8_3>docker build -t pipatsorn/lab8 .

[+] Building 0.0s (5/5) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
0.0s
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 220B
0.0s
=> [internal] load metadata for docker.io/library/busybox:latest
0.0s
=> CACHED [1/1] FROM docker.io/library/busybox
0.0s
=> exporting to image
0.0s
=> => exporting layers
0.0s
=> => writing image sha256:ca46b9a474bc779fba5967555ca12bf7b55f2c48c93643fa56c00fb8cd98443
0.0s
=> => naming to docker.io/pipatsorn/lab8
0.0s

C:\Users\computer\Lab8_3>docker run pipatsorn/lab8
พิ พ ท ภ ร ต ภ ร ต 653380208-8

C:\Users\computer\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 แล้วป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้
 คำสั่ง
 \$ 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)

```

C:\Users\computer\Lab8_3>docker push pipatsorn/lab8
Using default tag: latest
The push refers to repository [docker.io/pipatsorn/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:bf3a3e3f7bfb56dfbc8a91e0e3dc5ff580a8e516fc8c1462cfacb2567209835 size: 528

C:\Users\computer\Lab8_3>

```

Lab Worksheet

Docker commands

```
docker push pipatsorn/lab8:tagname
```

Tags

Tag	OS	Type	Pulled	Pushed
latest		Image	3 minutes ago	4 minutes ago

[See all](#)

Automated builds

Manually pushing Images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

Repository overview

An overview describes what your image does and how to run it. It displays in [the public view of your repository](#) once you have pushed some content.

[Add overview](#)

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

[https://hub.docker.com/repository/docker/pipatsorn/lab8/general](#)

Tags

TAG	Digest	OS/ARCH	Last pull
latest	b7fb	linux/amd64	4 minutes ago

[Copy](#)

Compressed size 2.06 MB

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

[Cookies Settings](#) [Reject All](#) [Accept All Cookies](#)

แบบฝึกปฏิบัติที่ 8.4: การ Build และ Update แอปพลิเคชันจาก Container image

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
2. ทำการ Clone ซอฟต์แวร์โค้ดของเว็บแอปพลิเคชันจาก GitHub repository
<https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง

Lab Worksheet

\$ 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

```
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vivobook>mkdir Lab8_4

C:\Users\Vivobook>cd Lab8_4

C:\Users\Vivobook\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.
remote: Total 980 (delta 5), reused 1 (delta 1), pack-reused 971 (from 2)
Receiving objects: 100% (980/980), 5.28 MiB | 830.00 KiB/s, done.
Resolving deltas: 100% (523/523), done.

C:\Users\Vivobook\Lab8_4>cd getting-started/app

C:\Users\Vivobook\Lab8_4\getting-started\app>dir
Volume in drive C is OS
Volume Serial Number is BA0B-C687

Directory of C:\Users\Vivobook\Lab8_4\getting-started\app

01/28/2025  03:40 PM    <DIR>          .
01/28/2025  03:40 PM    <DIR>          ..
01/28/2025  03:40 PM                  678 package.json
01/28/2025  03:40 PM    <DIR>          spec
01/28/2025  03:40 PM    <DIR>          src
01/28/2025  03:40 PM            150,541 yarn.lock
              2 File(s)       151,219 bytes
              4 Dir(s)   179,297,165,312 bytes free

C:\Users\Vivobook\Lab8_4\getting-started\app>notepad package.json
```

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"
  },
  "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"
  }
}
```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงในไฟล์


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

Lab Worksheet

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น myapp_รหัส
ศ. ไม่มีขีด

\$ docker build -t <myapp_รหัสศ. ไม่มีขีด> .

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

```
C:\Users\Vivobook\Lab8_4\getting-started\app>notepad Dockerfile

C:\Users\Vivobook\Lab8_4\getting-started\app>docker build -t myapp_6533802088 .
[+] Building 0.2s (1/1) FINISHED docker:desktop-linux
=> [internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 2B                            0.0s
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory

C:\Users\Vivobook\Lab8_4\getting-started\app>dir
Volume in drive C is OS
Volume Serial Number is BA0B-C687

Directory of C:\Users\Vivobook\Lab8_4\getting-started\app

01/28/2025  03:50 PM    <DIR>      .
01/28/2025  03:40 PM    <DIR>      ..
01/28/2025  03:51 PM            119 Dockerfile.txt
01/28/2025  03:40 PM            678 package.json
01/28/2025  03:40 PM    <DIR>      spec
01/28/2025  03:40 PM    <DIR>      src
01/28/2025  03:40 PM            150,541 yarn.lock
            3 File(s)        151,338 bytes
            4 Dir(s)   179,254,976,512 bytes free

C:\Users\Vivobook\Lab8_4\getting-started\app>ren Dockerfile.txt Dockerfile
```

Lab Worksheet

```
C:\Users\Vivobook\Lab8_4\getting-started\app>docker build -t myapp_6533802088 .
[+] Building 44.6s (10/10) FINISHED                               docker:desktop-linux
=> [internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 156B                           0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 3.8s
=> [auth] library/node:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                            0.1s
=> => transferring context: 2B                            0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cb 16.7s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc 0.1s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639 444B / 444B 0.8s
=> => sha256:5650d6de56fd0bb419872b876ac1df28f57 1.26MB / 1.26MB 2.9s
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b 3.64MB / 3.64MB 6.0s
=> => sha256:37892ffbfc当地871a10f813803949d18c 40.01MB / 40.01MB 15.3s
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e 0.1s
=> => extracting sha256:37892ffbfc当地871a10f813803949d18c3015a482 0.8s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b395 0.1s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d126398 0.0s
=> [internal] load build context                          1.0s
=> => transferring context: 4.62MB                      0.9s
=> [2/4] WORKDIR /app                                0.1s
=> [3/4] COPY . . .                                 0.1s
=> [4/4] RUN yarn install --production            17.9s
=> exporting to image                                5.6s
=> => exporting layers                             3.8s
=> => exporting manifest sha256:0bf3492d13a511b11ed15cc07155ceb6 0.0s
=> => exporting config sha256:9017fde37137b8d43e4611e157bdac89b8 0.0s
=> => exporting attestation manifest sha256:1884eea0337a0d40087d 0.1s
=> => exporting manifest list sha256:08a7da725f43883ee9f50d2c9b4 0.0s
=> => naming to docker.io/library/myapp_6533802088:latest 0.0s
=> => unpacking to docker.io/library/myapp_6533802088:latest 1.6s

C:\Users\Vivobook\Lab8_4\getting-started\app>docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
myapp_6533802088   latest   08a7da725f43   About a minute ago  342MB
hello-world         latest   d715f14f9eca   6 days ago   20.4kB
```

Lab Worksheet

```

# Install the base requirements for the app.
# This stage is to support development.
FROM --platform=$BUILDPLATFORM python:alpine AS base
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt

FROM --platform=$BUILDPLATFORM node:18-alpine AS app-base
WORKDIR /app
COPY app/package.json app/yarn.lock ./ 
COPY app/spec ./spec
COPY app/src ./src

# Run tests to validate app
FROM app-base AS test
RUN yarn install
RUN yarn test

# Clear out the node_modules and create the zip
FROM app-base AS app-zip-creator
COPY --from=test /app/package.json /app/yarn.lock ./ 
COPY app/spec ./spec
COPY app/src ./src
RUN apk add zip && \
    zip -r /app.zip /app

# Dev-ready container - actual files will be mounted in
FROM --platform=$BUILDPLATFORM base AS dev
CMD ["mkdocs", "serve", "-a", "0.0.0.0:8000"]

# Do the actual build of the mkdocs site
FROM --platform=$BUILDPLATFORM base AS build
COPY . .
RUN mkdocs build

# Extract the static content from the build
# and use a nginx image to serve the content
FROM --platform=$TARGETPLATFORM nginx:alpine
COPY --from=app-zip-creator /app.zip /usr/share/nginx/html/assets/app.zip
COPY --from=build /app/site /usr/share/nginx/html

```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

Lab Worksheet

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

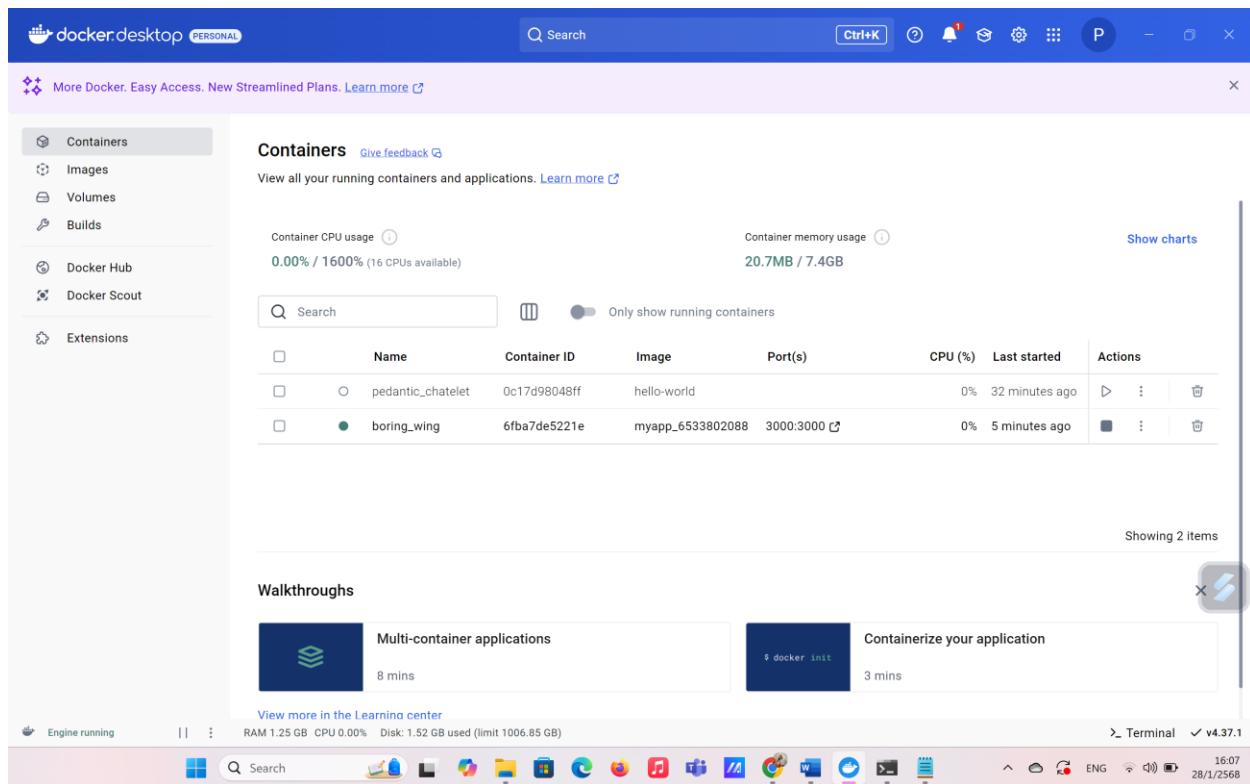
```
C:\Users\Vivobook\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533802088
6fba7de5221e1969298ff1af90068dfa72183dca5c3dc0b578bd3c4dbfb5a4c2

C:\Users\Vivobook\Lab8_4\getting-started\app>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
6fba7de5221e        myapp_6533802088   "docker-entrypoint.s..."   32 seconds ago    Up 31 seconds     0.0.0.0:3000->3000/tcp   boring_wing

C:\Users\Vivobook\Lab8_4\getting-started\app>
```

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several icons and a search bar. An open browser window displays a simple 'Todo App' interface with a green 'Add Item' button and a message 'No items yet! Add one above!'. Above the browser, a command-line terminal window shows the execution of Docker commands to run an application and list containers. The Docker Desktop interface is visible in the background, showing the running container details.

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

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

Lab Worksheet

```
C:\Users\Vivobook\Lab8_4\getting-started\app>cd src\static\js
The system cannot find the path specified.

C:\Users\Vivobook\Lab8_4\getting-started\app>notepad app.js

C:\Users\Vivobook\Lab8_4\getting-started\app>cd ../../..
C:\Users\Vivobook\Lab8_4\getting-started\app>docker build -t myapp_6533802088 .
[+] Building 23.2s (10/10) FINISHED
   docker:desktop-linux
=> [internal] load build definition from Dockerfile          0.0s
=> => transferring dockerfile: 156B                          0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 2.4s
=> [auth] library/node:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                            0.0s
=> => transferring context: 2B                            0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc 0.1s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc 0.1s
=> [internal] load build context                         0.1s
=> => transferring context: 8.14kB                      0.1s
=> CACHED [2/4] WORKDIR /app                           0.0s
=> [3/4] COPY . .                                     0.1s
=> [4/4] RUN yarn install --production                14.5s
=> exporting to image                                  5.8s
=> => exporting layers                                4.0s
=> => exporting manifest sha256:f8c9dac7ccd09c5b5472fb0bfaf4042e 0.0s
=> => exporting config sha256:a39f2839a92024195ade39dc341ca20330 0.0s
=> => exporting attestation manifest sha256:33821bd5ccb14ac22c72 0.1s
=> => exporting manifest list sha256:b3d8262d0a6e74967c47cb6ffa4 0.0s
=> => naming to docker.io/library/myapp_6533802088:latest 0.0s
=> => unpacking to docker.io/library/myapp_6533802088:latest 1.6s

C:\Users\Vivobook\Lab8_4\getting-started\app>docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
6fba7de5221e        08a7da725f43   "docker-entrypoint.s..."   15 minutes ago   Up 15 minutes   0.0.0.0:3000->3000/tcp   boring_wing

C:\Users\Vivobook\Lab8_4\getting-started\app>
C:\Users\Vivobook\Lab8_4\getting-started\app>docker stop <container_id>
The syntax of the command is incorrect.
C:\Users\Vivobook\Lab8_4\getting-started\app>

C:\Users\Vivobook\Lab8_4\getting-started\app>docker stop 6fba7de5221e
6fba7de5221e

C:\Users\Vivobook\Lab8_4\getting-started\app>docker rm 6fba7de5221e
6fba7de5221e

C:\Users\Vivobook\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533802088
cd61c2f6bcfedf1be4800363200c5c7ea6847cfe6484674e448992ff1b7e8791
|
C:\Users\Vivobook\Lab8_4\getting-started\app>
```

Lab Worksheet

```

File Edit View
    ...items.slice(index + 1),
  ],
},
[items],
);

const onItemRemoval = React.useCallback(
  item => {
    const index = items.findIndex(i => i.id === item.id);
    setItems([...items.slice(0, index), ...items.slice(index + 1)]);
},
[items],
);

if (items === null) return 'Loading...';

return (
  <React.Fragment>
    <AddItemForm onNewItem={onNewItem} />
    {items.length === 0 && (
      <p className="text-center">There is no TODO item. Please add one to the list. By [พิพัฒน์ มีตรเจริญรัตน์]</p>
    )}
    {items.map(item => (
      <ItemDisplay
        item={item}
        key={item.id}
        onItemUpdate={onItemUpdate}
        onItemRemoval={onItemRemoval}
      />
    ))}
  </React.Fragment>
);
}

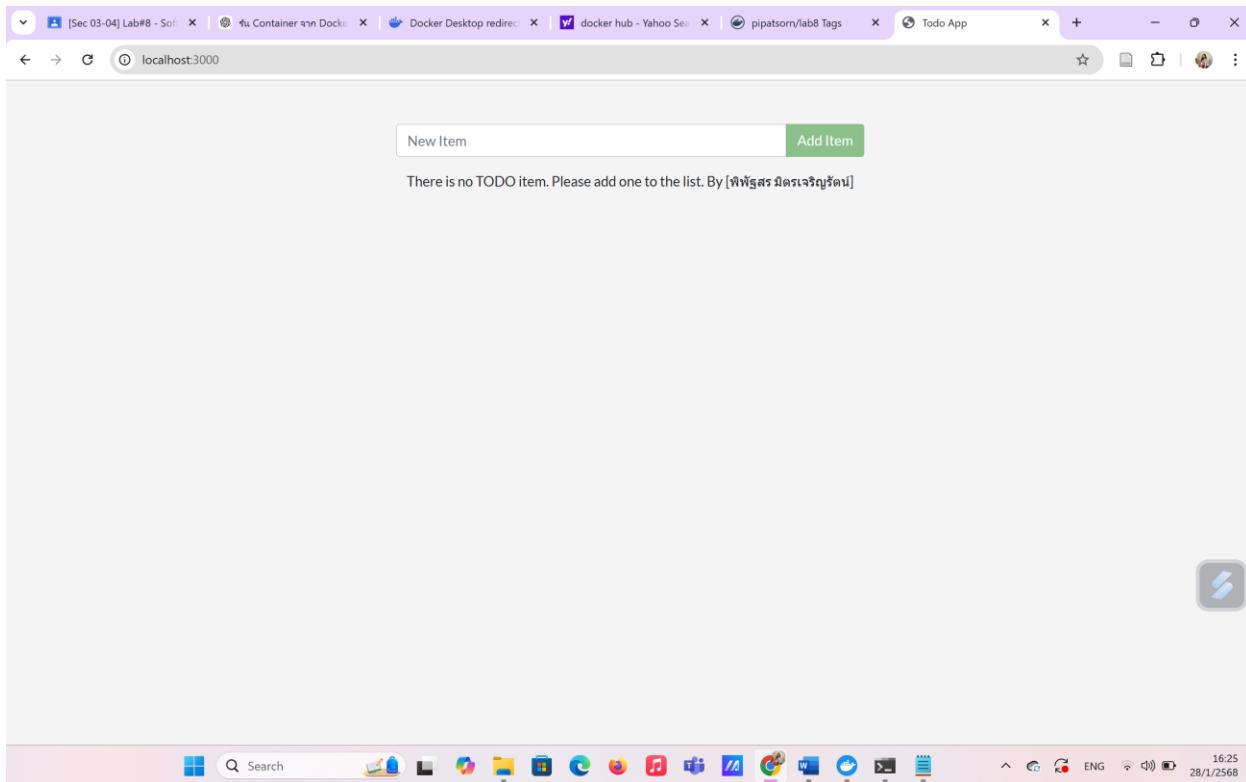
function AddItemForm({ onNewItem }) {
  const { Form, InputGroup, Button } = ReactBootstrap;

  const [newItem, setNewItem] = React.useState('');
  const [submitting, setSubmitting] = React.useState(false);

  const submitNewItem = e => {
    e.preventDefault();
    setSubmitting(true);
}

```

Lab Worksheet



(1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร
 เพราะ build ของ 8.9 ซึ่งเดียวกับ 8.5

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้เครื่องมือที่ต้องต่อไปนี้
 a. ผ่าน Command line interface

- i. ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
- 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. เลือกไอคอนถังขยะในแถบของ Container ที่ต้องการจะลบ
- iii. ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกรอบ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

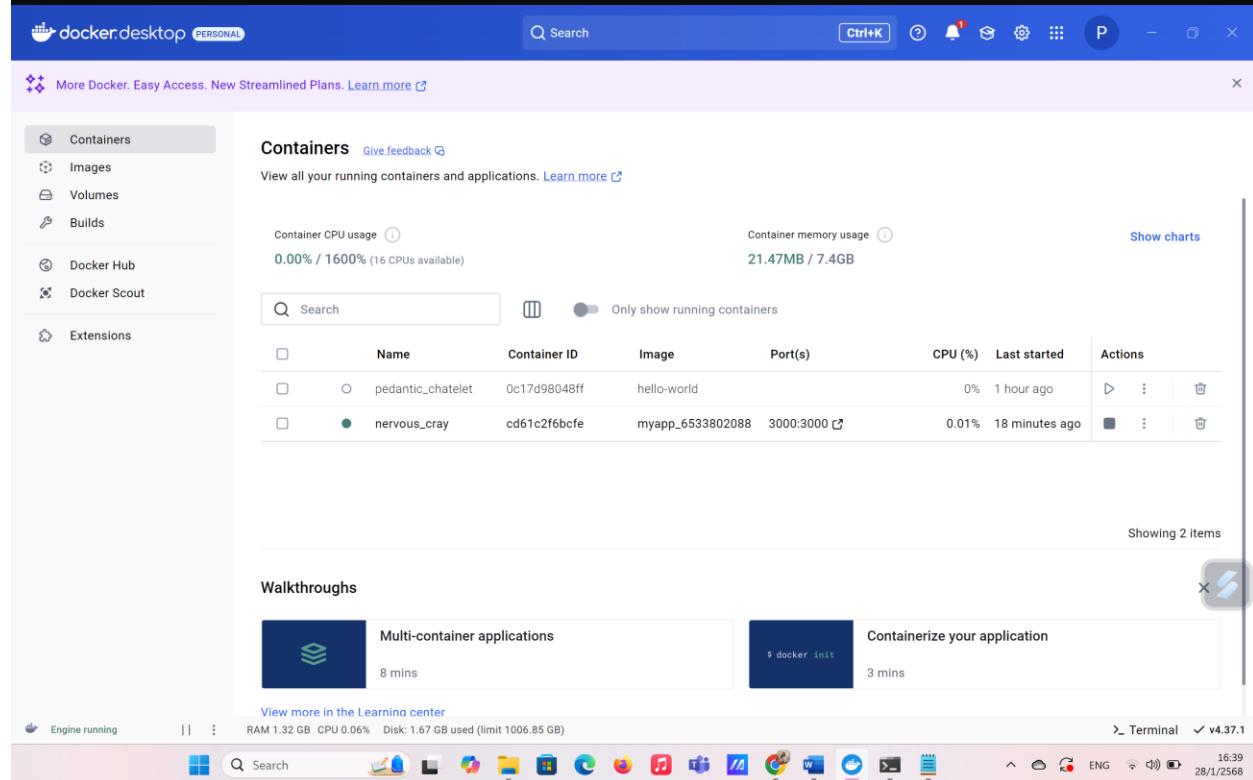
[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

Lab Worksheet

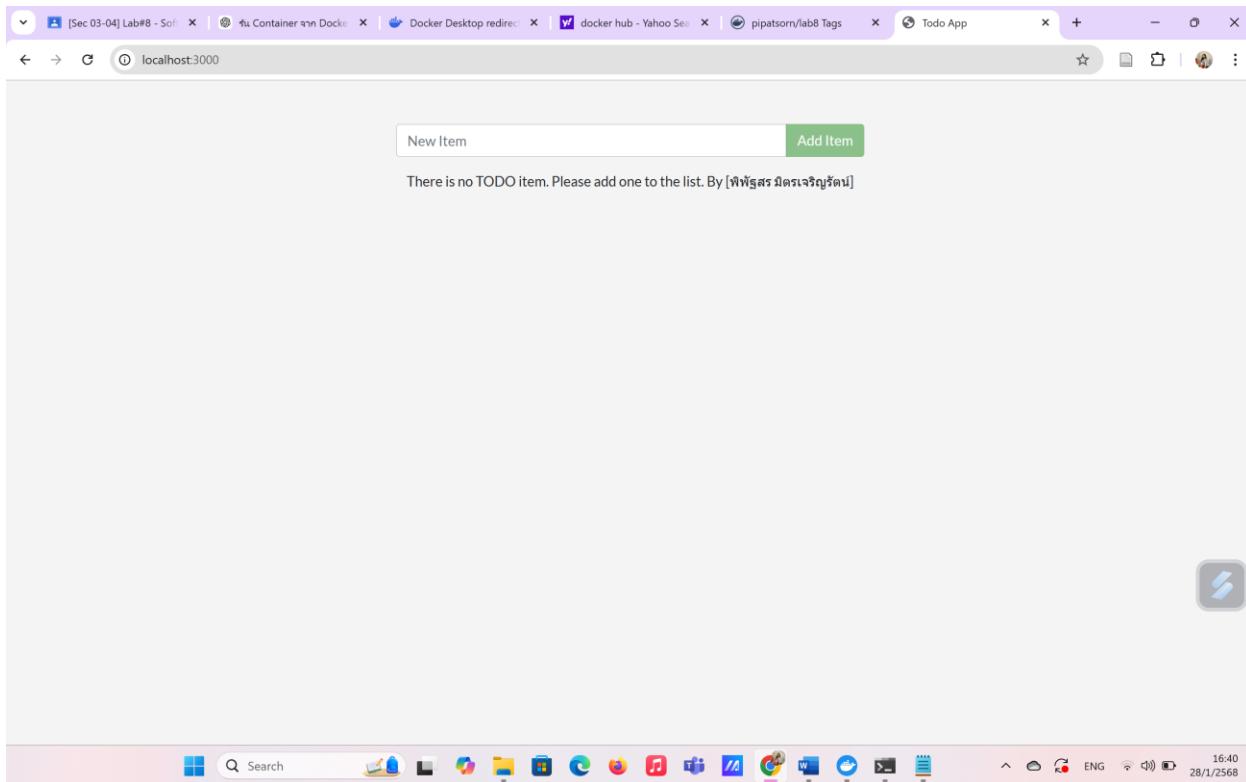
```
C:\Users\Vivobook\Lab8_4\getting-started\app>
C:\Users\Vivobook\Lab8_4\getting-started\app>docker stop <container_id>
The syntax of the command is incorrect.
C:\Users\Vivobook\Lab8_4\getting-started\app>

C:\Users\Vivobook\Lab8_4\getting-started\app>docker stop 6fba7de5221e
6fba7de5221e

C:\Users\Vivobook\Lab8_4\getting-started\app>docker rm 6fba7de5221e
6fba7de5221e
```



Lab Worksheet



แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop
2. ป้อนคำสั่งและทำการรัน container โดยผ่านพอร์ต
 - \$ 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
[Check point#12] Capture หน้าจอที่แสดงผล Admin password

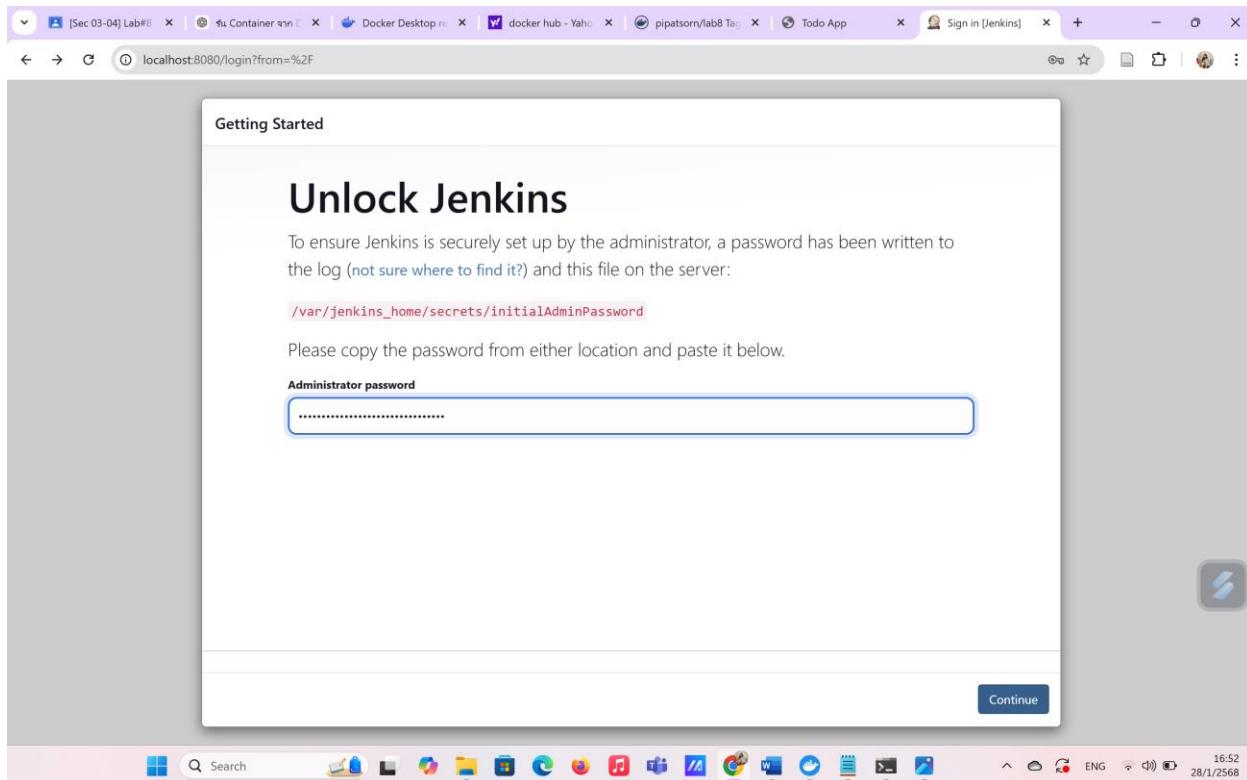
Lab Worksheet

```

Command Prompt - docker r  +  ~
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

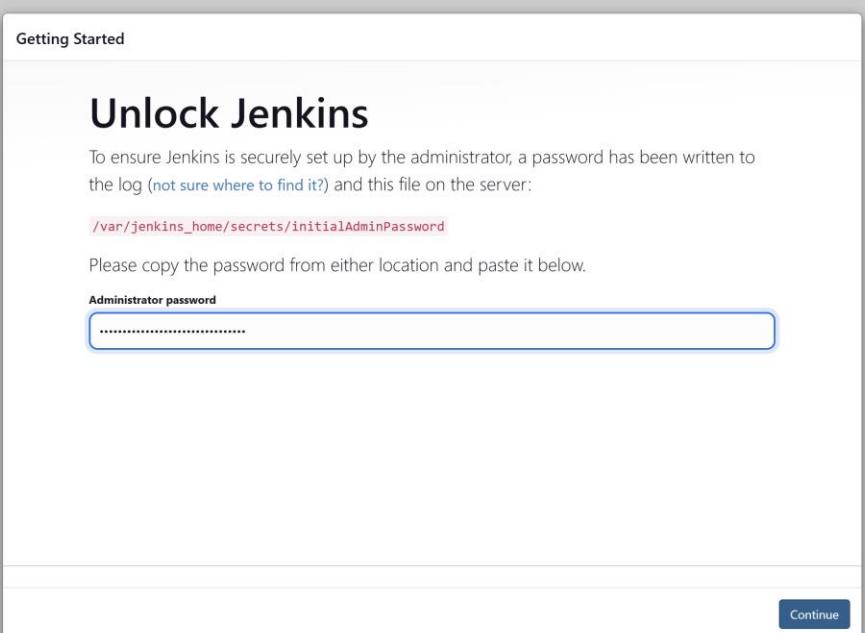
C:\Users\Vivobook>docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk17
Unable to find image 'jenkins/jenkins:lts-jdk17' locally
lts-jdk17: Pulling from jenkins/jenkins
3cfa79b75794: Download complete
0a96bdb82805: Download complete
c09c222384de: Download complete
1a17773e6697: Download complete
d281eadf0ae3: Download complete
b535e1c5efef: Download complete
b5e2db483aae: Download complete
be7ab5dc59d: Download complete
a91f2a3bb141: Download complete
72b137db2cdf: Download complete
c48e3974586c: Download complete
d0aac1e9e20c: Download complete
Digest: sha256:dc56634cc8fa476f36eba16d7db6c6bc1f5e3c7062432256738a008a339ee95c
Status: Downloaded newer image for jenkins/jenkins:lts-jdk17
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-01-28 09:43:22.067+0000 [id=1]     INFO    winstome.Logger#logInternal: Beginning extraction from war file
2025-01-28 09:43:22.780+0000 [id=1]     WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-01-28 09:43:22.826+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-12.0.16; built: 2024-12-09T21:02:54.535Z; git: c3f88b
afbf4e393f23204dc14dc57b092e84debc7; jvm 17.0.13+11
2025-01-28 09:43:23.199+0000 [id=1]     INFO    o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.j
etty.ee9.jsp.JettyJspServlet
2025-01-28 09:43:23.243+0000 [id=1]     INFO    o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-01-28 09:43:23.676+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.ma
sterEnvVars.get("JENKINS_HOME")
2025-01-28 09:43:23.497+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@772861aa{Je
nkins_v2.479.3.,/,,b=file:///var/jenkins_home/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreToNestedHandler@6631cb64[STARTED]}
2025-01-28 09:43:23.797+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@44550792[HTTP/1.1, (http/1.1){0.0.0
.0:8080}]
2025-01-28 09:43:23.813+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started oejs.Server@1c7fd41f{STARTING}[12.0.16,st0=0] @2458
ms
2025-01-28 09:43:23.814+0000 [id=35]     INFO    winstome.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-01-28 09:43:23.986+0000 [id=43]     INFO    jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-01-28 09:43:24.002+0000 [id=63]     INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-01-28 09:43:24.641+0000 [id=72]     INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-01-28 09:43:24.645+0000 [id=72]     INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-01-28 09:43:24.646+0000 [id=52]     INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-01-28 09:43:24.799+0000 [id=59]     INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-01-28 09:43:24.800+0000 [id=59]     INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-28 09:43:24.801+0000 [id=51]     INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-28 09:43:24.803+0000 [id=51]     INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-28 09:43:24.823+0000 [id=85]     INFO    hudson.util.Retriger#start: Attempt #1 to do the action check updates server
2025-01-28 09:43:25.162+0000 [id=62]     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:
dcfab9a8650147f58d590d3823a27219
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****
2025-01-28 09:43:31.549+0000 [id=62]     INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-28 09:43:31.569+0000 [id=33]     INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-28 09:43:34.053+0000 [id=85]     INFO    hudson.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenIn
staller
2025-01-28 09:43:34.054+0000 [id=85]     INFO    hudson.util.Retriger#start: Performed the action check updates server successfully at the attempt #1
|
```

Lab Worksheet



4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดбраузอร์ 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



Getting Started

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:

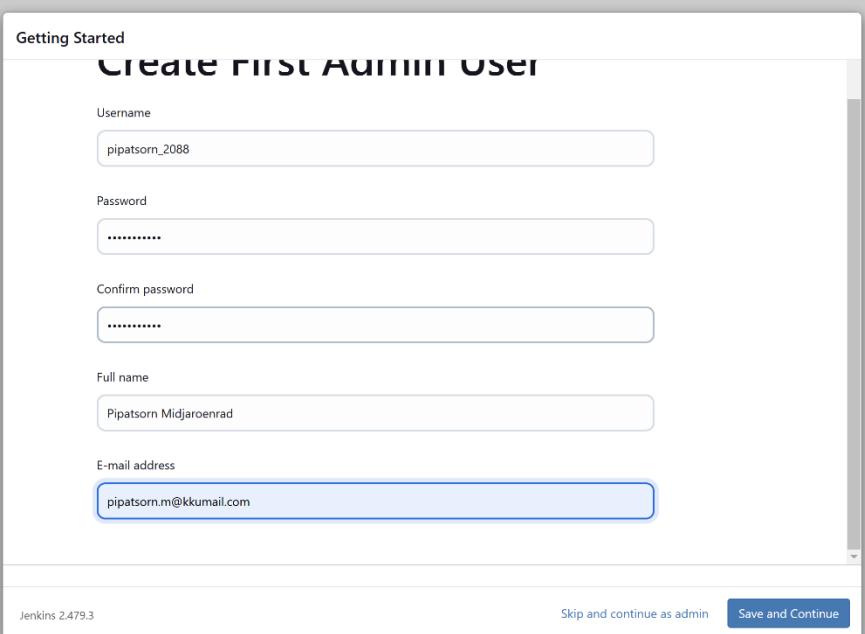
```
/var/jenkins_home/secrets/initialAdminPassword
```

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

Administrator password

.....

Continue



Getting Started

Create first Admin User

Username

pipatsorn_2088

Password

.....

Confirm password

.....

Full name

Pipatsorn Midjaroenrad

E-mail address

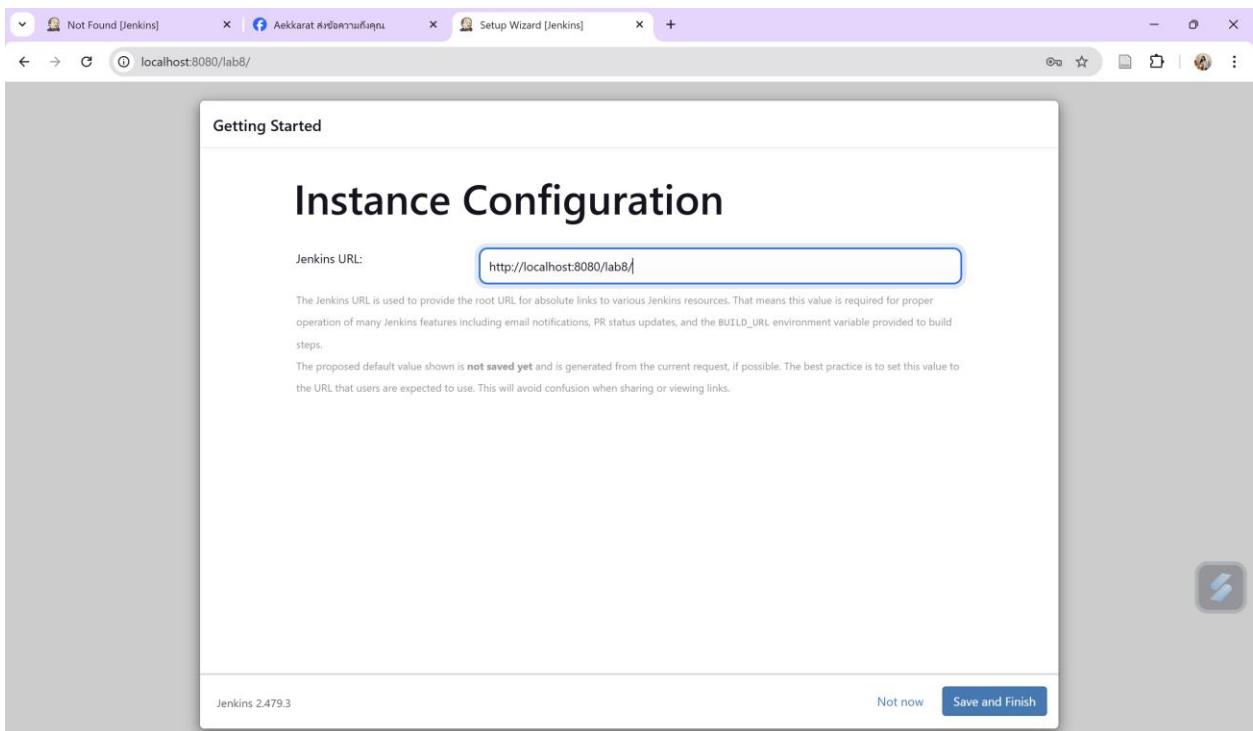
pipatsorn.m@kkumail.com

Jenkins 2.479.3

Skip and continue as admin

Save and Continue

Lab Worksheet

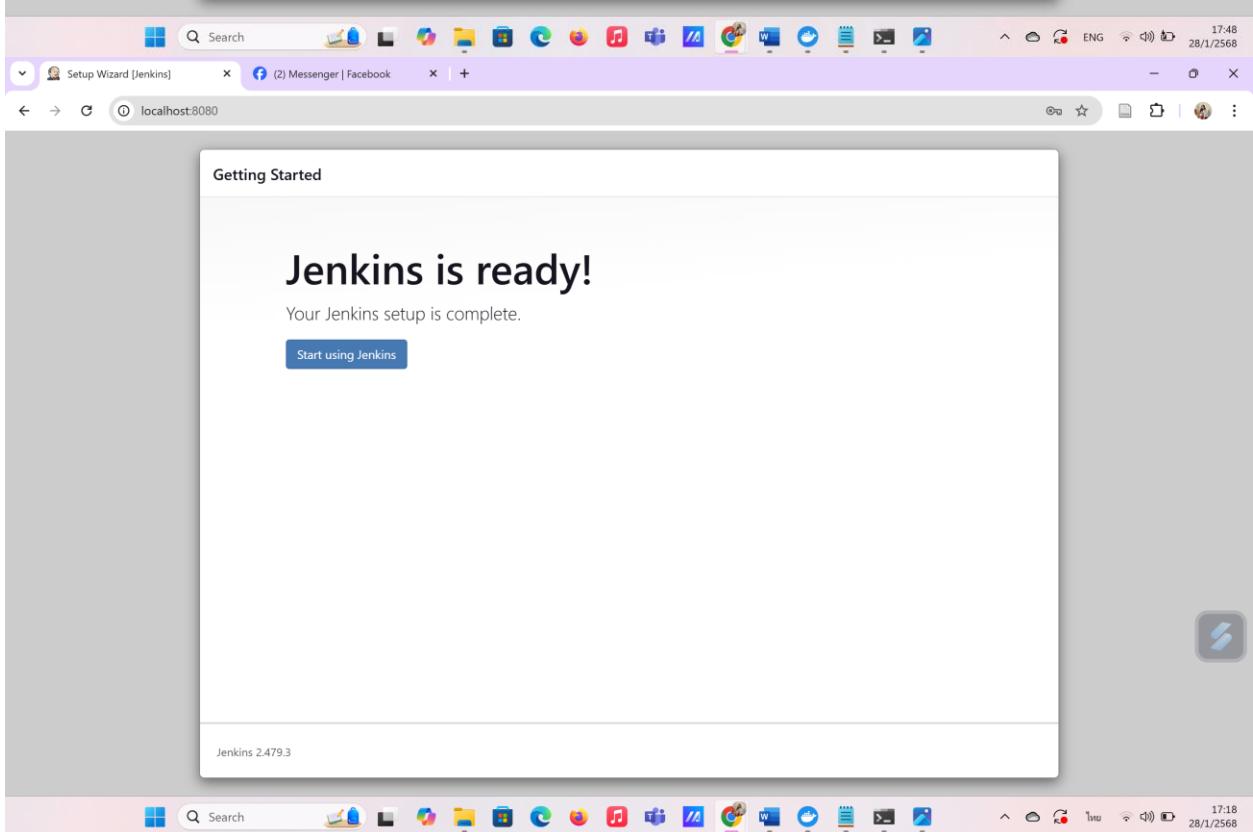


Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.479.3 Not now Save and Finish



Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.479.3

Lab Worksheet

The screenshot shows the Jenkins Dashboard at localhost:8080/lab8/. The page title is "Welcome to Jenkins!". It features a sidebar with links for "New Item", "Build History", "Manage Jenkins", and "My Views". Below the sidebar are two main sections: "Build Queue" (empty) and "Build Executor Status" (0/2). To the right, there's a "Start building your software project" section with "Create a job" and "Set up a distributed build" buttons, along with links for "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". The bottom right corner shows the Jenkins logo.

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

This screenshot of the Jenkins Dashboard is identical to the one above, showing the "Welcome to Jenkins!" page. It includes the same sidebar, "Build Queue" (empty), "Build Executor Status" (0/2), and the "Start building your software project" section with its various options. The bottom right corner again shows the Jenkins logo.

9. เลือก Manage Jenkins และไปที่เมนู Plugins

Lab Worksheet

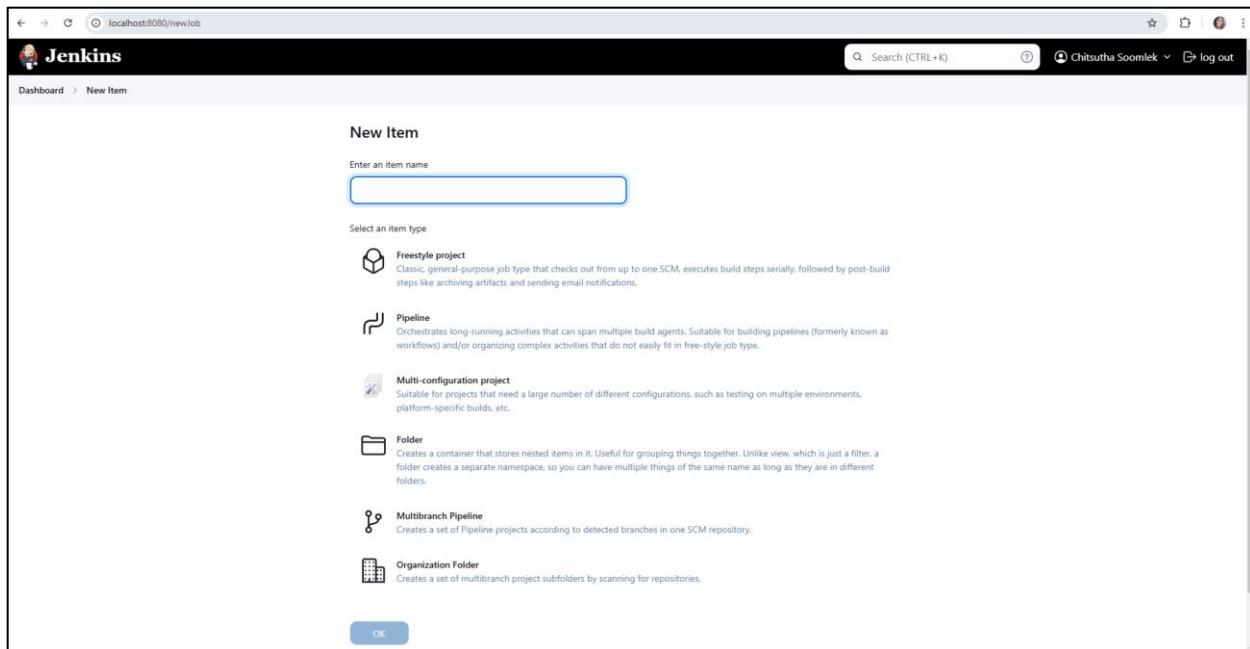
The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, Build History, Manage Jenkins (which is selected), and My Views. The main area is titled 'Manage Jenkins' and contains several sections: 'System Configuration' (Build Queue, Build Executor Status, System, Tools, Plugins, Nodes, Clouds, Appearance); 'Security' (Security, Credentials, Credential Providers, Users); and 'Status Information' (System Information, System Log, Load Statistics, About Jenkins). A message at the top says 'It appears that your reverse proxy set up is broken.' with 'More Info' and 'Dismiss' buttons.

10. ไปที่เมนู Available plugins และเลือกติดตั้ง Robotframework เพิ่มเติม

The screenshot shows the Jenkins Plugin Manager. The left sidebar has links for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area is titled 'Plugins' and shows a search bar with 'robot'. A table lists available plugins: 'Robot Framework 5.0.0' by 'Build Reports' (Released 2 mo 7 days ago). There's also a note: 'This publisher stores Robot Framework test reports for builds and shows summaries of them in project and build views along with trend graph.' An 'Install' button is visible at the top right of the plugin row.

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

Lab Worksheet



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปวิ่งบน Repository ของนักศึกษา จนนั่นตั้งค่าที่
จำเป็นในหน้านี้ทั้งหมด ดังนี้

Description:

```

Started by user Pipatsorn Midjaroenrad
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  Cloning repository https://github.com/Pipatsorn/Lab8.git
    > git init /var/jenkins_home/workspace/UAT # timeout=10
    > git --version # timeout=10
    > git --version # 'git version 2.39.5'
    > git fetch --tags --force --progress -- https://github.com/Pipatsorn/Lab8.git +refs/heads/*:refs/remotes/origin/* # timeout=10
    > git config remote.origin.url https://github.com/Pipatsorn/Lab8.git # timeout=10
    > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b4b31a1ce056e97c015e6e06ad09e052a378beb2 (refs/remotes/origin/main)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f b4b31a1ce056e97c015e6e06ad09e052a378beb2 # timeout=10
Commit message: "Add test cases for Lab7"
First time build. Skipping changelog.
[UAT] $ /bin/sh -xe /tmp/jenkins4707446537332842444.sh
+ robot UAT-LAB7.robot
/tmp/jenkins4707446537332842444.sh: 2: robot: not found

```

GitHub project: กดเลือกแล้วใส่ 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 ไปถึงไฟล์ให้เรียบร้อยด้วย)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows a GitHub repository named "Lab8". The repository page lists the following files:

- README.md: Initial commit, 1 hour ago
- UAT-LAB7-1.robot: "Add test cases for Lab7", 1 hour ago
- UAT-LAB7-2.robot: "Add test cases for Lab7", 1 hour ago
- UAT-LAB7.robot: "Add test cases for Lab7", 1 hour ago
- gherkin_login.robot: "Add test cases for Lab7", 1 hour ago
- invalid_login.robot: "Add test cases for Lab7", 1 hour ago
- resource.robot: "Add test cases for Lab7", 1 hour ago
- valid_login.robot: "Add test cases for Lab7", 1 hour ago

The repository has 2 commits and 1 branch. The "About" section is empty. The "Releases" and "Packages" sections also show no content.

Lab Worksheet

The screenshot shows two stacked Jenkins configuration pages for a job named "UAT".

Top Page (General Settings):

- General:** Enabled (checkbox checked).
- Description:** Lab 8.5.
- Source Code Management:** GitHub project selected, Project url: https://github.com/Pipatsorn/Lab8/.
- Build Triggers:** Discard old builds (checkbox checked).
- Build Environment:** None.
- Build Steps:** None.
- Post-build Actions:** None.

Bottom Page (Source Code Management):

- Source Code Management:** Git selected.
- Repositories:**
 - Repository URL: https://github.com/Pipatsorn/Lab8.git
 - Credentials: - none -
 - Add Repository button.
 - Branches to build: None.

Lab Worksheet

The screenshot shows the Jenkins configuration interface for a job named "UAT".

Source Code Management:

- Branch Specifier (blank for 'any'): */main
- Repository browser: (Auto)

Build Triggers:

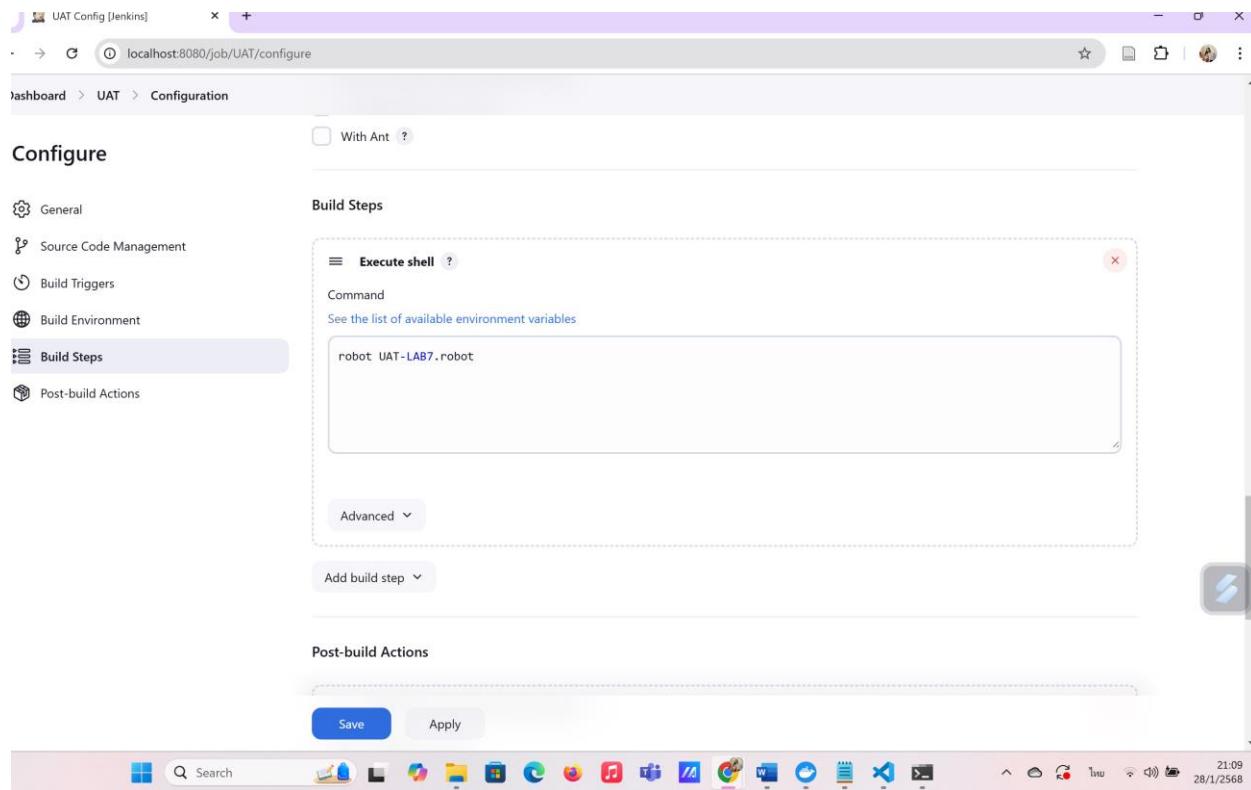
- Trigger builds remotely (e.g., from scripts):
- Build after other projects are built:
- Build periodically:
 - Schedule: H/15 * * * *
 - Note: Would last have run at Tuesday, January 28, 2025 at 2:05:53 PM Coordinated Universal Time; would next run at Tuesday, January 28, 2025 at 2:20:53 PM Coordinated Universal Time.
- GitHub hook trigger for GITScm polling:
- Poll SCM:

Build Environment:

- Delete workspace before build starts:
- Use secret text(s) or file(s):

Buttons: Save, Apply

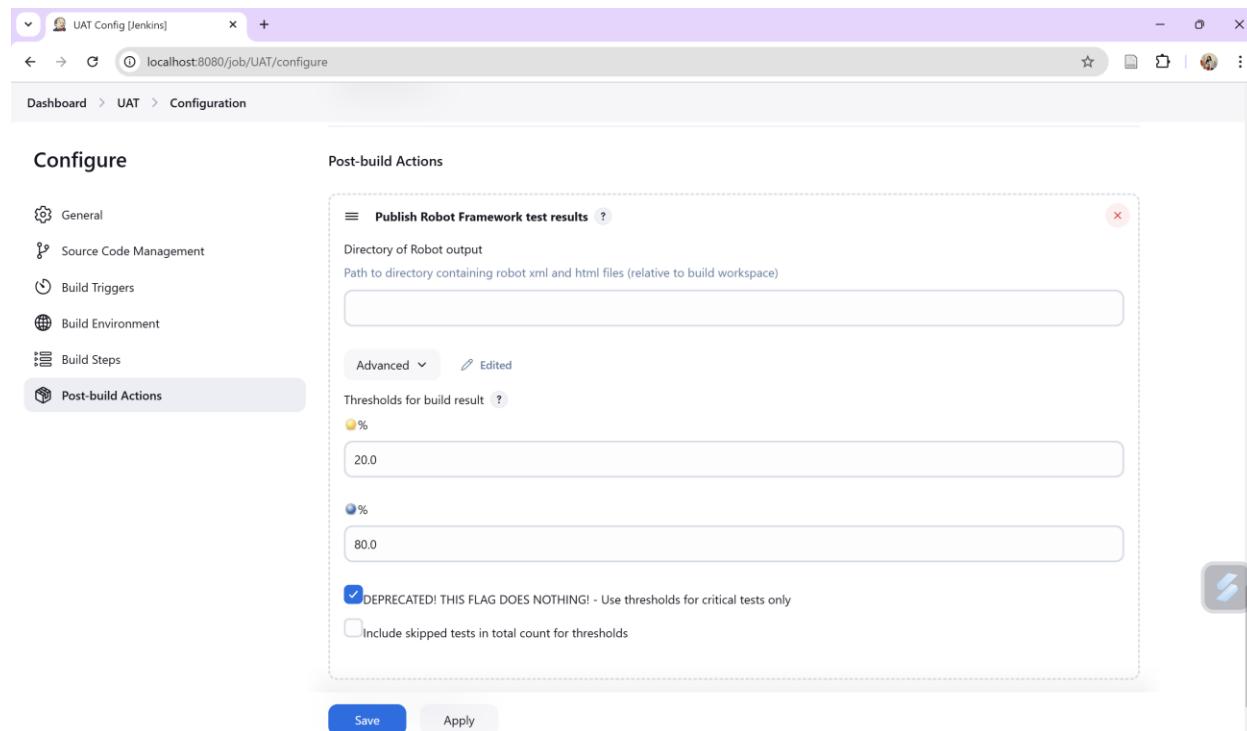
Lab Worksheet



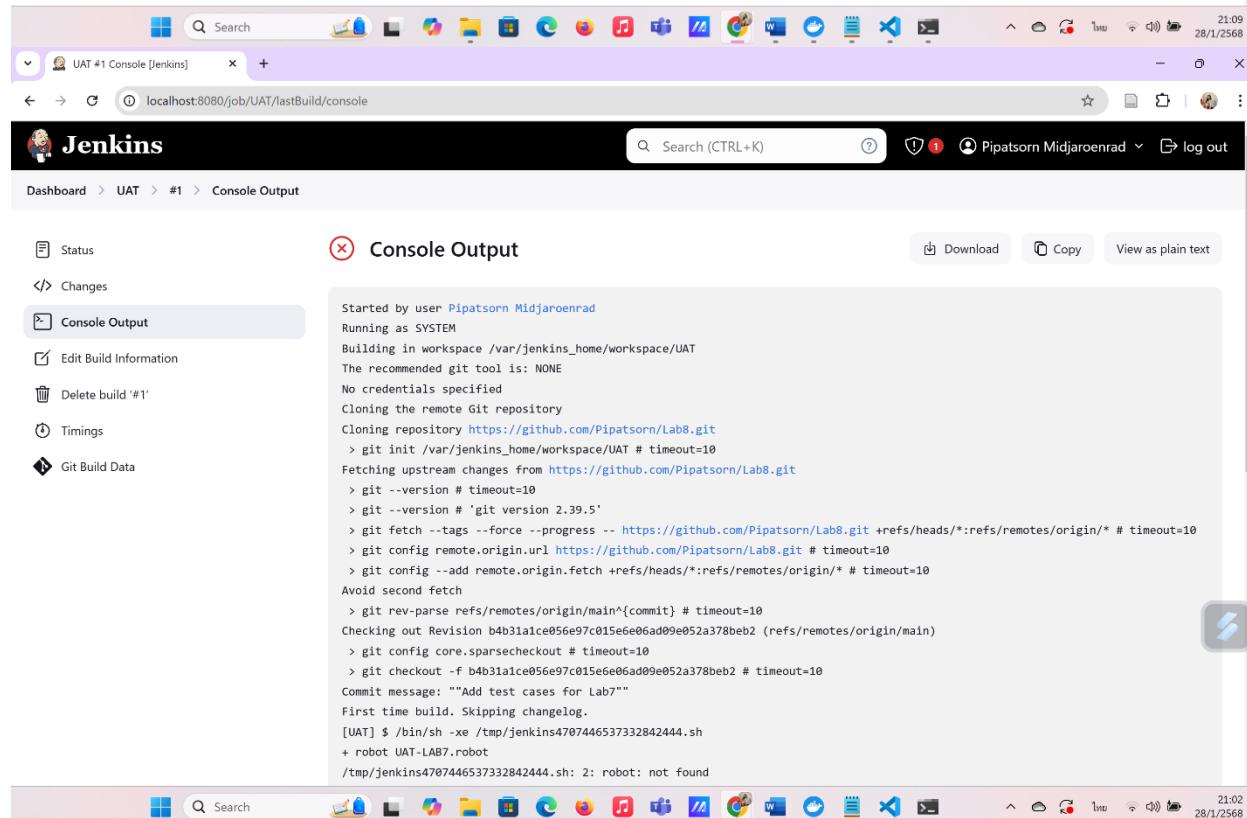
- (1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ robot UAT-LAB7.robot
 - (2) **Post-build action:** เพิ่ม Publish Robot Framework test results -> ระบุไดเร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)
13. กด Apply และ Save
 14. สร้าง Build Now

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

Lab Worksheet



The screenshot shows the Jenkins configuration interface for the 'UAT' job. The 'Post-build Actions' section is selected. It contains a configuration for 'Publish Robot Framework test results'. The 'Directory of Robot output' field is empty. Under 'Thresholds for build result', there are two entries: '20.0' and '80.0'. A checked checkbox says 'DEPRECATED! THIS FLAG DOES NOTHING - Use thresholds for critical tests only'. An unchecked checkbox says 'Include skipped tests in total count for thresholds'. At the bottom are 'Save' and 'Apply' buttons.



The screenshot shows the Jenkins console output for build #1. The 'Console Output' tab is selected. The output log is displayed, starting with the build configuration and progress through the git clone and fetch steps, followed by the execution of a robot test script which fails to find the expected file.

```

Started by user Pipatsorn Midjaroenrad
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  Cloning repository https://github.com/Pipatsorn/Lab8.git
    > git init /var/jenkins_home/workspace/UAT # timeout=10
Fetching upstream changes from https://github.com/Pipatsorn/Lab8.git
  > git --version # timeout=10
  > git --version # "git version 2.39.5"
  > git fetch --tags --force --progress -- https://github.com/Pipatsorn/Lab8.git +refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config remote.origin.url https://github.com/Pipatsorn/Lab8.git # timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b4b31a1ce056e97c015e6e06ad09e052a378beb2 (refs/remotes/origin/main)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f b4b31a1ce056e97c015e6e06ad09e052a378beb2 # timeout=10
Commit message: "Add test cases for Lab7"
First time build. Skipping changelog.
[UAT] $ /bin/sh -xe /tmp/jenkins4707446537332842444.sh
+ robot UAT-LAB7.robot
/tmp/jenkins4707446537332842444.sh: 2: robot: not found

```