

Lab Worksheet

ชื่อ-นามสกุล ปวีนวัฒน์ สุขร่วม รหัสนักศึกษา 653380136-7 Section 1

Lab#8 – Software Deployment Using Docker

วัตถุประสงค์การเรียนรู้

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

Pre-requisite

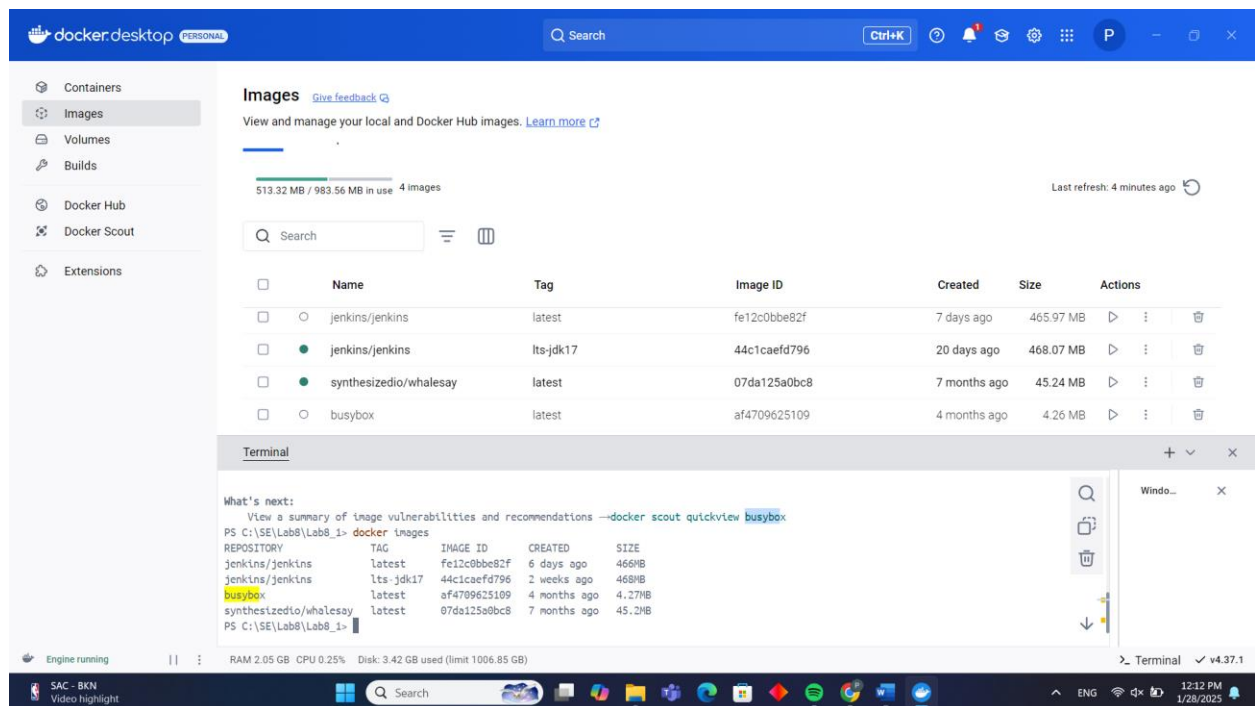
1. ติดตั้ง Docker desktop ลงบนเครื่องคอมพิวเตอร์ โดยดาวน์โหลดจาก <https://www.docker.com/get-started>
2. สร้าง Account บน Docker hub (<https://hub.docker.com/signup>)
3. กำหนดให้ \$ หมายถึง Command prompt และ <> หมายถึง ให้ป้อนค่าของพารามิเตอร์ที่กำหนด

แบบฝึกปฏิบัติที่ 8.1 Hello world - รัน Container จาก Docker image

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

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

Lab Worksheet



(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร ชื่อ Image

(2) Tag ที่ใช้บ่งบอกถึงอะไร Image Version

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

6.

Images [Give feedback](#)

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

517.59 MB / 983.56 MB in use 4 images Last refresh: 6 minutes ago

Search

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	jenkins/jenkins	latest	fe12c0bbe82f	7 days ago	465.97 MB	▶ ⋮ 🗑
<input type="checkbox"/>	jenkins/jenkins	lts-jdk17	44c1caefd796	20 days ago	468.07 MB	▶ ⋮ 🗑
<input type="checkbox"/>	synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB	▶ ⋮ 🗑
<input type="checkbox"/>	busybox	latest	af4709625109	4 months ago	4.26 MB	▶ ⋮ 🗑

Terminal

```
PS C:\SE\Lab8\Lab8_1> docker run -it busybox sh
/#
```

Engine running RAM 2.09 GB CPU 0.50% Disk: 3.42 GB used (limit 1006.85 GB) Terminal v4.37.1

7.

Images [Give feedback](#)

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

517.59 MB / 983.56 MB in use 4 images Last refresh: 6 minutes ago

Search

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	jenkins/jenkins	latest	fe12c0bbe82f	7 days ago	465.97 MB	▶ ⋮ 🗑
<input type="checkbox"/>	jenkins/jenkins	lts-jdk17	44c1caefd796	20 days ago	468.07 MB	▶ ⋮ 🗑
<input type="checkbox"/>	synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB	▶ ⋮ 🗑
<input type="checkbox"/>	busybox	latest	af4709625109	4 months ago	4.26 MB	▶ ⋮ 🗑

Terminal

```
PS C:\SE\Lab8\Lab8_1> docker run -it busybox sh
/# ls
bin  dev  etc  home  lib  lib64  proc  root  sys  tmp  usr  var
/#
```

Engine running RAM 2.11 GB CPU 0.33% Disk: 3.42 GB used (limit 1006.85 GB) Terminal v4.37.1

Lab Worksheet

8.

The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Containers, Images (selected), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area displays the 'Images' tab, which lists local Docker images. Below the list is a terminal window showing the output of the 'ls -la' command.

Image Name	Repository	Tag	Image ID	Created	Size
jenkins/jenkins	jenkins/jenkins	lts-jdk17	44c1caefd796	20 days ago	468.07 MB
synthesizedio/whalesay	synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB
busybox	busybox	latest	af4709625109	4 months ago	4.26 MB

Showing 4 items

```

Terminal
/ # 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 28 05:14 .
drwxr-xr-x 1 root root      4096 Jan 28 05:14 ..
-rwxr-xr-x 1 root root         0 Jan 28 05:14 .dockerenv
drwxr-xr-x 2 root root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root      3660 Jan 28 05:14 dev
drwxr-xr-x 1 root root      4096 Jan 28 05:14 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 245 root root         0 Jan 28 05:14 proc
drwxr-xr-x 1 root root      4096 Jan 28 05:15 root
dr-xr-xr-x 11 root root         0 Jan 28 05:14 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
/ #
  
```

Engine running | RAM 2.10 GB CPU 0.08% Disk: 3.42 GB used (limit 1006.85 GB) | Terminal v4.37.1

9.

The screenshot shows the Docker Desktop application window, similar to the previous one. The left sidebar contains navigation options: Containers, Images (selected), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area displays the 'Images' tab, which lists local Docker images. Below the list is a terminal window showing the output of the 'ls -la' command.

Image Name	Repository	Tag	Image ID	Created	Size
jenkins/jenkins	jenkins/jenkins	lts-jdk17	44c1caefd796	20 days ago	468.07 MB
synthesizedio/whalesay	synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB
busybox	busybox	latest	af4709625109	4 months ago	4.26 MB

Showing 4 items

```

Terminal
bin  dev  etc  home  lib  lib64  proc  root  sys  tmp  usr  var
/ # ls -la
total 48
drwxr-xr-x 1 root root      4096 Jan 28 05:14 .
drwxr-xr-x 1 root root      4096 Jan 28 05:14 ..
-rwxr-xr-x 1 root root         0 Jan 28 05:14 .dockerenv
drwxr-xr-x 2 root root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root      3660 Jan 28 05:14 dev
drwxr-xr-x 1 root root      4096 Jan 28 05:14 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 245 root root         0 Jan 28 05:14 proc
drwxr-xr-x 1 root root      4096 Jan 28 05:15 root
dr-xr-xr-x 11 root root         0 Jan 28 05:14 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
PS C:\SE\Lab8\Lab8_1>
  
```

Engine running | RAM 2.11 GB CPU 0.00% Disk: 3.42 GB used (limit 1006.85 GB) | Terminal v4.37.1

Lab Worksheet

10.

The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images (selected), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Images' and displays a list of local images:

Image	Repository	Tag	Created	Size	Actions
jenkins/jenkins	its-jdk17	44c1caefd796	20 days ago	468.07 MB	[Refresh] [More] [Delete]
synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB	[Refresh] [More] [Delete]
busybox	latest	af4709625109	4 months ago	4.26 MB	[Refresh] [More] [Delete]

Below the image list is a 'Terminal' window showing the output of a command:

```
total 48
drwxr-xr-x 1 root root 4096 Jan 28 05:14 .
drwxr-xr-x 1 root root 4096 Jan 28 05:14 ..
-rwxr-xr-x 1 root root 0 Jan 28 05:14 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 28 05:14 dev
drwxr-xr-x 1 root root 4096 Jan 28 05:14 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 245 root root 0 Jan 28 05:14 proc
drwxr-xr-x 1 root root 4096 Jan 28 05:15 root
dr-xr-xr-x 11 root root 0 Jan 28 05:14 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

PS C:\SE\Lab8\Lab8_1> docker run busybox echo "Hello Paweenawat Sukruam from busybox"
Hello Paweenawat Sukruam from busybox
PS C:\SE\Lab8\Lab8_1>
```

The bottom status bar shows 'Engine running', 'RAM 2.09 GB', 'CPU 0.17%', and 'Disk: 3.42 GB used (limit 1006.85 GB)'.

11.

The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images (selected), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Images' and displays a list of local images:

Image	Repository	Tag	Created	Size	Actions
jenkins/jenkins	its-jdk17	44c1caefd796	20 days ago	468.07 MB	[Refresh] [More] [Delete]
synthesizedio/whalesay	latest	07da125a0bc8	7 months ago	45.24 MB	[Refresh] [More] [Delete]
busybox	latest	af4709625109	4 months ago	4.26 MB	[Refresh] [More] [Delete]

Below the image list is a 'Terminal' window showing the output of a command:

```
PS C:\SE\Lab8\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS      PORTS
d44a2b30881    busybox    "echo 'Hello Paweenaw..." 25 seconds ago Exited (0) 24 seconds ago
c7a3f57ac8f8    busybox    "sh"                    2 minutes ago Exited (0) About a minute ago
059d318af5b0    busybox    "sh"                    3 minutes ago Exited (0) 2 minutes ago
777cf9122609    busybox    "sh"                    3 minutes ago Exited (0) 3 minutes ago
7b58af04c223    jenkins/jenkins:its-jdk17 "/usr/bin/tini -- /u..." 6 days ago    Up 8 minutes    0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp
e3ed0c799c7d    synthesizedio/whalesay "/usr/local/bin/cows..." 6 days ago    Exited (0) 6 days ago
6c3c68b5465f    synthesizedio/whalesay "/usr/local/bin/cows..." 6 days ago    Exited (0) 6 days ago
ad033ef1822b    synthesizedio/whalesay "/usr/local/bin/cows..." 6 days ago    Exited (0) 6 days ago
hungry_spence

PS C:\SE\Lab8\Lab8_1>
```

The bottom status bar shows 'Engine running', 'RAM 2.09 GB', 'CPU 0.00%', and 'Disk: 3.42 GB used (limit 1006.85 GB)'.

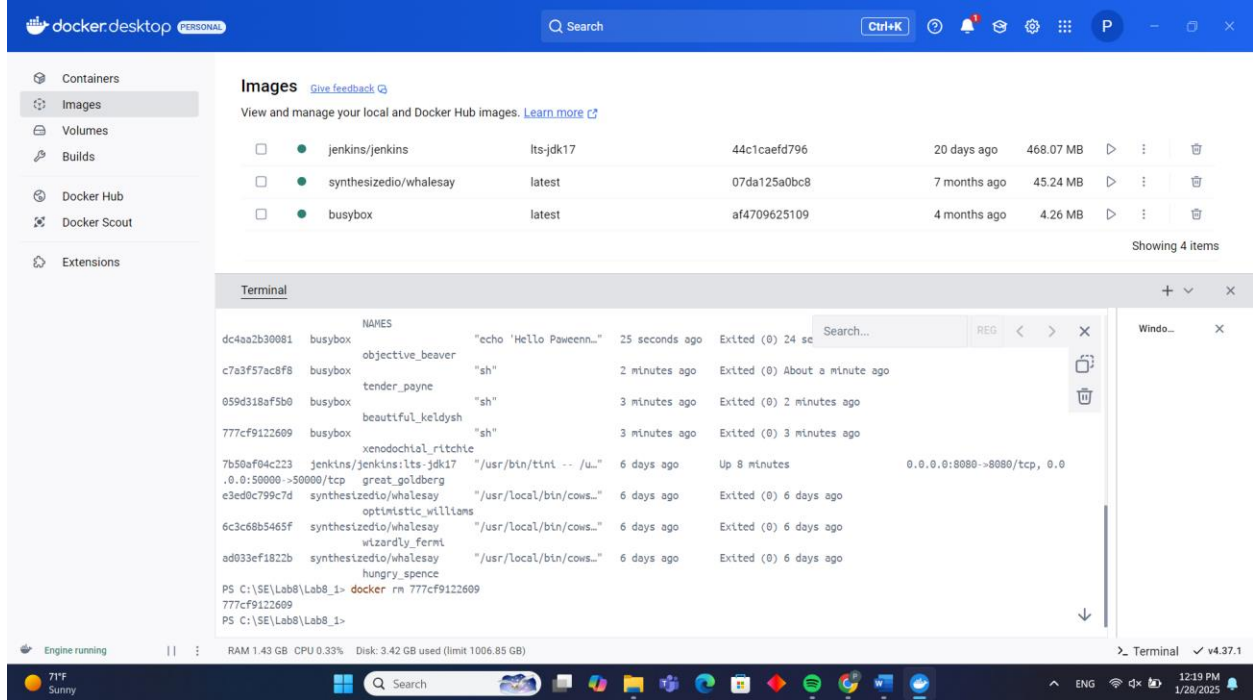
(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
รัน Shell ใน container

Lab Worksheet

- (2) คอลัมน์ STATUS จากการรันคำสั่ง `docker ps -a` แสดงถึงข้อมูลอะไร
สถานะใช้งานของ Container

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

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



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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ (Windows) บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

`$ cat > Dockerfile << EOF`

Lab Worksheet

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

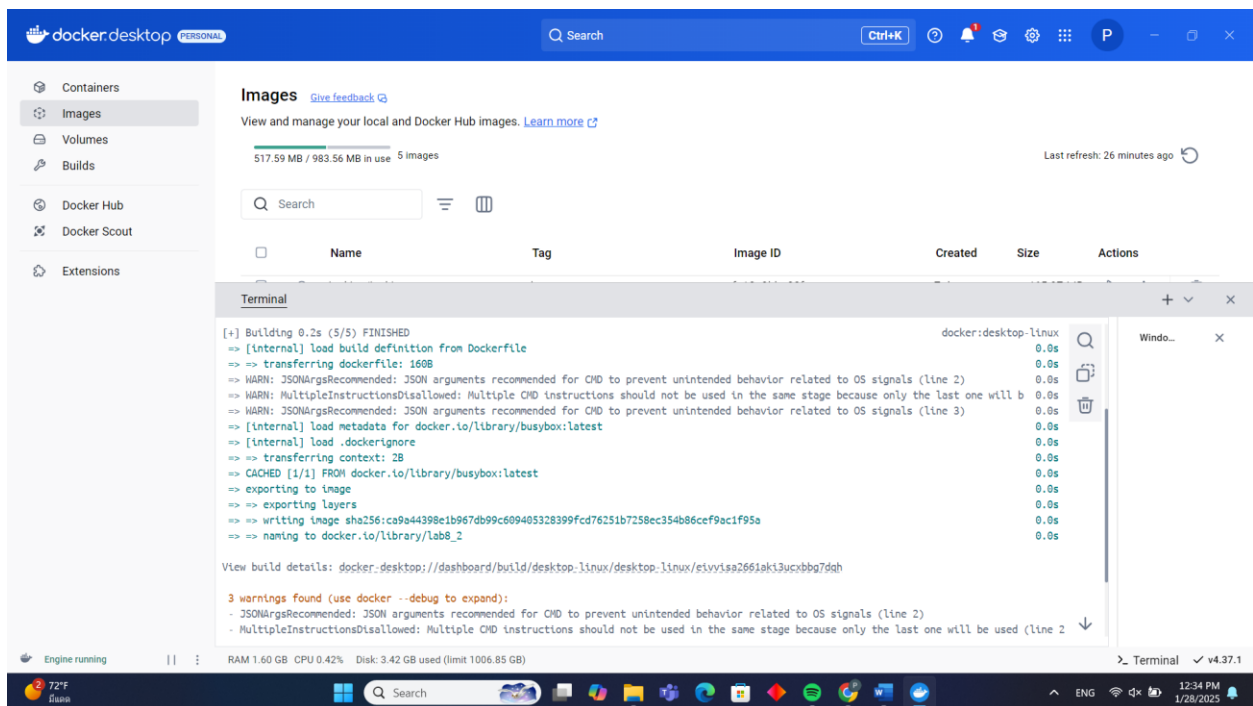
แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

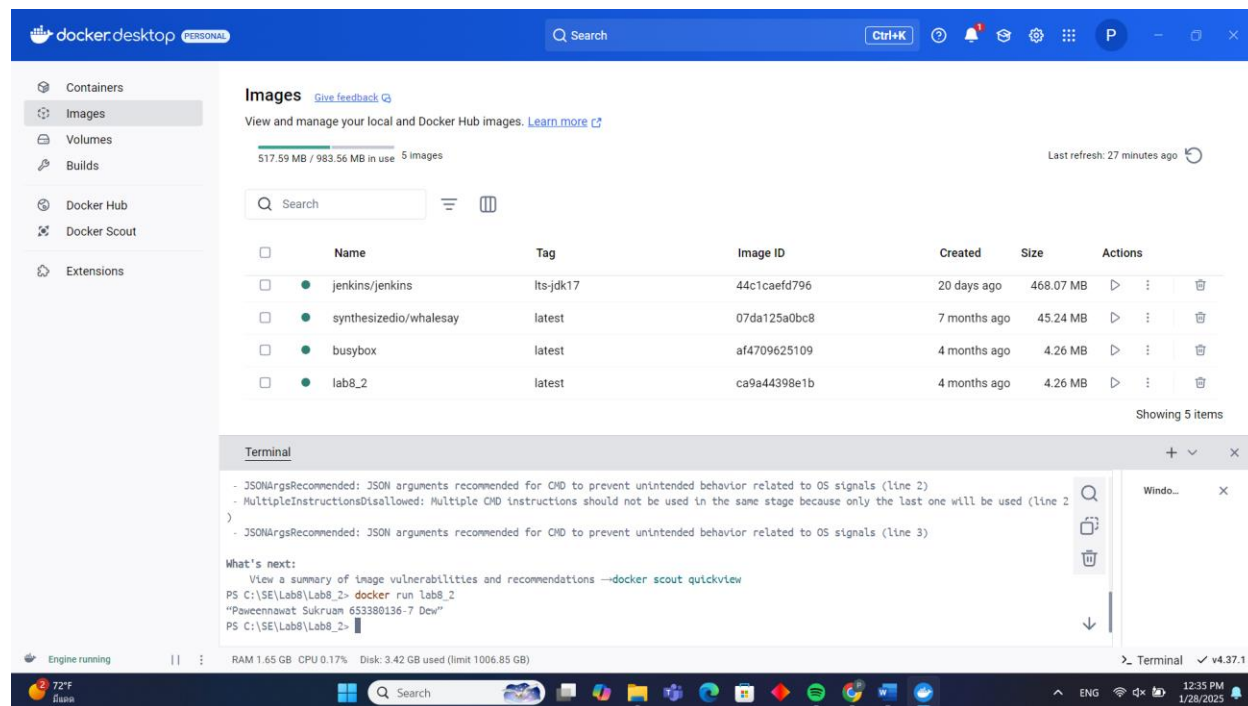
\$ docker build -t <ชื่อ Image> .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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



Lab Worksheet



(1) คำสั่งที่ใช้ในการ run คือ

`docker run lab8_2`

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

Build image โดยสามารถตั้งชื่อ image และ tag ได้

แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

`$ cat > Dockerfile << EOF`

Lab Worksheet

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

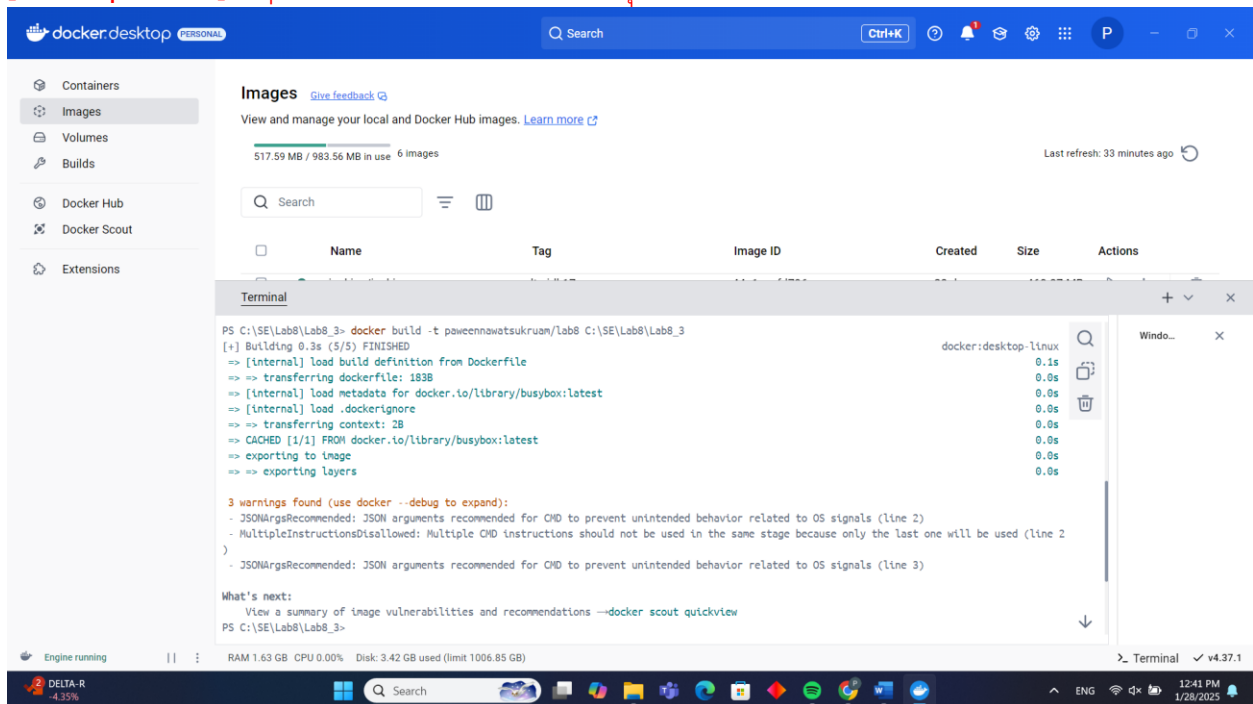
7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t <username ที่ลงทะเบียนกับ Docker Hub>/lab8

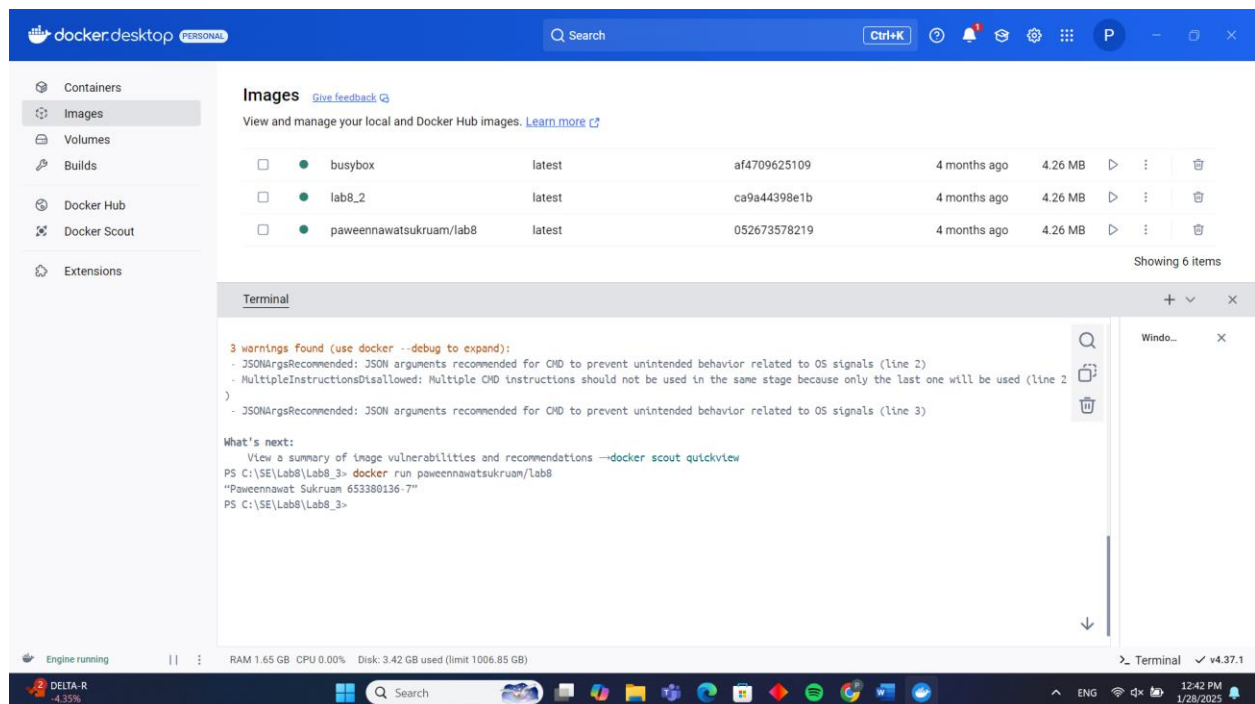
5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

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

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



Lab Worksheet



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 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)

Lab Worksheet

The screenshot shows the Docker Desktop application. On the left is a sidebar with navigation options: Containers, Images (selected), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Images' and contains a table of local images:

Image	Tag	ID	Pushed	Size	Actions
busybox	latest	af4709625109	4 months ago	4.26 MB	[Play] [Info] [Delete]
lab8_2	latest	ca9a44398e1b	4 months ago	4.26 MB	[Play] [Info] [Delete]
pawennawatsukruam/lab8	latest	052673578219	4 months ago	4.26 MB	[Play] [Info] [Delete]

Below the table, a terminal window is open, displaying the output of a Docker command. It shows warnings about JSON arguments and a summary of image vulnerabilities. The terminal text is as follows:

```
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)

What's next:
View a summary of image vulnerabilities and recommendations --docker scout quickview
PS C:\SE\Lab8\Lab8_3> docker run pawennawatsukruam/lab8
"Pawennawat Sukruam 653389136-7"
PS C:\SE\Lab8\Lab8_3> docker push pawennawatsukruam/lab8
Using default tag: latest
The push refers to repository [docker.io/pawennawatsukruam/lab8]
59654b79d9ad: Mounted from library/busybox
latest: digest: sha256:c9fa5a6af8908c4295eb9888a1e239dffc1847ee0e5701bcd9dd6e86531da46 size: 527
PS C:\SE\Lab8\Lab8_3>
```

The bottom status bar indicates the engine is running, with RAM at 1.65 GB, CPU at 0.08%, and disk usage at 3.42 GB. The system tray shows the date and time as 12:44 PM on 1/28/2025.

The screenshot shows the Docker Hub website. The browser address bar displays 'hub.docker.com/repositories/pawennawatsukruam'. The page header includes the Docker Hub logo and navigation links: Explore, Repositories (selected), Organizations, and Usage. A search bar is present with the text 'Search Docker Hub'. Below the header, there is a search filter for 'pawennawatsukruam' and a 'Create a repository' button. The main content area shows a table of repositories:

Name	Last Pushed	Contains	Visibility	Scout
pawennawatsukruam/lab8	1 minute ago	IMAGE	Public	Inactive

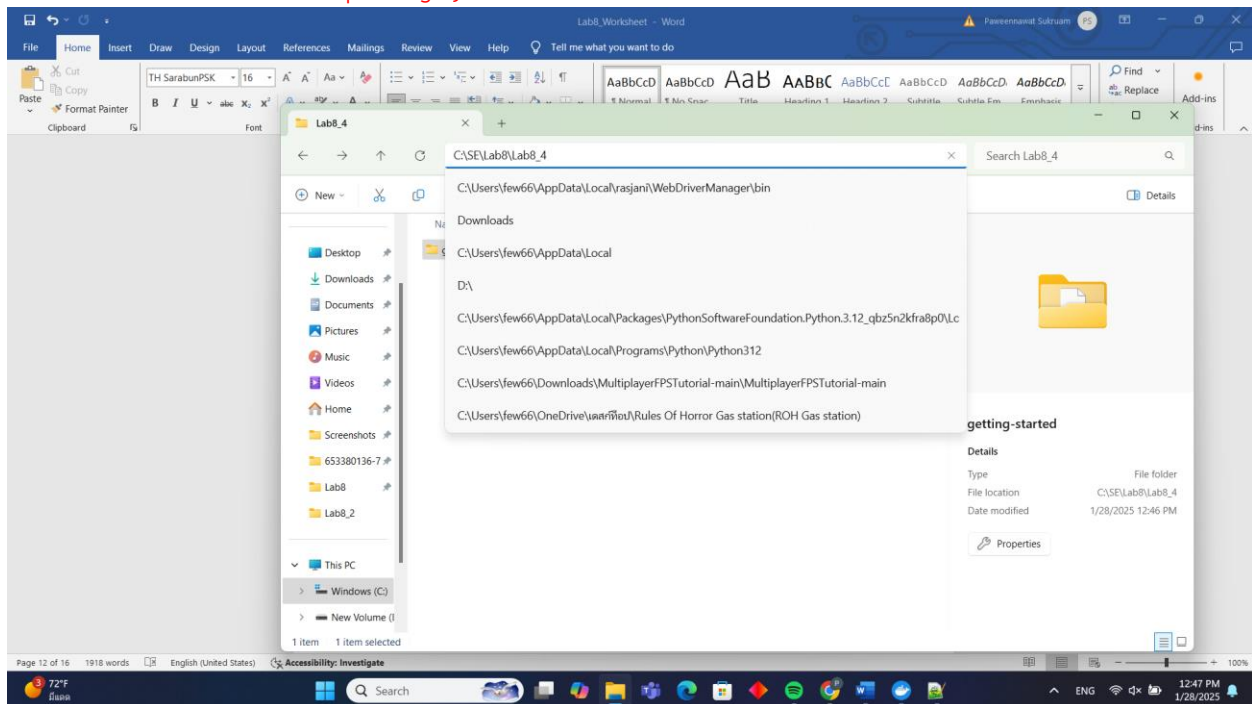
The page indicates '1-1 of 1' results. The system tray at the bottom shows the date and time as 12:44 PM on 1/28/2025.

Lab Worksheet

แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก GitHub repository <https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง
\$ git clone https://github.com/docker/getting-started.git
3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

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



Lab Worksheet

```

1  {
2    "name": "101-app",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "scripts": {
7      "prettify": "prettier -l --write \"**/*.js\"",
8      "test": "jest",
9      "dev": "nodemon src/index.js"
10   },
11   "dependencies": {
12     "express": "^4.18.2",
13     "mysql2": "^2.3.3",
14     "sqlite3": "^5.1.2",
15     "uuid": "^9.0.0",
16     "wait-port": "^1.0.4"
17   },
18   "resolutions": {
19     "ansi-regex": "5.0.1"
20   },
21   "prettier": {
22     "trailingComma": "all",
23     "tabWidth": 4,
24     "useTabs": false,
25     "semi": true,
26     "singleQuote": true
27   },
28   "devDependencies": {
29     "jest": "^29.3.1",
30     "nodemon": "^2.0.20",
31     "prettier": "^2.7.1"
32   }
33 }
34

```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น myapp_รหัสสนศ. ไม่มีขีด
\$ docker build -t <myapp_รหัสสนศ. ไม่มีขีด> .

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

Lab Worksheet

The screenshot shows the Docker Desktop interface. On the left, the sidebar includes Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Images' and displays a table of local images:

Image	Tag	Architecture	Size	Created
lab8_2	latest	ca9a44398e1b	4.26 MB	4 months ago
paweenawatsukruam/lab8	latest	052673578219	4.26 MB	4 months ago
myapp_6533801367	latest	fe8e468efc17	216.9 MB	14 seconds ago

Below the table, a terminal window shows the output of the 'docker build' command. The build process is complete, and the image is named 'docker.io/library/myapp_6533801367'.

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

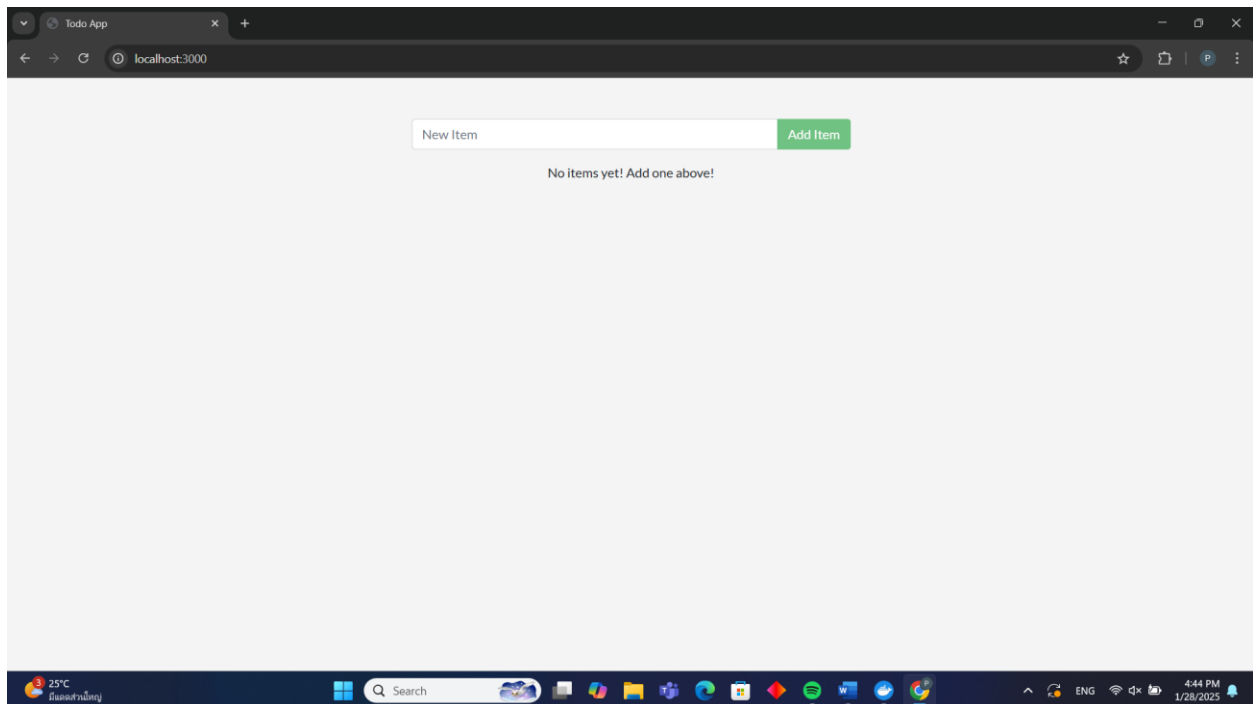
[Check point#9] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

The screenshot shows the Docker Desktop interface. On the left, the sidebar includes Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Images' and displays a table of local images:

Image	Tag	Architecture	Size	Created
lab8_2	latest	ca9a44398e1b	4.26 MB	4 months ago
paweenawatsukruam/lab8	latest	052673578219	4.26 MB	4 months ago
synthesizedio/whalesay	latest	07da125a0bc8	45.24 MB	7 months ago

Below the table, a terminal window shows the output of the 'docker run' command. The container is started successfully, and the image is named 'docker.io/library/myapp_6533801367'.

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

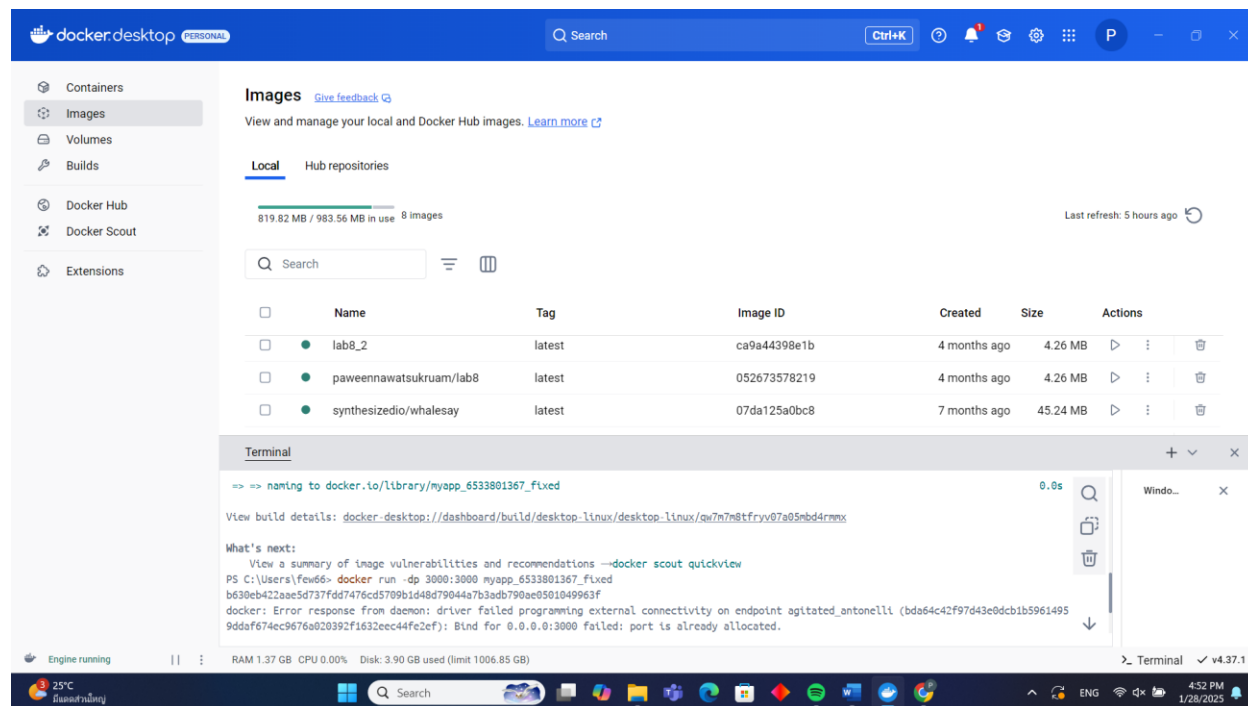
b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

Lab Worksheet



(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

ไม่สามารถเชื่อมต่อ port ได้ เกิดขึ้นเพราะ port ดังกล่าวถูกใช้งานอยู่

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

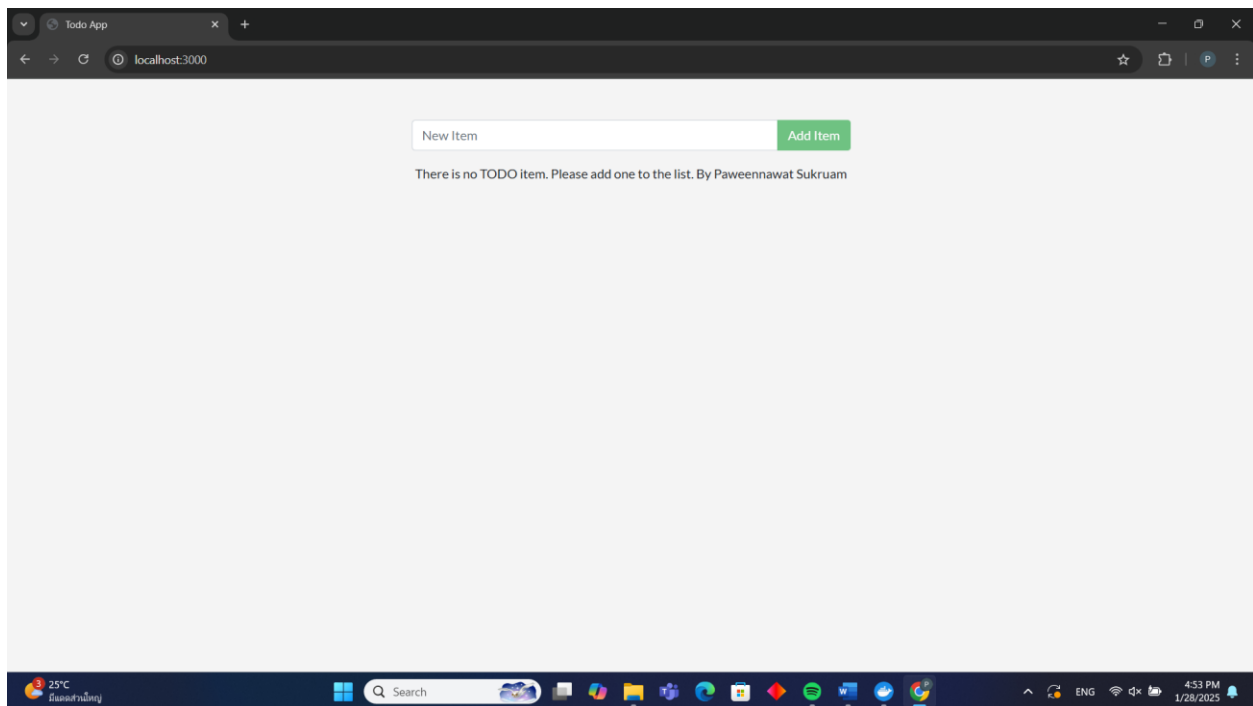
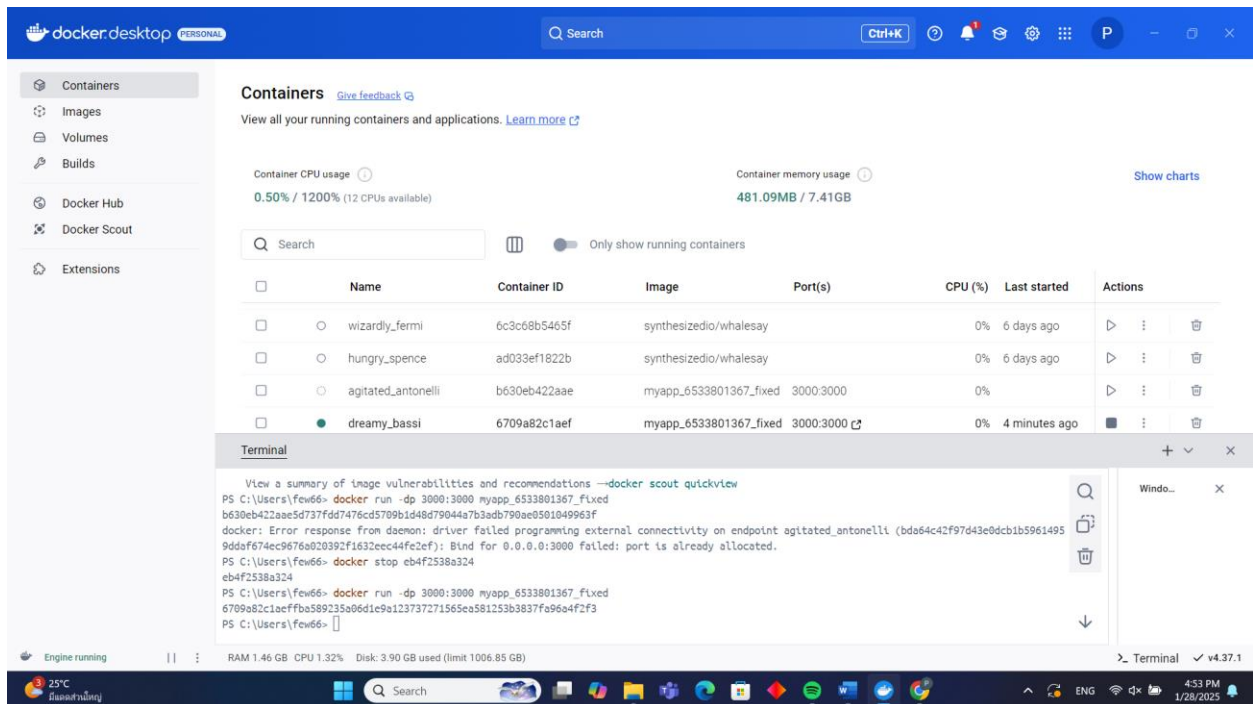
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

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

Lab Worksheet



แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

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

Lab Worksheet

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

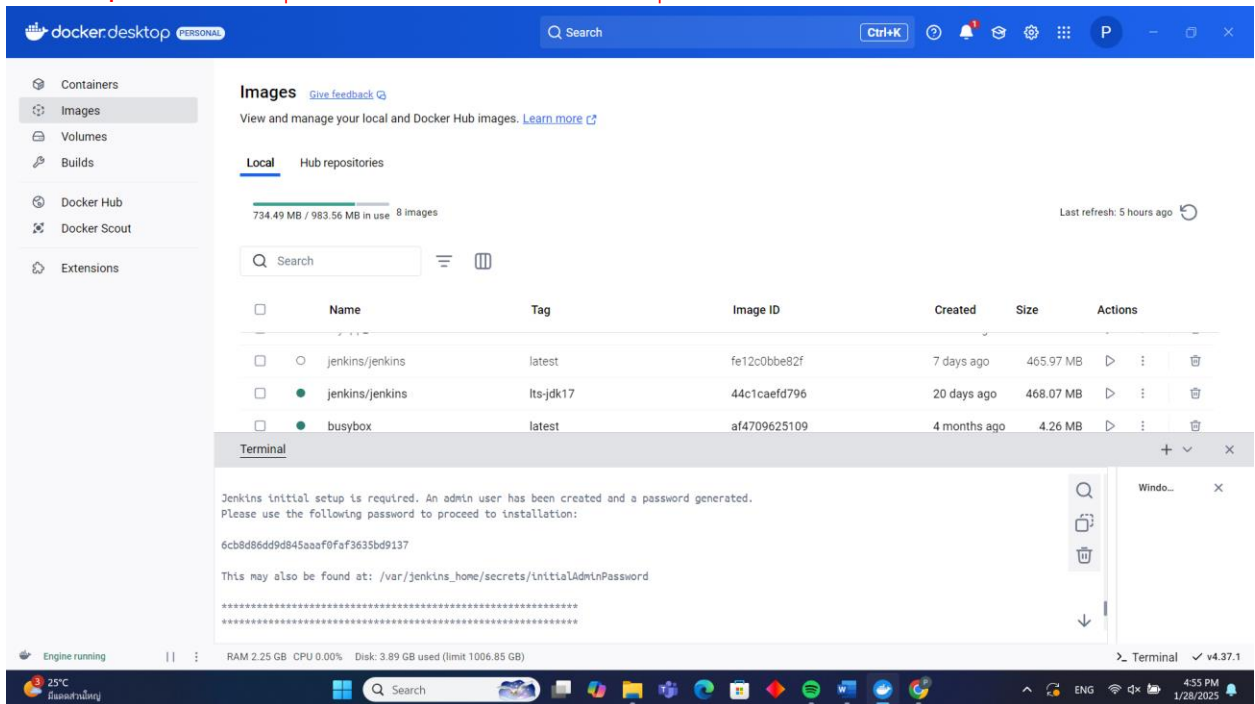
หรือ

```
$ 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



4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

Lab Worksheet

Getting Started

Username
paweennawat_1367

Password

Confirm password

Full name
Paweennawat Sukruam

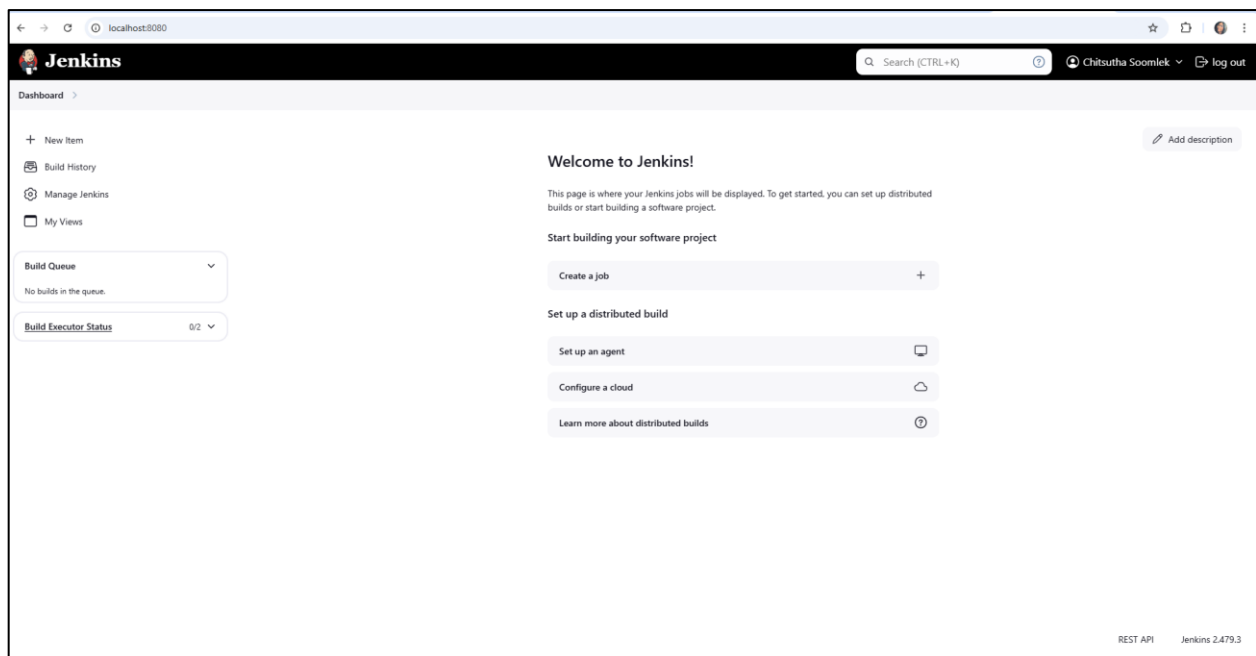
E-mail address
paweennawat.s@kkumail.com

Jenkins 2.479.3

[Skip and continue as admin](#) [Save and Continue](#)

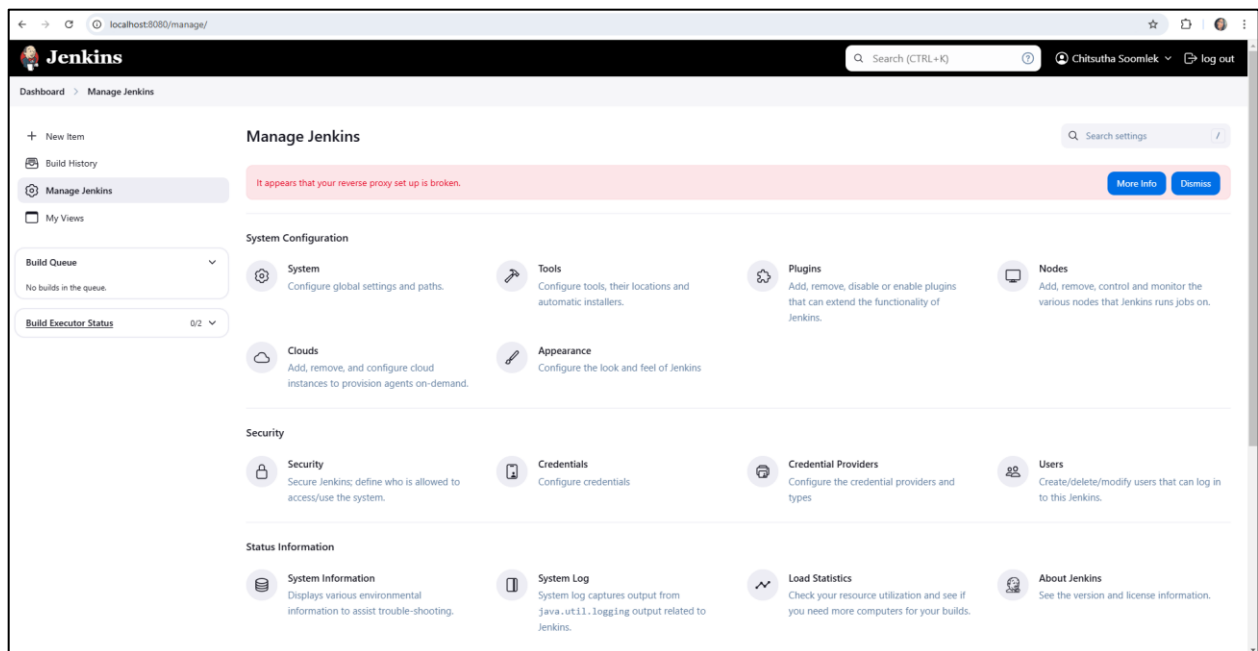
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบหน้าจอ Dashboard ดังแสดงในภาพ

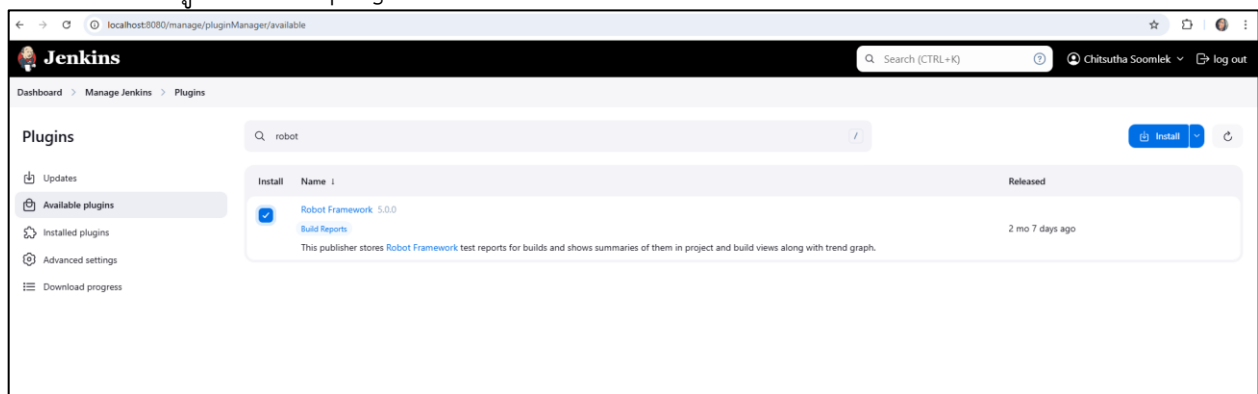


9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

Lab Worksheet

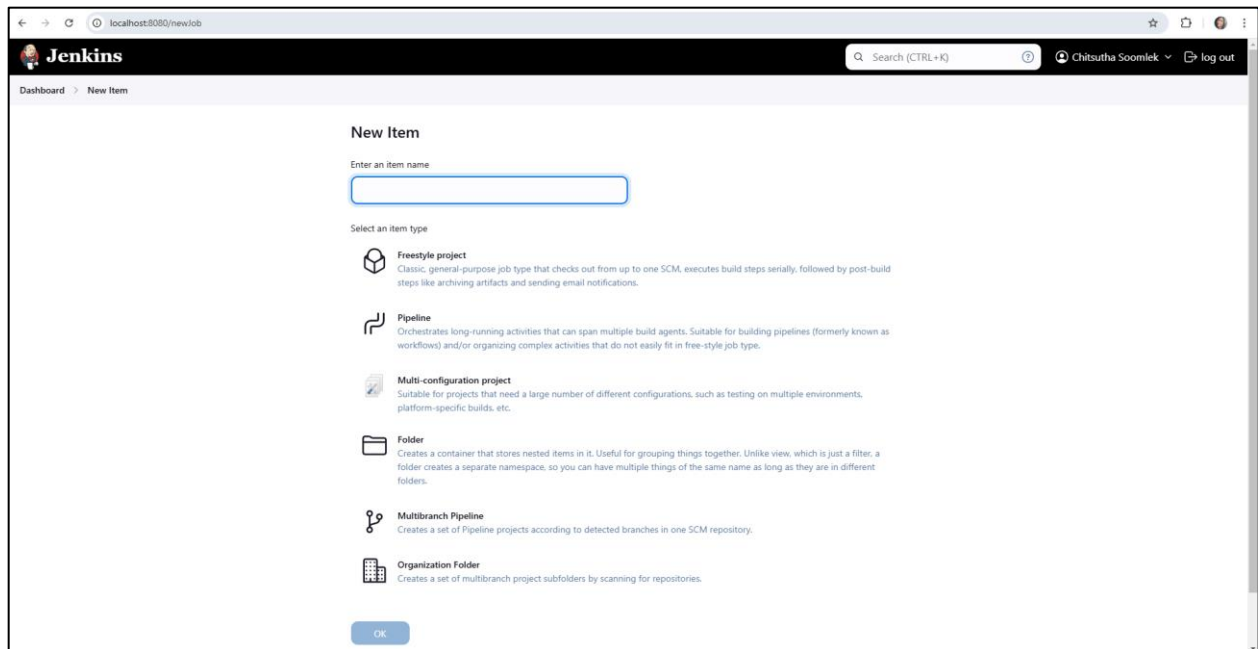


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

Lab Worksheet



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

Description: Lab 8.5

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

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Lab Worksheet

The screenshot shows the Jenkins configuration page for a job named 'UAT'. The 'General' tab is selected in the left sidebar. The 'Enabled' toggle is turned on. The 'Description' field contains 'Lab 8.5'. The 'GitHub project' checkbox is checked, and the 'Project url' is set to 'https://github.com/PaweennawatSukruam/Lab7.git/'. The 'Discard old builds' checkbox is unchecked. The 'This project is parameterized' and 'Throttle builds' checkboxes are also unchecked. The 'Save' button is highlighted.

The screenshot shows the Jenkins configuration page for the same 'UAT' job, but with the 'Build Triggers' tab selected. The 'Build periodically' checkbox is checked, and the 'Schedule' field contains 'H/15 * * * *'. Below the schedule field, it indicates the last run was on Tuesday, January 28, 2025 at 2:05:28 PM and the next run is at 2:05:28 PM. The 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'GitHub hook trigger for GITScm polling', and 'Poll SCM' checkboxes are all unchecked. The 'Delete workspace before build starts' and 'Use secret text(s) or file(s)' checkboxes under the 'Build Environment' section are also unchecked. The 'Save' button is highlighted.

Lab Worksheet

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

`cd Lab7/LabTest`

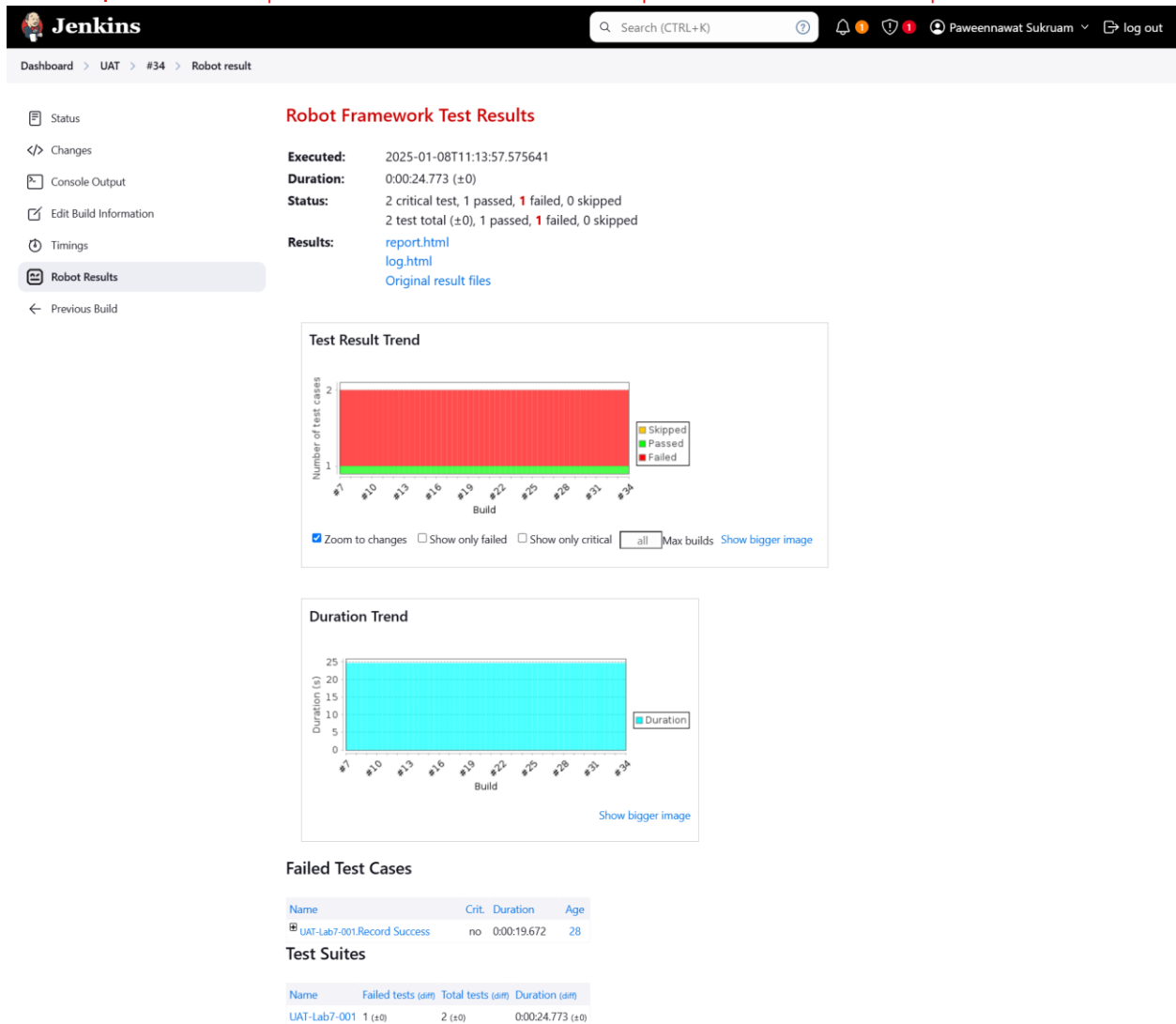
`Robot .`

Post-build action: เพิ่ม Publish Robot Framework test results -> ระบุได้เร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่าน แล้วนับว่าซอฟต์แวร์มีปัญหา -> 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

The screenshot shows the Jenkins web interface for build #34 of the 'UAT' job. The 'Console Output' tab is selected, displaying the following log:

```
Started by user Paweenawat Sukruam
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/sh -xe /tmp/jenkins7955968266142239600.sh
+ cd Lab7/LabTest
+ robot .
/tmp/jenkins7955968266142239600.sh: 3: robot: not found
Build step 'Execute shell' marked build as failure
Robot results publisher started...
INFO: Checking test criticality is deprecated and will be dropped in a future release!
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE
```

On the left sidebar, the 'Console Output' link is highlighted. The top navigation bar shows 'Dashboard > UAT > #34 > Console Output'. The bottom of the image shows a Windows taskbar with the date 1/28/2025 and time 10:08 PM.