

## Lab Worksheet

ชื่อ-นามสกุล นายพงษ์วรินทร์ แก้วสง่า รหัสนักศึกษา 653380207-0 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 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

```

Terminal

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\Users\phong> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\phong> mkdir Lab8_1

Directory: C:\Users\phong

Mode                LastWriteTime         Length Name
----                -
d-----          1/29/2025   7:56 PM             Lab8_1

RAM 2.02 GB  CPU 0.00%  Disk: 1.09 GB used (limit 1006.85 GB)

PS C:\Users\phong\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
PS C:\Users\phong\Lab8_1> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    74cc54e27dc4   7 days ago    10.1kB
busybox       latest    af4709625109   4 months ago  4.27MB
PS C:\Users\phong\Lab8_1>

```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร เป็นชื่อของอิมเมจ (Docker Image) ที่ถูกดึง (pulled) หรือสร้างขึ้น (built) บนระบบ ซึ่งในที่นี้มี hello-world และ busybox เป็นอิมเมจที่อยู่ในเครื่อง
- (2) Tag ที่ใช้บอกถึงอะไร เป็นตัวระบุเวอร์ชันของอิมเมจที่ถูกดึงมาใช้งาน เช่น latest หมายถึงอิมเมจเวอร์ชันล่าสุดที่อยู่บน Docker Hub หรือ Registry

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

**Terminal**

```

7 days ago 10.1kB busybox latest af4709625109 4 months ago 4.27MB
PS C:\Users\phong\Lab8_1> docker run busybox
PS C:\Users\phong\Lab8_1> docker run -it busybox sh
/ # ls
bin  etc  lib  proc  sys  usr
dev  home lib64 root  tmp  var
/ # ls -la
total 48
drwxr-xr-x 1 root root      4096 Jan 29 13:01 .
drwxr-xr-x 1 root root      4096 Jan 29 13:01 ..
-rwxr-xr-x 1 root root         0 Jan 29 13:01 .dockerenv
drwxr-xr-x 2 root root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root      360 Jan 29 13:01 dev
drwxr-xr-x 1 root root      4096 Jan 29 13:01 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
Hello Pongwarin Kaewsanga from busybox
RAM 0.88 GB CPU 0.13% Disk: 1.09 GB used (limit 1006.85 GB)

```

**Terminal**

```

drwxr-xr-x 2 root root      12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root      360 Jan 29 13:01 dev
drwxr-xr-x 1 root root      4096 Jan 29 13:01 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
Hello Pongwarin Kaewsanga from busybox
PS C:\Users\phong\Lab8_1> docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a7ac1a3dd242	busybox	"echo 'Hello Pongwarin...'"	17 seconds ago	Exited (0) 16 seconds ago		gallant_leavitt
d882f35a9b06	busybox	"sh"	About a minute ago	Exited (0) 34 seconds ago		elastic_noether
9707e83e7698	busybox	"sh"	About a minute ago	Exited (0) About a minute ago		agitated_pasteur
d64925237adb	hello-world	"/hello"	10 minutes ago	Exited (0) 10 minutes ago		hopeful_tu

```

PS C:\Users\phong\Lab8_1>
RAM 0.88 GB CPU 0.25% Disk: 1.09 GB used (limit 1006.85 GB)

```

- เมื่อใช้ option `-it` ในคำสั่ง `run` ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป `-i` (interactive) ทำให้คอนเทนเนอร์รับอินพุตจากผู้ใช้ในเทอร์มินัลตลอดเวลา ส่วน `-t` (tty) เปิดใช้งาน pseudo-terminal ทำให้สามารถโต้ตอบกับคอนเทนเนอร์ได้เหมือนกับการใช้งานเครื่องคอมพิวเตอร์ปกติ การใช้ `-it` พร้อมกับคำสั่งเชลล์อย่าง `sh` หรือ `bash` ทำให้ผู้ใช้สามารถเข้าถึงเชลล์ของคอนเทนเนอร์ได้ และทำงานต่างๆ ภายในนั้นได้ทันที
- คอลัมน์ STATUS จากการรันคำสั่ง `docker ps -a` แสดงถึงข้อมูลอะไร คอลัมน์ STATUS แสดงสถานะของคอนเทนเนอร์ เช่น "Exited (0) X seconds ago" หมายถึงคอนเทนเนอร์ทำงานเสร็จสิ้นแล้วและปิดตัวเองด้วยสถานะปกติ (exit code 0) หรือ "Up X seconds" หมายถึงคอนเทนเนอร์กำลังทำงานอยู่ในปัจจุบัน ข้อมูลนี้ช่วยให้ทราบว่าแต่ละคอนเทนเนอร์กำลังทำงานหรือหยุดทำงานอยู่

## Lab Worksheet

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

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



```

Terminal
Hello Pongwarin Kaewsanga from busybox
PS C:\Users\phong\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
a7ac1a3dd242   busybox   "echo 'Hello Pongwar..." 17 seconds ago Exited (0) 16 seconds ago      gallant_leavitt
d882f35a9b06   busybox   "sh"                    About a minute ago Exited (0) 34 seconds ago      elastic_noether
9707e83e7698   busybox   "sh"                    About a minute ago Exited (0) About a minute ago      agitated_pasteur
d64925237adb   hello-world "/hello"                10 minutes ago Exited (0) 10 minutes ago      hopeful_tu
PS C:\Users\phong\Lab8_1> docker rm a7ac1a3dd242
a7ac1a3dd242
PS C:\Users\phong\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
d882f35a9b06   busybox   "sh"                    8 minutes ago Exited (0) 7 minutes ago      elastic_noether
9707e83e7698   busybox   "sh"                    8 minutes ago Exited (0) 8 minutes ago      agitated_pasteur
d64925237adb   hello-world "/hello"                17 minutes ago Exited (0) 17 minutes ago      hopeful_tu
PS C:\Users\phong\Lab8_1>

RAM 0.98 GB   CPU 0.13%   Disk: 1.09 GB used (limit 1006.85 GB)
  
```

## 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

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

EOF

## Lab Worksheet

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

The screenshot shows a Windows File Explorer window titled "New-Item -Name 'Dockerfile' -ItemType 'File'" with the directory "C:\Users\phong\Lab8\_2". It lists a file "Dockerfile" with a size of 0 bytes, last written on 1/29/2025 at 8:17 PM.

Below the file explorer is the Docker Desktop interface. The terminal shows the command `docker build -t my-windows-image .` being executed. The output indicates the build is finished, and the image is being transferred and built. The terminal output is as follows:

```
PS C:\Users\phong\Lab8_2> docker build -t my-windows-image .
[+] Building 0.7s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 280B
RAM 1.12 GB CPU 0.25% Disk: 1.09 GB used (limit 1006.85 GB)
```

The "Terminal" tab shows the detailed build process:

```
=> => transferring dockerfile: 280B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:8add3b0a49df6c27b70f79dd297fc77946272e6dd63c24c8e297c8493b07ff53
=> => naming to docker.io/library/my-windows-image
```

At the bottom, it shows 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)

The bottom of the terminal shows the command `docker images` and a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE.

## Lab Worksheet

```

Terminal

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)
PS C:\Users\phong\Lab8_2> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world          latest              74cc54e27dc4        7 days ago         10.1kB
busybox              latest              af4709625109        4 months ago       4.27MB
my-windows-image     latest              8add3b0a49df        4 months ago       4.27MB
PS C:\Users\phong\Lab8_2> docker build -t images
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build

Terminal

Start a build
PS C:\Users\phong\Lab8_2> docker run my-windows-image
“นายพงษ์รินทร์ แก้วสง่า วิทยาลัยอาชีวศึกษา 653380207-0 ชื่อเล่น พงษ์”
PS C:\Users\phong\Lab8_2> docker run -it my-windows-image powershell
docker: Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "powershell": executable file not found in $PATH: unknown.
PS C:\Users\phong\Lab8_2> docker stop <container_id>
At line:1 char:13
+ docker stop <container_id>
+ ~
+ The '<' operator is reserved for future use.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : RedirectionNotSupported

PS C:\Users\phong\Lab8_2> docker run my-windows-image
“นายพงษ์รินทร์ แก้วสง่า วิทยาลัยอาชีวศึกษา 653380207-0 ชื่อเล่น พงษ์”
PS C:\Users\phong\Lab8_2>

```

- (1) คำสั่งที่ใช้ในการ run คือ `docker run my-windows-image` คำสั่งนี้จะรันคอนเทนเนอร์จากอิมเมจ `my-windows-image` ที่คุณสร้างขึ้น โดยไม่ระบุ option อื่นๆ ซึ่งจะทำให้คอนเทนเนอร์ทำงานตามคำสั่งที่ถูกกำหนดไว้ใน Dockerfile หรือภาพที่สร้างจากคอนเทนเนอร์นั้น ๆ
- (2) Option `-t` ในคำสั่ง `$ docker build` ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
 ส่งผลต่อการทำงานของคำสั่งโดยการตั้งชื่อและแท็ก (tag) ให้กับอิมเมจที่สร้างขึ้นจากการคอมไพล์ Dockerfile ที่กำหนดไว้ เช่น `docker build -t my-windows-image .` จะสร้างอิมเมจที่มีชื่อ `my-windows-image` และแท็กเป็น `latest` ซึ่งสามารถนำไปใช้ในการรันคอนเทนเนอร์จากอิมเมจนี้ในภายหลัง

## Lab Worksheet

## แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

```
FROM busybox
```

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

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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

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

```
FROM busybox
```

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

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

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

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

5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

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

## Lab Worksheet

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

```
C:\Users\phong\Lab8_3>docker build -t framemx/lab8 .
[+] Building 63.9s (8/8) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 641B                             0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 4.5s
=> [auth] library/ubuntu:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 17.4s
=> => resolve docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 0.0s
=> => sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab 6.69kB / 6.69kB 0.0s
=> => sha256:6e75a10070b0fcb0bead763c5118a369bc7cc30dfc1b0749c491bbb21f15c3c7 424B / 424B 0.0s
=> => sha256:b1d9df8ab81559494794e522b380878cf9ba82d4c1fb67293bcf931c3aa69ae4 2.30kB / 2.30kB 0.0s
=> => sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 29.75MB / 29.75MB 13.7s

RAM 0.81 GB CPU 0.50% Disk: 1.24 GB used (limit 1006.85 GB)

=> => sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 29.75MB / 29.75MB 13.7s
=> => extracting sha256:de44b265507ae44b212defcb50694d666f136b35c1090d9709068bc861bb2d64 3.1s
=> [2/3] RUN apt-get update && apt-get install -y locales fonts-dejavu 38.7s
=> [3/3] RUN locale-gen th_TH.UTF-8 2.4s
=> exporting to image 0.5s
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 10)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 13)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 13)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 14)

C:\Users\phong\Lab8_3>docker run framemx/lab8
นายพนัสนิษฐ์ แก้วสุว 653380207-0

C:\Users\phong\Lab8_3>
```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยใช้คำสั่ง

```
$ 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
```

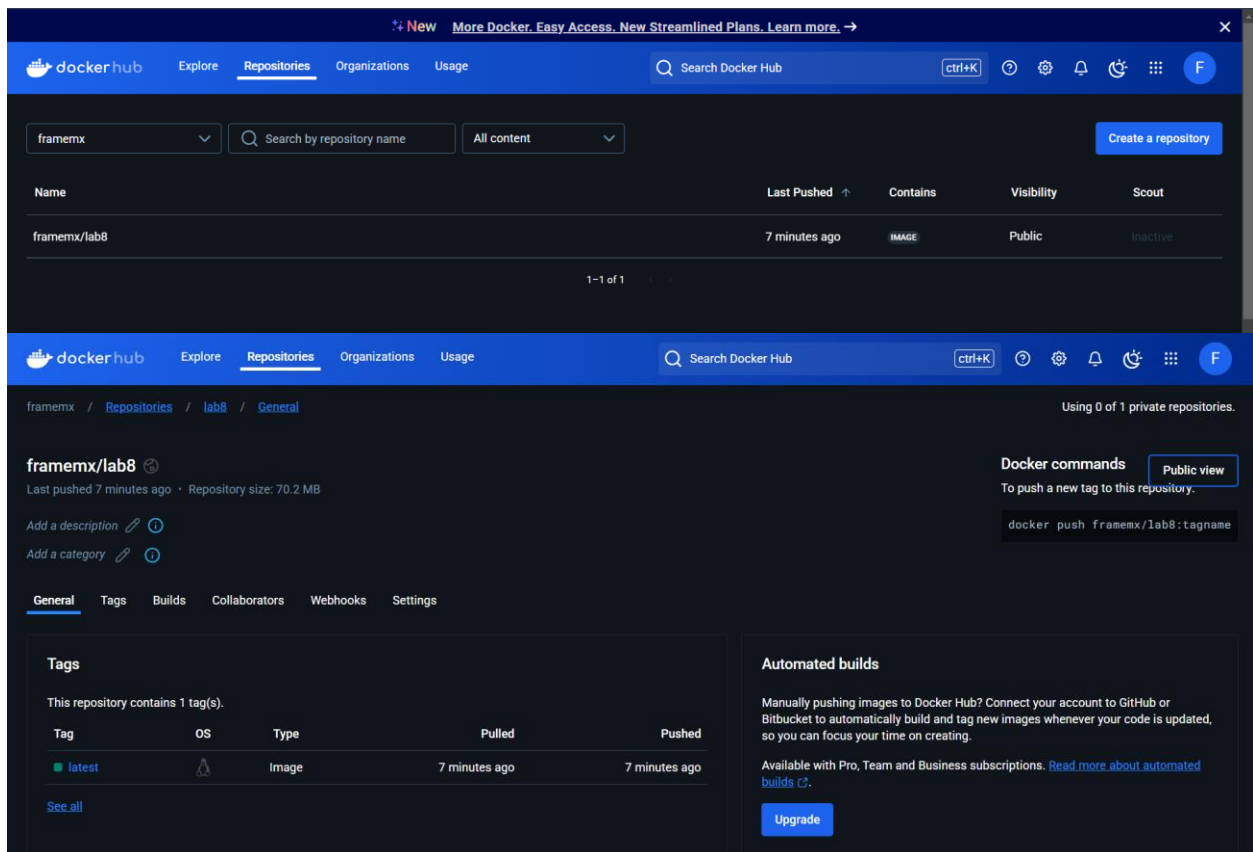
```
$ docker login -u <username> -p <password>
```

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

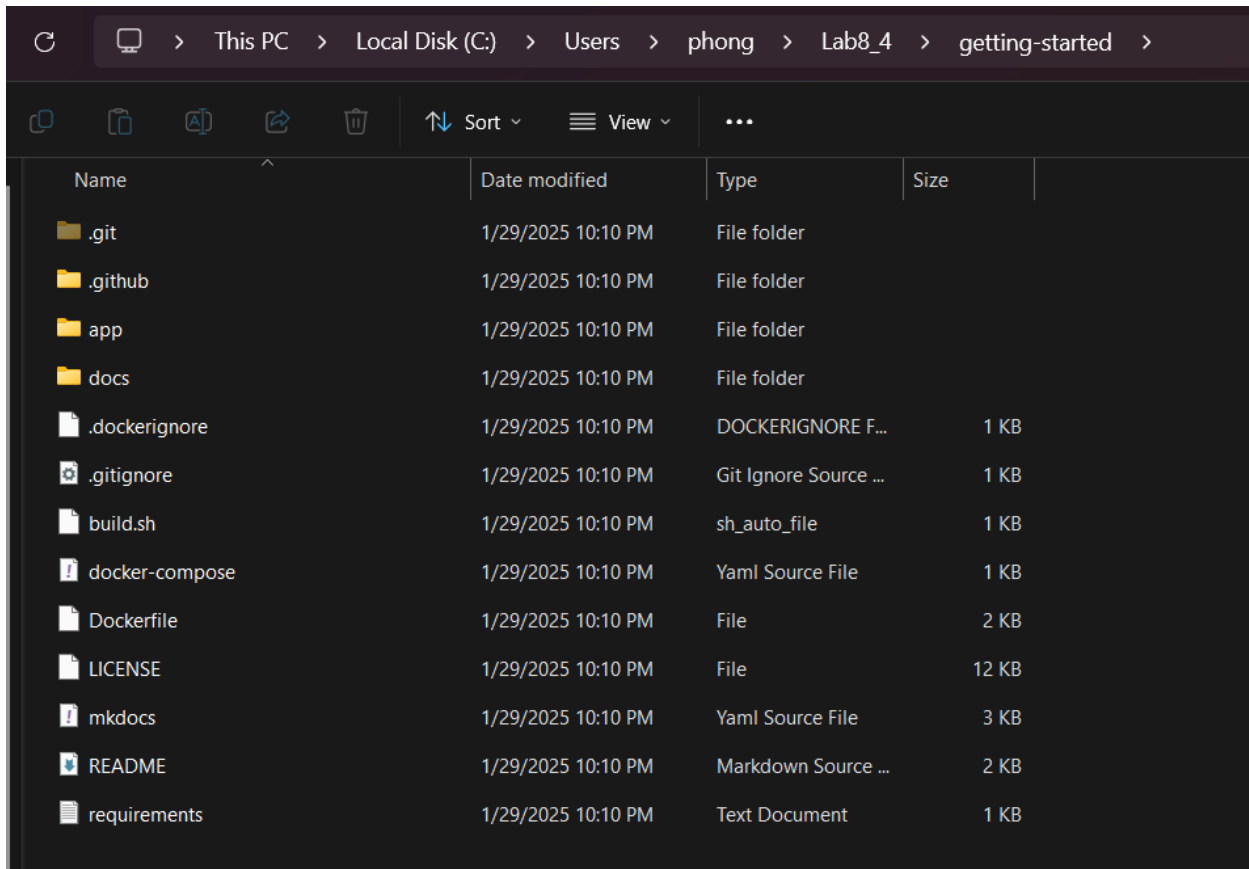


## แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

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

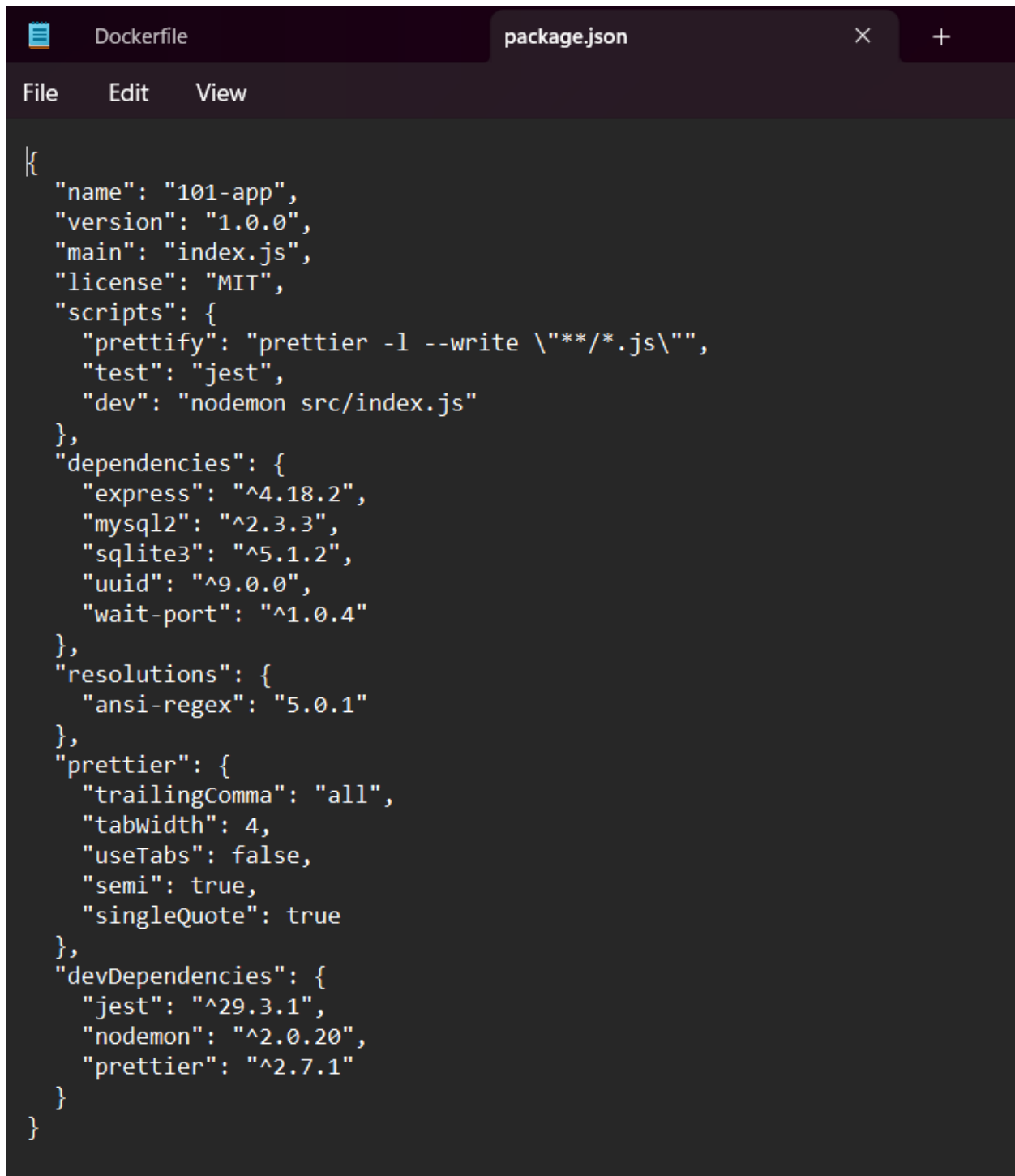
## Lab Worksheet



The screenshot shows a Windows File Explorer window with the address bar indicating the path: This PC > Local Disk (C:) > Users > phong > Lab8\_4 > getting-started >. The window displays a list of files and folders in a table format.

Name	Date modified	Type	Size
.git	1/29/2025 10:10 PM	File folder	
.github	1/29/2025 10:10 PM	File folder	
app	1/29/2025 10:10 PM	File folder	
docs	1/29/2025 10:10 PM	File folder	
.dockerignore	1/29/2025 10:10 PM	DOCKERIGNORE F...	1 KB
.gitignore	1/29/2025 10:10 PM	Git Ignore Source ...	1 KB
build.sh	1/29/2025 10:10 PM	sh_auto_file	1 KB
docker-compose	1/29/2025 10:10 PM	Yaml Source File	1 KB
Dockerfile	1/29/2025 10:10 PM	File	2 KB
LICENSE	1/29/2025 10:10 PM	File	12 KB
mkdocs	1/29/2025 10:10 PM	Yaml Source File	3 KB
README	1/29/2025 10:10 PM	Markdown Source ...	2 KB
requirements	1/29/2025 10:10 PM	Text Document	1 KB

## Lab Worksheet



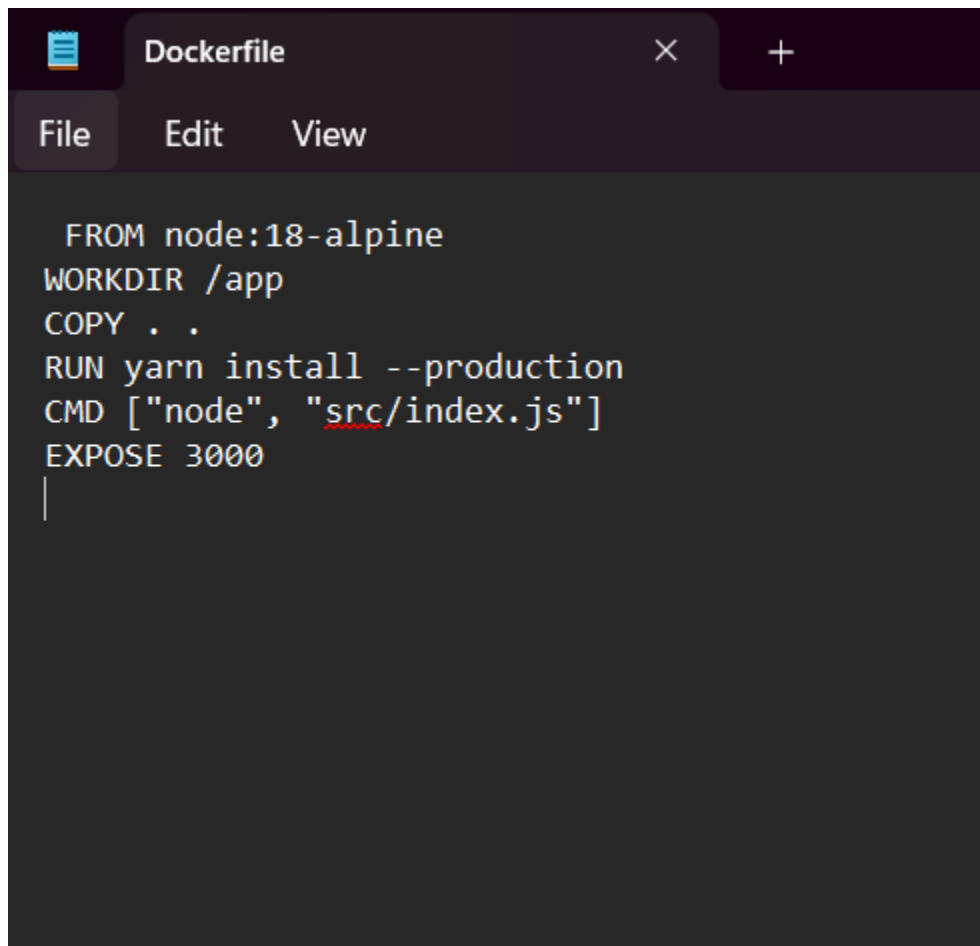
The image shows a code editor window with two tabs: 'Dockerfile' and 'package.json'. The 'package.json' tab is active, displaying the following JSON configuration:

```
{
  "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"
  }
}
```

## Lab Worksheet

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



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

## Lab Worksheet

```

C:\Users\phong\Lab8_4\getting-started\app>echo. > Dockerfile

C:\Users\phong\Lab8_4\getting-started\app>docker build -t myapp_6533802070 .
[+] Building 101.2s (10/10) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 159B                                0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine  4.8s
=> [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@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 61.9s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> => sha256:dcfb7b337595be6f4d214e4eed84f230eefe0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB 0.0s
=> => sha256:dcfb7b337595be6f4d214e4eed84f230eefe0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB 0.0s
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB 13.7s
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB 5.6s
=> => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB 0.0s
=> => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aeece8647b17a7bcebbcc385e 1.72kB / 1.72kB 0.0s
=> => sha256:37892ffbfc8a871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB 57.5s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B 6.5s
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 0.3s
=> => extracting sha256:37892ffbfc8a871a10f813803949d18c3015a482051d51b7e0da02525e63167c 3.3s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 0.1s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 0.0s
=> [internal] load build context                                   1.0s
=> => transferring context: 4.62MB                                    0.8s
=> [2/4] WORKDIR /app                                           0.6s
=> [3/4] COPY . .                                               0.2s
=> [4/4] RUN yarn install --production                          31.1s
=> exporting to image                                           2.2s
=> [2/4] WORKDIR /app                                           0.6s
=> [3/4] COPY . .                                               0.2s
=> [4/4] RUN yarn install --production                          31.1s
=> exporting to image                                           2.2s
=> => exporting layers                                           2.2s
=> => writing image sha256:72411cf3e060b4967c1ae56cec379fb96248011b32425553d0030f6b43ecc309 0.0s
=> => naming to docker.io/library/myapp_6533802070              0.0s

C:\Users\phong\Lab8_4\getting-started\app>
RAM 1.22 GB CPU 0.25% Disk: 1.49 GB used (limit 1006.85 GB)

```

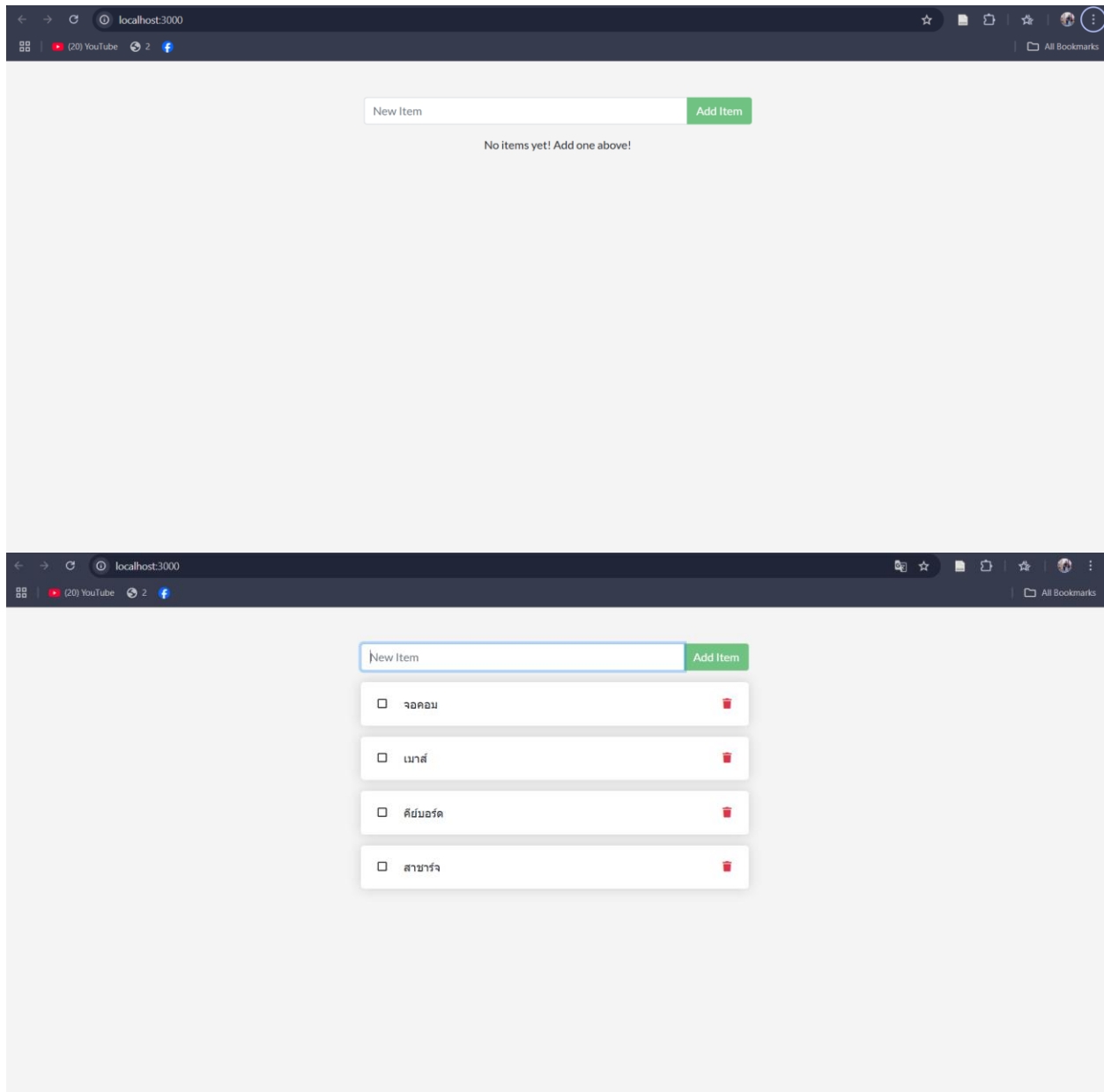
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

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

## Lab Worksheet



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

## Lab Worksheet

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

By ชื่อและนามสกุลของนักศึกษา</p>

b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

```
C:\Users\phong\Lab8_4\getting-started\app>docker build -t myapp_6533802070 .
[+] Building 56.7s (10/10) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 159B                                0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine  4.5s
=> [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@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> [internal] load build context                                   0.1s
=> => transferring context: 8.28kB                                   0.0s
=> CACHED [2/4] WORKDIR /app                                       0.0s
=> [3/4] COPY . .                                                  0.2s
=> [4/4] RUN yarn install --production                            49.4s
=> exporting to image                                              2.1s
=> => exporting layers                                              2.0s
=> => writing image sha256:f50a552af4ae30e62aececcce8d6cbd8b555dc87c7ab30bdddde0277a4828fdf 0.0s
=> => writing image sha256:f50a552af4ae30e62aececcce8d6cbd8b555dc87c7ab30bdddde0277a4828fdf 0.0s
=> => naming to docker.io/library/myapp_6533802070                0.0s

C:\Users\phong\Lab8_4\getting-started\app>docker build -t myapp_6533802070 .
[+] Building 2.2s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 159B                                0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine  1.9s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> [internal] load build context                                   0.0s
=> => transferring context: 2.49kB                                   0.0s
=> CACHED [2/4] WORKDIR /app                                       0.0s
=> CACHED [3/4] COPY . .                                          0.0s
=> CACHED [4/4] RUN yarn install --production                     0.0s
=> exporting to image                                              0.0s

=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> [internal] load build context                                   0.0s
=> => transferring context: 2.49kB                                   0.0s
=> CACHED [2/4] WORKDIR /app                                       0.0s
=> CACHED [3/4] COPY . .                                          0.0s
=> CACHED [4/4] RUN yarn install --production                     0.0s
=> exporting to image                                              0.0s
=> => exporting layers                                              0.0s
=> => writing image sha256:f50a552af4ae30e62aececcce8d6cbd8b555dc87c7ab30bdddde0277a4828fdf 0.0s
=> => naming to docker.io/library/myapp_6533802070                0.0s

C:\Users\phong\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533802070
de190c78b3bfd0bacc6485c0cf0a5aab5ce20c0de073d708df9ddac590ed5c3d
docker: Error response from daemon: driver failed programming external connectivity on endpoint lucid_euclid (d6e905c7fee2417a9069188a12389df6f2c384fbd58d90f397a3d5c58d95d0d6): Bind for 0.0.0.0:3000 failed: port is already allocated.

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

## Lab Worksheet

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

ข้อผิดพลาดเกิดจาก **port 3000** ถูกใช้งานอยู่แล้ว (โดย container หรือโปรแกรมอื่น) ทำให้ Docker ไม่สามารถใช้งาน port นี้ได้

การแก้ไข

-ตรวจสอบว่า container อื่นใช้ port นี้ด้วยคำสั่ง `docker ps` และหยุดมันด้วย `docker stop`

<container\_id>

-เปลี่ยน port binding เป็น port อื่น เช่น `docker run -dp 3001:3000 myapp_6533802070`

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

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. เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- iii. ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

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



## Lab Worksheet

```
C:\Users\phong\Lab8_4\getting-started\app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
8c062f690914   72411cf3e060   "docker-entrypoint.s..." 12 minutes ago Up 12 minutes   0.0.0.0:3000->3000/tcp   practical_yalow

C:\Users\phong\Lab8_4\getting-started\app>

C:\Users\phong\Lab8_4\getting-started\app>docker stop 8c062f690914
8c062f690914

C:\Users\phong\Lab8_4\getting-started\app>docker rm 8c062f690914
8c062f690914

C:\Users\phong\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533802070
dd7b7d9c24d5b9f3ffe57a4550f2e36f3335525770305e176499db0c86af2f06

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

localhost:3000

New Item

There is no TODO item. Please add one to the list. By นายพงษ์วรินทร์ แก้วสง่า 653380207-0

localhost:3000

New Item

- ☐ สมุด
- ☐ หนังสือ
- ☐ ปากกา
- ☐ ดินสอ

## Lab Worksheet

## แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop

2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต

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

หรือ

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

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

3. บันทึกที่รหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

```
C:\Users\phong\Lab8_4\getting-started\app>docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v jenkins_home:/var/jenkins_home jenkins/
jenkins:lts-jdk17
Unable to find image 'jenkins/jenkins:lts-jdk17' locally
lts-jdk17: Pulling from jenkins/jenkins
0a96bdb82805: Pulling fs layer
b5e2db483aae: Pull complete
72b137db2cdf: Pull complete
d0aac1e9e20c: Pull complete
c48e3974586c: Pull complete
d281eadf0ae3: Pull complete
be47ab5dc59d: Pull complete
c09c222384de: Pull complete
a91f2a3bb141: Pull complete
1a17773e6697: Pull complete

a91f2a3bb141: Pull complete
1a17773e6697: Pull complete
b535e1c5efef: Pull complete
3cfa79b75794: Pull 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
2025-01-29 15:53:59.042+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2025-01-29 15:54:00.796+0000 [id=1] WARNING o.e.jee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-01-29 15:54:00.909+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.16; built: 2024-12-09T21:02:54.535Z; git: c3
f88bafb4e393f23204dc14dc57b042e84debc7; jvm 17.0.13+11
2025-01-29 15:54:01.767+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclip
se.jetty.ee9.jsp.JettyJspServlet
2025-01-29 15:54:01.849+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-01-29 15:54:02.790+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVar
s.masterEnvVars.get("JENKINS_HOME")

s.masterEnvVars.get("JENKINS_HOME")
2025-01-29 15:54:03.003+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@772861a
a{Jenkins v2.479.3/,b=file:///var/jenkins_home/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@6631cb64{STARTED
}}
2025-01-29 15:54:03.030+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@c35af2a{HTTP/1.1, (http/1.1)}{0.
0.0.0:8080}
2025-01-29 15:54:03.054+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@1c7fd41f{STARTING}[12.0.16,sto=0] @
5148ms
2025-01-29 15:54:03.090+0000 [id=26] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-01-29 15:54:03.446+0000 [id=34] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-01-29 15:54:03.479+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-01-29 15:54:05.184+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-01-29 15:54:05.194+0000 [id=41] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-01-29 15:54:05.195+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-01-29 15:54:05.503+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-01-29 15:54:05.504+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-29 15:54:05.508+0000 [id=36] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
```

## Lab Worksheet

```

2025-01-29 15:54:05.508+0000 [id=36] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-29 15:54:05.514+0000 [id=33] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-29 15:54:05.558+0000 [id=60] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-01-29 15:54:06.142+0000 [id=40] INFO jenkins.install.SetupWizard#init:

*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.

*****
*****
*****

2025-01-29 15:54:19.583+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 15:54:19.621+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running

*****
*****
*****

2025-01-29 15:54:19.583+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 15:54:19.621+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-29 15:54:23.003+0000 [id=60] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2025-01-29 15:54:23.004+0000 [id=60] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1

```

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

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

**[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า**

- กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
- เมื่อติดตั้งเรียบร้อยแล้วจะพบหน้าจอ Dashboard ดังแสดงในภาพ

## Lab Worksheet

The screenshot shows the Jenkins Dashboard with a search bar and user profile. The left sidebar contains links for New Item, Build History, Manage Jenkins, and My Views. The main area displays a table of build history for a job named 'UAT'.

S	W	Name	Last Success	Last Failure	Last Duration	Robot Results + Duration Trend
		UAT	19 min #1	4 min 34 sec #3	5.7 sec	0/0 pass

Below the table, there are filters for 'Icon: S M L' and a 'Build Queue' section showing 'No builds in the queue.' and 'Build Executor Status' showing '0/2'.

The screenshot shows the Jenkins Dashboard with a search bar and user profile. The left sidebar contains links for New Item, Build History, Manage Jenkins, and My Views. The main area displays a 'Welcome to Jenkins!' message with instructions on how to get started. Below the message, there are buttons for 'Create a job', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

**Welcome to Jenkins!**

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

**Start building your software project**

Create a job

**Set up a distributed build**

Set up an agent

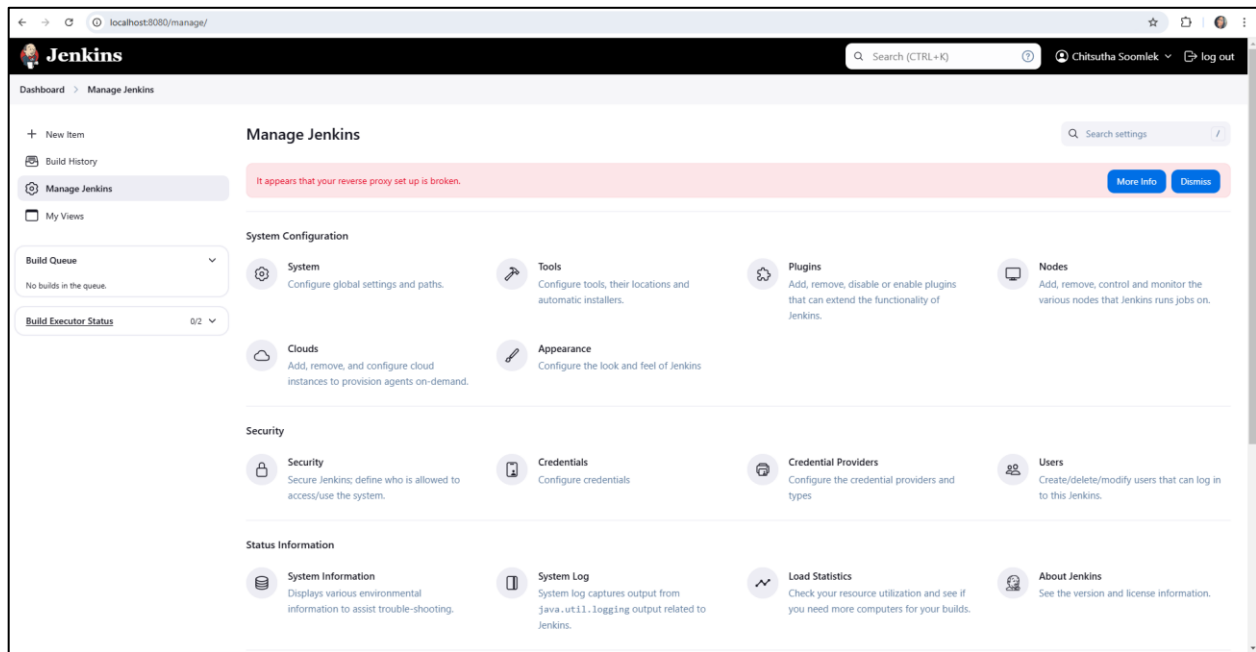
Configure a cloud

Learn more about distributed builds

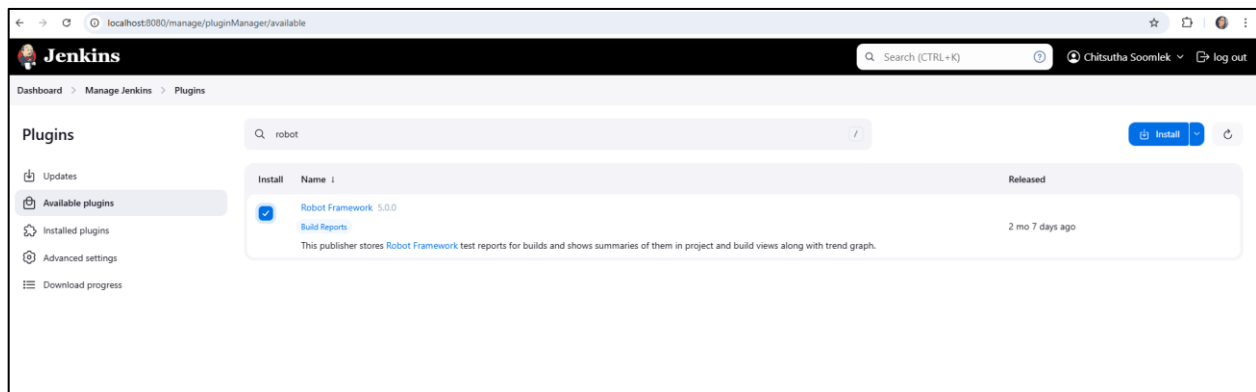
REST API Jenkins 2.479.3

9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

## Lab Worksheet

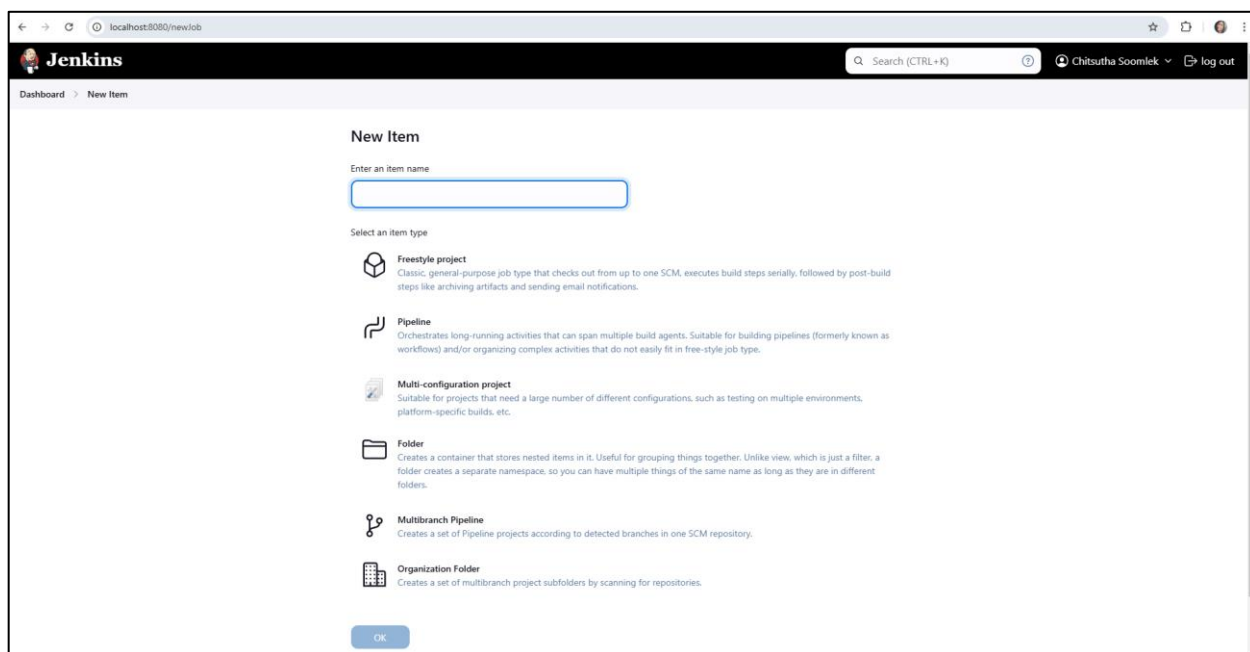


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

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

`./var/jenkins_home/workspace/UAT/robot_env/bin/activate`

`export PATH=$PATH:/var/jenkins_home/workspace/UAT`

`mkdir results`

`robot --outputdir results invalid_login.robot`

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

## Lab Worksheet

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

The screenshot shows the Jenkins Dashboard with the following elements:


- Header:** Jenkins logo, search bar (Search (CTRL+K)), user profile (phongwarin kaewsanga), and log out button.
- Left Sidebar:**
  - + New Item
  - Build History
  - Manage Jenkins
  - My Views
  - Build Queue: No builds in the queue.
  - Build Executor Status: 0/2
- Main Content Area:**
  - Filter: All +
  - Table of pipelines:
 

S	W	Name	Last Success	Last Failure	Last Duration	Robot Results + Duration Trend
		UAT	18 min #1	3 min 23 sec #3	5.7 sec	
  - Icon: S M L

The screenshot shows the Jenkins UAT pipeline details page with the following elements:

- Header:** Jenkins logo, search bar (Search (CTRL+K)), user profile (phongwarin kaewsanga), and log out button.
- Left Sidebar:**
  - Status
  - Changes
  - Workspace
  - Build Now
  - Configure
  - Delete Project
  - Robot Results
  - GitHub
  - Rename
- Main Content Area:**
  - UAT Pipeline Header:** UAT with a status icon and description button.
  - Latest Robot Results:**
    - Total Failed Passed Skipped Pass %
    - All tests: 6 | 6 | 0 | 0 | 0.0
    - Links: Browse results, Open report.html, Open log.html
  - Permalinks:**
    - Last build (#3), 3 min 6 sec ago
    - Last stable build (#1), 18 min ago
    - Last successful build (#1), 18 min ago
    - Last failed build (#3), 3 min 6 sec ago
    - Last unsuccessful build (#3), 3 min 6 sec ago
    - Last completed build (#3), 3 min 6 sec ago
  - Robot Framework Tests Trend (all tests):**
    - Bar chart showing Number of test cases (0 to 6) vs Build.
    - Legend: Skipped (yellow), Passed (green), Failed (red).
    - Buttons: Zoom to changes, Show only failed, Show only critical, all (selected), Max builds, Show bigger image.
  - Builds List:**
    - Filter: /
    - Today:
      - #3 9:35 AM
      - #2 9:34 AM

## Lab Worksheet

 #2 (Jan 30, 2025, 9:34:44 AM) Add description

Keep this build forever

Started by user [phongwarin kaewsanga](#)

Started 3 min 3 sec ago

Took [3.2 sec](#)

This run spent:

- 35 ms waiting;
- 3.2 sec build duration;
- 3.3 sec total from scheduled to completion.

**Revision:** 9cfba65808a589d1ec872b9823f8d60feedf4226**Repository:** [https://github.com/kku-computer-science/lab7\\_653380207-0.git](https://github.com/kku-computer-science/lab7_653380207-0.git)

- refs/remotes/origin/main

**Robot Test Summary:**

	Total	Failed	Passed	Skipped	Pass %
All tests	6	6	0	0	0.0

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



No changes.



## Lab Worksheet

```

-----
Empty Password | FAIL |
Parent suite setup failed:
NoSuchDriverException: Message: Unable to obtain driver for chrome; For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors/driver\_location
-----

Empty Username And Password | FAIL |
Parent suite setup failed:
NoSuchDriverException: Message: Unable to obtain driver for chrome; For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors/driver\_location
-----

Invalid Login :: A test suite containing tests related to invalid ... | FAIL |
Suite setup failed:
NoSuchDriverException: Message: Unable to obtain driver for chrome; For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors/driver\_location
-----

6 tests, 0 passed, 6 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
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:
documentation/webdriver/troubleshooting/errors/driver\_location

6 tests, 0 passed, 6 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
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

```