

## Lab Worksheet

ชื่อ-นามสกุล จีรวรรณ สิงห์กลั่น รหัสนักศึกษา 653380191-9 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 พร้อมกับตอบ

## Lab Worksheet

## คำถามต่อไปนี

```

Download an image from a registry
PS 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

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview busybox
PS C:\Users\computer\Lab8_1> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ai-final      latest    4fb4cf7f2382   3 months ago   7.64GB
busybox       latest    af4709625109   3 months ago   4.27MB
PS C:\Users\computer\Lab8_1> |

```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร ai-final busybox
- (2) Tag ที่ใช้บอกถึงอะไร ภาพ busybox ใช้แท็ก latest ซึ่งหมายถึงเป็นภาพเวอร์ชันล่าสุดใน repository busybox
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 พร้อมกับตอบคำถามต่อไปนี

## Lab Worksheet

```
C:\Users\lenovo\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\lenovo\Lab8_1>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
busybox              latest          af4709625109   4 months ago   4.27MB
docker/welcome-to-docker latest         c1f619b6477e   14 months ago  18.6MB

C:\Users\lenovo\Lab8_1>docker run busybox

C:\Users\lenovo\Lab8_1>docker run -it busybox sh
/ # ls
/ # ls -la
total 48
drwxr-xr-x 1 root    root      4096 Jan 29 13:04 .
drwxr-xr-x 1 root    root      4096 Jan 29 13:04 ..
-rwxr-xr-x 1 root    root        0 Jan 29 13:04 .dockerenv
drwxr-xr-x 2 root    root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root    root      360 Jan 29 13:04 dev
drwxr-xr-x 1 root    root      4096 Jan 29 13:04 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 237 root  root        0 Jan 29 13:04 proc
drwx----- 1 root    root      4096 Jan 29 13:04 root
dr-xr-xr-x 11 root    root        0 Jan 29 13:04 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\lenovo\Lab8_1>docker run busybox echo "Hello Jeerawan Singkin 653380191-9 from busybox"
Hello Jeerawan Singkin 653380191-9 from busybox

C:\Users\lenovo\Lab8_1>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS      PORTS   NAMES
ccbcfa277645   busybox   "echo 'Hello Jeerawa..." 22 seconds ago Exited (0) 21 seconds ago           quirky_bardeen
1ab13c31805b   busybox   "sh"                     2 minutes ago Exited (0) About a minute ago       interesting_jang
41e0e9cf6ccd   busybox   "sh"                     2 minutes ago Exited (0) 2 minutes ago           loving_brattain
```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

-i ทำให้ container ยังคงเปิดอยู่และสามารถรับ input จากผู้ใช้งานผ่าน terminal ได้

ใช้สำหรับ container ที่ต้องการรับคำสั่งจาก command line (เช่น sh, bash)

-t จำลอง terminal (TTY) ทำให้ container แสดงผลลัพธ์เหมือนกำลังใช้งาน shell จริง ๆ

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

แสดง สถานะของ container ในปัจจุบัน

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

```
C:\Users\lenovo\Lab8_1>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS      PORTS   NAMES
ccbcfa277645   busybox   "echo 'Hello Jeerawa..." 22 seconds ago Exited (0) 21 seconds ago           quirky_bardeen
1ab13c31805b   busybox   "sh"                     2 minutes ago Exited (0) About a minute ago       interesting_jang
41e0e9cf6ccd   busybox   "sh"                     2 minutes ago Exited (0) 2 minutes ago           loving_brattain

C:\Users\lenovo\Lab8_1>docker rm ccbcfa277645
ccbcfa277645

C:\Users\lenovo\Lab8_1>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS      PORTS   NAMES
1ab13c31805b   busybox   "sh"                     16 minutes ago Exited (0) 15 minutes ago       interesting_jang
41e0e9cf6ccd   busybox   "sh"                     17 minutes ago Exited (0) 17 minutes ago       loving_brattain

C:\Users\lenovo\Lab8_1>
```

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

## Lab Worksheet

## 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
```

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

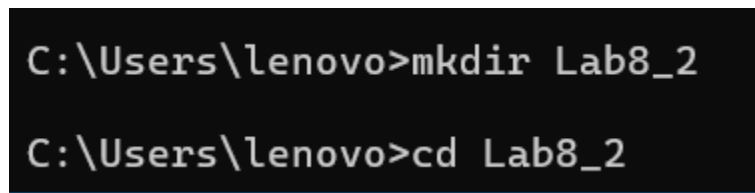
## Lab Worksheet

```

C:\Users\lenovo>mkdir Lab8_2

C:\Users\lenovo>cd Lab8_2

```



```

Dockerfile.swp X
C: > Users > lenovo > Lab8_2 > Dockerfile.swp
1 FROM busybox
2 CMD echo "Hi there. This is my first docker image. น.ส จีรารณ สิงห์กลั่น 653380191-9 จี พ"
3 |

C:\Users\lenovo\Lab8_2>docker build -t my-first-image -f Dockerfile.swp .
[+] Building 0.2s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 196B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2) 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                     0.0s
=> CACHED [1/1] FROM docker.io/library/busybox:latest          0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> writing image sha256:684a71e167c966425430d6578c439720417a3744a77aea2a5d401c628c5f0aad 0.0s
=> naming to docker.io/library/my-first-image                  0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/74x548kjett33z7z8jxw7rb88

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

C:\Users\lenovo\Lab8_2>docker run my-first-image
Hi there. This is my first docker image. น.ส จีรารณ สิงห์กลั่น น 653380191-9 จี พ

C:\Users\lenovo\Lab8_2>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
busybox              latest             af4709625100       4 months ago       4.27MB
my-first-image       latest             684a71e167c9       4 months ago       4.27MB
docker/welcome-to-docker latest            c1f619b6477e       14 months ago      18.6MB

```

(1) คำสั่งที่ใช้ในการ run คือ

docker run my-first-image

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

-t (--tag) ใช้ กำหนดชื่อและ tag ของ Image

ทำให้เรียกใช้งาน Image ได้ง่ายขึ้น

ถ้าไม่ใช้ -t Docker จะสร้าง Image ที่ไม่มีชื่อ (<none>)

### 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

## Lab Worksheet

## 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
```

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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\lenovo\Lab8_3> dir

Directory: C:\Users\lenovo\Lab8_3

Mode                LastWriteTime         Length Name
----                -
-a----             1/29/2025   9:03 PM             165 Dockerfile.swp
```

## Lab Worksheet

```

Dockerfile.swp
C: > Users > lenovo > Lab8_3 > Dockerfile.swp
1 FROM busybox
2 CMD echo "Hi there. My work is done. You can run them from my Docker image. น.ส จิรวรรณ สิงห์กลิ่น 653380191-9 จีฟ"
3
4 |

PS C:\Users\lenovo\Lab8_3> docker build -t jeerawansingkin/lab8 -f Dockerfile.swp .
[+] Building 0.2s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 223B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                   0.0s
=> CACHED [1/1] FROM docker.io/library/busybox:latest          0.0s
=> exporting to image                                          0.0s
=> => exporting layers                                          0.0s
=> => writing image sha256:170887735d5c1eb792e585e89d1b04d00d169bea64d43f2a15bfa63b98b3e842 0.0s
=> => naming to docker.io/jeerawansingkin/lab8                 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/xceunwbsa9eu9l3s01f96vsko

1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
PS C:\Users\lenovo\Lab8_3> docker run jeerawansingkin/lab8
Hi there. My work is done. You can run them from my Docker image. น.ส จิรวรรณ สิงห์กลิ่น น 653380191-9 จีฟ
PS C:\Users\lenovo\Lab8_3> docker push jeerawansingkin/lab8
Using default tag: latest
The push refers to repository [docker.io/jeerawansingkin/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:ab7b5d79b29f7efe2d09b80e17961dda38b4f01589dc2995da5ef783a22b410d size: 527

```

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

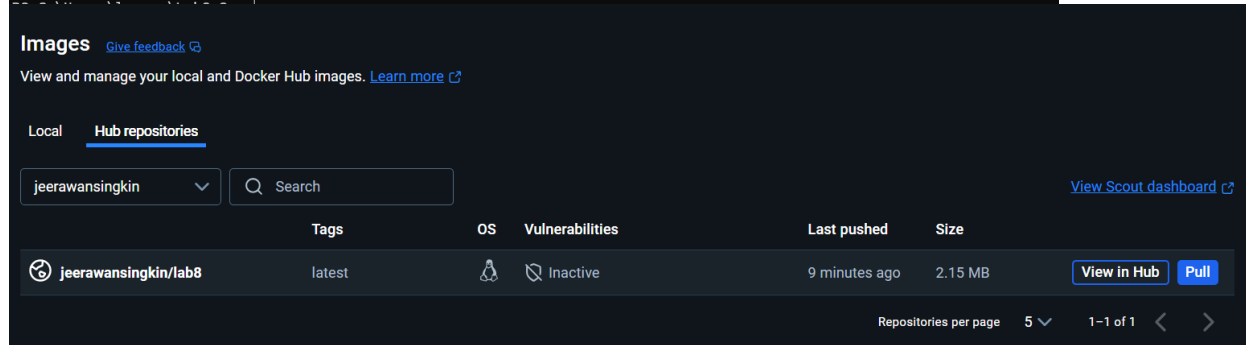
[Check point#6] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดง Repository ที่มี Docker image (<username>/lab8)

## Lab Worksheet

```
PS C:\Users\lenovo\Lab8_3> docker build -t jeerawansingkin/lab8 -f Dockerfile.swp .
[+] Building 0.2s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 223B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                    0.0s
=> CACHED [1/1] FROM docker.io/library/busybox:latest          0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:170887735d5c1eb792e585e89d1b04d00d169bea64d43f2a15bfa63b98b3e842 0.0s
=> => naming to docker.io/jeerawansingkin/lab8                 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/xceunwbsa9eu9l3s01f96vnko

1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
PS C:\Users\lenovo\Lab8_3> docker run jeerawansingkin/lab8
Hi there. My work is done. You can run them from my Docker image. น.ส จี รววรรณสิงห์ กลิ่น น 653380191-9 จี พ
PS C:\Users\lenovo\Lab8_3> docker push jeerawansingkin/lab8
Using default tag: latest
The push refers to repository [docker.io/jeerawansingkin/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:ab7b5d79b29f7efe2d09b80e17961dda38b4f01589dc2995da5ef783a22b410d size: 527
-- .....
```



## 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

```
C:\Users\lenovo>mkdir Lab8_4

C:\Users\lenovo>cd Lab8_4

C:\Users\lenovo\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 | 2.15 MiB/s, done.
Resolving deltas: 100% (523/523), done.
```

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

```
C:\Users\lenovo\Lab8_4\getting-started\app>code package.json
```

```
package.json X Dockerfile.swp
C: > Users > lenovo > Lab8_4 > getting-started > app > package.json > ...

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
```

## 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  
ศ. ไม่มีขีด

```
$ docker build -t <myapp_รหัสสนศ. ไม่มีขีด> .
```

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

```
C:\Users\lenovo\Lab8_4\getting-started\app>code Dockerfile
C:\Users\lenovo\Lab8_4\getting-started\app>docker build -t myapp_6533801919 .
[*] Building 29.1s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 15kB
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974af66cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> resolve docker.io/library/node:18-alpine@sha256:974af66cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB
=> sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB
=> sha256:974af66cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB
=> sha256:6e804119c3884fc5782795bf0d2ad89201c63105aece8647b17a7bcebb385e 1.72kB / 1.72kB
=> sha256:dcbf7b337595be6fd4d214e4eed84f238ee0e4ac83a50380d573e289b9e5e40 6.18kB / 6.18kB
=> sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB
=> extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0
=> sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B
=> extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c
=> extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1
=> extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771
=> [internal] load build context
=> => transferring context: 4.62MB
=> [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> writing image sha256:2bcd196328cb5e064a49e19b4d2b8642f56033c000bc49aad2a4e617604a2b6c
=> naming to docker.io/library/myapp_6533801919
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/av537h7yaiawoegolhg25qj5r
C:\Users\lenovo\Lab8_4\getting-started\app>
```

```
package.json × Dockerfile ×
C: > Users > lenovo > Lab8_4 > getting-started > app > Dockerfile
1 FROM node:18-alpine
2 WORKDIR /app
3 COPY . .
4 RUN yarn install --production
5 CMD ["node", "src/index.js"]
6 EXPOSE 3000
```

## Lab Worksheet

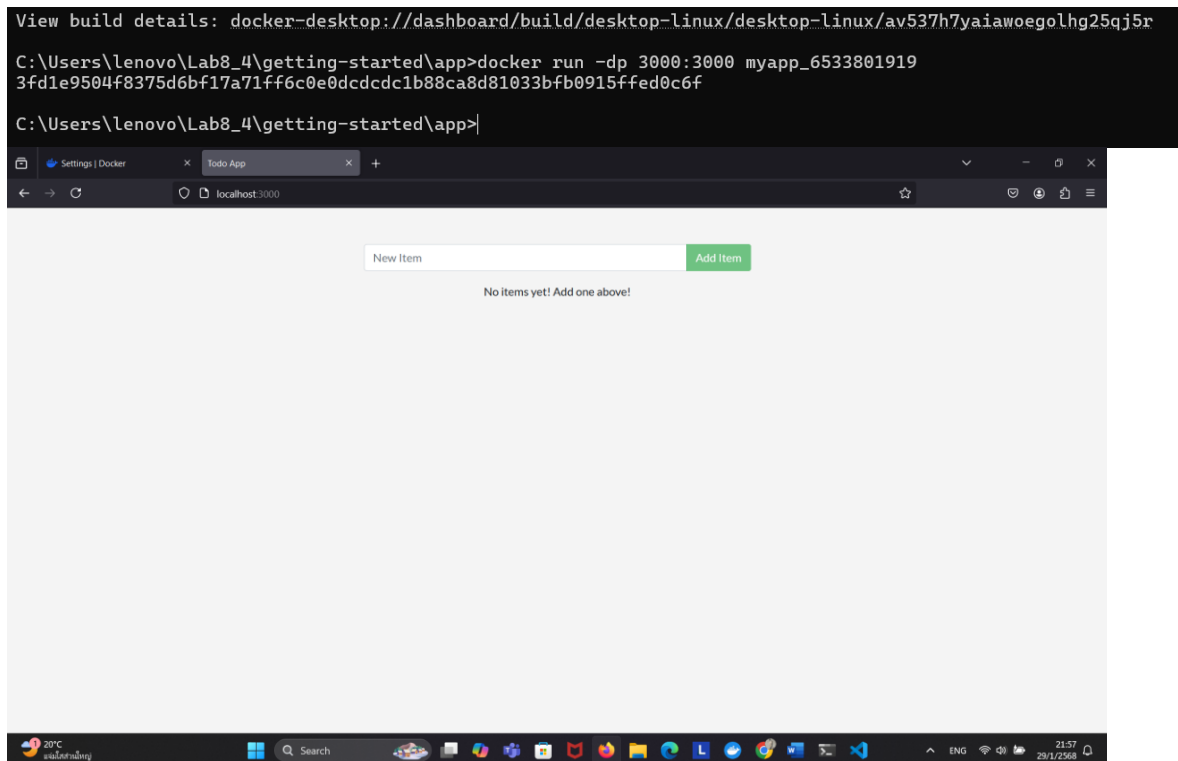
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

```
$ 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

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้



## Lab Worksheet

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</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\lenovo\Lab8_4\getting-started\app>code src/static/js/app.js

C:\Users\lenovo\Lab8_4\getting-started\app>docker build -t myapp_6533801919 .
[+] Building 12.9s (10/10) FINISHED
=> => transferring dockerfile: 154B
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> [internal] load build context
=> => transferring context: 8.14kB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting image
=> exporting layers
=> writing image sha256:a8a04395f3b99326d3a40797c605d2f60fec751e3097762b1d58654f2ab1a870
=> naming to docker.io/library/myapp_6533801919

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/pnje6lsic20zlojr9zj95spfb
C:\Users\lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801919 .
docker: invalid reference format.
See 'docker run --help'.

C:\Users\lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801919
389489fb11a6e811d2f027c6acff7df14deb28dc2e14506c8f2f5185fb6f30e1
docker: Error response from daemon: driver failed programming external connectivity on endpoint sharp_saha (06d16f54ae38cae11c5a19d418a89ad1af21ca4d629487bd23451b72a0d5b646): Bind for 0.0.0.0:3000 failed: port is already allocated.
```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร  
Container ใหม่พยายามใช้ Port 3000 แต่ Docker ไม่สามารถ Bind Port ได้ เพราะ Container เก่ายังรันอยู่

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

## Lab Worksheet

b. ผ่าน Docker desktop

- i. ไปที่หน้าต่าง Containers
- ii. เลือกไอคอนถึงขยะในแถวของ 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

```
C:\Users\lenovo\Lab8_4\getting-started\app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                   NAMES
166b91591b10   myapp_6533801919  "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  0.0.0.0:3000->3000/tcp  sad_tesla

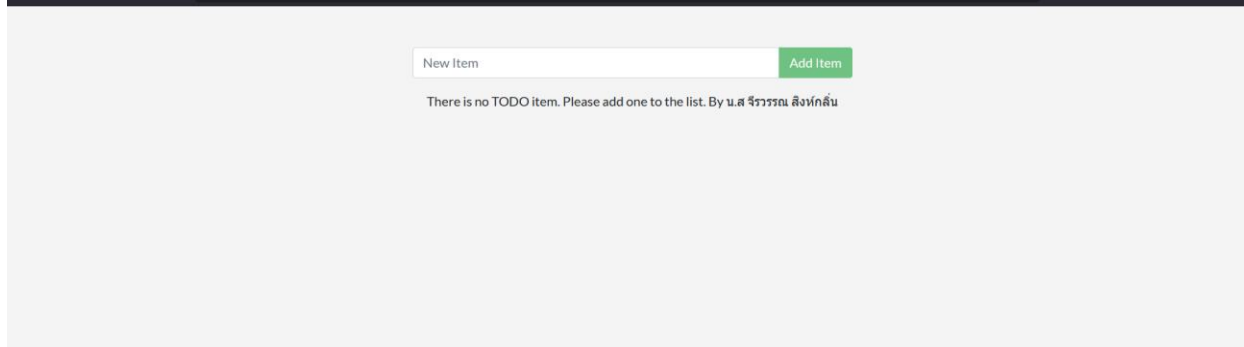
C:\Users\lenovo\Lab8_4\getting-started\app>docker stop 166b91591b10
166b91591b10

C:\Users\lenovo\Lab8_4\getting-started\app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                   NAMES

C:\Users\lenovo\Lab8_4\getting-started\app>docker rm 166b91591b10
166b91591b10

C:\Users\lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801919
3eaa3a65624d82b74f49e7fbc478d443dd517250816c0973527718822d6a011

C:\Users\lenovo\Lab8_4\getting-started\app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                   NAMES
3eaa3a65624d   myapp_6533801919  "docker-entrypoint.s..." 5 seconds ago  Up 4 seconds  0.0.0.0:3000->3000/tcp  gifted_curran
```



## Lab Worksheet

## แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop

2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต

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

หรือ

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

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

3. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

[Check point#12] Capture หน้าจอที่แสดงผล Admin password

```
C:\Users\lenovo>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: Pull complete
b5e2db493aae: Pull complete
72b137db2cdf: Pull complete
d8aac1e9e28c: Pull complete
c48e3974586c: Pull complete
d281eadf8ae3: Pull complete
be47ab5dc59d: Pull complete
c89c222384de: Pull complete
a91f2a3bb141: Pull complete
1a17773e6697: Pull complete
b535e1c5efef: Pull complete
3cfa79b75794: Pull complete
Digest: sha256:dc56634cc8fa476f36ebal6d7db6c6bc1f5e3c7062432256738a008a339ee95c
Status: Downloaded newer image for jenkins/jenkins:lts-jdk17
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-01-29 15:55:04.445+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2025-01-29 15:55:04.971+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-01-29 15:55:05.006+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.16; built: 2024-12-09T21:02:54.535Z; git: c3f88baf4e393
f23280dc1ddc57b042e84debc7; jvm 17.0.13+11
2025-01-29 15:55:05.251+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9
.jsp.JettyJspServlet
2025-01-29 15:55:05.284+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-01-29 15:55:05.635+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvV
ars.get("JENKINS_HOME")
2025-01-29 15:55:05.714+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@772861aa{Jenkins v2
.479.3,/,b=file:///var/jenkins_home/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@86631cb64{STARTED}}
2025-01-29 15:55:05.725+0000 [id=1] INFO o.e.j.s.server.AbstractConnector#doStart: Started ServerConnector@27df0f3d{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2025-01-29 15:55:05.735+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@1c7fd41f{STARTING}[12.0.16,sto=0] @1669ms
2025-01-29 15:55:05.736+0000 [id=26] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-01-29 15:55:05.879+0000 [id=34] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-01-29 15:55:05.897+0000 [id=54] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-01-29 15:55:06.385+0000 [id=33] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-01-29 15:55:06.399+0000 [id=48] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-01-29 15:55:06.391+0000 [id=55] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

b6cb8a1a556344a2b6b0634f1b7a0198

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

*****
*****
*****
```

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

## Lab Worksheet

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

The top screenshot shows the Jenkins Setup Wizard 'Getting Started' screen. It displays a list of plugins to be installed, including Folders Plugin, OWASP Markup Formatter Plugin, Build Timeout, Credentials Binding Plugin, Timestamper, Workspace Cleanup Plugin, Ant Plugin, Gradle, Pipeline, GitHub Branch Source, Pipeline: GitHub Groovy Libraries, Pipeline Graph View, Git, SSH Build Agents, Matrix Authorization Strategy, PAM Authentication, LDAP, Email Extension, and Mailer. The bottom screenshot shows the same screen with a form to enter user details: Username (jeerawan\_1919), Password (\*\*\*\*\*), Confirm password (\*\*\*\*\*), Full name (Jeerawan Singkin), and E-mail address (jeerawan.s@kkumail.com). The 'Save and Continue' button is visible at the bottom right.

## Lab Worksheet

Getting Started

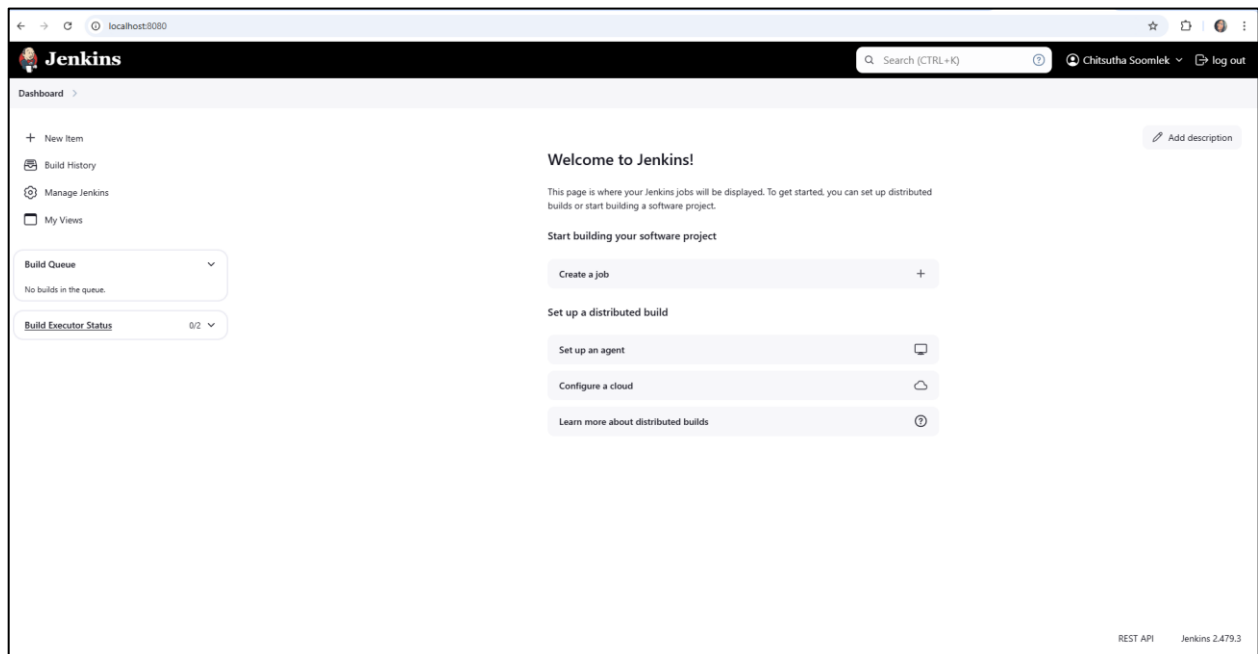
# Instance Configuration

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.

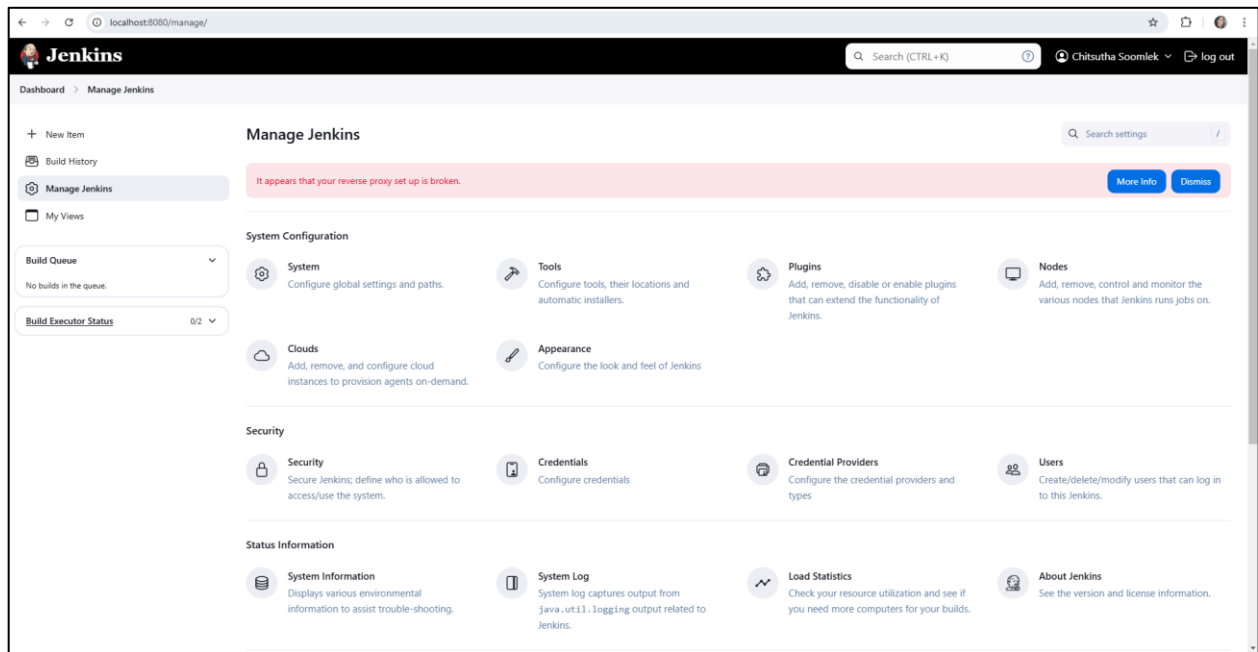
- กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
- เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



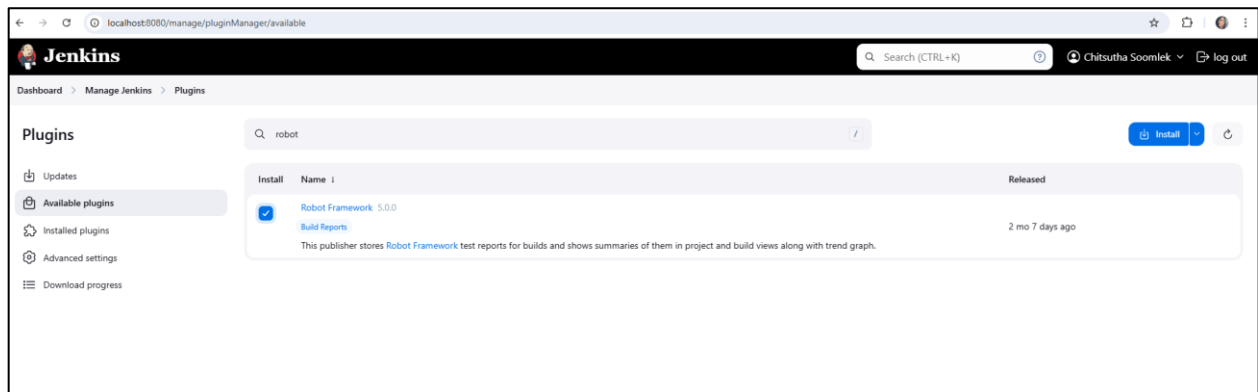
- เลือก Manage Jenkins แล้วไปที่เมนู Plugins



## Lab Worksheet

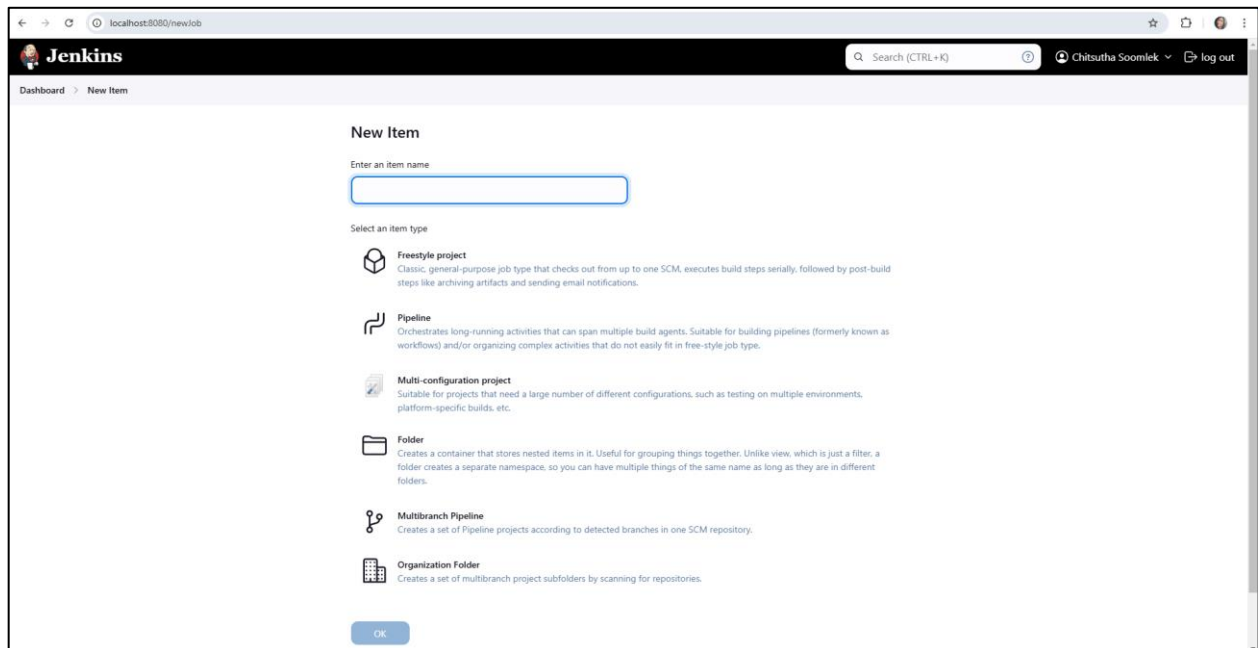


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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 image shows two screenshots of the Jenkins web interface. The top screenshot displays the 'General' configuration tab for a job named 'UAT'. The 'Enabled' toggle is turned on. The 'Description' field contains the text: 'This is a simple Jenkins Pipeline for UAT testing.' The 'GitHub project' checkbox is checked, and the 'Project url' is set to 'https://github.com/jeerawan-singkin/lab7'. The bottom screenshot displays the 'Source Code Management' configuration tab for the same job. The 'None' radio button is selected under 'Source Code Management'. Under 'Build Triggers', the 'Build periodically' checkbox is checked, and the 'Schedule' is set to 'H/15 \* \* \* \*'. The 'GitHub hook trigger for GITScm polling' and 'Poll SCM' checkboxes are unchecked.

**Jenkins** Search (CTRL+K) Jeerawan Singkin log out

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

#### General

Enabled ☒

Description

This is a simple Jenkins Pipeline for UAT testing.

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

https://github.com/jeerawan-singkin/lab7

Advanced ▾

☐ This project is parameterized ?

Save Apply

localhost:8080/job/UAT/configure

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

#### Source Code Management

☒ None

☐ Git ?

#### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

H/15 \* \* \* \* \*

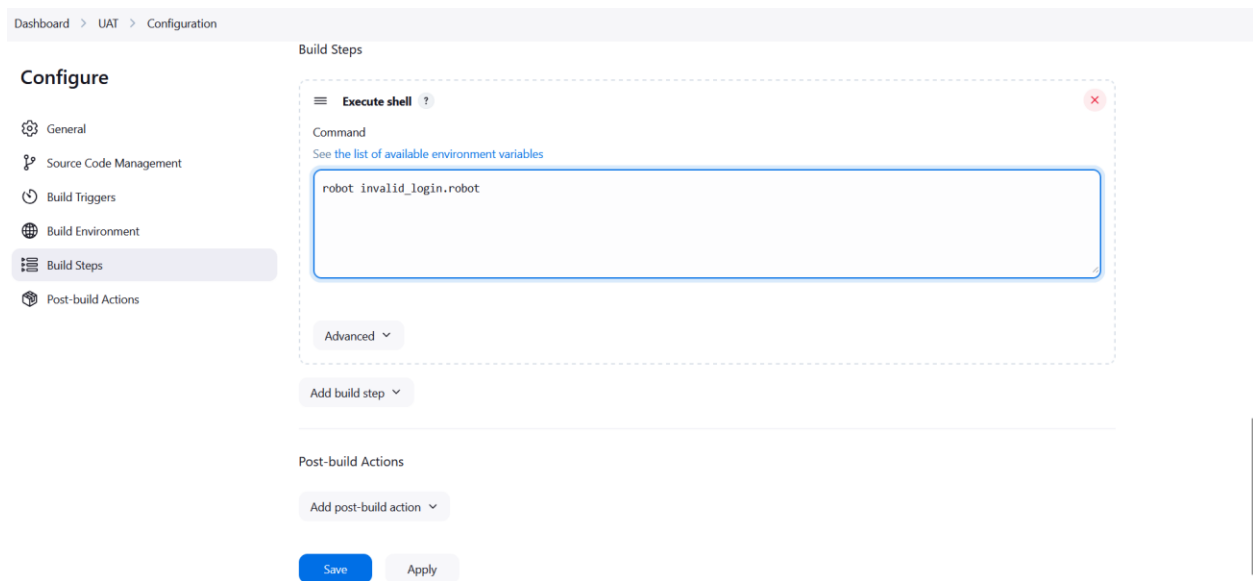
Would last have run at Wednesday, January 29, 2025 at 4:35:37 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 4:50:37 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Save Apply

## Lab Worksheet



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

**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