

## Lab Worksheet

ชื่อ-นามสกุล \_\_\_\_\_ ปวโรสุตา ทิพนาสา \_\_\_\_\_ รหัสนักศึกษา \_\_\_\_\_ 653380137-5 \_\_\_\_\_ 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

## Lab Worksheet

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

```

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

E:\year3_2\softwareEng\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

E:\year3_2\softwareEng\Lab8_1>

E:\year3_2\softwareEng\Lab8_1>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
dpag/pgadmin4       latest          199b8ca63523    6 weeks ago    507MB
mongo               latest          f08e39122805    6 weeks ago    855MB
postgres            latest          810c36706d00    2 months ago   435MB
busybox             latest          af4709625109    3 months ago   4.27MB

E:\year3_2\softwareEng\Lab8_1>docker run busybox
E:\year3_2\softwareEng\Lab8_1>

E:\year3_2\softwareEng\Lab8_1>docker run busybox echo "Hello Paweesuda Thippayanasa from busybox"
Hello Paweesuda Thippayanasa from busybox

E:\year3_2\softwareEng\Lab8_1>docker run busybox

E:\year3_2\softwareEng\Lab8_1>docker run -it busybox sh
/ # ls
/ # ls -la
total 48
drwxr-xr-x  1 root    root          4096 Jan 22 03:57 .
drwxr-xr-x  1 root    root          4096 Jan 22 03:57 ..
-rwxr-xr-x  1 root    root           0 Jan 22 03:57 .dockerenv
drwxr-xr-x  2 root    root        12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root         360 Jan 22 03:57 dev
drwxr-xr-x  1 root    root         4096 Jan 22 03:57 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 238 root    root           0 Jan 22 03:57 proc
drwx----- 1 root    root         4096 Jan 22 03:57 root
dr-xr-xr-x 11 root    root           0 Jan 22 03:57 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

E:\year3_2\softwareEng\Lab8_1>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS      PORTS          NAMES
8fc3fc814066   busybox    "echo 'Hello Paweesuda'" 28 seconds ago Exited (0) 26 seconds ago           recursing_wing
792e29f74d09   busybox    "sh"                    3 minutes ago Exited (0) 2 minutes ago           reverent_tu
84f088a0a6f5c   busybox    "sh"                    3 minutes ago Exited (0) 3 minutes ago           interesting_swartz
4fb8ea04da09   mongo     "docker-entrypoint.s..." 38 hours ago  Exited (0) 37 minutes ago           mongg0db-mongo-1
2b257b4a593d   dpag/pgadmin4:latest "/entrypoint.sh"         4 weeks ago   Exited (0) 38 hours ago           pgadmin
43dd7963b5e8   postgres:latest "docker-entrypoint.s..." 4 weeks ago   Exited (0) 38 hours ago           postgres

```

## Lab Worksheet

- (1) สิ่งที่อยู่ภายใต้คอนเทนเนอร์ Repository คืออะไร\_\_แสดงชื่อของ Docker image ที่ถูกดึงมาจาก Docker Hub หรือที่สร้างขึ้นมาเอง\_\_\_\_\_
- (2) Tag ที่ใช้บ่งบอกถึงอะไร\_\_ใช้เพื่อระบุเวอร์ชันของ Docker image หากไม่ระบุ Tag จะใช้ค่า default เป็น latest\_\_\_\_\_
5. ป้อนคำสั่ง \$ docker run busybox
6. ป้อนคำสั่ง \$ docker run -it busybox sh
7. ป้อนคำสั่ง ls
8. ป้อนคำสั่ง ls -la
9. ป้อนคำสั่ง exit
10. ป้อนคำสั่ง \$ docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"
11. ป้อนคำสั่ง \$ docker ps -a

[Check point#2] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ตั้งแต่ขั้นตอนที่ 6-12 พร้อมกับตอบคำถามต่อไปนี้

```
E:\year3_2\softwareEng\Lab8_1>docker run -it busybox sh
/ # echo "Hello Paweesuda Thippayanasa from busybox"
Hello Paweesuda Thippayanasa from busybox
/ # |

E:\year3_2\softwareEng\Lab8_1>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
8fc3fc814066   busybox    "echo 'Hello Paweesu..." 28 seconds ago Exited (0)    26 seconds ago    recurring_wing
792e29f74d09   busybox    "sh"                    3 minutes ago  Exited (0)    2 minutes ago    reverent_tu
84f08a0a6f5c   busybox    "sh"                    3 minutes ago  Exited (0)    3 minutes ago    interesting_swartz
4fb8ea04da09   mongo     "docker-entrypoint.s..." 38 hours ago   Exited (0)    37 minutes ago    monggg0db-mongo-1
2b257b4a593d   dpkg/pgadmin4:latest "/entrypoint.sh"        4 weeks ago   Exited (0)    38 hours ago    pgadmin
43dd7963b5e8   postgres:latest "docker-entrypoint.s..." 4 weeks ago   Exited (0)    38 hours ago    postgres

E:\year3_2\softwareEng\Lab8_1>docker run -it busybox sh
/ # echo "Hello Paweesuda Thippayanasa from busybox"
Hello Paweesuda Thippayanasa from busybox
/ # exit

E:\year3_2\softwareEng\Lab8_1>docker rm 2b257b4a593d
2b257b4a593d

E:\year3_2\softwareEng\Lab8_1>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
9dc32a467fab   busybox    "sh"                    4 minutes ago  Exited (0)    About a minute ago    naughty_pare
8fc3fc814066   busybox    "echo 'Hello Paweesu..." 23 minutes ago Exited (0)    23 minutes ago    recurring_wing
792e29f74d09   busybox    "sh"                    25 minutes ago Exited (0)    25 minutes ago    reverent_tu
84f08a0a6f5c   busybox    "sh"                    26 minutes ago Exited (0)    26 minutes ago    interesting_swartz
4fb8ea04da09   mongo     "docker-entrypoint.s..." 38 hours ago   Exited (0)    About an hour ago    monggg0db-mongo-1
43dd7963b5e8   postgres:latest "docker-entrypoint.s..." 4 weeks ago   Exited (0)    38 hours ago    postgres

E:\year3_2\softwareEng\Lab8_1>
```

- (1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
-it ใช้เพื่อเปิด interactive terminal ใน container ทำให้สามารถโต้ตอบกับ container ได้ผ่าน command line
- (2) คอนเทนเนอร์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร  
แสดงสถานะของ container เช่น กำลังทำงาน (Up), หยุดทำงาน (Exited), หรือสร้างแล้วแต่ยังไม่เริ่มทำงาน (Created)\_\_\_\_\_

## Lab Worksheet

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

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

```
E:\year3_2\softwareEng\Lab8_1>docker rm 2b257b4a593d
2b257b4a593d
```

## 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

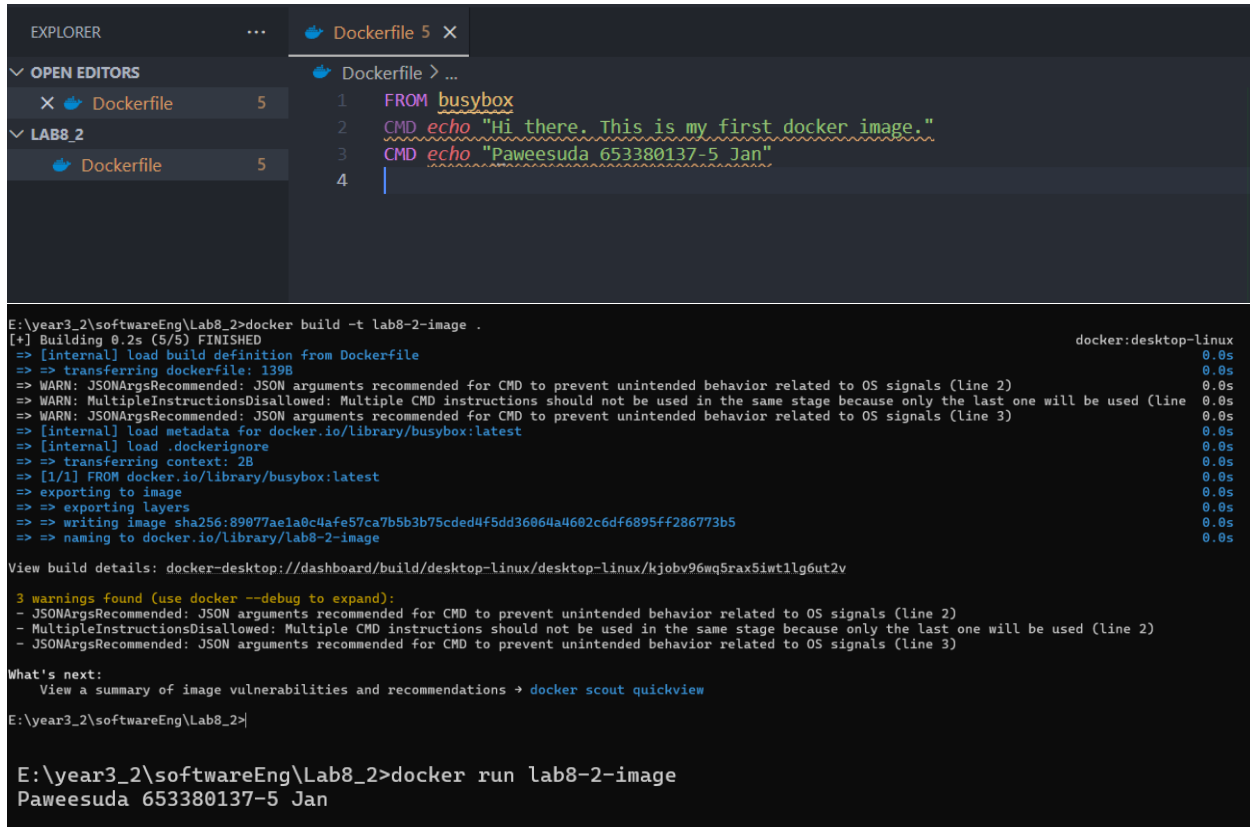
5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

```
$ docker build -t <ชื่อ Image> .
```

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

## Lab Worksheet

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



```

EXPLORER
...
Dockerfile 5 X

OPEN EDITORS
Dockerfile 5
LAB8_2
Dockerfile 5

1 FROM busybox
2 CMD echo "Hi there. This is my first docker image."
3 CMD echo "Paweesuda 653380137-5 Jan"
4

E:\year3_2\softwareEng\Lab8_2>docker build -t lab8-2-image .
[*] Building 0.2s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 139B
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 3)
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:89877a6a0c4afe57ca7b5b3b75cded4f5dd36864a4682c6df6895ff286773b5
=> => naming to docker.io/library/lab8-2-image

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/kjobv96mq5rax5iwtllg6ut2v

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

E:\year3_2\softwareEng\Lab8_2>

E:\year3_2\softwareEng\Lab8_2>docker run lab8-2-image
Paweesuda 653380137-5 Jan
  
```

(1) คำสั่งที่ใช้ในการ run คือ

docker run lab8-2-image

(2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
ใช้ระบุชื่อให้กับ Docker image ที่สร้างขึ้น โดยจะช่วยให้การเรียกใช้หรือการจัดการ Docker image นั้น  
ง่ายขึ้น แทนที่จะใช้ ID ของ image ที่ยาวและยากต่อการจำ การใช้ -t จะกำหนดชื่อที่สามารถเข้าใจได้ เช่น  
-t lab8-2-image

## 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
```

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

## Lab Worksheet

```
E:\year3_2\softwareEng\Lab8_3>docker build -t 74an/lab8 .
[+] Building 0.1s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 139B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because 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 sha256:89077a1a0c4afe57ca7b5b3b75cded4f5dd36064a4602c6df6895ff286773b5 0.0s
=> => naming to docker.io/74an/lab8                             0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/dt#f8e8yznm04f6v2ovv7tuk7

3 warnings found (use docker --debug --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
```

E:\year3\_2\softwareEng\Lab8\_3>docker run 74an/lab8  
Paweesuda 653380137-5 Jan

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการให้คำสั่ง

```
$ 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)

```
E:\year3_2\softwareEng\Lab8_3>docker push 74an/lab8
Using default tag: latest
The push refers to repository [docker.io/74an/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:df37c7873c0980ee0b2c99b92a1ee266f29191119cc740ed6b42fbbbc5467c82 size: 527
```

The screenshot shows the Docker Hub profile of a user named Paweesuda Thipayanasa. The profile includes a blue circular avatar with the number 7, the username, and a link to edit the profile. Below the profile information, there are tabs for 'Repositories', 'Starred', and 'Contributed'. The 'Repositories' tab is active, showing a search bar and a list of repositories. One repository, '74an/lab8', is listed with a Docker icon, the username '74an', and the text 'Updated a minute ago'. The repository has 0 stars and 0 forks.

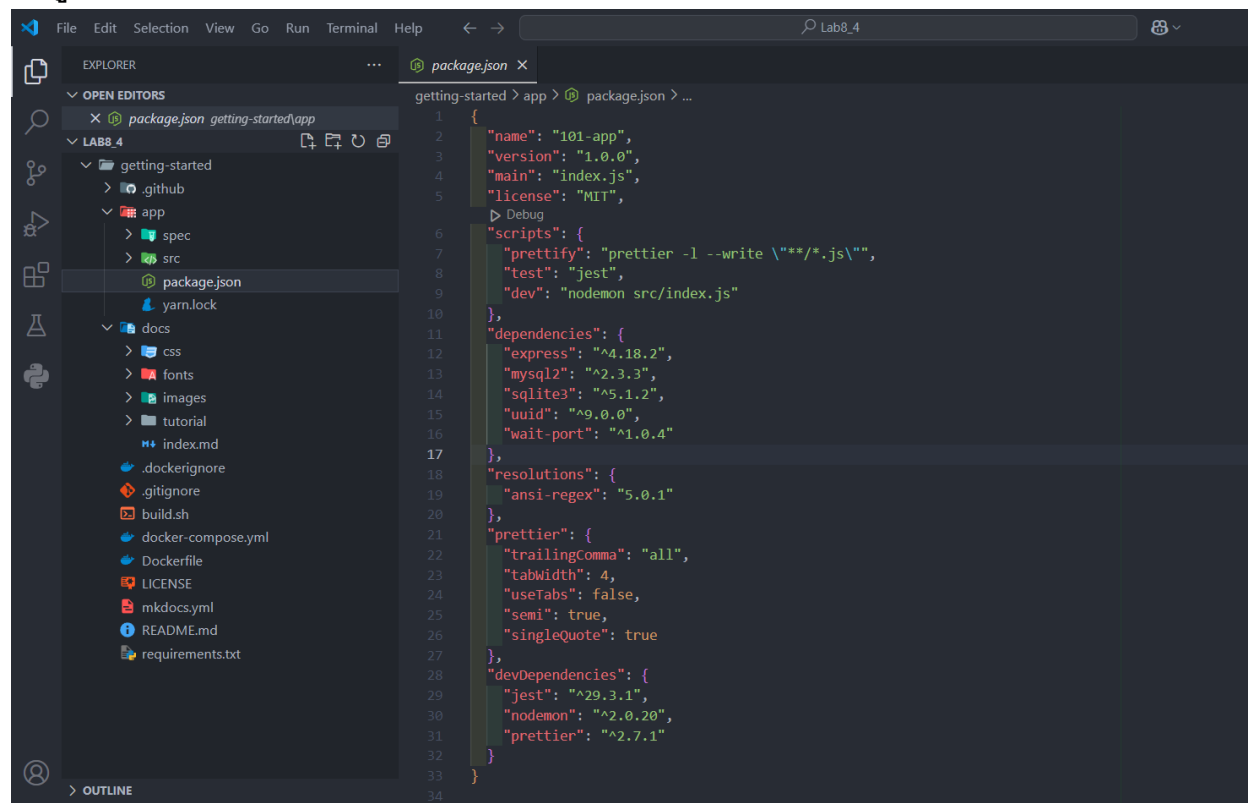
## Lab Worksheet

## 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

ที่อยู่: E:\year3\_2\softwareEng\Lab8\_4\getting-started



```
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 }
```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์  
FROM node:18-alpine



## Lab Worksheet

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดให้ชื่อ image เป็น myapp\_รหัสสนศ. ไม่มีขีด

\$ docker build -t <myapp\_รหัสสนศ. ไม่มีขีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

```
E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker build -t myapp_6533801375 .
[+] Building 20.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 150B
=> [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
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aee8647b1747bcebbbc385e 1.72kB / 1.72kB
=> => sha256:dc5f78337599b66f4d214e4ed804f230ee9e04ac03a59380d572e289b9e5a0 6.13kB / 6.13kB
=> => sha256:1f3e46996e2966e4faa5846e5e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB
=> => sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB
=> => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB
=> => sha256:6504e29600c8d5213b52cda808370abb3d12639802d06b46b6f3e368990ca771 444B / 444B
=> => extracting sha256:1f3e46996e2966e4faa5846e5e76e3748b7315e2ded61476c24403d592134f0
=> => extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1
=> => extracting sha256:6504e29600c8d5213b52cda808370abb3d12639802d06b46b6f3e368990ca771
=> [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 sha256:30ea3b92ca7a001b2928a205d42a5d458878484127339920c42b22f3d7ac27bd
=> => naming to docker.io/library/myapp_6533801375

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ujh3fkffnbt4buq2tfcolyac6

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview

E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801375
9e1426f5ad645ef1b6bfdf07110031a604b56c597304c883cc81260485e40aea
```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

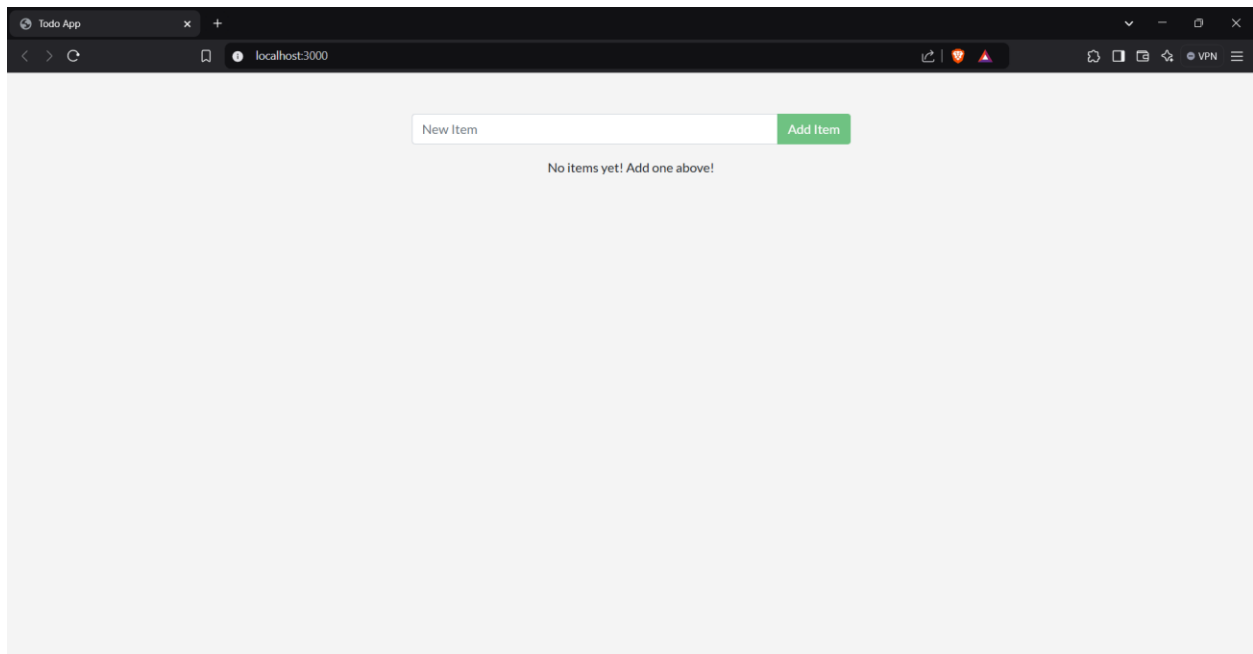
\$ docker run -dp 3000:3000 <myapp\_รหัสสนศ. ไม่มีขีด>

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

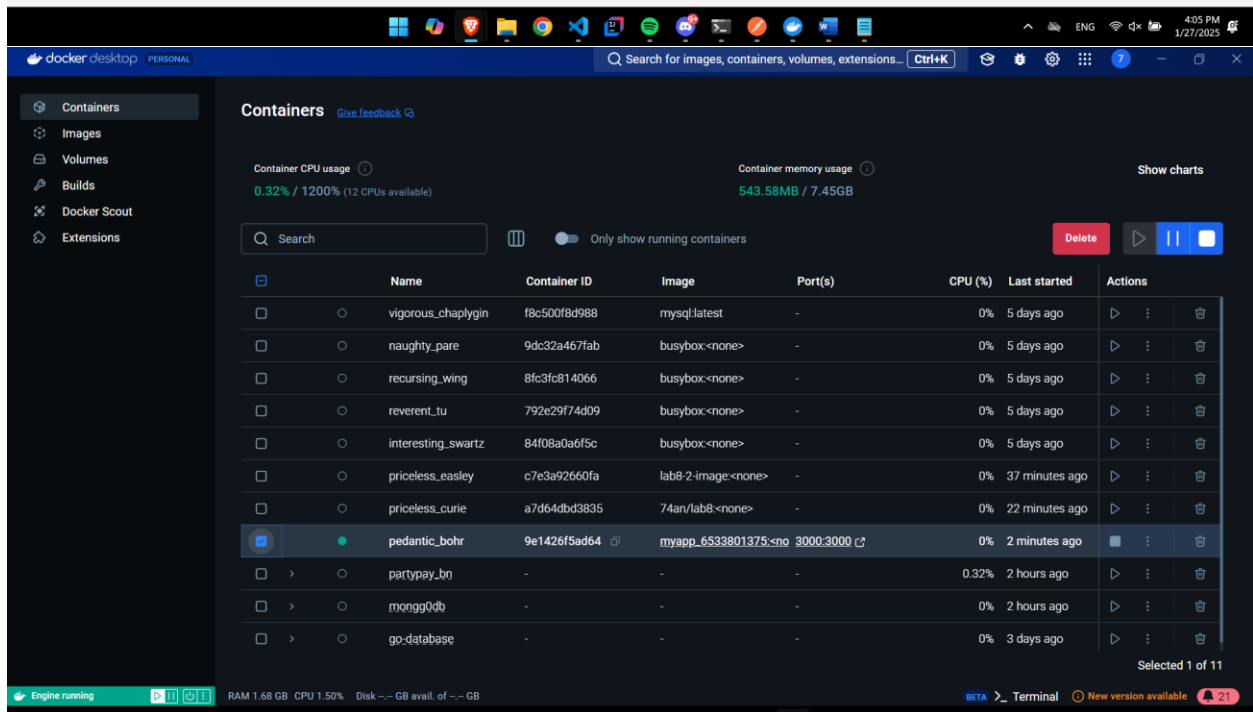
[Check point#9] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้นบน Browser

และ Dashboard ของ Docker desktop

## Lab Worksheet



The screenshot shows a web browser window with the address bar set to 'localhost:3000'. The page displays a simple 'Todo App' interface. At the top, there is a text input field labeled 'New Item' and a green button labeled 'Add Item'. Below the input field, a message reads 'No items yet! Add one above!'.

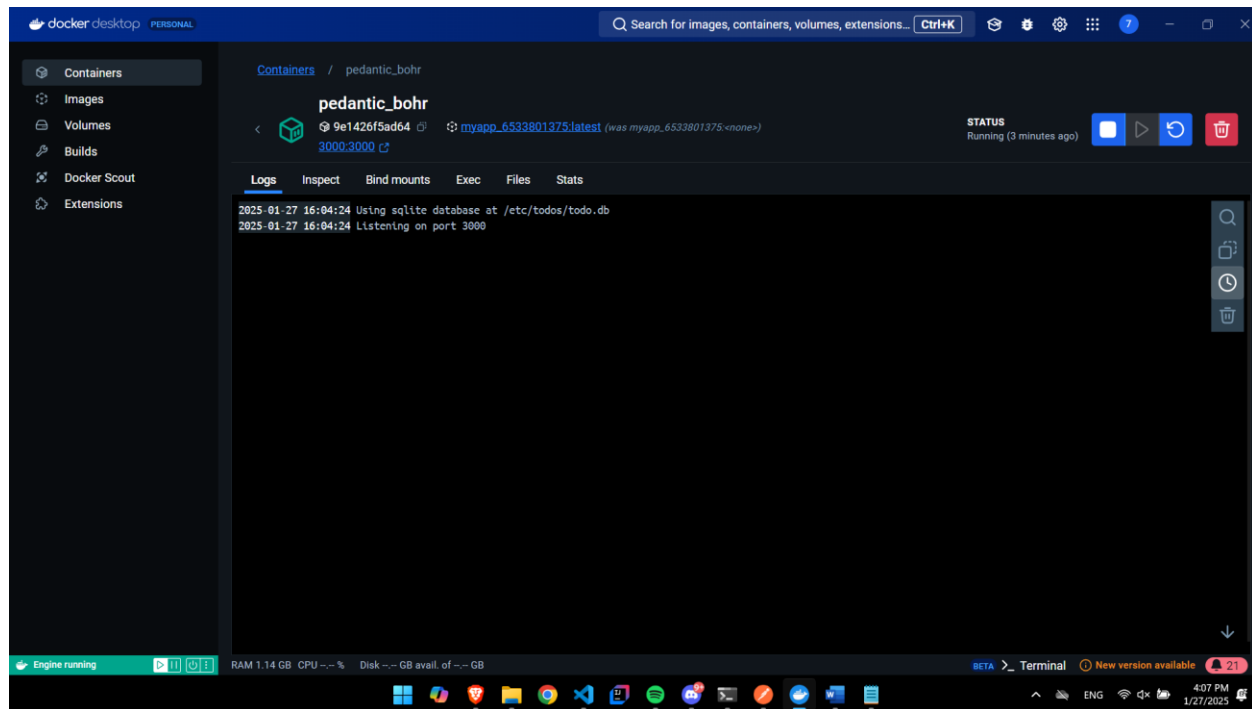
The screenshot shows the Docker Desktop interface. The 'Containers' tab is selected, displaying a list of containers. The 'pedantic\_bohr' container is highlighted. The interface shows the following details:

- Container CPU usage:** 0.32% / 1200% (12 CPUs available)
- Container memory usage:** 543.58MB / 7.45GB
- Search bar:** Search for images, containers, volumes, extensions...
- Filters:** Only show running containers (toggle on)
- Containers Table:**

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
vigorous_chaplygin	f8c50f8d988	mysql:latest	-	0%	5 days ago	[Play] [Stop] [Delete]
naughty_pare	9dc32a467fab	busybox:<none>	-	0%	5 days ago	[Play] [Stop] [Delete]
recurring_wing	8fc3fc814066	busybox:<none>	-	0%	5 days ago	[Play] [Stop] [Delete]
reverent_tu	792e29f74d09	busybox:<none>	-	0%	5 days ago	[Play] [Stop] [Delete]
interesting_swartz	84f08a0a6f5c	busybox:<none>	-	0%	5 days ago	[Play] [Stop] [Delete]
priceless_easley	c7e3a92660fa	lab8-2-image:<none>	-	0%	37 minutes ago	[Play] [Stop] [Delete]
priceless_curie	a7d64dbd3835	74an/lab8:<none>	-	0%	22 minutes ago	[Play] [Stop] [Delete]
pedantic_bohr	9e1426f5ad64	myapp_6533801375:<no>	3000:3000	0%	2 minutes ago	[Play] [Stop] [Delete]
partypay_bn	-	-	-	0.32%	2 hours ago	[Play] [Stop] [Delete]
mongggodb	-	-	-	0%	2 hours ago	[Play] [Stop] [Delete]
go-database	-	-	-	0%	3 days ago	[Play] [Stop] [Delete]

At the bottom, the status bar shows 'Engine running', 'RAM 1.68 GB', 'CPU 1.50%', and 'Disk -- GB avail. of -- GB'. There is also a notification for 'New version available' and a 'Terminal' button.

## 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`</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

## Lab Worksheet

```
E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker build -t myapp_6533801375 .
[+] Building 13.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 150B
=> [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: 8.10kB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> writing image sha256:71f8da7696c1c05d322c8e8e76cd4994ffede1daf46293aca2a930fb08bd0941
=> => naming to docker.io/library/myapp_6533801375

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/qxlbmxy2z3jq5mxjpdv1ilo8x

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview

E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801375
9a27cc03efaa42231c011a8d06271bfe7b0678da088f7c447926db99e69b0f19
docker: Error response from daemon: driver failed programming external connectivity on endpoint sharp_greider (ce2da05d31139a9800a8197ce026ef7e29b06082323136003c565e971f244376): Bind for 0.0.0.0:3000 failed: port is already allocated.
```

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

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

พอร์ต 3000 ที่ต้องการใช้นั้น ถูกใช้ไปแล้ว

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

## Lab Worksheet

- ii. เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
  - iii. ยืนยันโดยการกด Delete forever
12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6
13. เปิด Browser ไปที่ URL = <http://localhost:3000>

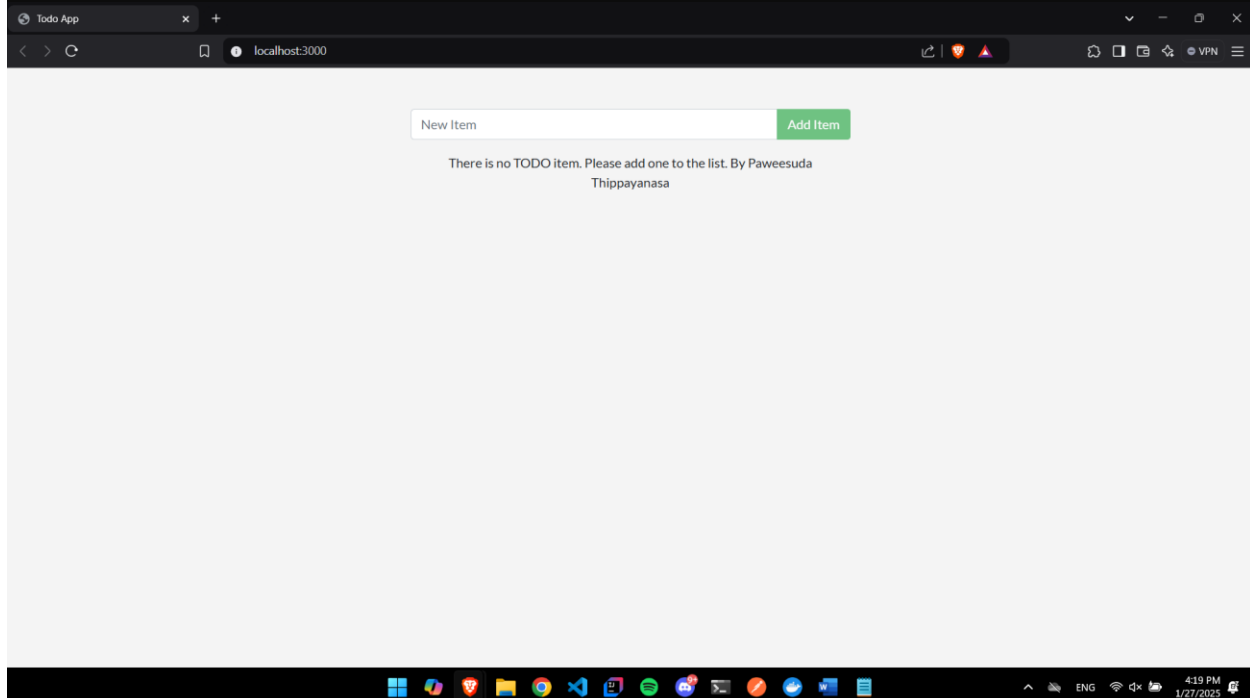
[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

```
E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
9a27cc03efaa   myapp_6533801375  "docker-entrypoint.s..." 5 minutes ago  Up 4 seconds  0.0.0.0:3000->3000/tcp    sharp_greider

E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker stop 9a27cc03efaa
9a27cc03efaa

E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker rm 9a27cc03efaa
9a27cc03efaa

E:\year3_2\softwareEng\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533801375
94b021280767936241e68de8233758b036a1d90d6ba6445079538f4db37ed1e9
```



## 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

```
E:\year3_2\softwareEng\Lab\docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v jenkins_home:/var/jenkins_home jenkins-robotframework
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-01-27 14:59:46.287+0000 [id=1] INFO winstone.Logger$LogInternal: Beginning extraction from war file
2025-01-27 14:59:47.038+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler$setContextPath: Empty contextPath
2025-01-27 14:59:47.094+0000 [id=1] INFO org.eclipse.jetty.server.Server$doStart: jetty-12.0.16; built: 2024-12-09T21:02:54.535Z; git: c3f88bafba393f23204dc14dc57b842e84debc7; jvm 17.0.13+11
2025-01-27 14:59:47.566+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor$visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2025-01-27 14:59:47.625+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager$doStart: Session workerName=node0
2025-01-27 14:59:48.032+0000 [id=1] INFO hudson.WebAppMain$contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2025-01-27 14:59:48.170+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,
hmcje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@6631cb04[STARTED])
2025-01-27 14:59:48.183+0000 [id=1] INFO o.e.j.server.AbstractConnector$doStart: Started ServerConnector@27d0f3d(HTTP/1.1, (http/1.1)){0.0.0.0:8080}
2025-01-27 14:59:48.198+0000 [id=1] INFO org.eclipse.jetty.server.Server$doStart: Started oejs.Server@1c7fd01f[STARTING]{12.0.16,sto=0} @2533ms
2025-01-27 14:59:48.200+0000 [id=20] INFO winstone.Logger$LogInternal: Winstone Servlet Engine running: controlPort=disabled
2025-01-27 14:59:48.362+0000 [id=340] INFO jenkins.InitReactorRunner$1onAttained: Started initialization
2025-01-27 14:59:48.387+0000 [id=855] INFO jenkins.InitReactorRunner$1onAttained: Listed all plugins
2025-01-27 14:59:48.987+0000 [id=40] INFO jenkins.InitReactorRunner$1onAttained: Prepared all plugins
2025-01-27 14:59:48.993+0000 [id=37] INFO jenkins.InitReactorRunner$1onAttained: Started all plugins
2025-01-27 14:59:48.994+0000 [id=53] INFO jenkins.InitReactorRunner$1onAttained: Augmented all extensions
2025-01-27 14:59:49.134+0000 [id=42] INFO jenkins.InitReactorRunner$1onAttained: System config loaded
2025-01-27 14:59:49.135+0000 [id=40] INFO jenkins.InitReactorRunner$1onAttained: System config adapted
2025-01-27 14:59:49.140+0000 [id=52] INFO jenkins.InitReactorRunner$1onAttained: Loaded all jobs
2025-01-27 14:59:49.143+0000 [id=43] INFO jenkins.InitReactorRunner$1onAttained: Configuration for all jobs updated
2025-01-27 14:59:49.171+0000 [id=60] INFO hudson.util.Retrier$start: Attempt #1 to do the action check updates server
2025-01-27 14:59:49.537+0000 [id=43] INFO jenkins.install.SetupWizard$init:

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

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

71923e8c5ea94ccd8833ba37a8bcff72

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

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

## Lab Worksheet

The image displays two sequential screenshots of the Jenkins Setup Wizard interface, accessed via a web browser at localhost:8080.

**Top Screenshot: Create First Admin User**

This screen is titled "Getting Started" and "Create First Admin User". It contains the following fields:

- Username: paweesuda\_1375
- Password: [masked]
- Confirm password: [masked]
- Full name: Paweesuda Thippayanasa
- E-mail address: paweesuda.t@kkumail.com

At the bottom, it indicates "Jenkins 2.479.3" and provides two buttons: "Skip and continue as admin" and "Save and Continue".

**Bottom Screenshot: Instance Configuration**

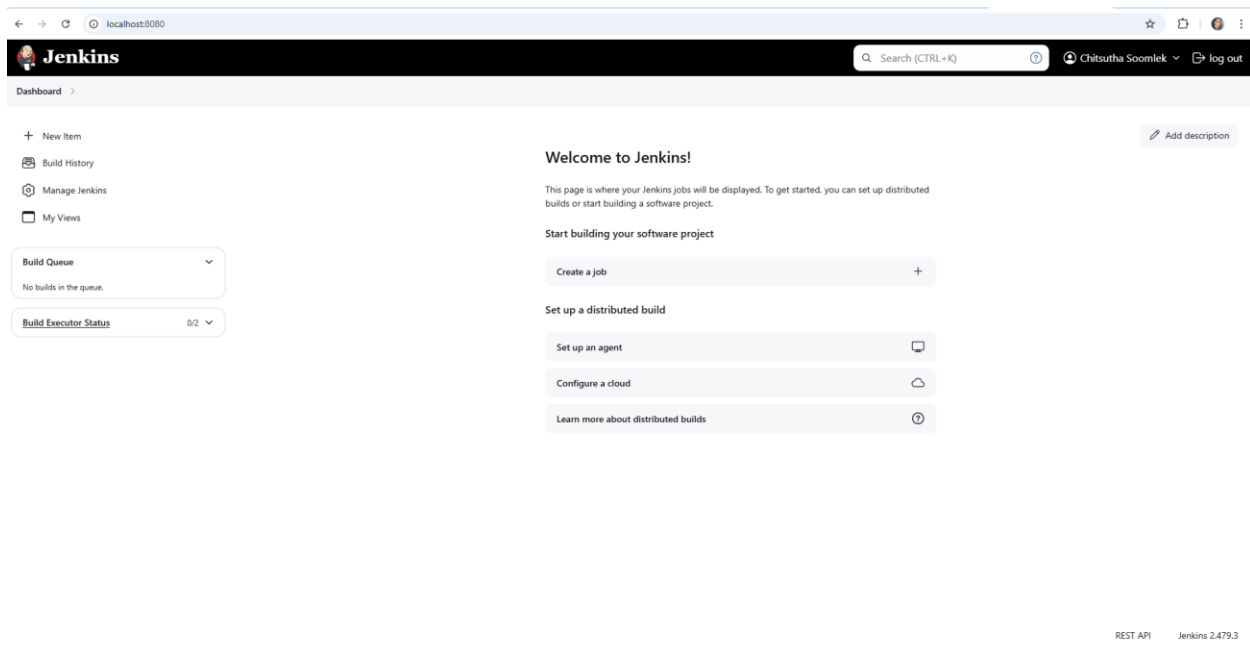
This screen is also titled "Getting Started" and "Instance Configuration". It contains the following field:

- Jenkins URL: http://localhost:8080/lab8

Below the field, there is explanatory text about the Jenkins URL. At the bottom, it indicates "Jenkins 2.479.3" and provides two buttons: "Not now" and "Save and Finish".

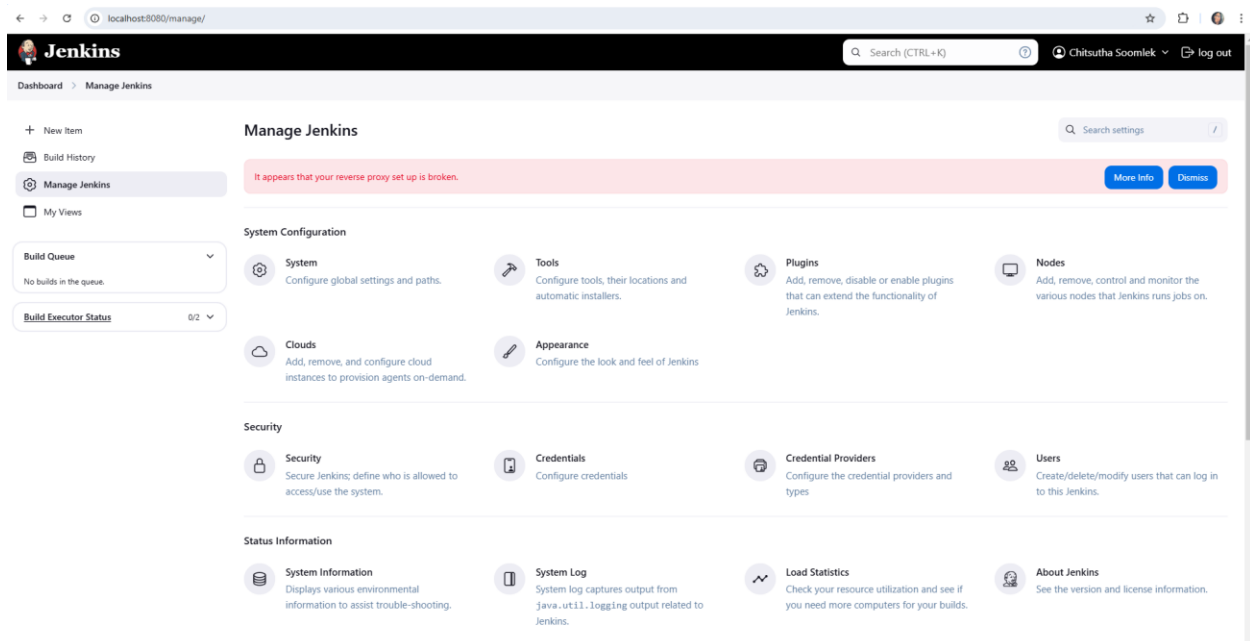
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบหน้าจอ Dashboard ดังแสดงในภาพ

## Lab Worksheet



The screenshot shows the Jenkins Dashboard at localhost:8080. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for Chitsutha Soomlek with a log out button. The left sidebar contains links for New Item, Build History, Manage Jenkins, and My Views. The main content area features a 'Welcome to Jenkins!' message, a 'Start building your software project' section with a 'Create a job' button, and a 'Set up a distributed build' section with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The bottom right corner displays 'REST API' and 'Jenkins 2.479.3'.

## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/manage/. The left sidebar is the same as the dashboard. The main content area has a 'Manage Jenkins' title and a search settings bar. A red notification banner at the top states 'It appears that your reverse proxy set up is broken.' with 'More Info' and 'Dismiss' buttons. Below this, the 'System Configuration' section includes links for System, Tools, Plugins, Nodes, Clouds, and Appearance. The 'Security' section includes links for Security, Credentials, Credential Providers, and Users. The 'Status Information' section includes links for System Information, System Log, Load Statistics, and About Jenkins.

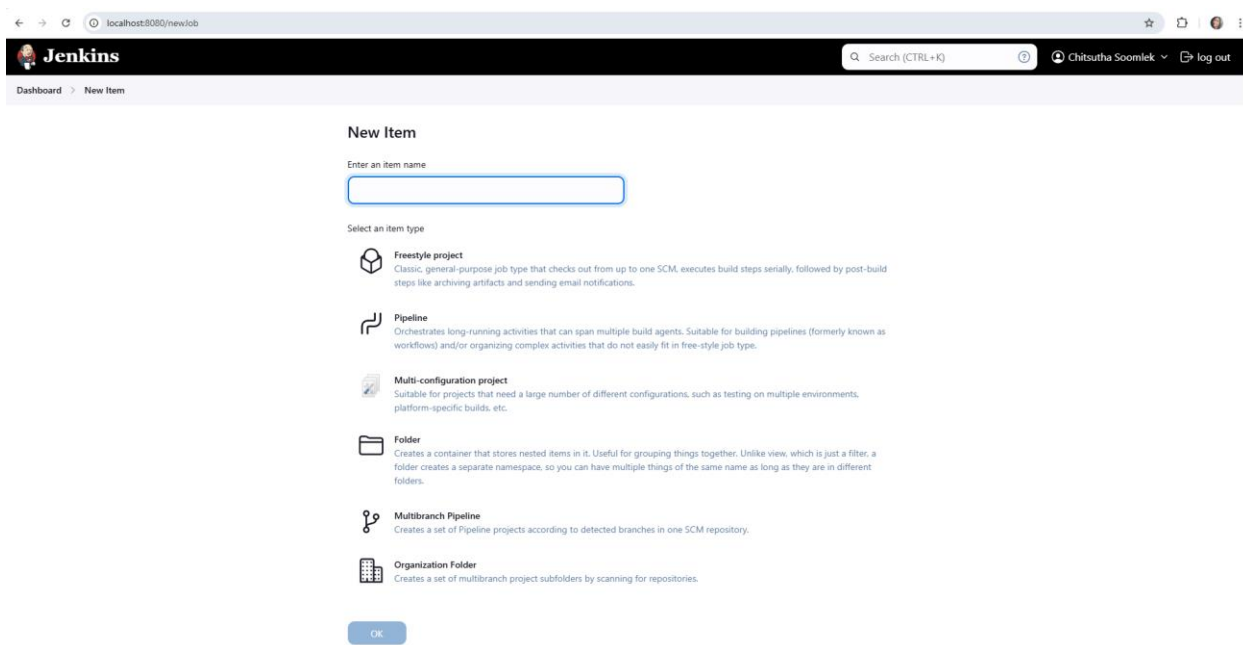


## Lab Worksheet

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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



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

## Lab Worksheet

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows the Jenkins web interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information. Below this is a breadcrumb trail: Dashboard > UAT > Configuration. The main content area is titled 'Configure' and shows the 'General' tab selected. The job name 'Lab 8.5' is displayed. Under the 'Description' field, there's a text area containing 'Lab 8.5'. Below this, there are several checkboxes: 'Discard old builds' (unchecked), 'GitHub project' (checked), 'This project is parameterized' (unchecked), 'Throttle builds' (unchecked), and 'Execute concurrent builds if necessary' (unchecked). The 'Project url' field is filled with 'https://github.com/Paweesuda-Thippayanasa/lab7\_TestAutomation\_No2.git/'. At the bottom, there are 'Save' and 'Apply' buttons.

Jenkins

Dashboard > UAT > Configuration

Configure

General

Enabled

Description

Lab 8.5

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

https://github.com/Paweesuda-Thippayanasa/lab7\_TestAutomation\_No2.git/

Advanced ▾

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced ▾

Source Code Management

Save Apply

## Lab Worksheet

**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 Monday, January 27, 2025 at 7:05:18 PM Coordinated Universal Time; would next run at Monday, January 27, 2025 at 7:20:18 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

---

**Build Environment**

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

**Save** **Apply**

---

UAT Config [Jenkins]

localhost:8080/job/UAT/configure

Dashboard > UAT > Configuration

**Configure**

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

**Build Steps**

**Execute shell** ?

Command

See the list of available environment variables

robot no7\_2/UATLab7\_1.robot

Advanced ▾

Add build step ▾

---

**Post-build Actions**

**Publish Robot Framework test results** ?

Directory of Robot output

Path to directory containing robot.xml and html files (relative to build workspace)

\*\*/output.xml

Advanced ▾ Edited

Thresholds for build result ?

**Save** **Apply**

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

\_\_\_\_\_robot no7\_2/UATLab7\_1.robot\_\_\_\_\_

## Lab Worksheet

Post-build action: เพิ่ม Publish Robot Framework test results ->

ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

13. กด Apply และ Save

14. สั่ง Build Now

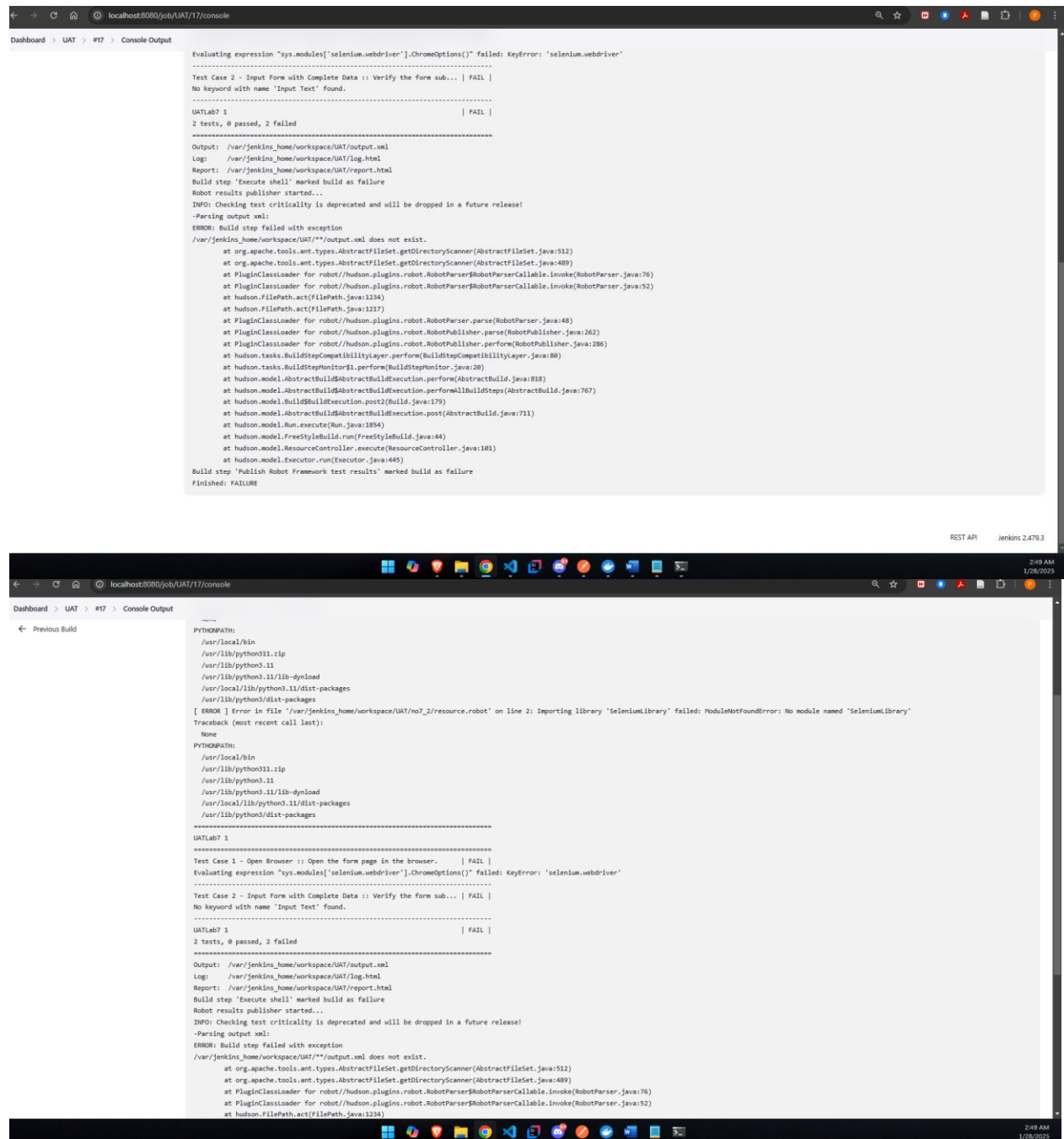
[Check point#15] Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output

```

Started by timer
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/sh -xe /tmp/jenkins3250794007314774519.sh
+ robot no7_2/UATLab7_1.robot
[ ERROR ] Error in file '/var/jenkins_home/workspace/UAT/no7_2/UATLab7_1.robot' on line 2: Importing library 'SeleniumLibrary' failed: ModuleNotFoundError: No module named 'SeleniumLibrary'
Traceback (most recent call last):
  None
PYTHONPATH:
  /usr/local/bin
  /usr/lib/python3.11.zip
  /usr/lib/python3.11
  /usr/lib/python3.11/lib-dynload
  /usr/local/lib/python3.11/dist-packages
  /usr/lib/python3/dist-packages
[ ERROR ] Error in file '/var/jenkins_home/workspace/UAT/no7_2/resource.robot' on line 2: Importing library 'SeleniumLibrary' failed: ModuleNotFoundError: No module named 'SeleniumLibrary'
Traceback (most recent call last):
  None
PYTHONPATH:
  /usr/local/bin
  /usr/lib/python3.11.zip
  /usr/lib/python3.11
  /usr/lib/python3.11/lib-dynload
  /usr/local/lib/python3.11/dist-packages
  /usr/lib/python3/dist-packages
=====
UATLab7 1
=====
Test Case 1 - Open Browser :: Open the form page in the browser. | FAIL |
Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: KeyError: 'selenium.webdriver'
=====
Test Case 2 - Input Form with Complete Data :: Verify the form sub... | FAIL |
No keyword with name 'Input Text' found.
=====
UATLab7 1 | FAIL |
2 tests. 0 passed. 2 failed

```

## Lab Worksheet



Dashboard > UAT > #17 > Console Output

```

Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: KeyError: 'selenium.webdriver'

Test Case 2 - Input Form with Complete Data :: Verify the form sub... | FAIL |
No keyword with name 'Input Text' found.

-----
UATLab7 1 | FAIL |
2 tests, 0 passed, 2 failed
-----

Output: /var/jenkins_home/workspace/UAT/output.xml
Log: /var/jenkins_home/workspace/UAT/log.html
Report: /var/jenkins_home/workspace/UAT/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:
ERROR: Build step failed with exception
/var/jenkins_home/workspace/UAT/**/output.xml does not exist.
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:512)
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:489)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:76)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:52)
at hudson.FilePath.act(FilePath.java:1234)
at hudson.FilePath.act(FilePath.java:1217)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser.parse(RobotParser.java:48)
at PluginClassLoader for robot/hudson.plugins.robot.RobotPublisher.parse(RobotPublisher.java:262)
at PluginClassLoader for robot/hudson.plugins.robot.RobotPublisher.perform(RobotPublisher.java:286)
at hudson.tasks.BuildStepCompatibilityLayer.perform(BuildStepCompatibilityLayer.java:88)
at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:20)
at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
at hudson.model.AbstractBuild$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:767)
at hudson.model.AbstractBuild$AbstractBuildExecution.postBuild(AbstractBuild.java:179)
at hudson.model.AbstractBuild$AbstractBuildExecution.post(AbstractBuild.java:711)
at hudson.model.Run.execute(Run.java:1854)
at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:44)
at hudson.model.ResourceController.execute(ResourceController.java:181)
at hudson.model.Executor.run(Executor.java:445)
Build step 'Publish Robot Framework test results' marked build as failure
Finished: FAILURE

```

REST API Jenkins 2.479.3

Dashboard > UAT > #17 > Console Output

Previous Build

```

PYTHONPATH:
/usr/local/bin
/usr/lib/python3.11.zip
/usr/lib/python3.11
/usr/lib/python3.11/lib-dynload
/usr/local/lib/python3.11/dist-packages
/usr/lib/python3.11/dist-packages
[ ERROR ] Error in file "/var/jenkins_home/workspace/UAT/no?_2/resource.robot" on line 2: Importing library 'SeleniumLibrary' failed: ModuleNotFoundError: No module named 'SeleniumLibrary'
Traceback (most recent call last):
None
PYTHONPATH:
/usr/local/bin
/usr/lib/python3.11.zip
/usr/lib/python3.11
/usr/lib/python3.11/lib-dynload
/usr/local/lib/python3.11/dist-packages
/usr/lib/python3.11/dist-packages

UATLab7 1
-----
Test Case 1 - Open Browser :: Open the form page in the browser... | FAIL |
Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: KeyError: 'selenium.webdriver'

Test Case 2 - Input Form with Complete Data :: Verify the form sub... | FAIL |
No keyword with name 'Input Text' found.

-----
UATLab7 1 | FAIL |
2 tests, 0 passed, 2 failed
-----

Output: /var/jenkins_home/workspace/UAT/output.xml
Log: /var/jenkins_home/workspace/UAT/log.html
Report: /var/jenkins_home/workspace/UAT/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:
ERROR: Build step failed with exception
/var/jenkins_home/workspace/UAT/**/output.xml does not exist.
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:512)
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:489)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:76)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:52)
at hudson.FilePath.act(FilePath.java:1234)

```

2:49 AM 1/28/2025