

Lab Worksheet

ชื่อ-นามสกุล _____ ศิววิทย์ ชั่มนาเสีย _____ รหัสนักศึกษา _____ 663380402-3 _____ 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

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\JJay> mkdir Lab8_1

Directory: C:\Users\JJay

Mode                LastWriteTime         Length Name
----                -----          ---- -  
d----       2/4/2026   6:28 PM           0 Lab8_1
latest: Pulling from library/busybox
61dfb50712f5: Pull complete

Terminal

Mode                LastWriteTime         Length Name
----                -----          ---- -  
d----       2/4/2026   6:28 PM           0 Lab8_1
latest: Pulling from library/busybox
61dfb50712f5: Pull complete
96cfb76e59bd: Download complete
Digest: sha256:b3255e7dfbcd10cb367af0d409747d511aeb66dfac98cf30e97e87e4207dd76f
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
PS C:\Users\JJay\Lab8_1> docker images
 $\text{IMAGE} \quad \text{ID} \quad \text{DISK USAGE} \quad \text{CONTENT SIZE} \quad \text{EXTRA}$ 
busybox:latest  b3255e7dfbcd  6.77MB    2.22MB
PS C:\Users\JJay\Lab8_1>
```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร _____ คือ busybox _____

(2) Tag ที่ใช้บ่งบอกถึงอะไร _____ คือ tag latest _____

5. ป้อนคำสั่ง \$ docker run busybox
6. ป้อนคำสั่ง \$ docker run -it busybox sh
7. ป้อนคำสั่ง ls
8. ป้อนคำสั่ง ls -la
9. ป้อนคำสั่ง exit

Lab Worksheet

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

```
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\JJay> mkdir Lab8_1

Directory: C:\Users\JJay

Mode                LastWriteTime         Length Name
----                -              -          -
d-----       2/4/2026   6:28 PM           0 Lab8_1
latest: Pulling from library/busybox
61dfb50712f5: Pull complete
96cfb76e59bd: Download complete
Digest: sha256:b3255e7dfbcd10cb367af0d409747d511aeb66dfac98cf30e97e87e4207dd76f
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
PS C:\Users\JJay\Lab8_1> docker images

drwxr-xr-x    1 root      root        4096 Feb  4 11:34 ..
-rw-r--r--    1 root      root          0 Feb  4 11:34 .dockercfg
drwxr-xr-x    2 root      root     12288 Sep 26 2024 bin
drwxr-xr-x    5 root      root        360 Feb  4 11:34 dev
```

Lab Worksheet

Terminal

```

drwxr-xr-x  1 root    root        4096 Feb  4 11:34 ..
-rw xr-xr-x  1 root    root        0 Feb  4 11:34 .dockerenv
drwxr-xr-x  2 root    root      12288 Sep 26 2024 bin
drwxr-xr-x  5 root    root       360 Feb  4 11:34 dev
drwxr-xr-x  1 root    root       4096 Feb  4 11:34 etc
drwxr-xr-x  2 nobody  nobody     4096 Sep 26 2024 home
drwxr-xr-x  2 root    root       4096 Sep 26 2024 lib
lrwxrwxrwx  1 root    root        3 Sep 26 2024 lib64 -> lib
dr-xr-xr-x  319 root   root       0 Feb  4 11:34 proc
drwx-----  1 root    root       4096 Feb  4 11:34 root
dr-xr-xr-x  13 root   root       0 Feb  4 11:34 sys
drwxrwxrwt  2 root    root       4096 Sep 26 2024 tmp
drwxr-xr-x  4 root    root       4096 Sep 26 2024 usr
drwxr-xr-x  4 root    root       4096 Sep 26 2024 var
/ # exit          > docker run busybox echo "Hello ตัววิธี ชื่อภาษาไทย from busybox"
Hello ตัววิธี ชื่อภาษาไทย from busybox
PS C:\Users\JJay\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED          STATUS
              PORTS NAMES
dd2652c5b59b   busybox    "echo 'Hello ตัววิธี'"  10 seconds ago   Exited (0) 9 seconds ago
                  zen_raman
7d92a1c599ff   busybox    "sh"
go              keen_davinci
a9312f41dea8   busybox    "sh"
                  affectionate_bardeen
PS C:\Users\JJay\Lab8_1> []

```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสั้งๆ ง่ายๆ

ทำให้เราสามารถเข้าไปพิมพ์คำสั่งงาน ภายใน Container ได้เหมือนใช้คอมอีกเครื่อง _____

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a และงดลิงข้อมูลอะไร

สถานะปัจจุบันของ Container เช่น Up (กำลังทำงานอยู่) หรือ Exited (ทำงานเสร็จแล้วและปิดตัวลงแล้ว) _____

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

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

Lab Worksheet

```
PS C:\Users\JJay\Lab8_1> docker rm dd2652c5b59b
dd2652c5b59b
PS C:\Users\JJay\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED          STATUS          PORTS     NAMES
MES
7d92a1c599ff   busybox    "sh"      11 minutes ago  Exited (0) 10 minutes ago
en_davinci
a9312f41dea8   busybox    "sh"      11 minutes ago  Exited (0) 11 minutes ago
fectionate_bardeen
PS C:\Users\JJay\Lab8_1>
```

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
สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ส (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

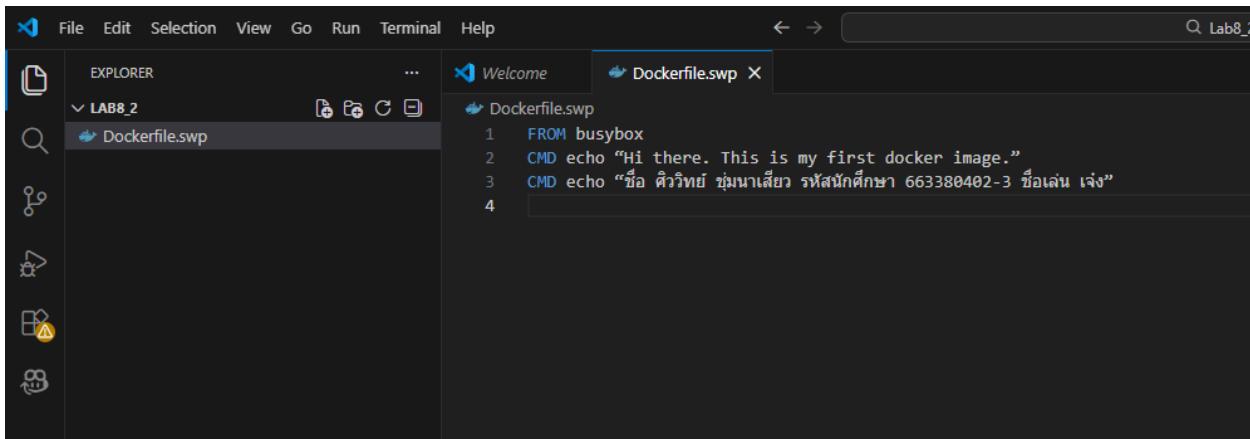
5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

Lab Worksheet

\$ docker build -t <ชื่อ Image> .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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



```

File Edit Selection View Go Run Terminal Help
Q Lab8_2

EXPLORER ... Dockerfile.swp X
LAB8_2 Dockerfile.swp
1 FROM busybox
2 CMD echo "Hi there. This is my first docker image."
3 CMD echo "ชื่อ คิววิทย์ ภูมนาเสียง รหัสนักศึกษา 663380402-3 ชื่อเล่น เจ่ง"
4

PS C:\Users\JJay\Lab8_1> cd ..
PS C:\Users\JJay> mkdir Lab8_2

Directory: C:\Users\JJay

Mode                LastWriteTime         Length Name
----                -              -          -
d---        2/4/2026   6:50 PM            Lab8_2

PS C:\Users\JJay> cd lab8_2

```

Lab Worksheet

```
PS C:\Users\JJay\lab8_2> docker build -t busybox .
[+] Building 0.8s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 285B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/busybox:latest@sha256:b3255e7dfbcd10cb367af0d409747d 0.2s
=> => resolve docker.io/library/busybox:latest@sha256:b3255e7dfbcd10cb367af0d409747d 0.1s
=> exporting to image 0.2s
=> => exporting layers 0.0s
=> => exporting manifest sha256:7014aca105218568515ffbbba47fe3c5ec2ff8ab5a5c59174973 0.0s
=> => exporting config sha256:bccc2143abca80f5562bad8800031a4deb94bf885d6f98a161c3e4 0.0s
=> => exporting attestation manifest sha256:2e311b5004b0972e60429ff79ea4fb035dba7d62 0.1s
=> => exporting manifest list sha256:40d52de0bca7977ccb0d5b3c0040af84784e4ff3d923b0c 0.0s
=> => naming to docker.io/library/busybox:latest 0.0s
=> => unpacking to docker.io/library/busybox:latest 0.0s

=> => exporting manifest sha256:ec214ba066cac4e5d0cf15974c5062128237ead6e7b362ee7 0.0s
=> => exporting config sha256:a60a0f2403007c68b8bfacfd4509cc759244ca75eaa133e1811 0.0s
=> => exporting attestation manifest sha256:ee11010e1cb3998f0204c37ee9793053cc27d 0.1s
=> => exporting manifest list sha256:9258792af83ad28173e4e9bf999f59bdf4fab2c999f7 0.0s
=> => naming to docker.io/library/busybox:latest 0.0s
=> => unpacking to docker.io/library/busybox:latest 0.0s

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

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/uudilqlu_uha8enwcdgufh06ar
PS C:\Users\JJay\lab8_2> docker run busybox
Hi there. This is my first docker image.
接到 คิวไวท์ ภูมานา เส็ง รหัสลับคือ 663380402-3 ข้อมูลนี้จะถูกลบใน 7 วัน
PS C:\Users\JJay\lab8_2>
```

(1) คำสั่งที่ใช้ในการ run คือ

docker run busybox

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

ใช้สำหรับตั้งชื่อ (Tag) ให้กับ Docker Image ที่เราสร้างขึ้น เพื่อให้สามารถอ้างอิงและเรียกใช้งาน Image นั้นได้ง่ายในภายหลัง เช่น ใช้กับคำสั่ง docker run หากไม่กำหนด -t ระบบจะสร้าง Image

Lab Worksheet

โดยใช้ค่า Image ID ซึ่งเป็นตัวอักษรยาวและยากต่อการจดจำ _____

แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เก่าไว้
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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการwinโดว์ส์ 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

```
PS C:\Users\JJay\lab8_2> cd ..
```

Lab Worksheet

```
PS C:\Users\JJay> mkdir Lab8_3
PS C:\Users\JJay> cd Lab8_3
PS C:\Users\JJay\Lab8_3> docker build -t busybox .
[+] Building 0.6s (5/5) FINISHED
      docker :desktop-linux
      => [internal] load build definition from Dockerfile          0.0s
      => => transferring dockerfile: 249B                         0.0s
      => WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior 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@sha256:ac9565943a196c89be9c         0.0s
      => => resolve docker.io/library/busybox:latest@sha256:ac9565943a196c89be9c91bcab4          0.0s
      => exporting to image                                         0.2s
      => => exporting layers                                       0.0s
      => => exporting manifest sha256:f24e3e5d017789fb36bf3f76279bfae4d23f503dd2591c66b 0.0s
      => => exporting config sha256:08b0bd7b30a01a3b392d194b1703e47ece7444637a14b26af00 0.0s
      => => exporting attestation manifest sha256:365f5a668c2fc5db0611997258672b1c4d764 0.2s
      => => exporting manifest list sha256:ac9565943a196c89be9c91bcab46bc801d30be8bb7e6 0.0s
      => => naming to docker.io/library/busybox:latest           0.0s
      => => unpacking to docker.io/library/busybox:latest        0.0s

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

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/wxsahrh1w4a7xgfkfszj08wp7
PS C:\Users\JJay\Lab8_3> docker run busybox
Hi there. My work is done. You can run them from my Docker image.
คิววิทย์ ชุมพาเสือ รหัสนักศึกษา 663380402-3
PS C:\Users\JJay\Lab8_3>
```

6. ทำการ Push ตัว Docker image ไปเว็บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

\$ docker login และป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้คำสั่ง

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

Lab Worksheet

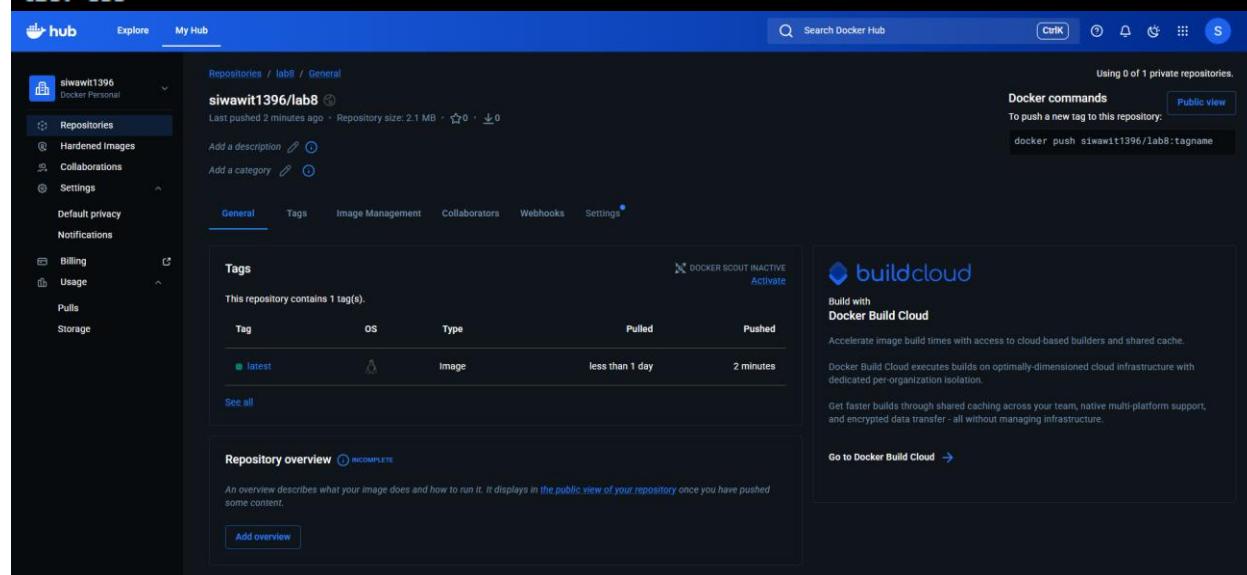
7. ไปที่ Docker Hub กด Tab ชื่อ Tags หรือไปที่ Repository ก็ได้

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

```
PS C:\Users\JJay\Lab8_3> docker login
Authenticating with existing credentials... [Username: siwawit1396]

! Info →To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
PS C:\Users\JJay\Lab8_3> docker tag busybox siwawit1396/lab8
PS C:\Users\JJay\Lab8_3> docker push siwawit1396/lab8
Using default tag: latest
The push refers to repository [docker.io/siwawit1396/lab8]
61dfb50712f5: Mounted from library/busybox
ae2c998fa1f1: Pushed
latest: digest: sha256:6850b9a2dd4ee469f3725d2a4b4d68af4a7faa3d599edeb5333f3a4a42af3ea size: 855


```

แบบฝึกปฏิบัติที่ 8.4: การ Build และ Update Container image และการ Update Container image

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

Lab Worksheet

```
$ git clone https://github.com/docker/getting-started.git
```

3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพับไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

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

```
PS C:\Users\JJay\Lab8_3> cd ..
PS C:\Users\JJay> mkdir Lab8_4

Directory: C:\Users\JJay

Mode                LastWriteTime         Length Name
----                -              -          -
d-----        2/4/2026   8:28 PM           Lab8_4

PS C:\Users\JJay> cd Lab8_4
PS C:\Users\JJay\Lab8_4> git clone https://github.com/Siwa-dev/Jenkins-Robot-Lab8.git
Cloning into 'Jenkins-Robot-Lab8'...
warning: You appear to have cloned an empty repository.
PS C:\Users\JJay\Lab8_4> git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 987, done.
remote: Counting objects: 100% (274/274), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 987 (delta 247), reused 230 (delta 230), pack-reused 713 (from 1)
Receiving objects: 100% (987/987), 5.28 MiB | 5.93 MiB/s, done.
Resolving deltas: 100% (539/539), done.
PS C:\Users\JJay\Lab8_4> ls

Directory: C:\Users\JJay\Lab8_4

Mode                LastWriteTime         Length Name
----                -              -          -
d-----        2/4/2026   8:30 PM           getting-started
d-----        2/4/2026   8:28 PM           Jenkins-Robot-Lab8

PS C:\Users\JJay\Lab8_4> cd ^C
PS C:\Users\JJay\Lab8_4> cd getting-started
```

Lab Worksheet

```
PS C:\Users\JJay\Lab8_4\getting-started> ls
```

```
Directory: C:\Users\JJay\Lab8_4\getting-started
```

Mode	LastWriteTime	Length	Name
-	-	-	-
d----	2/4/2026 8:30 PM		.github
d----	2/4/2026 8:30 PM		app
d----	2/4/2026 8:30 PM		docs
-a---	2/4/2026 8:30 PM	28	.dockerignore
-a---	2/4/2026 8:30 PM	28	.gitignore
-a---	2/4/2026 8:30 PM	267	build.sh
-a---	2/4/2026 8:30 PM	179	docker-compose.yml
-a---	2/4/2026 8:30 PM	1223	Dockerfile
-a---	2/4/2026 8:30 PM	11556	LICENSE
-a---	2/4/2026 8:30 PM	2076	mkdocs.yml
-a---	2/4/2026 8:30 PM	1739	README.md
-a---	2/4/2026 8:30 PM	110	requirements.txt

```
PS C:\Users\JJay\Lab8_4\getting-started> cd app
```

```
PS C:\Users\JJay\Lab8_4\getting-started\app> ls
```

```
Directory: C:\Users\JJay\Lab8_4\getting-started\app
```

Mode	LastWriteTime	Length	Name
-	-	-	-
d----	2/4/2026 8:30 PM		spec
d----	2/4/2026 8:30 PM		src
-a---	2/4/2026 8:30 PM	678	package.json
-a---	2/4/2026 8:30 PM	150541	yarn.lock

```
PS C:\Users\JJay\Lab8_4\getting-started\app> pwd
```

```
Path
```

```
----
```

```
C:\Users\JJay\Lab8_4\getting-started\app
```

Lab Worksheet

```
PS C:\Users\JJay\Lab8_4\getting-started\app> type package.json
{
  "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"
  }
}
PS C:\Users\JJay\Lab8_4\getting-started\app> []
```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไว้ในไฟล์

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

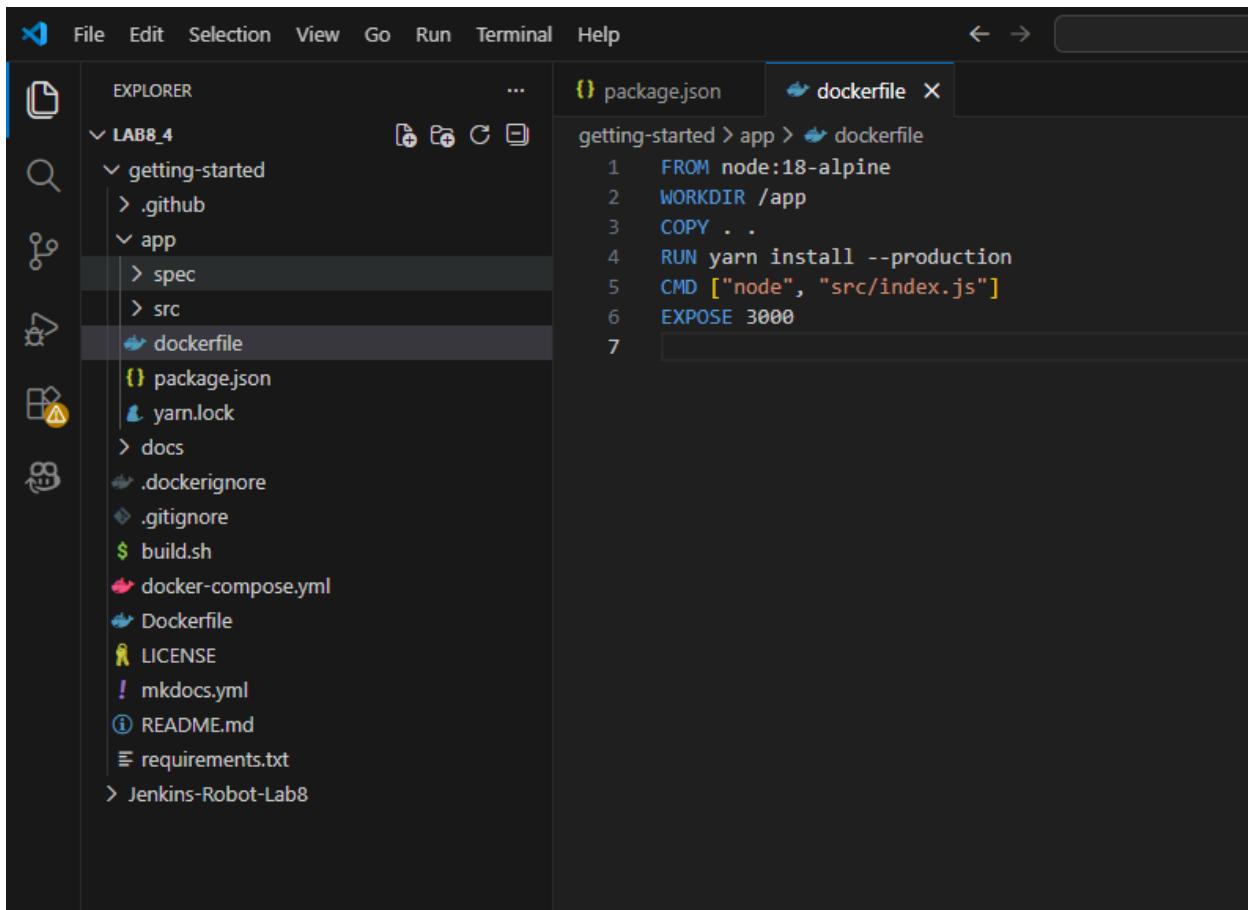
Lab Worksheet

EXPOSE 3000

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

```
$ docker build -t <myapp_รหัสศ. 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

```
PS C:\Users\JJay\Lab8_4\getting-started\app> docker build -t myapp_6633804023 .
[+] Building 29.1s (10/10) FINISHED
          docker:desktop-linux
=> [internal] load build definition from dockerfile                      0.1s
=> => transferring dockerfile: 156B                                     0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine        3.8s
=> [auth] library/node:pull token for registry-1.docker.io               0.0s
=> [internal] load .dockerignore                                         0.1s
=> => transferring context: 2B                                         0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332 9.8s
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332 0.0s
=> => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47 1.26MB / 1.26MB 1.6s
=> => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df 446B / 446B 1.6s
=> => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4ef 40.01MB / 40.01MB 8.6s
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e9 3.64MB / 3.64MB 2.6s
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c 0.1s
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2 0.6s
=> => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26 0.1s
=> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5 0.0s
=> [internal] load build context                                         0.7s
=> => transferring context: 4.62MB                                    0.5s
=> [2/4] WORKDIR /app                                              0.1s
=> [3/4] COPY . .                                                 0.1s
=> [4/4] RUN yarn install --production                           11.0s
=> exporting to image                                               3.9s

=> => exporting manifest sha256:71cca04546a06cf516982aa303705eb5610bf7037754ab4e4 0.0s
=> => exporting config sha256:b14288ad4c1146cb65971b86142d524370d94127acb524dba69 0.0s
=> => exporting attestation manifest sha256:41f02f5a79c05891d2f9ae1bb5401eb7c5500 0.1s
=> => exporting manifest list sha256:3c643f769a406b8c0d9076ea5e35bd8a04b074025c06 0.0s
=> => naming to docker.io/library/myapp_6633804023:latest           0.0s
=> => unpacking to docker.io/library/myapp_6633804023:latest          1.3s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/va80x43w3i68fsklz6aebv9cn
```

- ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

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

Lab Worksheet

PS C:\Users\JJay\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6633804023
b73e93fc95a948cfc6b7501e34a6673553edb55efec0a5b58b28e0d102b847e3
PS C:\Users\JJay\Lab8_4\getting-started\app>

The browser window shows a "Todo App" interface with a header bar containing "New Item" and "Add Item" buttons. Below the header, a message says "No items yet! Add one above!". There is a large empty area for listing items.

หมายเหตุ: นศ.สามารถทดสอบเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้

a. เปิดไฟล์ src/static/js/app.js ด้วย Editor และแก้ไขบรรทัดที่ 56 จาก

<p className="text-center">No items yet! Add one above!</p> เป็น

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

By ชื่อและนามสกุลของนักศึกษา

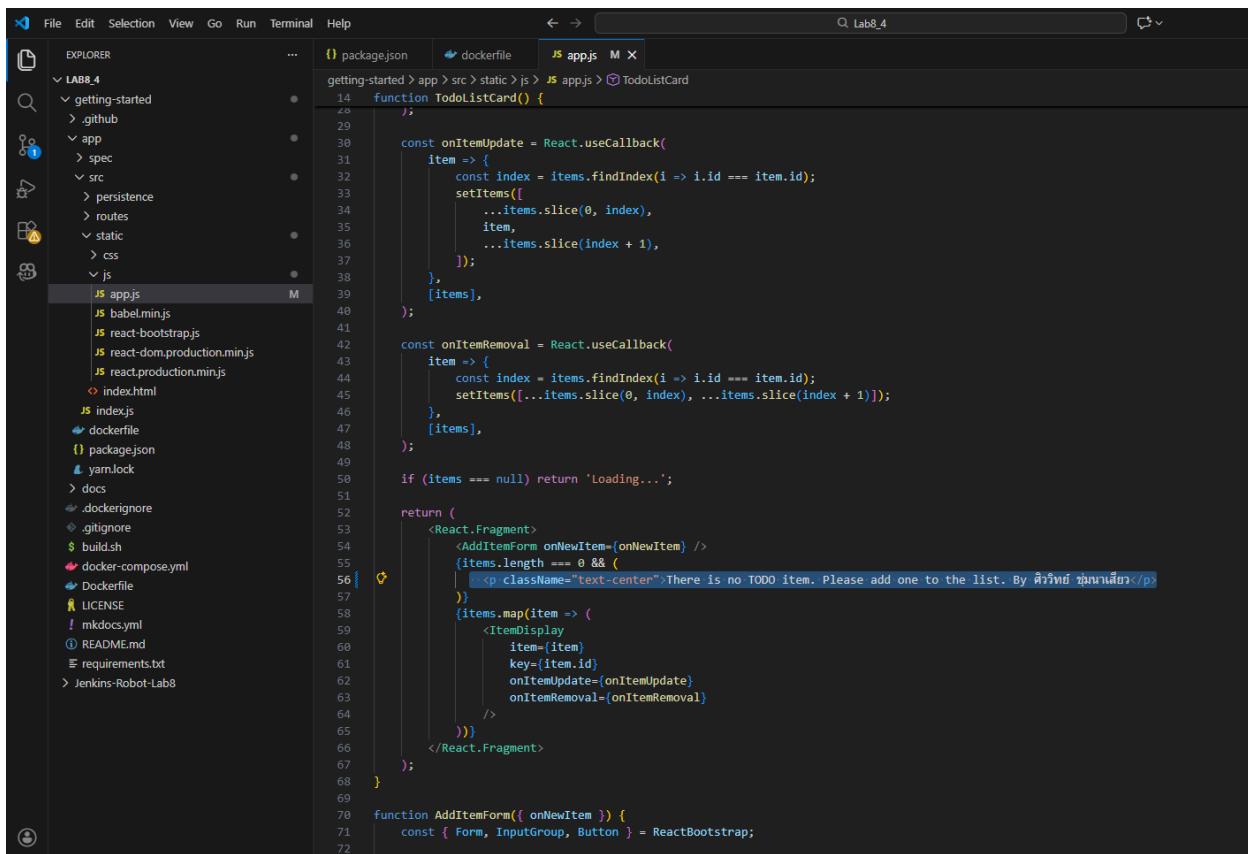
b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

Lab Worksheet



```

File Edit Selection View Go Run Terminal Help Q Lab8_4
EXPLORER package.json dockerfile app.js M X
LAB8_4
getting-started
.github
app
spec
src
persistence
routes
static
css
js
app.js M
babel.min.js
react-bootstrap.js
react-dom.production.min.js
react.production.min.js
index.html
index.js
dockerfile
package.json
yarn.lock
docs
.dockerignore
.gitignore
build.sh
docker-compose.yml
Dockerfile
LICENSE
mkdocs.yml
README.md
requirements.txt
Jenkins-Robot-Lab8
10  function TodoListCard() {
11    ...
12  }
13
14  const onItemUpdate = React.useCallback(
15    item => {
16      const index = items.findIndex(i => i.id === item.id);
17      setItems([
18        ...items.slice(0, index),
19        item,
20        ...items.slice(index + 1),
21      ]);
22    },
23    [items],
24  );
25
26  const onItemRemoval = React.useCallback(
27    item => {
28      const index = items.findIndex(i => i.id === item.id);
29      setItems([...items.slice(0, index), ...items.slice(index + 1)]);
30    },
31    [items],
32  );
33
34  if (items === null) return 'Loading...';
35
36  return (
37    <React.Fragment>
38      <AddItemForm onNewItem={onNewItem} />
39      {items.length === 0 && (
40        <p className="text-center">There is no TODO item. Please add one to the list. By หัวใจ คุณนี้</p>
41      )}
42      {items.map(item => (
43        <ItemDisplay
44          item={item}
45          key={item.id}
46          onItemUpdate={onItemUpdate}
47          onItemRemoval={onItemRemoval}
48        />
49      ))}
50    </React.Fragment>
51  );
52
53  function AddItemForm({ onNewItem }) {
54    const { Form, InputGroup, Button } = ReactBootstrap;
55
56    return (
57      <Form>
58        <InputGroup>
59          <Form.Control type="text" placeholder="Enter todo item" />
60          <Button variant="primary" onClick={onNewItem}>Add</Button>
61        </InputGroup>
62      </Form>
63    );
64  }
65
66  export default TodoListCard;
67
68  
```

Lab Worksheet

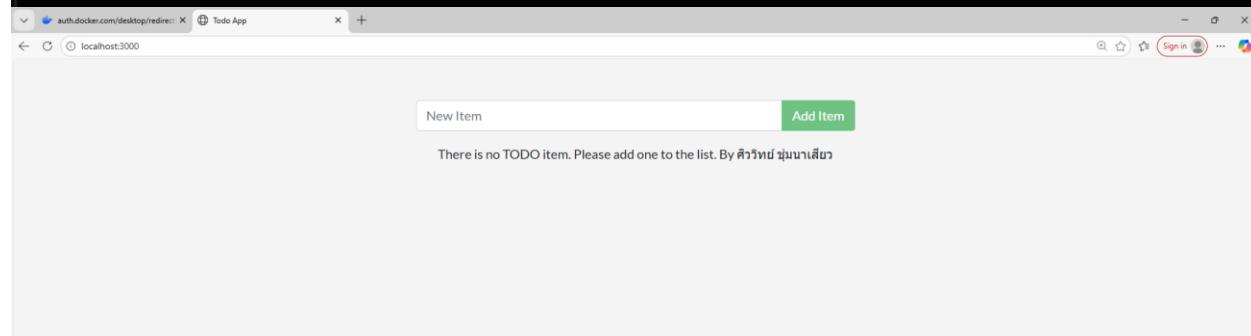
```
PS C:\Users\JJay\Lab8_4\getting-started\app> docker stop b73e93fc95a9
b73e93fc95a9
PS C:\Users\JJay\Lab8_4\getting-started\app> docker build -t myapp_6633804023 .
[+] Building 17.8s (10/10) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from dockerfile                      0.0s
=> => transferring dockerfile: 156B                                         0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine          2.2s
=> [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@sha256:8d6421d663b4c28fd3ebc498332 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332 0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 8.13kB                                       0.0s
=> CACHED [2/4] WORKDIR /app                                         0.0s
=> [3/4] COPY . .                                                 0.1s
=> [4/4] RUN yarn install --production                                11.3s
=> exporting to image                                              3.8s
=> => exporting layers                                         2.3s
=> => exporting manifest sha256:e7c5f64a74e9022c2d6cbe5e612728464f6e850279956b132 0.0s
=> => exporting config sha256:6a0a6330a832912fcceb6ad5dcfc5bbc509d40341e254e2bd688 0.0s
=> => exporting attestation manifest sha256:2435412868a4f876741e00c647459655ad3d5 0.1s
=> => exporting manifest list sha256:9a5c898e013328f83971dc549e4ad1a5507bd389b250 0.0s
=> => naming to docker.io/library/myapp_6633804023:latest           0.0s
=> => unpacking to docker.io/library/myapp_6633804023:latest         1.3s
```

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/l9axg37duzuhavsdks3y8aff

```
PS C:\Users\JJay\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6633804023
```

```
4ffd41ffe9321d6e71b995280814e9ca16aef4e1563da55d838822f924f2204d
```

```
PS C:\Users\JJay\Lab8_4\getting-started\app> []
```



- (1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร

Lab Worksheet

```

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/sw4n5kys
0s22hl0b7yxagkh85
PS C:\Users\JJay\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6633804023
0b95fb45a572ba5963e3896cf1ab23429217c5762ef91344c14df5052a74fdca
docker: Error response from daemon: failed to set up container networking: driver failed
programming external connectivity on endpoint beautiful_lamarr (310f71645933f4796790570f5
6c58fcf2168c2b0002dfa9d530732a4579e2a60): Bind for 0.0.0.0:3000 failed: port is already a
llocated

Run 'docker run --help' for more information
PS C:\Users\JJay\Lab8_4\getting-started\app>

```

_____port ได้ถูกใช้ไปแล้ว ต้องทำการ docker stop <container id> เพื่อหยุดการ run_____

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้เครื่องมือที่ดังต่อไปนี้

- ผ่าน Command line interface
 - ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
 - Copy หรือบันทึก Container ID ไว้
 - ใช้คำสั่ง \$ docker stop <Container ID> ที่ต้องการจะลบ เพื่อยุติการทำงานของ Container ดังกล่าว
 - ใช้คำสั่ง \$ docker rm <Container ID> ที่ต้องการจะลบ เพื่อทำการลบ

- ผ่าน Docker desktop

- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกรัง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

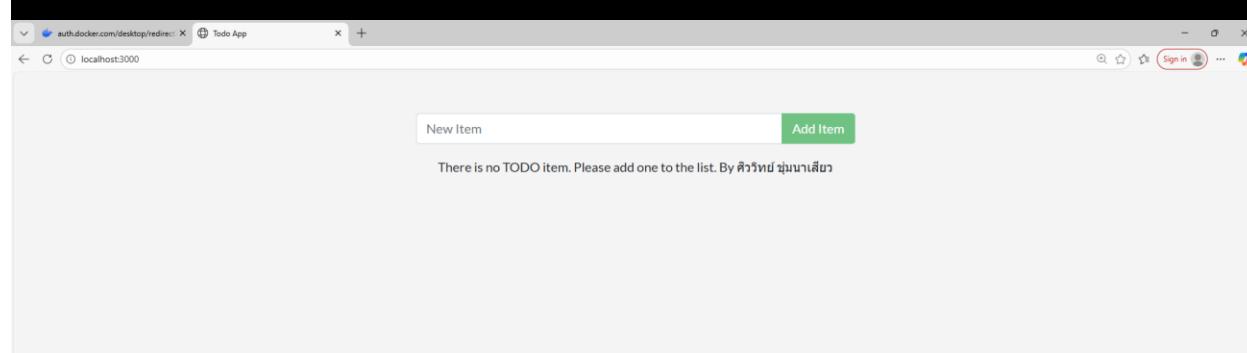
[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่เดือน Browser และ Dashboard ของ Docker desktop

Lab Worksheet

```

PS C:\Users\JJay\Lab8_4\getting-started\app> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
              PORTS
0b95fb45a572   myapp_6633804023 "docker-entrypoint.s..."  3 minutes ago   Created
                                         beautiful_lamarr
4ffd41ffe932   9a5c898e0133    "docker-entrypoint.s..."  7 minutes ago   Up 7 minu
tes           0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp  romantic_goldberg
b73e93fc95a9   3c643f769a40    "docker-entrypoint.s..."  18 minutes ago  Exited (0
) 8 minutes ago
6f4fa4817df8   busybox        "/bin/sh -c 'echo \"H..'"  About an hour ago  Exited (0
) About an hour ago
89735fc9816   9258792af83a   "/bin/sh -c 'echo \"H..'"  About an hour ago  Exited (0
) About an hour ago
7d92a1c599ff   b3255e7dfbcd  "sh"                   3 hours ago    Exited (0
) 3 hours ago
a9312f41dea8   b3255e7dfbcd  "sh"                   3 hours ago    Exited (0
) 3 hours ago
PS C:\Users\JJay\Lab8_4\getting-started\app> docker rm b73e93fc95a9
b73e93fc95a9
PS C:\Users\JJay\Lab8_4\getting-started\app> 
PS C:\Users\JJay\Lab8_4\getting-started\app> docker stop 4ffd41ffe932
4ffd41ffe932
PS C:\Users\JJay\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6633804023
9f508c7728d05a0f896f1ef7df9222ef7f2e255bfcab87bb068ff95af6add4b
PS C:\Users\JJay\Lab8_4\getting-started\app> 

```

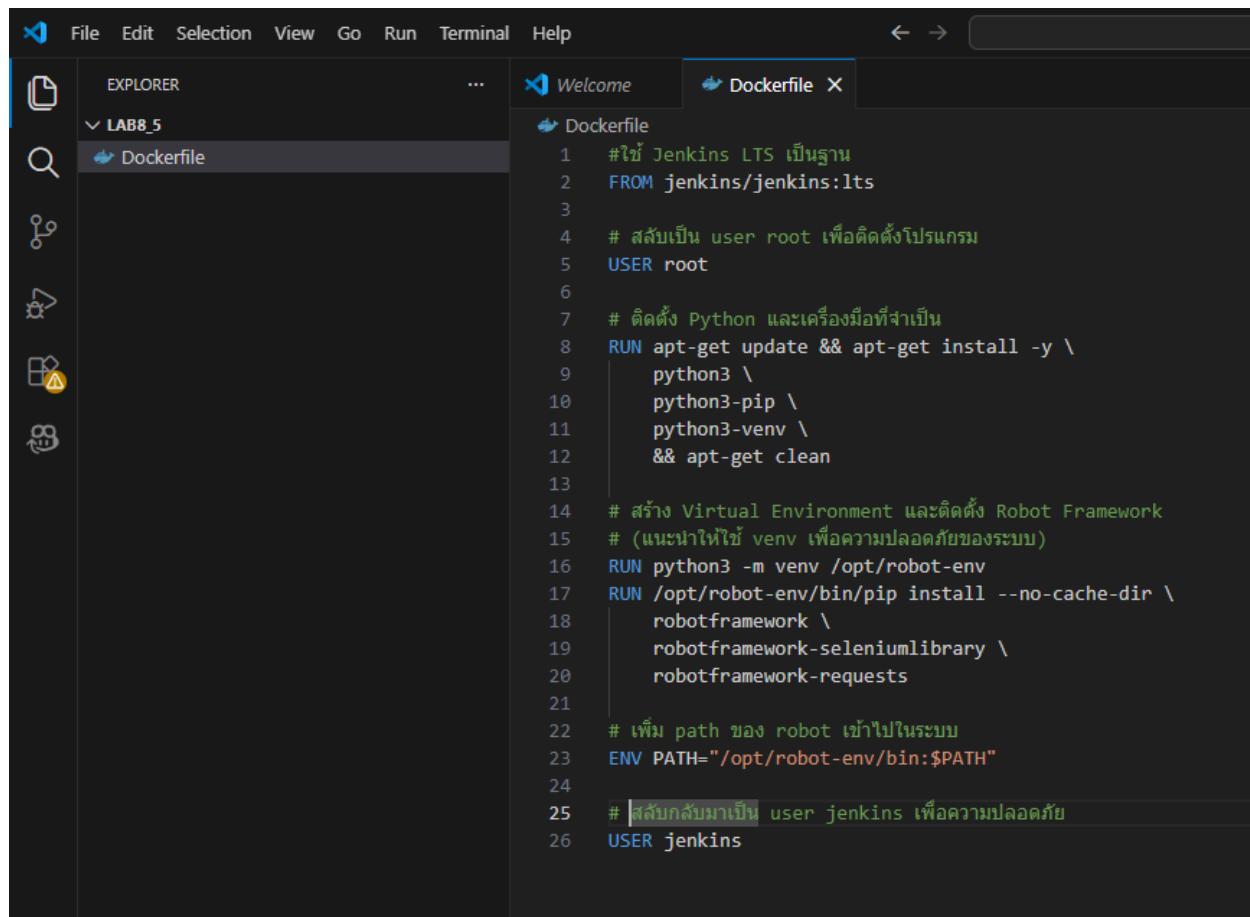


แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

- สร้าง Dockerfile เพื่อสร้าง Jenkins และ Environment ที่เหมาะสมกับการรัน Robot framework ใน Container

Lab Worksheet

[Check point#12] ส่ง Dockerfile ที่ใส่คำสั่งที่เกี่ยวข้องไว้



```

Dockerfile
1 #ใช้ Jenkins LTS เป็นฐาน
2 FROM jenkins/jenkins:lts
3
4 # สับเป็น user root เพื่อติดตั้งโปรแกรม
5 USER root
6
7 # ติดตั้ง Python และเครื่องมือที่จำเป็น
8 RUN apt-get update && apt-get install -y \
9     python3 \
10    python3-pip \
11    python3-venv \
12    && apt-get clean
13
14 # สร้าง Virtual Environment และติดตั้ง Robot Framework
15 # (แนะนำให้ใช้ venv เพื่อความปลอดภัยของระบบ)
16 RUN python3 -m venv /opt/robot-env
17 RUN /opt/robot-env/bin/pip install --no-cache-dir \
18     robotframework \
19     robotframework-seleniumlibrary \
20     robotframework-requests
21
22 # เพิ่ม path ของ robot ให้ไว้ในระบบ
23 ENV PATH="/opt/robot-env/bin:$PATH"
24
25 # ลับยกลับมาเป็น user jenkins เพื่อความปลอดภัย
26 USER jenkins

```

2. เปิด Command line หรือ Terminal บน Docker Desktop
3. Build image ที่สร้างขึ้นในข้อที่ 1 พร้อมกับตั้งชื่อของ image เป็น jenkins-robot-local

\$ docker build -t jenkins-robot-local .

4. รัน container โดยผูกพอร์ตให้เรียบร้อย เช่น

```

$ docker run -d \
--name jenkins-robot \
-p 8080:8080 -p 50000:50000 \
-v jenkins_home:/var/jenkins_home \
-v /var/run/docker.sock:/var/run/docker.sock \
jenkins-robot-local

```

หรือ

Lab Worksheet

```
$ docker run -it \
--name jenkins-debug \
-p 8080:8080 \
-v /var/run/docker.sock:/var/run/docker.sock \
jenkins-robot-local
```

[Check point#13] Capture หน้าจอที่แสดงผล Admin password

Lab Worksheet

```
PS C:\Users\JJay\Lab8_4\getting-started\app> cd ..
PS C:\Users\JJay\Lab8_4\getting-started> cd ..
PS C:\Users\JJay\Lab8_4> cd ..
PS C:\Users\JJay> mkdir Lab8_5
```

Directory: C:\Users\JJay

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
d----	2/4/2026 9:19 PM		Lab8_5

```
PS C:\Users\JJay> cd lab8_5
PS C:\Users\JJay\lab8_5> docker build -t jenkins-robot-local .
[+] Building 117.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.06kB
=> [internal] load metadata for docker.io/jenkins/jenkins:lts
=> [auth] jenkins/jenkins:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/jenkins/jenkins:lts@sha256:d1ea795c6facd7f549a21c40e5e43f
=> => resolve docker.io/jenkins/jenkins:lts@sha256:d1ea795c6facd7f549a21c40e5e43f
=> => sha256:7ad60f966cc569b8cd2fb9743561aefc702562607734da24feed2fe0 392B / 392B
=> => sha256:ed559592686a018bd4133dbddefe59a393c76879d7de48988391 2.23kB / 2.23kB
=> => sha256:57364cf49183c5a348ee44f64d97824f3675ca755e443acb7428e9 0B / 1.29kB
=> => sha256:99ddb13f3ebf7c016ff3c4e840f7f94e8ae2ee11e76ddd684 72.05MB / 72.05MB
=> => sha256:39483126681f342b217818e59842ad78a21576eec7da2432104f 6.29MB / 6.29MB
=> => sha256:406f94702a48c7fc4bd96e198cace8344da8ad779b02cc0daba4a09b 189B / 189B
=> => sha256:e92c5cf141dc90da610685a63439962c05cd31a20c926e06f 96.12MB / 96.12MB
=> => sha256:213d96e0ed060c4bce7c21be00d4ee2bc539c70ecec57c801ed20620 182B / 182B
=> => sha256:5b302cbb8ad382255e4945ccb4b538a40d327abab74e027a7f4f 1.23kB / 1.23kB
=> => sha256:a1c44ba8dd7aa3d28eb657a8b982f61639d88be1b791be797bcf 5.24MB / 5.24MB
=> => sha256:17e22d939f1732029823f86562d22d277dd321da6f370e212 61.59MB / 61.59MB
=> => sha256:53c88f1dfcb79b2f207f7f1a03a45e0dc5ed208b9f496de16 49.29MB / 49.29MB
=> => extracting sha256:53c88f1dfcb79b2f207f7f1a03a45e0dc5ed208b9f496de16b98f8118
=> => extracting sha256:17e22d939f1732029823f86562d22d277dd321da6f370e2126eeb2577
=> => extracting sha256:a1c44ba8dd7aa3d28eb657a8b982f61639d88be1b791be797bcf4bb82
=> => extracting sha256:5b302cbb8ad382255e4945ccb4b538a40d327abab74e027a7f4f4f431
=> => extracting sha256:213d96e0ed060c4bce7c21be00d4ee2bc539c70ecec57c801ed206202
=> => extracting sha256:e92c5cf141dc90da610685a63439962c05cd31a20c926e06f3289c070
=> => extracting sha256:406f94702a48c7fc4bd96e198cace8344da8ad779b02cc0daba4a09bf
```

Lab Worksheet

```

=> => extracting sha256:39483126681f342b217818e59842ad78a21576eec7da2432104f8362d 0.1s
=> => extracting sha256:99ddb13f3ebf7c016ff3c4e840f7f94e8ae2ee11e76ddd68473ec41c4 0.4s
=> => extracting sha256:ed559592686a018bd4133dbddefe59a393c76879d7de4898839197efe 0.0s
=> => extracting sha256:57364cf49183c5a348ee44f64d97824f3675ca755e443acb7428e9b0f 0.0s
=> => extracting sha256:7ad60f966cc569b8cd2fb9743561aefc702562607734da24feed2fe0a 0.0s
=> [2/4] RUN apt-get update && apt-get install -y python3 python3-pip 40.7s
=> [3/4] RUN python3 -m venv /opt/robot-env 1.8s
=> [4/4] RUN /opt/robot-env/bin/pip install --no-cache-dir robotframework 6.2s
=> exporting to image 13.6s
=> => exporting layers 10.7s
=> => exporting manifest sha256:64f284acf1e646af6e2cf2b42acdcc5f2829bb5a1a6d5ac77 0.0s
=> => exporting config sha256:2c467d385de0a1570592b8d57ae64beef1915017873f8b2c795 0.0s
=> => exporting attestation manifest sha256:9aee3db424108d558887cd7a47a864ba4f8dc 0.1s
=> => exporting manifest list sha256:df6723cfed3828b6893a0b7ff50a4466cea3e5b4e734 0.0s
=> => naming to docker.io/library/jenkins-robot-local:latest 0.0s
=> => unpacking to docker.io/library/jenkins-robot-local:latest 2.7s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/67sb1n7af6znp1tdffyg7supv

PS C:\Users\JJay\lab8_5> docker run -d --name jenkins-robot -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home -v //var/run/docker.sock:/var/run/docker.sock ` jenkins-robot-local
b8101b3871122da96d96cdc3c2468d17c952d0a6c201e1d30049e7579c0a649f
PS C:\Users\JJay\lab8_5> docker exec jenkins-robot cat /var/jenkins_home/secrets/initialAdminPassword
d62cee3dfb3f4910bec1bc988fb40470
PS C:\Users\JJay\lab8_5> []

```

5. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก
6. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดเบราว์เซอร์ และป้อนที่อยู่เป็น <http://localhost:8080>
7. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
8. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

Lab Worksheet

[Check point#14] Capture หน้าจอที่แสดงผลการตั้งค่า

The image contains two screenshots of a web browser. The top screenshot shows a 'Getting Started' page titled 'Unlock Jenkins'. It instructs the user to copy a password from a log file located at '/var/jenkins_home/secrets/initialAdminPassword'. A text input field is provided for pasting the password, followed by a 'Continue' button. The bottom screenshot shows a Jenkins user profile page for 'siwawit chumnasiao'. The sidebar menu includes 'Profile' (which is selected), 'Builds', 'My Views', 'Account', 'Appearance', 'Preferences', 'Security', 'Experiments', and 'Credentials'. The main content area displays the Jenkins User ID: siwawit_4023. At the bottom right of the page, there are links for 'REST API' and 'Jenkins 2.541.1'.

- เมื่อติดตั้งเรียบร้อยแล้วจะพบกับหน้า Dashboard ดังแสดงในภาพ

Lab Worksheet

The screenshot shows the Jenkins homepage at localhost:8080. The top navigation bar includes links for 'New Item', 'Build History', 'Add description', and user status. 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 links for 'Set up an agent' and 'Configure a cloud'. On the left, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2). The bottom right corner shows 'REST API' and 'Jenkins 2.541.1'.

10. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

The screenshot shows the 'Manage Jenkins' page at localhost:8080/manage/. The top navigation bar includes links for 'System Configuration', 'Security', and 'Status Information'. The main content area is divided into several sections: 'System Configuration' (System, Nodes), 'Security' (Security, Users), 'Status Information' (System Information, System Log, Load Statistics), and 'Plugins' (highlighted in blue). The 'Plugins' section contains a link to 'Available plugins'.

11. ไปที่เมนู Plugins > Available plugins แล้วเลือกติดตั้ง Robot Framework เพิ่มเติม

Lab Worksheet

Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Robot

Install

Install	Name	Released	Health
<input type="checkbox"/>	Robot Framework 6.2.0	Build Reports	3 mo 15 days ago

This publisher stores Robot Framework test reports for builds and shows summaries of them in project and build views along with trend graph.

91

12. เมื่อติดตั้งสำเร็จจะพบกับรายการ Plugins ทั้งหมดที่คุณติดตั้ง ถ้าติดตั้งสำเร็จให้เลือก “Restart Jenkins...” และกด Go back to the top page

Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Pipeline: GitHub Groovy Libraries	<input checked="" type="checkbox"/>	Success
Metrics	<input checked="" type="checkbox"/>	Success
Pipeline Graph View	<input checked="" type="checkbox"/>	Success
Git	<input checked="" type="checkbox"/>	Success
EDDSA API	<input checked="" type="checkbox"/>	Success
Trilead API	<input checked="" type="checkbox"/>	Success
SSH Build Agents	<input checked="" type="checkbox"/>	Success
Matrix Authorization Strategy	<input checked="" type="checkbox"/>	Success
LDAP	<input checked="" type="checkbox"/>	Success
jsoup API	<input checked="" type="checkbox"/>	Success
Email Extension	<input checked="" type="checkbox"/>	Success
Mailer	<input checked="" type="checkbox"/>	Success
Theme Manager	<input checked="" type="checkbox"/>	Success
Dark Theme	<input checked="" type="checkbox"/>	Success
Loading plugin extensions	<input checked="" type="checkbox"/>	Success
Robot Framework	<input checked="" type="checkbox"/>	Success
Loading plugin extensions	<input checked="" type="checkbox"/>	Success

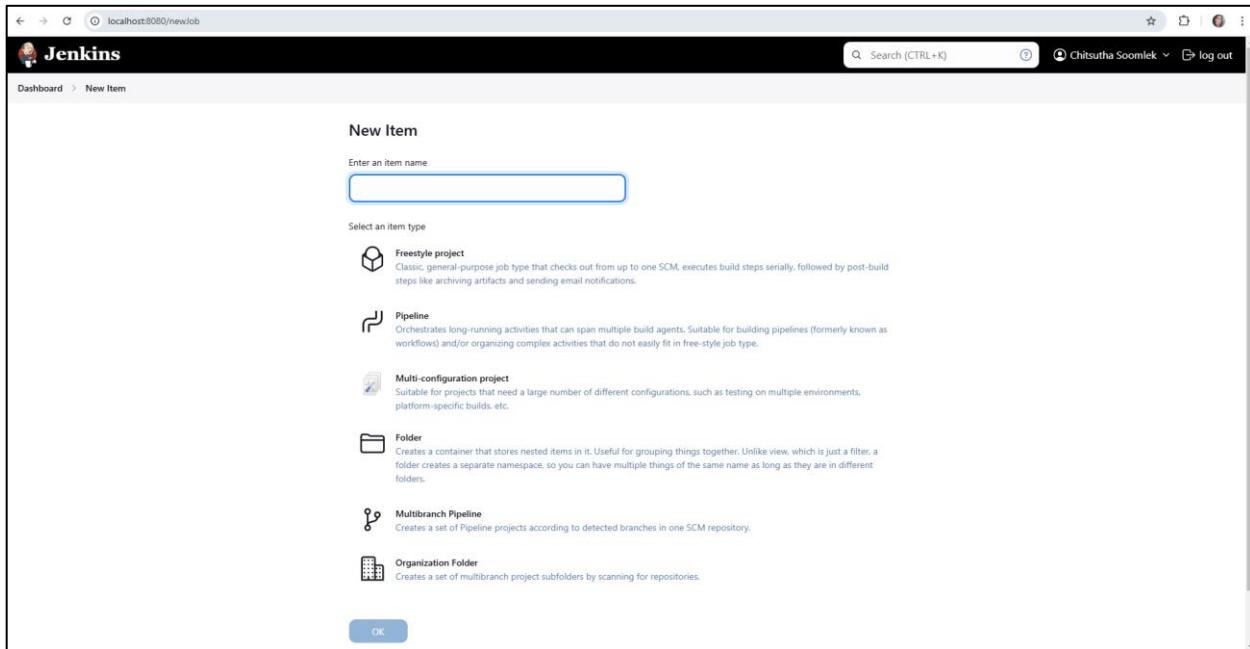
→ Go back to the top page
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.541.1

13. สร้างไฟล์ Jenkinsfile ไม่มีนามสกุล เพื่อ execute คำสั่งต่าง ๆ กับ built-in agent และเอาไฟล์ดังกล่าว เก็บไว้ที่ root ของ GitHub Repository ของนักศึกษา
14. สร้าง folder ชื่อ tests/ บน GitHub Repository ของนักศึกษา และสร้างไฟล์ Lab8.robot และนำไฟล์ไป放ใน folder ที่สร้าง
15. กลับไปที่หน้า Dashboard ของ Jenkins และสร้าง Pipeline อย่างง่าย โดยกำหนด New item เป็น Freestyle project และตั้งชื่อเป็น UAT

Lab Worksheet



16. ตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

Description: Lab 8.5

GitHub project: กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 14)

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

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

Lab Worksheet

[Check point#15] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?

Schedule ?

H/15 * * * *

Would last have run at Wednesday, February 4, 2026, 3:20:00 PM Coordinated Universal Time; would next run at Wednesday, February 4, 2026, 3:35:00 PM Coordinated Universal Time.

- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell ?

Command

See the list of available environment variables

```
mkdir -p results
robot -d results tests/Lab8.robot
```

Advanced ▾

+ Add build step

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

_____ robot -d results tests/Lab8.robot (robot คือคำสั่งเรียกโปรแกรม, -d results คือบอกให้เก็บไฟล์ report ไว้ในโฟลเดอร์ results, tests/Lab8.robot คือไฟล์ที่ต้องการทดสอบ)

Post-build action: เลือก Publish Robot Framework test results -> ระบุไดร์คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

17. กด Apply และ Save

18. สร้าง Build Now

Lab Worksheet

[Check point#16] Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output

Jenkins / UAT / #4

Status Changes Console Output Edit Build Information Delete build #4 Timings Git Build Data Robot Results Previous Build

```

Started by user siawit chumnasao
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/UAT/.git # timeout=10
> git config remote.origin.url https://github.com/Siwa-dev/Jenkins-Robot-Lab8.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.fetch +refs/heads/*:refs/remotes/origin/*
Fetching upstream changes from https://github.com/Siwa-dev/Jenkins-Robot-Labs.git
> git --version # timeout=10
> git -v version 2.47.3'
> git fetch --tags --force --progress .. https://github.com/Siwa-dev/Jenkins-Robot-Lab8.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 1b0fc96f05aab07270b2dacf71dc38023ef8fea (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1b0fc96f05aab07270b2dacf71dc38023ef8fea # timeout=10
Commit message: "Add Jenkins pipeline and Robot test for KGU site"
> git rev-list --no-walk 1b0fc96f05aab07270b2dacf71dc38023ef8fea # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins9824864509319192593.sh
+ mkdir -p results
+ robot -d results tests/Lab8.robot
=====
Lab8
=====
Open KGU Computing Website | PASS |
-----
Lab8 | PASS |
-----
1 test, 1 passed, 0 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
Robot results publisher started...
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: SUCCESS

```

Jenkins / UAT

Status UAT Lab 8.5 Edit description

Latest Robot Results:

	All	Failed	Passed	Skipped	Pass %
All tests	1	0	1	0	100.0

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

Robot Framework Tests Trend (all tests)

Build

Zoom to changes Show only failed all Max builds Show bigger image

Permalinks

Builds > Q Filter

Today

- Last build (#4), 14 min ago
- Last stable build (#4), 14 min ago
- Last unstable build (#3), 26 min ago
- Last failed build (#3), 26 min ago
- Last unsuccessful build (#3), 26 min ago
- Last completed build (#4), 14 min ago