

Lab Worksheet

ชื่อ-นามสกุล _____ นายธนพนธ์ ฝาศิริ _____ รหัสนักศึกษา _____ 653380200-4 _____ 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

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\computer> cd C:\Users\computer\Desktop\653380200-4\Lab\LAB8_1
PS C:\Users\computer\Desktop\653380200-4\Lab\LAB8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
```

What's next:

View a summary of image vulnerabilities and recommendations → `docker scout quickview busybox`PS C:\Users\computer\Desktop\653380200-4\Lab\LAB8_1> `docker images`

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
bash	latest	2a658e2e2bab	6 weeks ago	14.5MB
ai-final	latest	8c1b59b859f3	3 months ago	7.64GB
busybox	latest	af4709625109	3 months ago	4.27MB
dpage/pgadmin4	latest	90ec73daf8e7	5 months ago	481MB
postgres	latest	07a4ee949b9e	8 months ago	432MB

PS C:\Users\computer\Desktop\653380200-4\Lab\LAB8_1>

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร รายชื่อของ images ทั้งหมด
- (2) Tag ที่ใช้บ่งบอกถึงอะไร แสดง version ของ image นั้นๆ

5. ป้อนคำสั่ง \$ `docker run busybox`
6. ป้อนคำสั่ง \$ `docker run -it busybox sh`
7. ป้อนคำสั่ง `ls`
8. ป้อนคำสั่ง `ls -la`
9. ป้อนคำสั่ง `exit`
10. ป้อนคำสั่ง \$ `docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"`
11. ป้อนคำสั่ง \$ `docker ps -a`

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

Lab Worksheet

```

PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run busybox
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run -it busybox sh
/ # ls
bin dev etc home lib lib64 proc root sys tmp usr var
/ # ls -la
total 48
drwxr-xr-x 1 root root 4096 Jan 23 05:01 .
drwxr-xr-x 1 root root 4096 Jan 23 05:01 ..
-rwxr-xr-x 1 root root 0 Jan 23 05:01 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 23 05:01 dev
drwxr-xr-x 1 root root 4096 Jan 23 05:01 etc
drwxr-xr-x 2 nobody nobody 4096 Sep 26 21:31 home
drwxr-xr-x 2 root root 4096 Sep 26 21:31 lib
lrwxrwxrwx 1 root root 3 Sep 26 21:31 lib64 -> lib
dr-xr-xr-x 357 root root 0 Jan 23 05:01 proc
drwx----- 1 root root 4096 Jan 23 05:01 root
dr-xr-xr-x 11 root root 0 Jan 23 05:01 sys
drwxrwxrwt 2 root root 4096 Sep 26 21:31 tmp
drwxr-xr-x 4 root root 4096 Sep 26 21:31 usr
drwxr-xr-x 4 root root 4096 Sep 26 21:31 var
/ # exit
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run busybox echo "Hello นายณพนธ์ ผาคติ 653380200-4 from busybox"
Hello นายณพนธ์ ผาคติ 653380200-4 from busybox
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
68de63ae28fa	busybox	"echo 'Hello นายณพนธ์ ผาคติ'"	7 seconds ago	Exited (0) 6 seconds ago		jolly_aryabhata
110b689db7b7	busybox	"sh"	52 seconds ago	Exited (0) 41 seconds ago		busy_galois
7b3fe19848ce	busybox	"sh"	About a minute ago	Exited (0) 59 seconds ago		adoring_yalow
0d52ee808452	busybox	"sh"	3 minutes ago	Up 3 minutes		adoring_beaver
242dcffda855	busybox	"sh"	3 minutes ago	Exited (0) 3 minutes ago		optimistic_ishizaka
4a58ecf6761b	bash	"docker-entrypoint.s..."	55 minutes ago	Exited (137) 29 minutes ago		pensive_edison
9fe3534a3b32	ai-final:latest	"jupyter lab --ip=0..."	3 months ago	Exited (255) About an hour ago	0.0.0.0:8000->8000/tcp	loving_black

```

PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1>

```

- เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
ทำให้ container ทำงานในโหมด interactive พร้อมกับการแสดงผลคอลัมน์
- STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร
แสดงข้อมูลของ container

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

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

Lab Worksheet

```

PS C:\Users\computer> cd C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest

What's next:
View a summary of image vulnerabilities and recommendations --docker scout quickview busybox
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
bash                 latest          2a658e2e2bab    6 weeks ago    14.5MB
al-final             latest          8c1b59b859f3    3 months ago    7.64GB
busybox              latest          af4709625109    3 months ago    4.27MB
dpkg/pgadmin4        latest          90ec73daf8e7    5 months ago    481MB
postgres             latest          07a4ee949b9e    8 months ago    432MB
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run busybox
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run -it busybox sh
/ # ls
bin    dev    etc    home  lib    lib64  proc  root  sys    tmp    usr    var
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker run busybox echo "Hello นายธนพล ภาศิริ 653380200-4 from busybox"
Hello นายธนพล ภาศิริ 653380200-4 from busybox
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
68de63ae28fa   busybox    "echo 'Hello นายธนพล'" 7 seconds ago  Exited (0) 6 seconds ago           jolly_aryabhata
110b689db7b7   busybox    "sh"                    52 seconds ago Exited (0) 41 seconds ago           busy_galois
7b3fe19848ce   busybox    "sh"                    About a minute ago Exited (0) 59 seconds ago           adoring_yalow
0d52ee808452   busybox    "sh"                    3 minutes ago   Up 3 minutes                admiring_beaver
242dcffda855   busybox    "sh"                    3 minutes ago   Exited (0) 3 minutes ago           optimistic_ishizaka
4a58ecf6761b   bash      "docker-entrypoint.s..." 55 minutes ago  Exited (137) 29 minutes ago           pensive_edison
9fe3534a3b32   al-final:latest "jupyter lab --ip=0..." 3 months ago   Exited (255) About an hour ago   0.0.0.0:8000->8000/tcp loving_black
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker rm 110b689db7b7a9ad1b312492adf230e83cb0614509f0863bfb935f157cb10632
110b689db7b7a9ad1b312492adf230e83cb0614509f0863bfb935f157cb10632
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1>
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
68de63ae28fa   busybox    "echo 'Hello นายธนพล'" 7 seconds ago  Exited (0) 6 seconds ago           jolly_aryabhata
110b689db7b7   busybox    "sh"                    52 seconds ago Exited (0) 41 seconds ago           busy_galois
7b3fe19848ce   busybox    "sh"                    About a minute ago Exited (0) 59 seconds ago           adoring_yalow
0d52ee808452   busybox    "sh"                    3 minutes ago   Up 3 minutes                admiring_beaver
242dcffda855   busybox    "sh"                    3 minutes ago   Exited (0) 3 minutes ago           optimistic_ishizaka
4a58ecf6761b   bash      "docker-entrypoint.s..." 55 minutes ago  Exited (137) 29 minutes ago           pensive_edison
9fe3534a3b32   al-final:latest "jupyter lab --ip=0..." 3 months ago   Exited (255) About an hour ago   0.0.0.0:8000->8000/tcp loving_black
PS C:\Users\computer\Desktop\653380200-4\Lab\Lab8_1> docker rm 110b689db7b7a9ad1b312492adf230e83cb0614509f0863bfb935f157cb10632
110b689db7b7a9ad1b312492adf230e83cb0614509f0863bfb935f157cb10632
PS C:\Users\computer\Desktop\653380200-4\Lab\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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ (Windows) บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

\$ cat > Dockerfile << EOF

Lab Worksheet

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t <ชื่อ Image> .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_2> docker build -t lab .
[+] Building 2.6s (6/6) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 189B                                              0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2) 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will b 0.0s
=> => unpacking to docker.io/library/lab:latest                                  0.0s

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_2> docker run lab
"ธนพนธ์ ผศศิริ 653380200-4 ครีว"
```

- (1) คำสั่งที่ใช้ในการ run คือ

`docker run lab`

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

เพื่อดีงชื่อ และ ระบุ tag ของ 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 ในการใส่เนื้อหาหาแทน

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\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_3> docker build -t 6533802004/lab8 .
[+] Building 0.4s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile             0.0s
=> => transferring dockerfile: 210B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2) 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will b 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3) 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:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1 0.0s
=> => resolve docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:4efd06018f1c1fb4fa5896a2a01b9cff76ebecfdda0ce8ef8a98274fd31a29cd 0.0s
=> => exporting config sha256:570df9c533e9ba6b8e0153e6e5a0692d07f7c889eb5a1ac0ea3211b7577f823 0.0s
```

Lab Worksheet

```

=> exporting to image                                0.1s
=> => exporting layers                                0.0s
=> => exporting manifest sha256:4efd06018f1c1fb4fa5896a2a01b9c7f76ebecfdda0ce8ef8a98274fd31a29cd 0.0s
=> => exporting config sha256:570df9c533e9ba6b8e0153e6e65a0692d07f7c889eb5a1ac0ea3211b7577f823 0.0s
=> => exporting attestation manifest sha256:1f05801f115e91c8b933706c0b55916ed14f54f366db6ac284d7386225b3b184 0.0s
=> => exporting manifest list sha256:5bdec87475a0e04e50eb642afaa2099b723fab06811fa8d977ccfc117545f5e 0.0s
=> => naming to docker.io/6533802004/lab8:latest      0.0s
=> => unpacking to docker.io/6533802004/lab8:latest   0.0s

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
PS C:\Users\ACE\OneDrive\เอกสาร\solfen\Lab8_3> docker run 6533802004/lab8
"นายสมณณ์ ผศจิ 653380200-4"

```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

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

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

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

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

```

PS C:\Users\ACE\OneDrive\เอกสาร\solfen\Lab8_3> docker push 6533802004/lab8
Using default tag: latest
The push refers to repository [docker.io/6533802004/lab8]
9c0abc9c5bd3: Mounted from library/busybox
1309966f3a1a: Pushed
latest: digest: sha256:5bdec87475a0e04e50eb642afaa2099b723fab06811fa8d977ccfc117545f5e size: 855

```

Digest	OS/ARCH	Size	Last pull	
4efd06018f1c	linux / amd64	2.15 MB	1 minute ago	Pull image
1f05801f115e	unknown / unknown	1 KB		Pull image

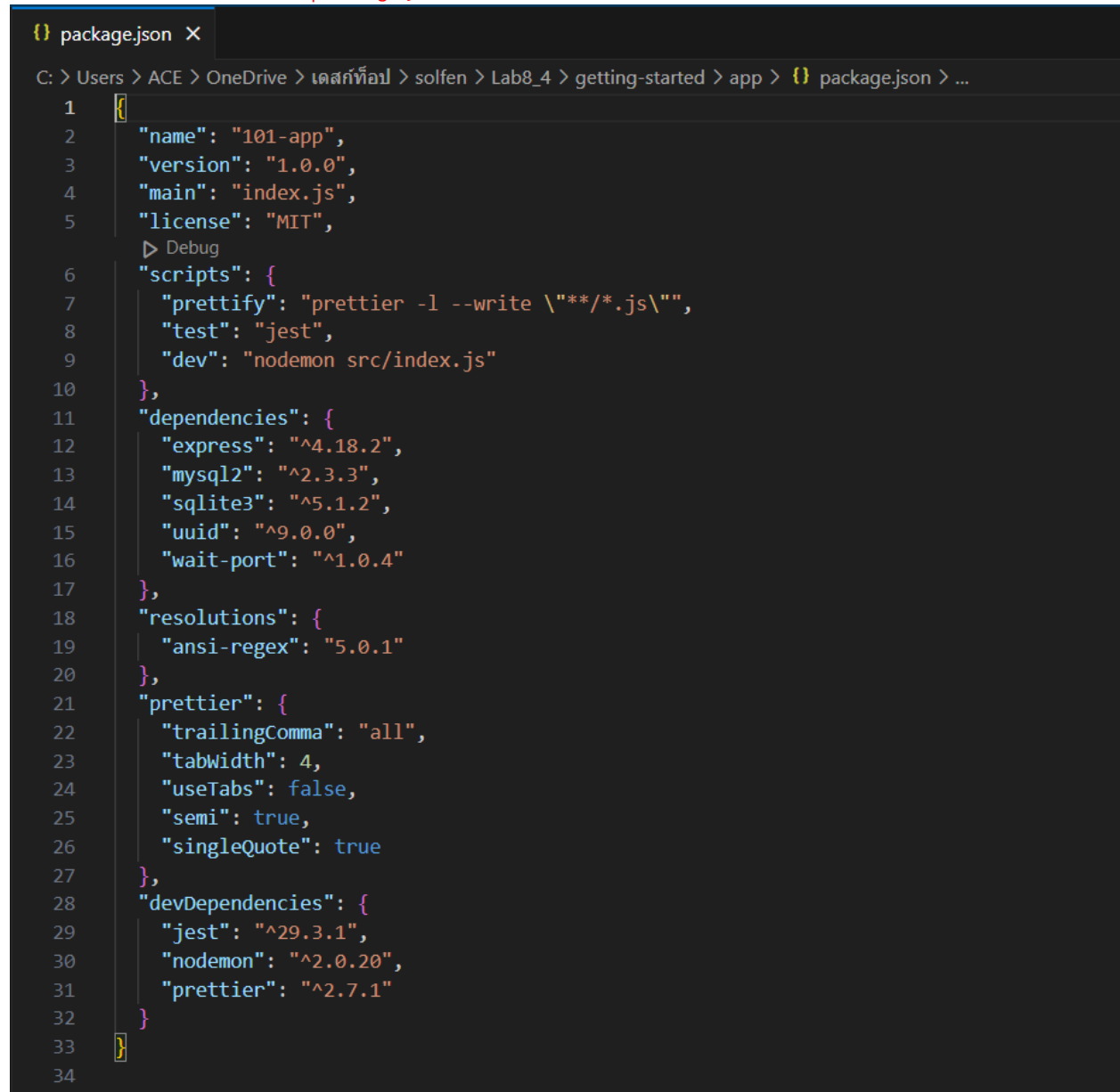
แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4

Lab Worksheet

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



```
{ package.json X
C: > Users > ACE > OneDrive > เดสก์ท็อป > solfen > Lab8_4 > getting-started > app > {} package.json > ...
1  {
2    "name": "101-app",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "scripts": {
7      "prettify": "prettier -l --write \"**/*.js\"",
8      "test": "jest",
9      "dev": "nodemon src/index.js"
10   },
11   "dependencies": {
12     "express": "^4.18.2",
13     "mysql2": "^2.3.3",
14     "sqlite3": "^5.1.2",
15     "uuid": "^9.0.0",
16     "wait-port": "^1.0.4"
17   },
18   "resolutions": {
19     "ansi-regex": "5.0.1"
20   },
21   "prettier": {
22     "trailingComma": "all",
23     "tabWidth": 4,
24     "useTabs": false,
25     "semi": true,
26     "singleQuote": true
27   },
28   "devDependencies": {
29     "jest": "^29.3.1",
30     "nodemon": "^2.0.20",
31     "prettier": "^2.7.1"
32   }
33 }
```


Lab Worksheet

```

PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4> cdcd getting-started
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started> ls

Directory: C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started

PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started> cd app
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> start package.json
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app>
[main 2025-01-29T12:31:49.833Z] update#setState idle
(node:29500) [DEP0168] DeprecationWarning: Uncaught N-API callback exception detected, please run node with option --force-node-api-uncaught-exceptions-policy=true to handle those exceptions properly.
(Use 'Code --trace-deprecation ...' to show where the warning was created)

```

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

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

```

PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker build -t myapp_6533802004 .
[+] Building 20.2s (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   3.6s
=> [auth] library/node:pull token for registry-1.docker.io         0.0s
=> [internal] load .dockerignore                                    0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 2.7s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB 0.4s
=> => sha256:37892ffbfc0aa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB 1.8s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B 1.2s
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB 1.4s
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 0.1s
=> => extracting sha256:37892ffbfc0aa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 0.6s

```

Lab Worksheet

```

=> => extracting sha256:37892ffbfca871a10f813803949d18c3015a482051d51b7e0da02525e63167c 0.6s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 0.0s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 0.0s
=> [internal] load build context 0.3s
=> => transferring context: 4.62MB 0.2s
=> [2/4] WORKDIR /app 0.2s
=> [3/4] COPY . . 0.1s
=> [4/4] RUN yarn install --production 9.7s
=> exporting to image 3.7s
=> => exporting layers 2.6s
=> => exporting manifest sha256:56f649dc004edb5faa1f8a7bcc7b0624e1d868a98166e2bbdd91f646e99f41f9 0.0s
=> => exporting config sha256:88b220e4b52cf2fb53112a2ea3ef9fba8036941620588f1bcb1732e3a6e344ae 0.0s
=> => exporting attestation manifest sha256:a3880965a7304f9277b95c07b06e685106ef71ba1b7e3ec9b81040c5ca98955d 0.0s
=> => exporting manifest list sha256:1485e5b81c7e8c76f0a8cc8affcfca7ed8c84c3d30d202269d1fa6176a1a368a 0.0s
=> => naming to docker.io/library/myapp_6533802004:latest 0.0s
=> => unpacking to docker.io/library/myapp_6533802004:latest 1.0s
PS C:\Users\ACE\OneDrive\Desktop\solfen\Lab8_4\getting-started\app>

```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

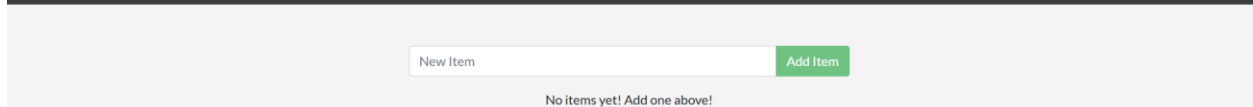
7. เปิด Browser ไปที่ URL = <http://localhost:3000>

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

```

PS C:\Users\ACE\OneDrive\Desktop\solfen\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802004
1acdb573f5151fb84c232fb580a1fbb363a3c4ee3ae5cfb68af5b42dc48ba9cc
PS C:\Users\ACE\OneDrive\Desktop\solfen\Lab8_4\getting-started\app>

```



หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้

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

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

Lab Worksheet

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

By ชื่อและนามสกุลของนักศึกษา</p>

b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

```
PS C:\Users\ACE\OneDrive\เดสทอป\solfen\Lab8_4\getting-started\app> docker build -t myapp_6533802004 .
[+] Building 17.4s (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  2.0s
=> [auth] library/node:pull token for registry-1.docker.io        0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 8.12kB                                    0.0s
=> CACHED [2/4] WORKDIR /app                                       0.0s
=> [3/4] COPY . .                                                 0.1s
=> [4/4] RUN yarn install --production                            11.4s
=> exporting to image                                              3.8s
=> => exporting layers                                              2.5s
=> => exporting manifest sha256:50448f281c9215224535fe38311d8c76443f762aa2d740aac86e0290e6cc1530 0.0s
=> => exporting config sha256:e072842e3c726317d72dc6e116b28cfb397a949b83f7ef4cc35b033e3c9d85d2 0.0s
=> => exporting attestation manifest sha256:9383a8c47dbf10d8cb5fcaa350a77c64a220d2ad06c3568fd6f5bd000511e2a3 0.0s
=> => exporting manifest list sha256:f6ab249d926792235d49787519883c5aa0fccc85b0ff0f9c83b7209b9d0a0958 0.0s
=> => naming to docker.io/library/myapp_6533802004:latest         0.0s
=> => unpacking to docker.io/library/myapp_6533802004:latest      1.2s
PS C:\Users\ACE\OneDrive\เดสทอป\solfen\Lab8_4\getting-started\app> dodocker run -dp 3000:3000 myapp_6533802004
e7bcb8be7429ea222462eaf7a31fa9df0442091b2c7c674278a27a5dcb1a957a
docker: Error response from daemon: driver failed programming external connectivity on endpoint dazzling_mccarthy (a55ef97cb030f2d97bebb02bb4a6a508dfa9782e95776f1251e7d4055fbc22f6): Bind for 0.0.0.0:3000 failed: port is already allocated.
```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

มีการรันอยู่ที่port 3000 อยู่จึงไม่สามารถเปิดได้

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

- ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
- Copy หรือบันทึก Container ID ไว้
- ใช้คำสั่ง \$ docker stop <Container ID ที่ต้องการจะลบ> เพื่อหยุดการทำงานของ Container ดังกล่าว
- ใช้คำสั่ง \$ docker rm <Container ID ที่ต้องการจะลบ> เพื่อทำการลบ

b. ผ่าน Docker desktop

- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

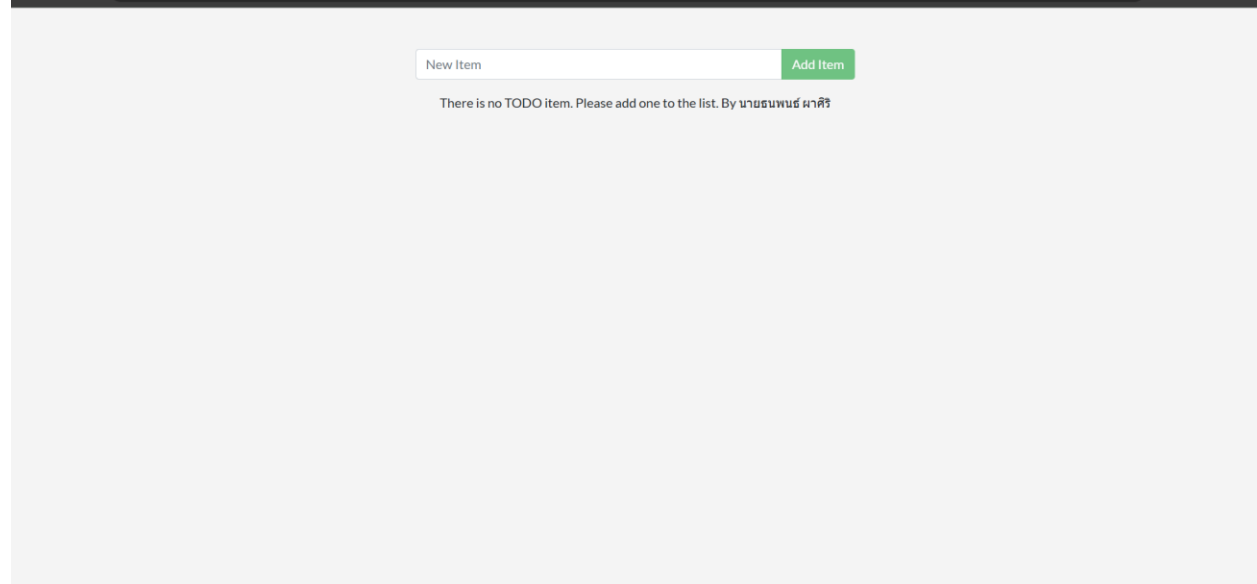
12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

Lab Worksheet

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

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

```
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
a394a9c1f386   myapp_6533802004   "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  0.0.0.0:3000->3000/tcp   xenodochial_bartik
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker s^C
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker stop a394a9c1f386
a394a9c1f386
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker rm a394a9c1f386
a394a9c1f386
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker build -t myapp_6533802004 .
PS C:\Users\ACE\OneDrive\เดสก์ท็อป\solfen\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802004
80a48254b7c754184191c9847a1984176a9e9142361e1cf3a6a825bb575b45b3
```



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

Lab Worksheet

```

2025-01-29 12:59:06.048+0000 [id=61] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-29 12:59:06.049+0000 [id=61] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-29 12:59:06.050+0000 [id=79] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-29 12:59:06.064+0000 [id=95] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-01-29 12:59:06.295+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:

bd85c16da15f48c5a7e6d83f420e958f

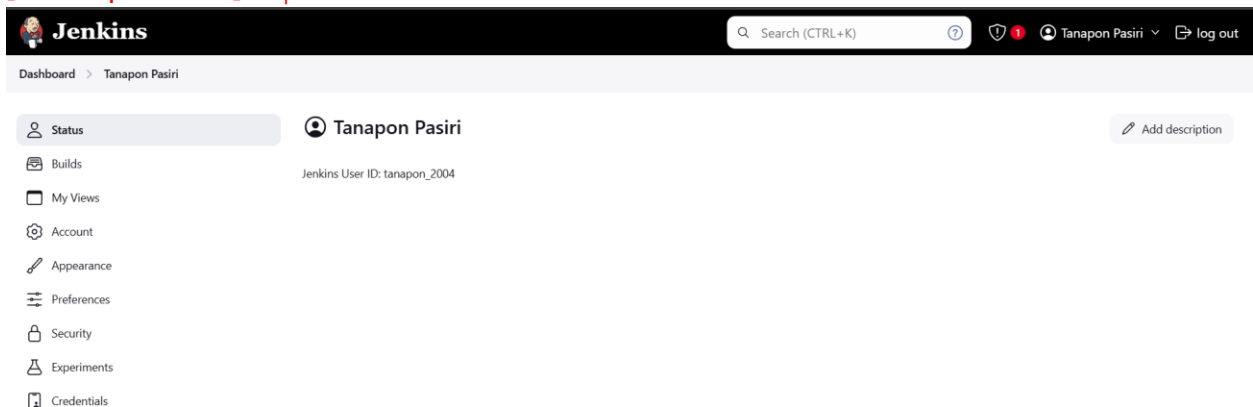
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

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

```

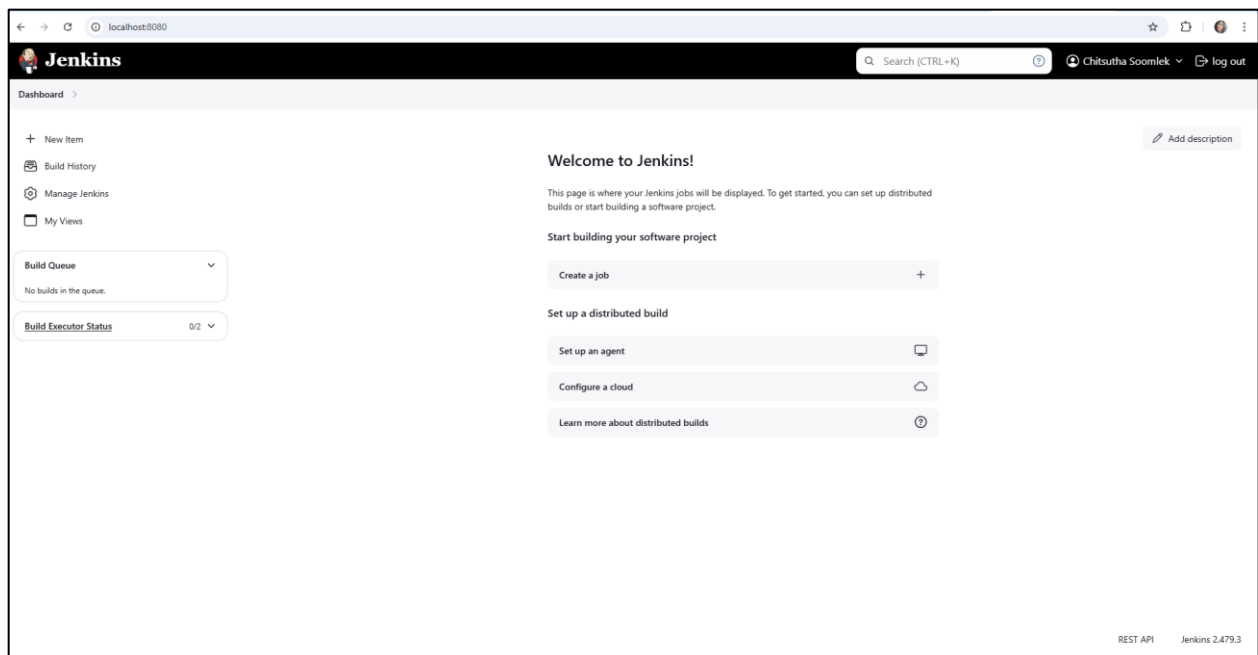
4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

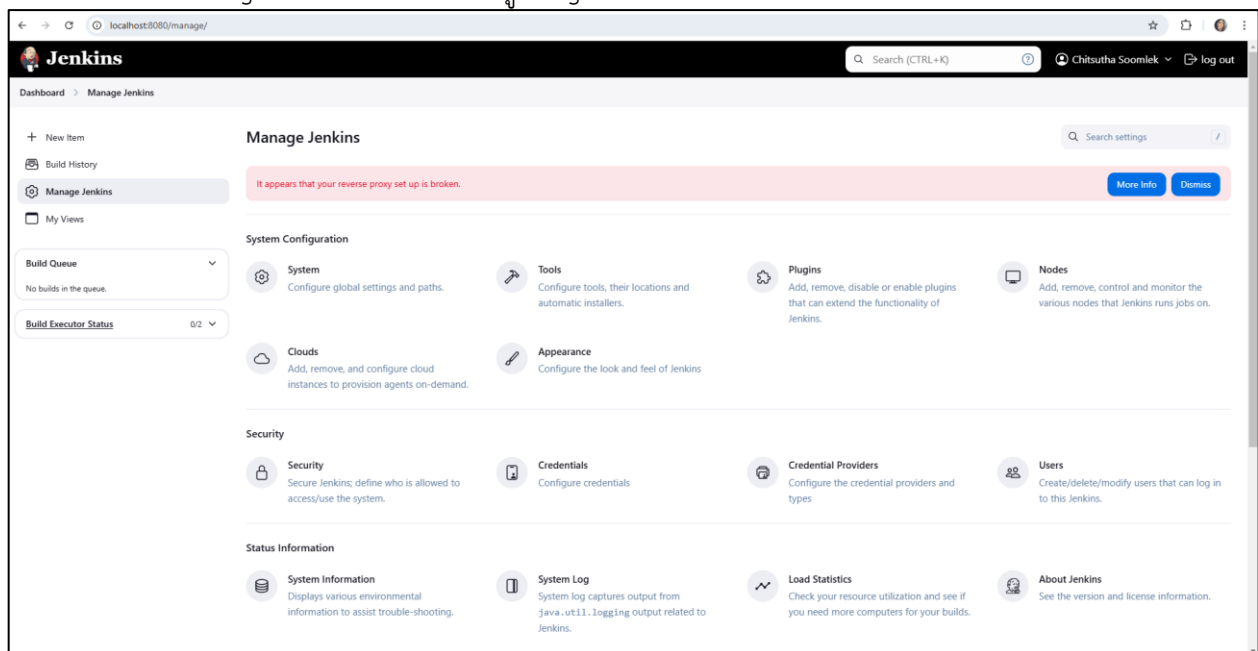


7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

Lab Worksheet

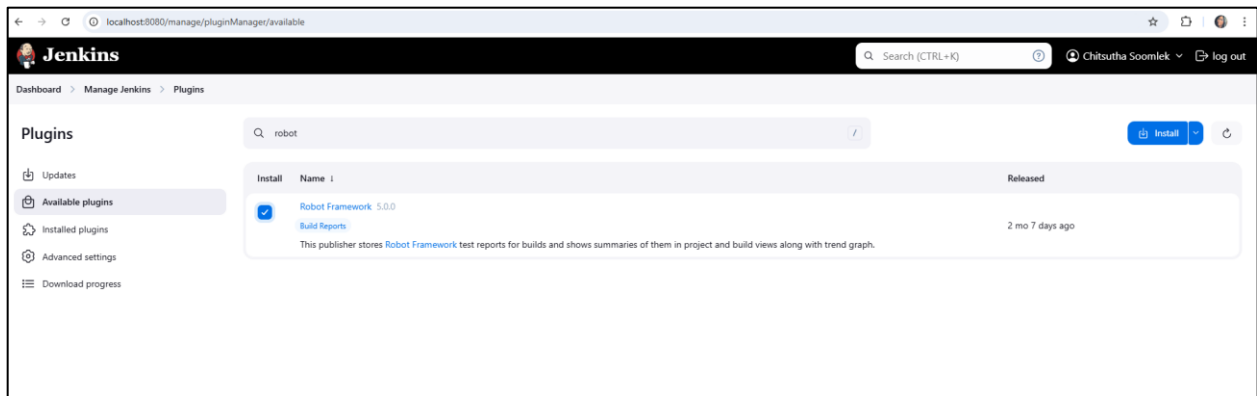


9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

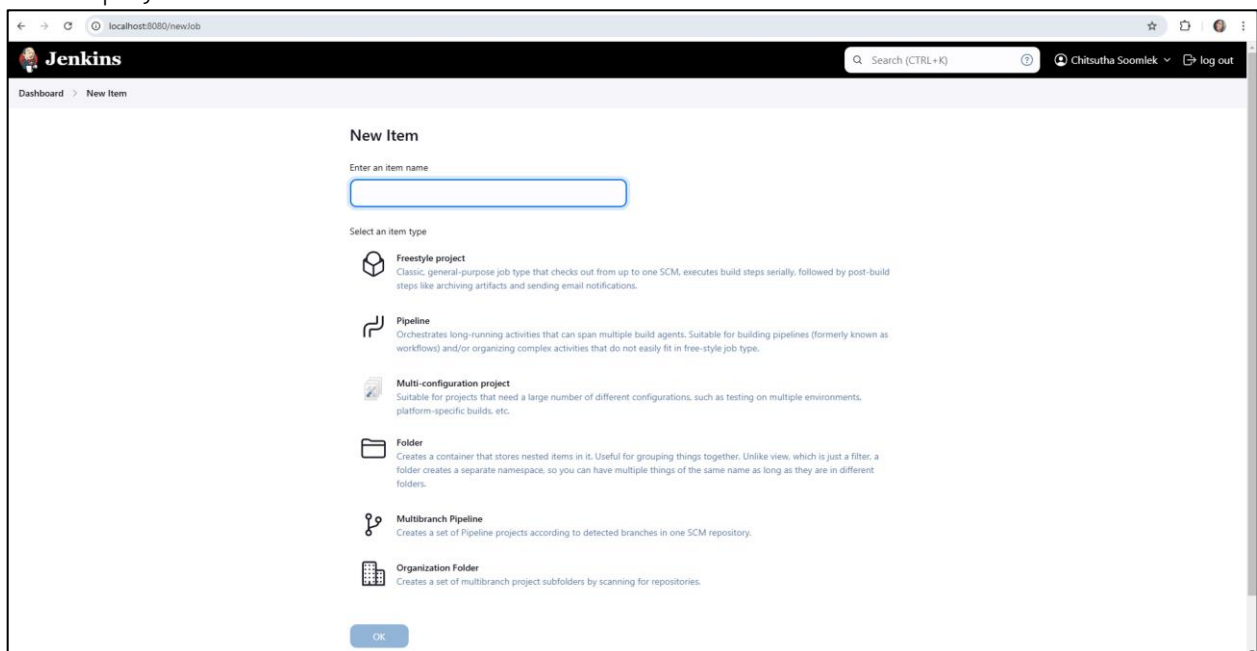


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม

Lab Worksheet



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

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Lab Worksheet

The screenshot shows the Jenkins web interface for configuring a job named 'UAT'. The top navigation bar includes the Jenkins logo, a search bar, and user information (Tanapon Pasiri). The breadcrumb trail is 'Dashboard > UAT > Configuration'.

Configure

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

General (Enabled)

Description

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

<https://github.com/drewtanapon/SolfEn.git/>

Advanced ▾

☐ This project is parameterized ?

[Save](#) [Apply](#)

localhost:8080/job/UAT/configure

Dashboard > UAT > Configuration

Configure

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

☐ Git ?

Build Triggers

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

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

[H/15 * * * * *](#)

Would last have run at Wednesday, January 29, 2025 at 1:20:38 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 1:35:38 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Build Environment

☐ Delete workspace before build starts

[Save](#) [Apply](#)

Lab Worksheet

Build Steps

Execute shell

?

Command

See [the list of available environment variables](#)

```

./var/jenkins_home/workspace/UAT/env/bin/activate
export PATH=$PATH:/var/jenkins_home/workspace/UAT
mkdir -p results
robot --outputdir results valid_login.robot

```

Advanced

Add build step

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

./var/jenkins_home/workspace/UAT/env/bin/activate
export PATH=$PATH:/var/jenkins_home/workspace/UAT
mkdir -p results
robot --outputdir results valid_login.robot

```

**Post-build action:** เพิ่ม Publish Robot Framework test results -> ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่าน แล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

13. กด Apply และ Save
14. สั่ง Build Now

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

## Lab Worksheet

```
=====
Valid Login :: A test suite with a single test for valid login.
=====
Valid Login | FAIL |
NoSuchDriverException: Message: Unable to obtain driver for chrome; For documentation on this error, please visit:
https://www.selenium.dev/documentation/webdriver/troubleshooting/errors/driver_location

Valid Login :: A test suite with a single test for valid login. | FAIL |
1 test, 0 passed, 1 failed
=====
Output: /var/jenkins_home/workspace/UAT/results/output.xml
Log: /var/jenkins_home/workspace/UAT/results/log.html
Report: /var/jenkins_home/workspace/UAT/results/report.html
Build step 'Execute shell' marked build as failure
Robot results publisher started...
INFO: Checking test criticality is deprecated and will be dropped in a future release!
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE
```