

Lab Worksheet

ชื่อ-นามสกุล_วิชากร บุญประคม_รหัสนักศึกษา_653380024-8_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

```

PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker image

Usage: docker image COMMAND

Manage images

Commands:
  build      Build an image from a Dockerfile
  history    Show the history of an image
  import     Import the contents from a tarball to create a filesystem image
  inspect    Display detailed information on one or more images
  load       Load an image from a tar archive or STDIN
  ls         List images
  prune      Remove unused images
  pull       Download an image from a registry
  push       Upload an image to a registry
  rm         Remove one or more images
  save       Save one or more images to a tar archive (streamed to STDOUT by default)
  tag        Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
default-gi-e7e67-php	latest	f55c6901d3bb	4 days ago	2.15GB
phpmyadmin/phpmyadmin	latest	44328e7738eb	7 days ago	799MB
mysql	8.0	d58ac93387f6	3 months ago	811MB
busybox	latest	a5d0ce49aa80	3 months ago	6.56MB

```

PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1>

```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร: ชื่อของ Images ของ container นั้นๆ
- (2) Tag ที่ใช้บอกถึงอะไร: version ของ docker images นั้น เช่น latest นั้นหมายถึง version ล่าสุดที่ releases
5. ป้อนคำสั่ง \$ docker run busybox

```

PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x  1 root    root          4096 Jan 22 06:17 ..
-rwxr-xr-x  1 root    root           0 Jan 22 06:17 .dockerenv
drwxr-xr-x  2 root    root        12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root         360 Jan 22 06:17 dev
drwxr-xr-x  1 root    root         4096 Jan 22 06:17 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 309 root    root           0 Jan 22 06:17 proc
drwx----- 1 root    root         4096 Jan 22 06:17 root
dr-xr-xr-x 11 root    root           0 Jan 22 06:17 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
/ #

```

Lab Worksheet

6. ป้อนคำสั่ง \$ docker run -it busybox sh

```

PowerShell
Run 'docker image COMMAND --help' for more information on a command.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
default-gi-e7e67-php latest          f55c6901d3bb   4 days ago     2.15GB
phpmyadmin/phpmyadmin latest          44328e7738eb   7 days ago     799MB
mysql               8.0            d58ac93387f6   3 months ago   811MB
busybox             latest         a5d0ce49aa80   3 months ago   6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x 1 root    root      4096 Jan 22 06:17 ..
-rwxr-xr-x 1 root    root        0 Jan 22 06:17 .dockerenv
drwxr-xr-x 2 root    root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root    root      360 Jan 22 06:17 dev
drwxr-xr-x 1 root    root      4096 Jan 22 06:17 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 309 root    root        0 Jan 22 06:17 proc
drwx----- 1 root    root      4096 Jan 22 06:17 root
dr-xr-xr-x 11 root    root        0 Jan 22 06:17 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
/ # |

```

7. ป้อนคำสั่ง ls

```

PowerShell
Run 'docker image COMMAND --help' for more information on a command.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
default-gi-e7e67-php latest          f55c6901d3bb   4 days ago     2.15GB
phpmyadmin/phpmyadmin latest          44328e7738eb   7 days ago     799MB
mysql               8.0            d58ac93387f6   3 months ago   811MB
busybox             latest         a5d0ce49aa80   3 months ago   6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x 1 root    root      4096 Jan 22 06:17 ..
-rwxr-xr-x 1 root    root        0 Jan 22 06:17 .dockerenv
drwxr-xr-x 2 root    root     12288 Sep 26 21:31 bin
drwxr-xr-x 5 root    root      360 Jan 22 06:17 dev
drwxr-xr-x 1 root    root      4096 Jan 22 06:17 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 309 root    root        0 Jan 22 06:17 proc
drwx----- 1 root    root      4096 Jan 22 06:17 root
dr-xr-xr-x 11 root    root        0 Jan 22 06:17 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
/ # |

```

8. ป้อนคำสั่ง ls -la

Lab Worksheet

```

PowerShell
Run 'docker image COMMAND --help' for more information on a command.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
default-gi-e7e67-php latest             f55c6901d3bb       4 days ago         2.15GB
phpmyadmin/phpmyadmin latest            44328e7738eb       7 days ago         799MB
mysql               8.0               d58ac93387f6       3 months ago       811MB
busybox             latest            a5d0ce49aa80       3 months ago       6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x  1 root    root      4096 Jan 22 06:17 ..
-rwxr-xr-x  1 root    root        0 Jan 22 06:17 .dockerenv
drwxr-xr-x  2 root    root     12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root      360 Jan 22 06:17 dev
drwxr-xr-x  1 root    root      4096 Jan 22 06:17 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 309 root    root        0 Jan 22 06:17 proc
drwx----- 1 root    root      4096 Jan 22 06:17 root
dr-xr-xr-x 11 root    root        0 Jan 22 06:17 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
/ # |

```

9. ป้อนคำสั่ง exit

```

PowerShell
Run 'docker image COMMAND --help' for more information on a command.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
default-gi-e7e67-php latest             f55c6901d3bb       4 days ago         2.15GB
phpmyadmin/phpmyadmin latest            44328e7738eb       7 days ago         799MB
mysql               8.0               d58ac93387f6       3 months ago       811MB
busybox             latest            a5d0ce49aa80       3 months ago       6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x  1 root    root      4096 Jan 22 06:17 ..
-rwxr-xr-x  1 root    root        0 Jan 22 06:17 .dockerenv
drwxr-xr-x  2 root    root     12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root      360 Jan 22 06:17 dev
drwxr-xr-x  1 root    root      4096 Jan 22 06:17 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 309 root    root        0 Jan 22 06:17 proc
drwx----- 1 root    root      4096 Jan 22 06:17 root
dr-xr-xr-x 11 root    root        0 Jan 22 06:17 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 E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> |

```

10. ป้อนคำสั่ง \$ docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"

Lab Worksheet

```

PowerShell
default-gi-e7e67-php latest f55c6901d3bb 4 days ago 2.15GB
phpmyadmin/phpmyadmin latest 44328e7738eb 7 days ago 799MB
mysql 8.0 d58ac93387f6 3 months ago 811MB
busybox latest a5d0ce49aa80 3 months ago 6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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 22 06:17 .
drwxr-xr-x 1 root root 4096 Jan 22 06:17 ..
-rwxr-xr-x 1 root root 0 Jan 22 06:17 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 22 06:17 dev
drwxr-xr-x 1 root root 4096 Jan 22 06:17 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 309 root root 0 Jan 22 06:17 proc
drwx----- 1 root root 4096 Jan 22 06:17 root
dr-xr-xr-x 11 root root 0 Jan 22 06:17 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 E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox echo "Hello ชี อและนามสกุล ของฉัน คคือ ภาษา fro
m busybox"
Hello ชี อและนามสกุล ของฉัน คคือ ภาษา from busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> |

```

11. ป้อนคำสั่ง \$ docker ps -a

```

PowerShell
/ # ls -la
total 48
drwxr-xr-x 1 root root 4096 Jan 22 06:17 .
drwxr-xr-x 1 root root 4096 Jan 22 06:17 ..
-rwxr-xr-x 1 root root 0 Jan 22 06:17 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 22 06:17 dev
drwxr-xr-x 1 root root 4096 Jan 22 06:17 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 309 root root 0 Jan 22 06:17 proc
drwx----- 1 root root 4096 Jan 22 06:17 root
dr-xr-xr-x 11 root root 0 Jan 22 06:17 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 E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker run busybox echo "Hello ชี อและนามสกุล ของฉัน คคือ ภาษา fro
m busybox"
Hello ชี อและนามสกุล ของฉัน คคือ ภาษา from busybox
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS          NAMES
32a859d68841   busybox   "echo 'Hello ชี อและ..." 19 seconds ago      Exited (0) 17 seconds ago          modest_
mcclintock
9963c983ab33   busybox   "sh"                    About a minute ago   Exited (0) 34 seconds ago          boring_be
nz
201560c3bc46   busybox   "sh"                    About a minute ago   Exited (0) About a minute ago          vibrant_m
axwell
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> |

```

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

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

_____ -it ใช้ในคำสั่ง docker run เพื่อเปิด terminal แบบ interactive

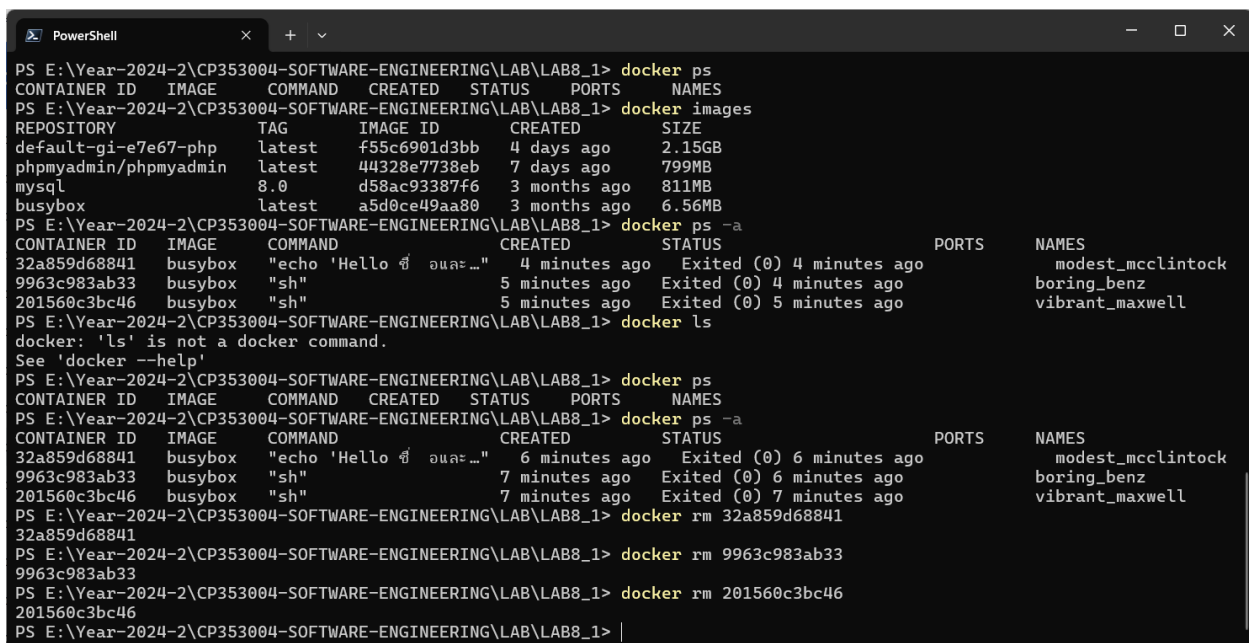
Lab Worksheet

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

_____ คอลัมน์ STATUS ใน docker ps -a แสดงสถานะของ container เช่น Up (กำลังทำงาน), Exited (หยุดทำงาน), หรือสถานะอื่นๆ พร้อมเวลา.

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

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



```

PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
default-gi-e7e67-php   latest   f55c6901d3bb   4 days ago   2.15GB
phpmyadmin/phpmyadmin   latest   44328e7738eb   7 days ago   799MB
mysql          8.0      d58ac93387f6   3 months ago   811MB
busybox        latest   a5d0ce49aa80   3 months ago   6.56MB
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
32a859d68841   busybox   "echo 'Hello ชี อและ..." 4 minutes ago   Exited (0) 4 minutes ago   modest_mcclintock
9963c983ab33   busybox   "sh"      5 minutes ago   Exited (0) 4 minutes ago   boring_benz
201560c3bc46   busybox   "sh"      5 minutes ago   Exited (0) 5 minutes ago   vibrant_maxwell
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
32a859d68841   busybox   "echo 'Hello ชี อและ..." 6 minutes ago   Exited (0) 6 minutes ago   modest_mcclintock
9963c983ab33   busybox   "sh"      7 minutes ago   Exited (0) 6 minutes ago   boring_benz
201560c3bc46   busybox   "sh"      7 minutes ago   Exited (0) 7 minutes ago   vibrant_maxwell
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker rm 32a859d68841
32a859d68841
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker rm 9963c983ab33
9963c983ab33
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_1> docker rm 201560c3bc46
201560c3bc46
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\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."

Lab Worksheet

CMD echo “ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น”

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

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

```
FROM busybox
```

```
CMD echo “Hi there. This is my first docker image.”
```

```
CMD echo “ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น”
```

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

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

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```
PowerShell
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_2> mv Dockerfile.swp Dockerfile
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_2> docker build -t my-app:1.0 .
[+] Building 4.5s (6/6) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 241B                             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
=> [1/1] FROM docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc82 4.3s
=> => resolve docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc82 4.3s
=> [auth] library/busybox:pull token for registry-1.docker.io 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:4c389074b438b22dd440a8b3b4c16da888e65ddd9d724be4cd7273f74454cc2 0.0s
=> => exporting config sha256:61da223f748c48f73f3174c6bee2da7145fc0b29c05af9c45c61b9f56db23808 0.0s
=> => exporting attestation manifest sha256:e9a37d58293179c23523692a70c149b7170c6a17876f231471947ae7ca4cd6e2 0.0s
=> => exporting manifest list sha256:3321c7bebf782eeb6f12afe69536fd642e34bd9da7698685624e3b4db638d213c 0.0s
=> => naming to docker.io/library/my-app:1.0                    0.0s
=> => unpacking to docker.io/library/my-app:1.0                 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 E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_2> |
```


Lab Worksheet

(1) คำสั่งที่ใช้ในการ run คือ

รัน Container จาก Docker Image

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

Option -t ในคำสั่ง docker build ใช้เพื่อกำหนด **ชื่อและแท็ก** ให้กับ Docker image ที่สร้างขึ้น เพื่อให้ง่ายต่อการระบุและใช้งานในภายหลัง. _____

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 และบันทึกคำสั่งต่อไปนี้

```
$ 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

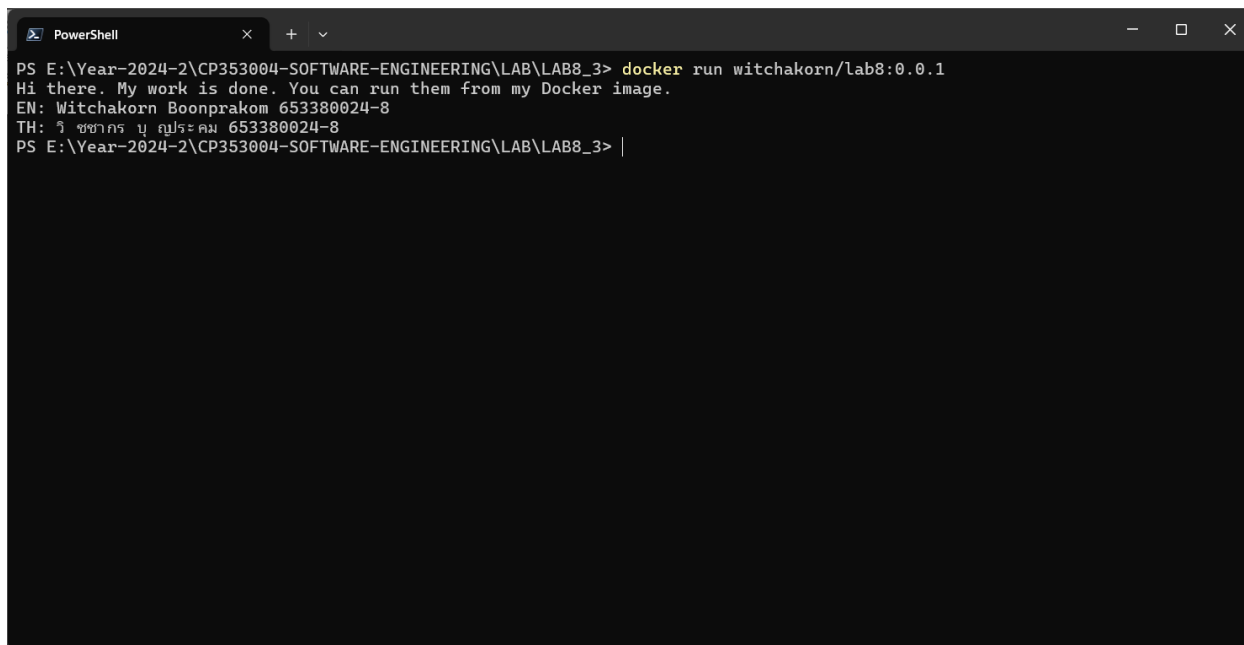
Lab Worksheet

\$ 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 E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_3> docker run witchakorn/lab8:0.0.1
Hi there. My work is done. You can run them from my Docker image.
EN: Witchakorn Boonprakom 653380024-8
TH: วิ ชชากร บุ ญประคณ 653380024-8
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_3> |

```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

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

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

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

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

Lab Worksheet

The screenshot shows a PowerShell terminal window at the top and a web browser window at the bottom. The terminal displays the following commands and output:

```
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_3> docker run witchakorn/lab8:0.0.1
Hi there. My work is done. You can run them from my Docker image.
EN: Witchakorn Boonprakom 653380024-8
TH: วิชิตกร บุญประคม 653380024-8
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_3> docker push witchakorn/lab8:0.0.1
The push refers to repository [docker.io/witchakorn/lab8]
e2bd77fdbd54: Pushed
9c0abc9c5bd3: Mounted from library/busybox
0.0.1: digest: sha256:9b6b575769363ce2b164cc81d798c6c27583fd11b6d130ccfcbbc9bb21bced28 size: 855
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_3> |
```

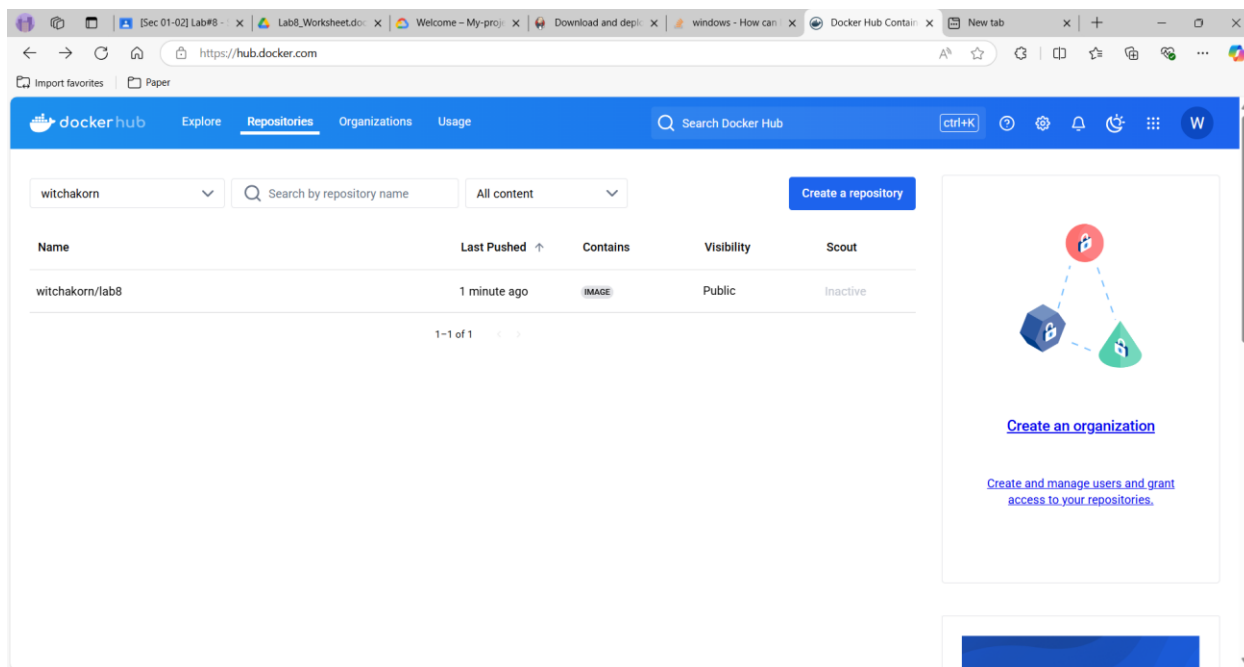
The web browser window shows the Docker Hub repository page for `witchakorn/lab8`. The page includes the following sections:

- General**: Shows the repository name `witchakorn/lab8`, last pushed 1 minute ago, and options to add a description and category (both marked as incomplete).
- Docker commands**: Provides a command to push a new tag: `docker push witchakorn/lab8:tagname`.
- Tags**: A table showing the repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
0.0.1		Image	a minute ago	a minute ago

Additional sections include **Automated builds** with instructions on connecting to GitHub or Bitbucket, and an **Upgrade** button.

Lab Worksheet



แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก GitHub repository
<https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง
`$ git clone https://github.com/docker/getting-started.git`
3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

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

Lab Worksheet

```

1  {
2    "name": "101-app",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "scripts": {
7      "prettify": "prettier -l --write '**/*.js'",
8      "test": "jest",
9      "dev": "nodemon src/index.js"
10   },
11   "dependencies": {
12     "express": "^4.18.2",
13     "mysql2": "^2.3.3",
14     "sqlite3": "^5.1.2",
15     "uuid": "^9.0.0",
16     "wait-port": "^1.0.4"
17   },
18   "resolutions": {
19     "ansi-regex": "5.0.1"
20   },
21   "prettier": {
22     "trailingComma": "all",
23     "tabWidth": 4,
24     "useTabs": false,
25     "semi": true,
26     "singleQuote": true
27   },
28   "devDependencies": {
29     "jest": "^29.3.1",
30     "nodemon": "^2.0.20",
31     "prettier": "^2.7.1"
32   }
33 }
34

```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงในไฟล์

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

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

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น

myapp_รหัสนศ. ไม่มีขีด

\$ docker build -t <myapp_รหัสนศ. ไม่มีขีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

Lab Worksheet

```

PowerShell
5 | COPY requirements.txt .
6 | >>> RUN pip install -r requirements.txt
7 |
8 | FROM --platform=$BUILDPLATFORM node:18-alpine AS app-base
-----
ERROR: failed to solve: process "/bin/sh -c pip install -r requirements.txt" did not complete successfully: exit code: 1
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> code .
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker build -t 6533800248 .\app\
[+] Building 15.3s (9/9) 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                1.1s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184 0.0s
=> [internal] load build context                                                  0.1s
=> => transferring context: 4.62MB                                                0.1s
=> CACHED [2/4] WORKDIR /app                                                     0.0s
=> [3/4] COPY . .                                                                0.0s
=> [4/4] RUN yarn install --production                                           10.2s
=> exporting to image                                                            3.7s
=> => exporting layers                                                            2.6s
=> => exporting manifest sha256:8a21f42a687c4b919245cb06aec743bc05af06a3961abeab2234cdd32ce5071f 0.0s
=> => exporting config sha256:33a3d6ac2ec47bffd6b6ba6734c545ef46ca5fd06d376f11a118b83e5ab5d8e6 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:6f85cca68effd51d1c8dd61451808935c4121ae7da93929e0fe1755670ca61b3 0.0s
=> => exporting attestation manifest sha256:50629b2e9eadd05adb367705650ac85e2e4869d4b9a2d0766deaa5fce874189e 0.0s
=> => exporting manifest list sha256:50629b2e9eadd05adb367705650ac85e2e4869d4b9a2d0766deaa5fce874189e 0.0s
=> => naming to docker.io/library/6533800248:latest                             0.0s
=> => unpacking to docker.io/library/6533800248:latest                           1.1s
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> |

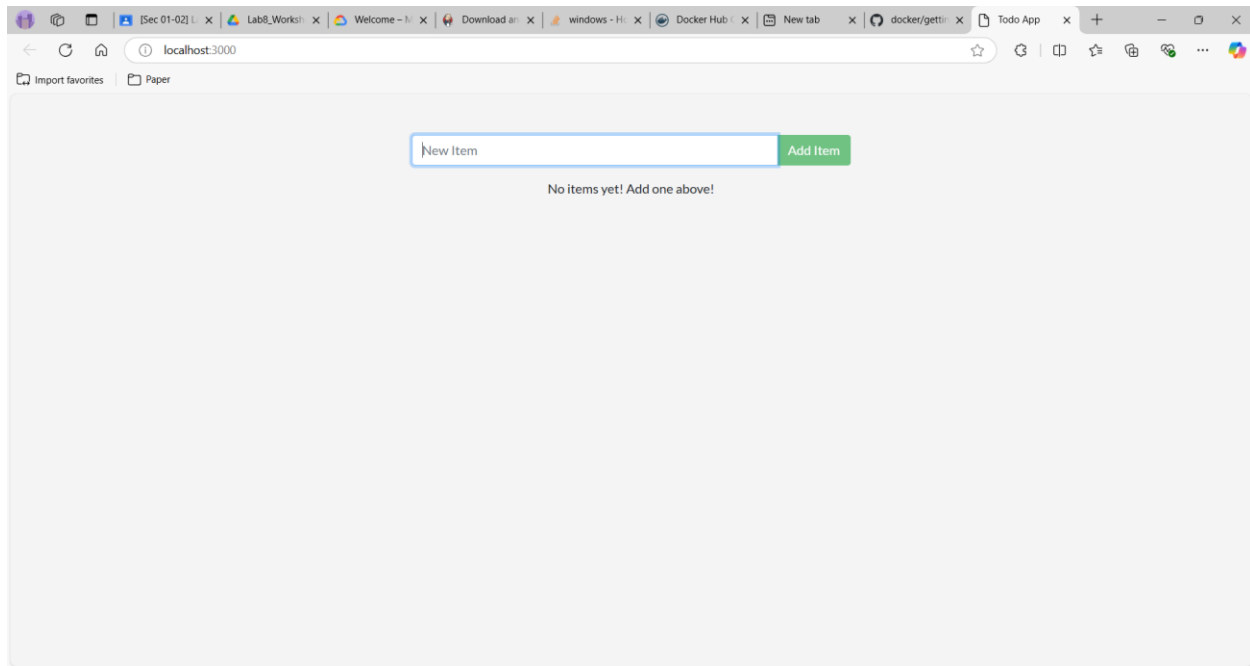
```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

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



Lab Worksheet

```

PowerShell
7 |
8 | FROM --platform=$BUILDPLATFORM node:18-alpine AS app-base
-----
ERROR: failed to solve: process "/bin/sh -c pip install -r requirements.txt" did not complete successfully: exit code: 1
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> code .
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker build -t 6533800248 .\app\
[+] Building 15.3s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 156B 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 1.1s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 4.62MB 0.1s
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY . . 0.0s
=> [4/4] RUN yarn install --production 10.2s
=> exporting to image 3.7s
=> => exporting layers 2.6s
=> => exporting manifest sha256:8a21f42a687c4b919245cb06aec743bc05af06a3961abeab2234cdd32ce5071f 0.0s
=> => exporting config sha256:33a3d6ac2ec47bffd6b6ba6734c545ef46ca5fd06d376f11a118b83e5ab5d8e6 0.0s
=> => exporting attestation manifest sha256:6f85cca68effd51d1c8dd61451808935c4121ae7da93929e0fe1755670ca61b3 0.0s
=> => exporting manifest list sha256:50629b2e9ead05adb367705650ac85e2e4869d4b9a2d0766deaa5fce874189e 0.0s
=> => naming to docker.io/library/6533800248:latest 0.0s
=> => unpacking to docker.io/library/6533800248:latest 1.1s
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
05cd0c097c3f14c0d6a567834eb80a691b10d1584487269473b2d4f93951d4f6
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> |

```

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

```

PowerShell
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
05cd0c097c3f14c0d6a567834eb80a691b10d1584487269473b2d4f93951d4f6
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker build -t 6533800248 .\app\
[+] Building 18.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 156B
=> [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:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184
=> => resolve docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184
=> [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
=> => exporting manifest sha256:fe7443335318d129c52d2c8e815792167f58820cdea805f6adc8bc05304b6f20
=> => exporting config sha256:531e8abcfbde39938a1e2d203bf721f77d6efbf4cb1a30ebe5fcb34ff42f0c
=> => exporting attestation manifest sha256:74e0e6d6ca64968d493d3ae3cf5d18ca002eb2d21091cdb64a9730806f5b415
=> => exporting manifest list sha256:b624302f6ae964f5b41a16a2f3628e5de2544ae85a37fdb6ef0660156bca7de
=> => naming to docker.io/library/6533800248:latest
=> => unpacking to docker.io/library/6533800248:latest
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
831ddfc1cb2cd562cf9be1264c7cb770e3a72bcaad19af2542f33dcce7b510c
docker: Error response from daemon: driver failed programming external connectivity on endpoint sharp_black (c1708416af8
46128084092860dfd8c9deb6dc027543c0cd42e11b2cec8cb01b): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> |

```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

หมายความว่า Port ของ host ในที่นี้หมายถึง 3000 มีการใช้งานอยู่ เลยไม่สามารถทำการสร้างได้ ถ้าเปิดเปลี่ยน port เป็นอย่างอื่น หรือ ปิด port 3000 ของเครื่องหลักลง ก็สามารถสร้างได้

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

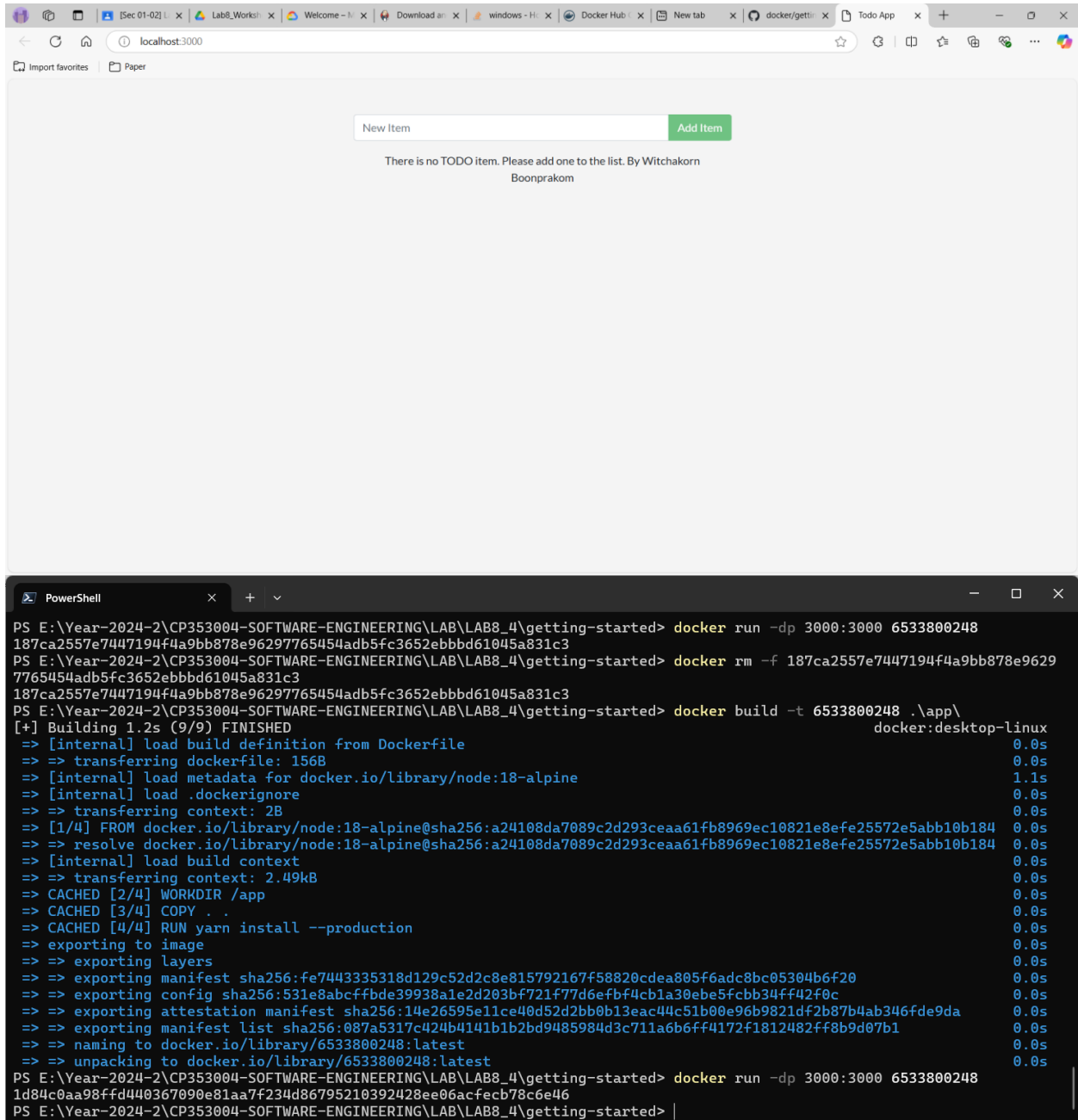
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

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



The screenshot shows a web browser window at localhost:3000 displaying a 'New Item' form with an 'Add Item' button. Below the form, a message states: 'There is no TODO item. Please add one to the list. By Witchakorn Boonprakom'.

Below the browser window, a PowerShell terminal window shows the following commands and output:

```
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
187ca2557e7447194f4a9bb878e96297765454adb5fc3652ebbbd61045a831c3
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker rm -f 187ca2557e7447194f4a9bb878e9629
7765454adb5fc3652ebbbd61045a831c3
187ca2557e7447194f4a9bb878e96297765454adb5fc3652ebbbd61045a831c3
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker build -t 6533800248 .\app\
[+] Building 1.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 156B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184
=> => resolve docker.io/library/node:18-alpine@sha256:a24108da7089c2d293ceaa61fb8969ec10821e8efe25572e5abb10b184
=> [internal] load build context
=> => transferring context: 2.49kB
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY .
=> CACHED [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:fe7443335318d129c52d2c8e815792167f58820cdea805f6adc8bc05304b6f20
=> => exporting config sha256:531e8abcfbbde39938a1e2d203bf721f77d6efbf4cb1a30ebe5fcb34ff42f0c
=> => exporting attestation manifest sha256:14e26595e11ce40d52d2bb0b13eac44c51b00e96b9821df2b87b4ab346fde9da
=> => exporting manifest list sha256:087a5317c424b4141b1b2bd9485984d3c711a6b6ff4172f1812482ff8b9d07b1
=> => naming to docker.io/library/6533800248:latest
=> => unpacking to docker.io/library/6533800248:latest
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
1d84c0aa98fdd40367090e81aa7f234d86795210392428ee06acfecb78c6e46
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> |
```

Lab Worksheet

```

PowerShell
=> => exporting attestation manifest sha256:74e0e6d6ca64968d493d3ae3cfff5d18ca002eb2d21091cdb64a9730806f5b415 0.0s
=> => exporting manifest list sha256:b624302f6ae964f5b41a16a2f3628e5de2544ae85a37fdba6ef0660156bca7de 0.0s
=> => naming to docker.io/library/6533800248:latest 0.0s
=> => unpacking to docker.io/library/6533800248:latest 1.1s
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
831ddfc1cb2cd562cf9be1264c7cb770e3a72bcaad19af2542f33dcce7b510c
docker: Error response from daemon: driver failed programming external connectivity on endpoint sharp_black (c1708416af8
46128084092860dfd8c9deb6dcd027543c0cd42e11b2cec8cb01b): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
05cd0c097c3f   50629b2e9ead   "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes  0.0.0.0:3000->3000/tcp   upbeat_ca
nnon
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker rm 05cd0c097c3f
Error response from daemon: cannot remove container "/upbeat_cannon": container is running: stop the container before re
moving or force remove
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker rm -rf 05cd0c097c3f
unknown shorthand flag: 'r' in -rf
See 'docker rm --help'.
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker rm -f 05cd0c097c3f
05cd0c097c3f
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker run -dp 3000:3000 6533800248
187ca2557e7447194f4a9bb878e9629776545adb5fc3652ebbbd61045a831c3
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker rm -f 187ca2557e7447194f4a9bb878e9629
776545adb5fc3652ebbbd61045a831c3
187ca2557e7447194f4a9bb878e9629776545adb5fc3652ebbbd61045a831c3
PS E:\Year-2024-2\CP353004-SOFTWARE-ENGINEERING\LAB\LAB8_4\getting-started> docker build -t 6533800248 .\app\
[+] Building 1.2s (9/9) 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 1.1s

```

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

```

PowerShell
2025-01-22 07:02:02.781+0000 [id=63] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-22 07:02:02.781+0000 [id=70] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-22 07:02:02.783+0000 [id=68] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-22 07:02:02.800+0000 [id=95] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-01-22 07:02:03.044+0000 [id=81] 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:

378088f5da0c4b7399d1c99c0fc5c1a1

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

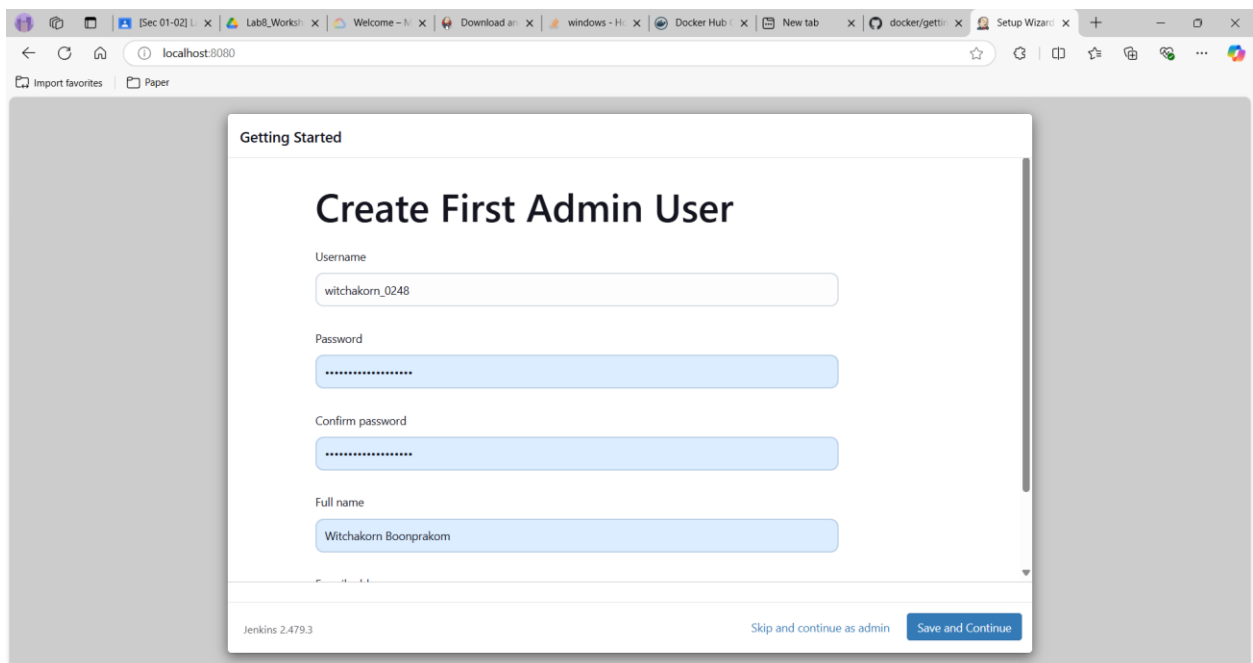
*****
*****
*****

2025-01-22 07:02:09.328+0000 [id=81] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-22 07:02:09.344+0000 [id=33] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-22 07:02:11.051+0000 [id=95] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2025-01-22 07:02:11.051+0000 [id=95] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1

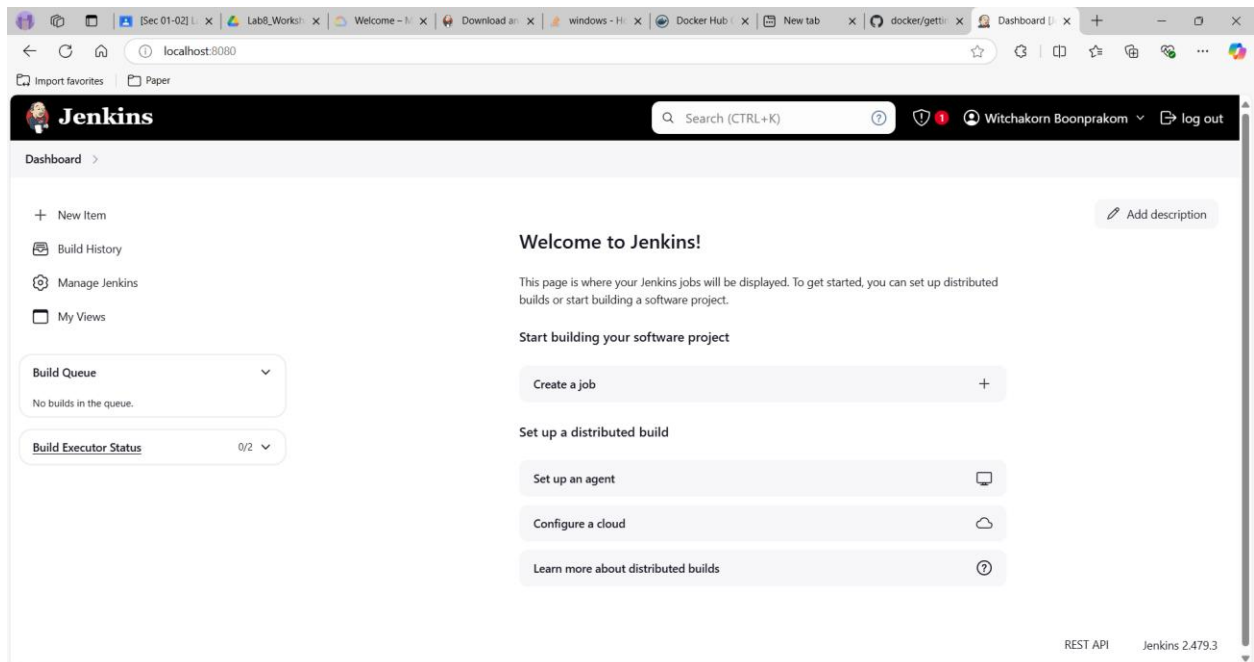
```

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

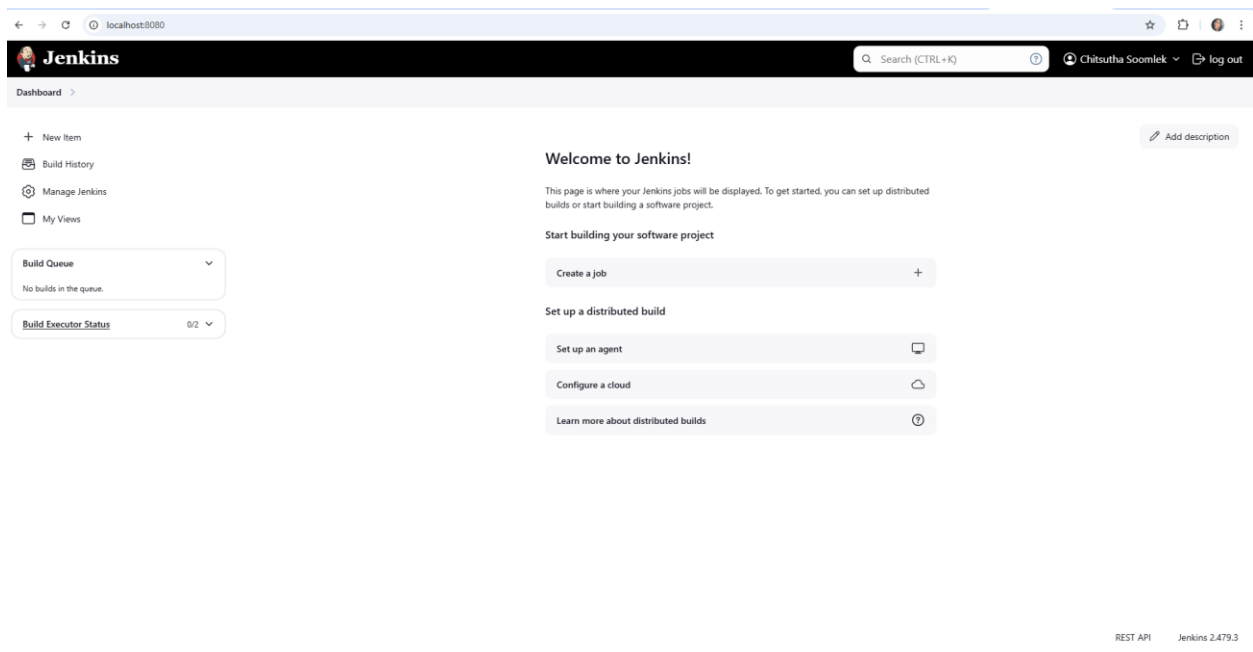


Lab Worksheet



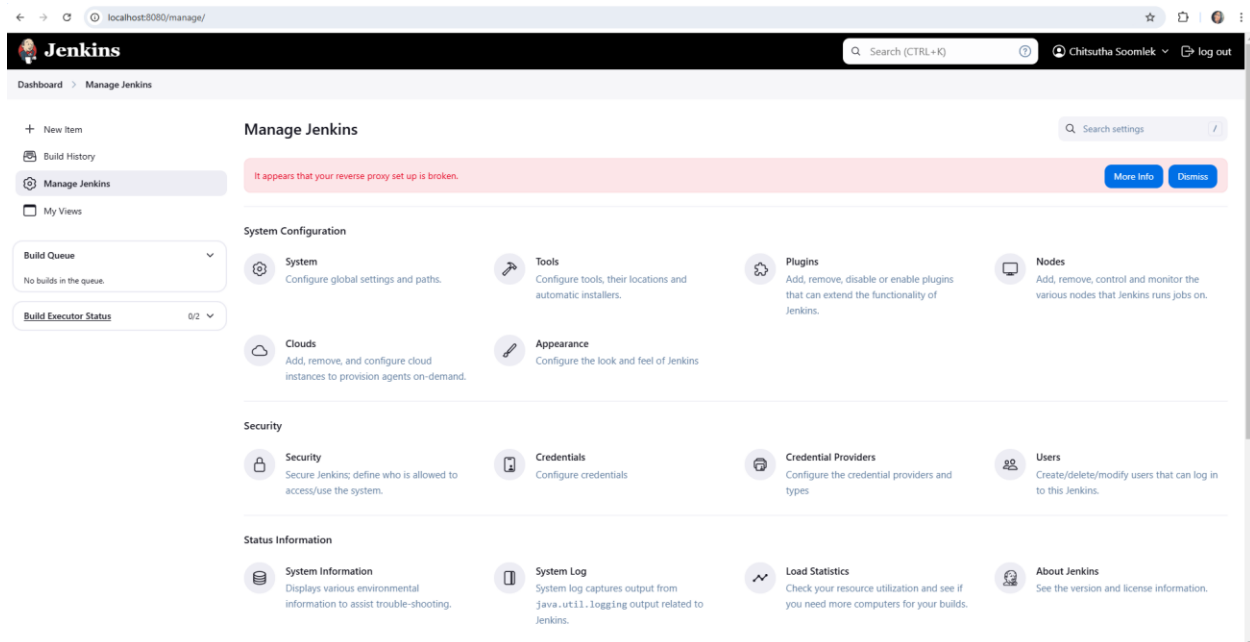
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

Lab Worksheet

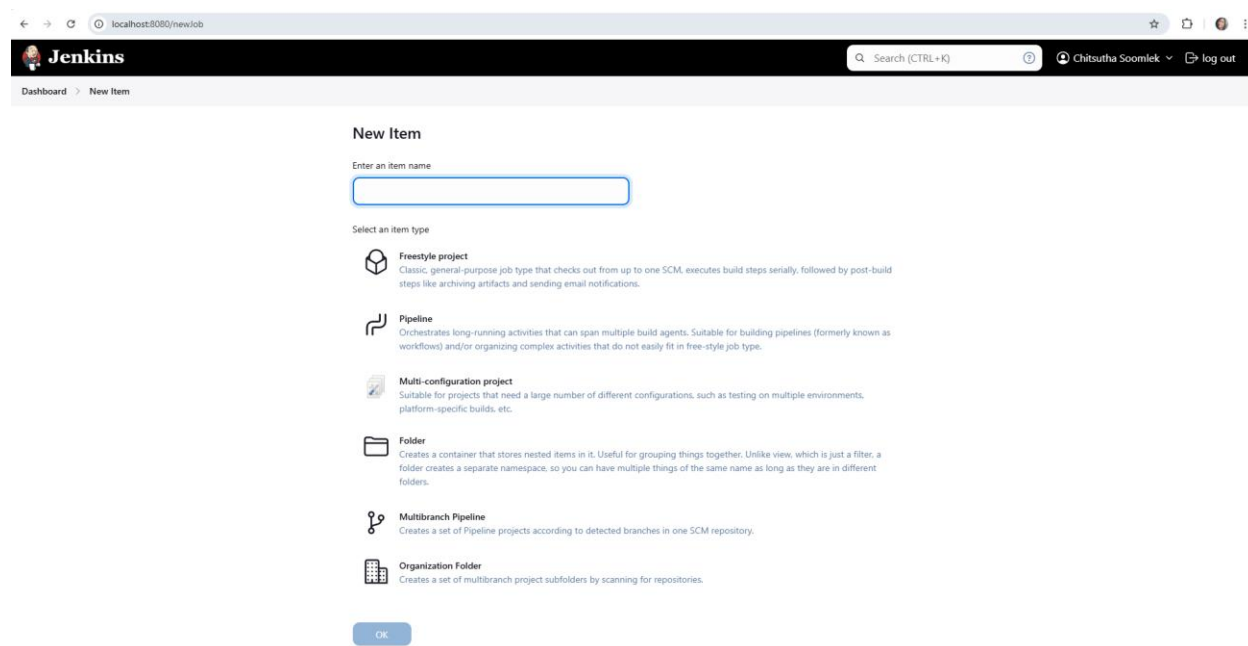


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

Lab Worksheet



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้นตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

Description: Lab 8.5

GitHub project: กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Lab Worksheet

The image displays two screenshots of the Jenkins configuration interface for a job named 'UAT'.

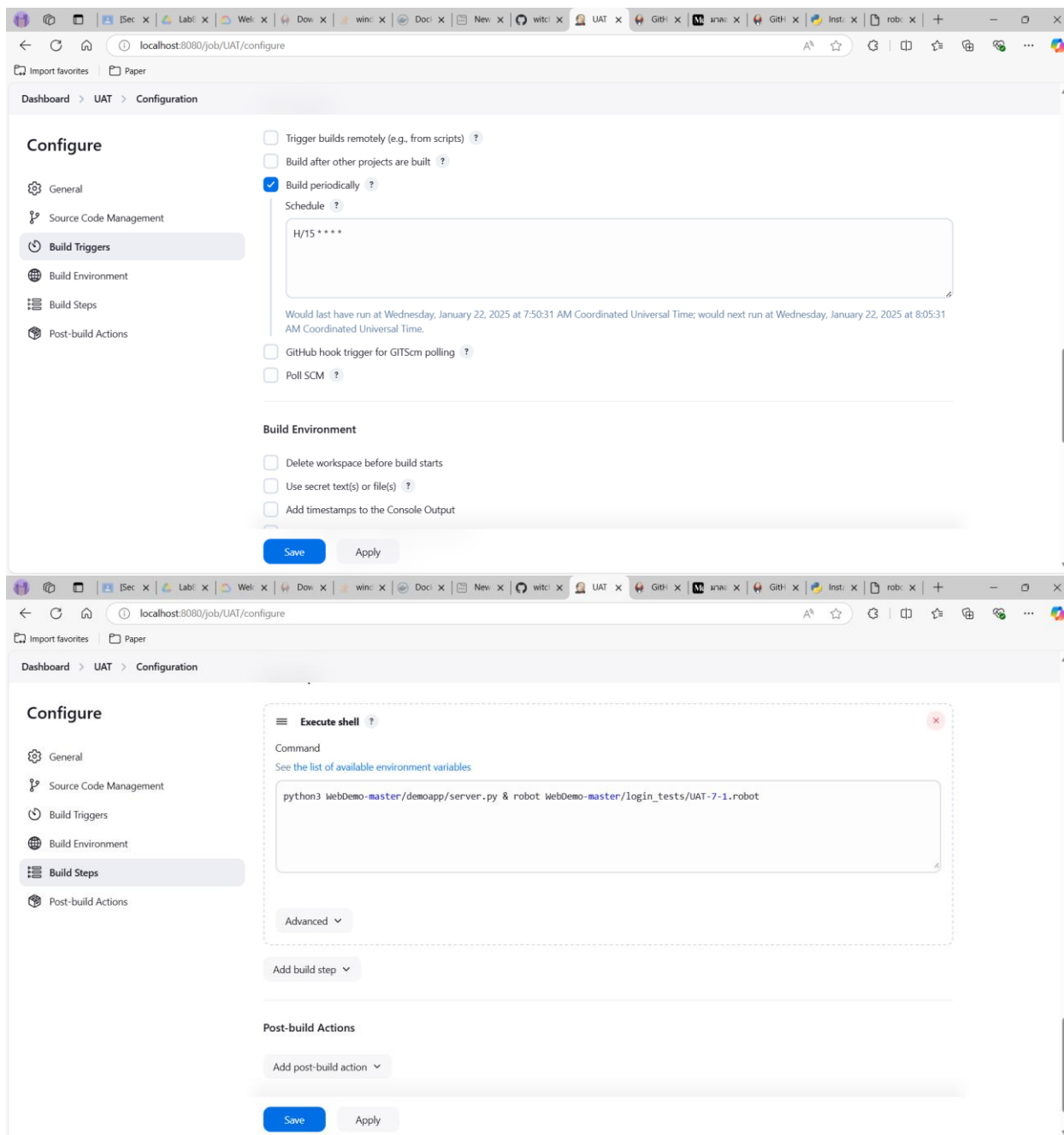
Top Screenshot: General Tab

- Configuration:** General (selected), Source Code Management, Build Triggers, Build Environment, Build Steps, Post-build Actions.
- General:** Enabled (checked).
- Description:** lab8.5
- Plain text:** [Preview](#)
- ☐ Discard old builds ?
- ☒ **GitHub project**
- Project url ?**
https://github.com/witchakornb/LAB-7-Test-Automation.git/
- Advanced** ▾
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- Buttons:** Save, Apply

Bottom Screenshot: Source Code Management Tab

- Configuration:** General, **Source Code Management** (selected), Build Triggers, Build Environment, Build Steps, Post-build Actions.
- Repositories ?**
- Repository URL ?**
https://github.com/witchakornb/LAB-7-Test-Automation.git
- Credentials ?**
- none -
- + Add**
- Advanced** ▾
- Add Repository**
- Branches to build ?**
- Branch Specifier (blank for 'any') ?**
*/main
- Buttons:** Save, Apply

Lab Worksheet



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

`python3 WebDemo-master/demoapp/server.py & robot WebDemo-master/login_tests/UAT-7-1.robot`

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

ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น

Lab Worksheet

% 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

Robot Framework Test Results

Executed: 2025-01-28T13:17:04.373760
Duration: 0:00:00.021 (+0:00:00.021)
Status: 2 critical test, 0 passed, 2 failed, 0 skipped
 2 test total (±0), 0 passed, 2 failed, 0 skipped
Results: [report.html](#), [log.html](#), [Original result files](#)

Test Result Trend

Number of test cases

Build

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

Duration Trend

Duration (ms)

Build

Show bigger image

Failed Test Cases

Name	Crit.	Duration	Age
UAT-7-1.1. Open Form	no	0:00:00.001	1
UAT-7-1.2. Record Success	no	0:00:00.001	1

Lab Worksheet

The screenshot shows a web browser window displaying the Jenkins console output for a job named 'UAT'. The browser's address bar shows 'localhost:8080/job/UAT/9/console'. The console output is as follows:

```

Commit message: "Remove obsolete XML output and console output files"
> git rev-list --no-walk e9925a86a2096afd2b1e5fb8c77db318d844fa80 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins32803154868867145.sh
+ robot WebDemo-master/login_tests/UAT-7-1.robot
+ python3 WebDemo-master/demoapp/server.py
[ ERROR ] Error in file '/var/jenkins_home/workspace/UAT/WebDemo-master/login_tests/UAT.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/python311.zip
  /usr/lib/python3.11
  /usr/lib/python3.11/lib-dynload
  /usr/local/lib/python3.11/dist-packages
  /usr/lib/python3/dist-packages
=====
UAT-7-1
=====
1. Open Form :: Opens the form page. | FAIL |
Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: ModuleNotFoundError: No
module named 'selenium'
-----
2. Record Success :: Fills out the contact information form with v... | FAIL |
No keyword with name 'Input Text' found.
-----
UAT-7-1 | 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
Process leaked file descriptors. See https://www.jenkins.io/redirect/troubleshooting/process-leaked-file-descriptors for more information
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

```

REST API Jenkins 2.479.3

Lab Worksheet

Dashboard > UAT > #9

Status ✖ #9 (Jan 28, 2025, 1:17:03 PM) ✎ Add description Keep this build forever

Changes

Console Output

Edit Build Information

Delete build '#9'

Timings

Git Build Data

Robot Results

Previous Build

Started by user [Witchakorn Boonprakom](#) Started 40 sec ago
Took **11 sec**

This run spent:

- 24 ms waiting;
- 11 sec build duration;
- 11 sec total from scheduled to completion.

git **Revision:** e9925a86a2096afd2b1e5fb8c77db318d844fa80
Repository: <https://github.com/witchakornb/LAB-7-Test-Automation.git>

- refs/remotes/origin/main

Robot Test Summary:

	Total	Failed	Passed	Skipped	Pass %
All tests	2	2	0	0	0.0

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

Changes No changes.

REST API Jenkins 2.479.3