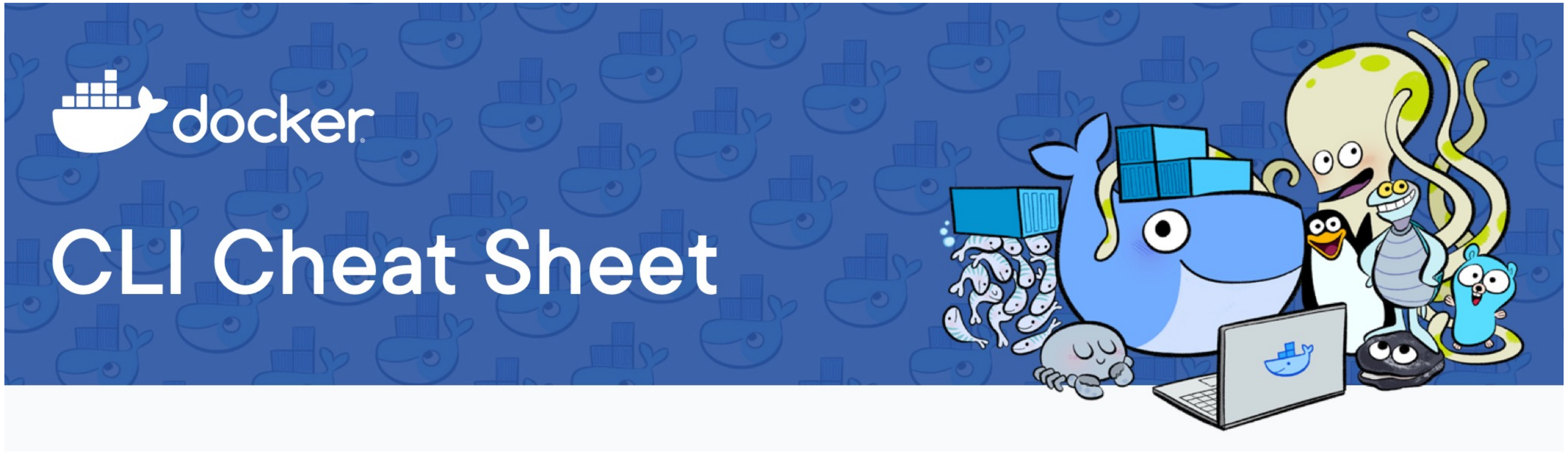
LAB-02
Docker CLI Basics

Olarn Rojanapornpun

Docker Cheat Sheet

- https://docs.docker.com/get-started/docker_cheatsheet.pdf



run command : primary process, --name

: create and run a new container from an image

```
docker container run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

- ❍ Runs a command in a new container as primary process (PID 1)
- ❍ If a command is not given, runs a command specified in the image
- ❍ When the primary process is terminated, the container is exited

```
# Run Nginx in background and set the container name to ‘ngx1’
docker run --name ngx1 -d nginx:alpine
docker exec ngx1 ps -ef
PID   USER     TIME  COMMAND
    1 root      0:00 nginx: master process nginx -g daemon off;
   30 nginx     0:00 nginx: worker process
   31 nginx     0:00 nginx: worker process
   32 root      0:00 ps -ef

# Run echo command in foreground and the container is exited
docker run --name ngx2 nginx:alpine echo hello world
hello world

# List the last created container
docker ps -n 1
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                                 PORTS     NAMES
23184390068f   nginx:alpine   "/docker-entrypoint.…"   1 second ago     Exited (0) Less than a second ago                ngx2
```

https://docs.docker.com/reference/cli/docker/container/run/

run command : -it; --rm; -d

- If the primary process is interactive, the option -it must be used, otherwise the process is terminated

```
# Run busybox without -it option
docker run busybox
docker ps -n 1
CONTAINER ID   IMAGE     COMMAND   CREATED            STATUS                       PORTS     NAMES
50ce2d947c0a   busybox   "sh"      24 seconds ago     Exited (0) 24 seconds ago              elegant_carver

# Run busybox image with -it option
docker run --name bbx1 -it busybox
/ # exit

# Run ps command and remove the container afterward
docker run --rm busybox ps -ef
PID   USER     TIME  COMMAND
    1 root      0:00 ps -ef

docker ps -n 1
CONTAINER ID   IMAGE     COMMAND   CREATED            STATUS                       PORTS     NAMES
44c62638c8cf   busybox   "sh"      18 seconds ago     Exited (0) 15 seconds ago              bbx1

# Run echo command in background, the text is shown in 'logs' instead
docker run --name bbx2 -d busybox echo hello world
docker logs bbx2
hello world
```

run command : publish port

```
docker run -p 127.0.0.1:80:80 nginx:alpine
```

- This binds port 80 of the container to TCP port 80 on 127.0.0.1 of the host
- If IP is not specified, it binds to all interfaces (0.0.0.0)
- Docker manages its own iptables rules (override firewall setting)

```
# Bind port 80 on container to port 3000 on host
docker run -d -p 3000:80 nginx:alpine

# Bind all exposed ports on container to random ports on host
docker run --name ngx3 -d -P nginx:alpine
docker ps -n 1
CONTAINER ID   IMAGE          COMMAND                   CREATED          STATUS          PORTS                       NAMES
9552eb2f6aa7   nginx:alpine   "/docker-entrypoint.…"    48 seconds ago   Up 47 seconds   0.0.0.0:32768->80/tcp,      ngx3
```

https://docs.docker.com/reference/cli/docker/container/run/#publish

run command : restart and env

can use TAB to find available restart policy

the container may restart indefinitely, leads to high CPU usage!

```
# Restart the container when daemon startup
docker run –name ngx4 -d --restart unless-stopped nginx:alpine
docker inspect --format='{{.HostConfig.RestartPolicy.Name}}' ngx4
unless-stopped
sudo systemctl restart docker
docker ps -n 1
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS         PORTS     NAMES
15be737b9adc   nginx:alpine   "/docker-entrypoint.…"   39 seconds ago   Up 2 seconds   80/tcp    ngx4

# Update restart policy on existing container
docker update --restart no ngx4
docker inspect --format='{{.HostConfig.RestartPolicy.Name}}' ngx4
no

# Set environment variables in the container. Can use variables exported in host.
# can use -e or --env
export firstname=Jack
docker run --rm -e firstname --env lastname busybox env | grep name
firstname=Jack

# The argument override the host variable
export firstname=Jack
docker run --rm -e firstname=Tim --env lastname=Russell busybox env | grep name
firstname=Tim
lastname=Russell
```

https://docs.docker.com/reference/cli/docker/container/run/#restart

ps command

: list containers

- alias of `docker container ls`
- -a, --all show all containers (default shows just running)
- -f, --filter filter output based on conditions provided
  e.g. {name, status, ancestor, publish, expose}

can use TAB to find available key and value

- -q, --quiet display container IDs only

```
# List all containers with the name ‘busybox’
docker ps -a --filter name=busybox

# List all containers with the image ‘ubuntu’ or has it as base
docker ps --filter ancestor=ubuntu

# List all containers that expose port 80
docker ps -f expose=80

# List all containers that publish port 80
docker ps -f publish=80
```

with expose and publish, only running and paused containers are listed

https://docs.docker.com/reference/cli/docker/container/ls/

ps command

Exercises

1. List alll exited containers
2. Delete all exited containers in one command [hint: use ‘$( )’]

exec command

: execute a command in a running container

- need `-it` when execute interactive command
- may use `sh -c` to execute chained commands
- may use `-d -e` as `run`

```
# Run busybox in background with the name ‘bbx3’
docker run --name bbx3 -it -d busybox

# Execute ‘cat’ in bbx3 container. Without -it, cat process is terminated immediately.
docker exec bbx3 cat

# Execute ‘cat’ in bbx3 container. With -it, cat works as expected.
docker exec -it bbx3 cat
hello
hello

# Execute a chained commands, only the first command is executed in container
docker exec bbx3 hostname -i && hostname -i
172.17.0.4
127.0.1.1

# Execute a chained commands with sh -c,
docker exec bbx3 sh -c "hostname -i && hostname -i"
172.17.0.4
172.17.0.4
```

https://docs.docker.com/reference/cli/docker/container/exec/

history command

: show the history of an image

- alias of `docker image history`

```
# View image history
docker history nginx:alpine

IMAGE          CREATED         CREATED BY                                      SIZE
COMMENT
02ffd439b71d   11 months ago   /bin/sh -c set -x     && apkArch="$(cat /etc…   30.8MB
<missing>      11 months ago   /bin/sh -c #(nop)  ENV NJS_VERSION=0.7.11       0B
<missing>      11 months ago   /bin/sh -c #(nop)  CMD ["nginx" "-g" "daemon…   0B
<missing>      11 months ago   /bin/sh -c #(nop)  STOPSIGNAL SIGQUIT           0B
<missing>      11 months ago   /bin/sh -c #(nop)  EXPOSE 80                    0B
<missing>      11 months ago   /bin/sh -c #(nop)  ENTRYPOINT ["/docker-entr…   0B
<missing>      11 months ago   /bin/sh -c #(nop) COPY file:e57eef017a414ca7…   16.4kB
<missing>      11 months ago   /bin/sh -c #(nop) COPY file:abbcbf84dc17ee44…   12.3kB
<missing>      11 months ago   /bin/sh -c #(nop) COPY file:5c18272734349488…   12.3kB
<missing>      11 months ago   /bin/sh -c #(nop) COPY file:7b307b62e82255f0…   8.19kB
<missing>      11 months ago   /bin/sh -c set -x     && addgroup -g 101 -S …   6.85MB
<missing>      11 months ago   /bin/sh -c #(nop)  ENV PKG_RELEASE=1            0B
<missing>      11 months ago   /bin/sh -c #(nop)  ENV NGINX_VERSION=1.23.4     0B
<missing>      11 months ago   /bin/sh -c #(nop)  LABEL maintainer=NGINX Do…   0B
<missing>      11 months ago   /bin/sh -c #(nop)  CMD ["/bin/sh"]              0B
<missing>      11 months ago   /bin/sh -c #(nop) ADD file:e51d4089e73ad6dee…   8.14MB
```

https://docs.docker.com/reference/cli/docker/image/history/

history command

```
# View busybox image history
docker history busybox
IMAGE          CREATED         CREATED BY                             SIZE      COMMENT
650fd573e056   10 months ago   BusyBox 1.36.1 (glibc), Debian 12      4.17MB

# Cannot run busybox without a command
docker run --name busybox -d --rm busybox
docker ps -a -f name=busybox
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

# May run busybox with sh
# NB: need -it option
docker run --name busybox -d --rm -it busybox sh
docker ps -a -f name=busybox
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
4e09b507fe92   busybox   "sh"      6 seconds ago   Up 5 seconds          busybox
```

busybox has no default command!

cp command

: copy files/folders between a container and the local filesystem

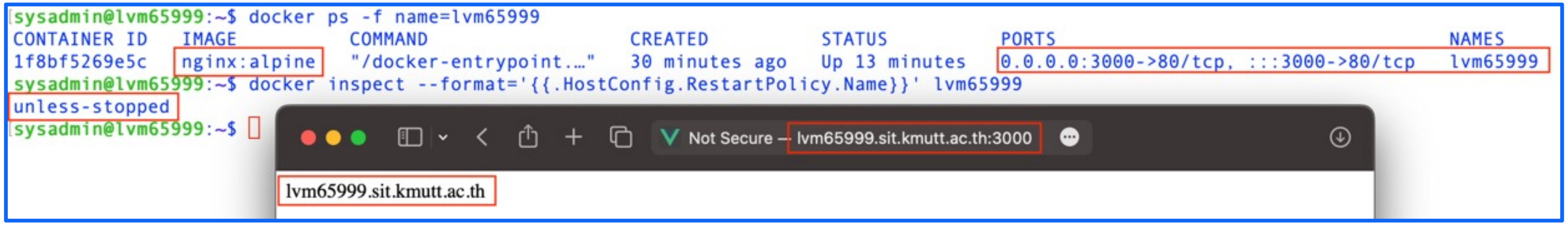
- container paths are relative to the container's root directory

```
docker run --name busybox -d --rm -it busybox sh
echo hello world > hello

# Copy file ‘hello’ to ‘busybox’ container at /tmp
docker cp hello busybox:tmp
                                        Successfully copied 2.05kB to busybox:tmp

docker exec busybox cat /tmp/hello
hello world

# Copy file ‘index.html’ in container to host
docker cp nginx:usr/share/nginx/html/index.html .
```

https://docs.docker.com/reference/cli/docker/container/cp/

Exercise 1: deploy lvmxxyyy in container

- Create and run a container to host 'lvmxxyyy'
  - Base-image: nginx:alpine
  - Name: lvm[xxyyy]
  - Restart policy: unless-stopped
  - port: 0.0.0.0:3000
  - web-page: lvmxxyyy
    - content: lvmxxyyy.sit.kmutt.ac.th



```
sysadmin@lvm65999:~$ docker ps -f name=lvm65999
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                    NAMES
1f8bf5269e5c   nginx:alpine   "/docker-entrypoint.…"   30 minutes ago   Up 13 minutes   0.0.0.0:3000->80/tcp, :::3000->80/tcp   lvm65999
sysadmin@lvm65999:~$ docker inspect --format='{{.HostConfig.RestartPolicy.Name}}' lvm65999
unless-stopped
sysadmin@lvm65999:~$
```

Not Secure — lvm65999.sit.kmutt.ac.th:3000

lvm65999.sit.kmutt.ac.th

commit command

: create a new image from a container's changes

```
# commit the changes to the image 'nginx:test'
docker commit nginx nginx:test

# List all images with the image name 'nginx'
docker images nginx

# View history of nginx:test image
docker history nginx:test
```

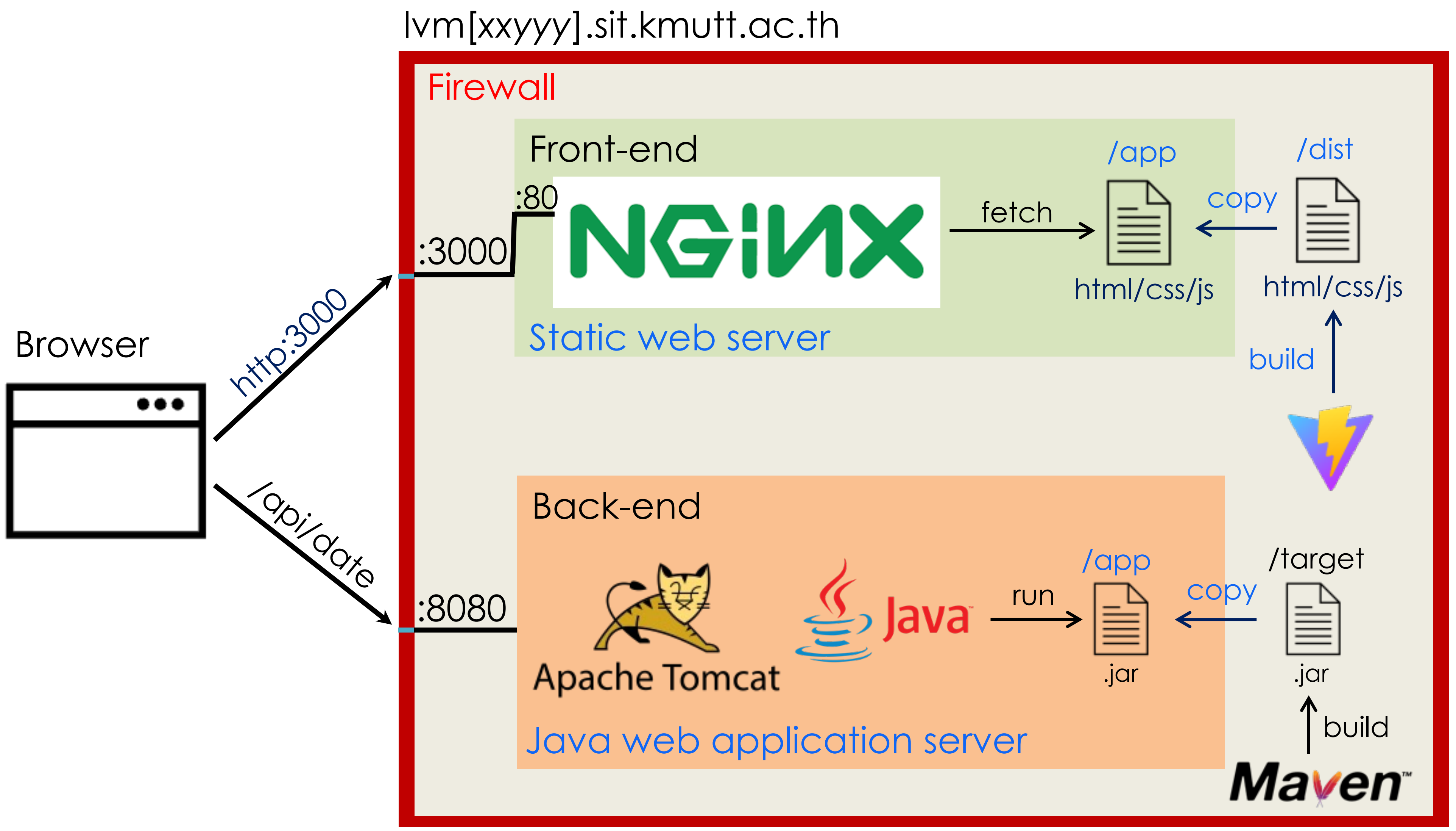
https://docs.docker.com/reference/cli/docker/container/commit/

commit command

Exercise

1. commit lvmxxyyy to ‘lvmxxyyy:bare’

Exercise 2: fetchdate/dateapi in container



dateapi container

- Create and run a container to host ‘dateapi’
  - Base-image: openjdk:17-jdk-alpine
  - Name: dateapi
  - Restart policy: unless-stopped
  - port: 0.0.0.0:8080

fetchdate container

- Create and run a container to host ‘dateapi’
  - Base-image: nginx:alpine
  - Name: fetchdate
  - Restart policy: unless-stopped
  - port: 0.0.0.0:3000

Bind mounts with --mount flag

- With bind mount, a file or directory on the host machine is mounted into a container.
- The file or directory is referenced by its absolute path on the host machine.

```
# bind mount current directory to /src
docker run -it --rm --mount type=bind,source=$(pwd),target=/src busybox
# run the following commands in container
cd /src; ls
touch newfile
exit

# newfile is listed with ‘root’ ownership
ls -l
```

https://docs.docker.com/storage/bind-mounts/

Exercise 3

- Redo Exercise 2 with bind mount