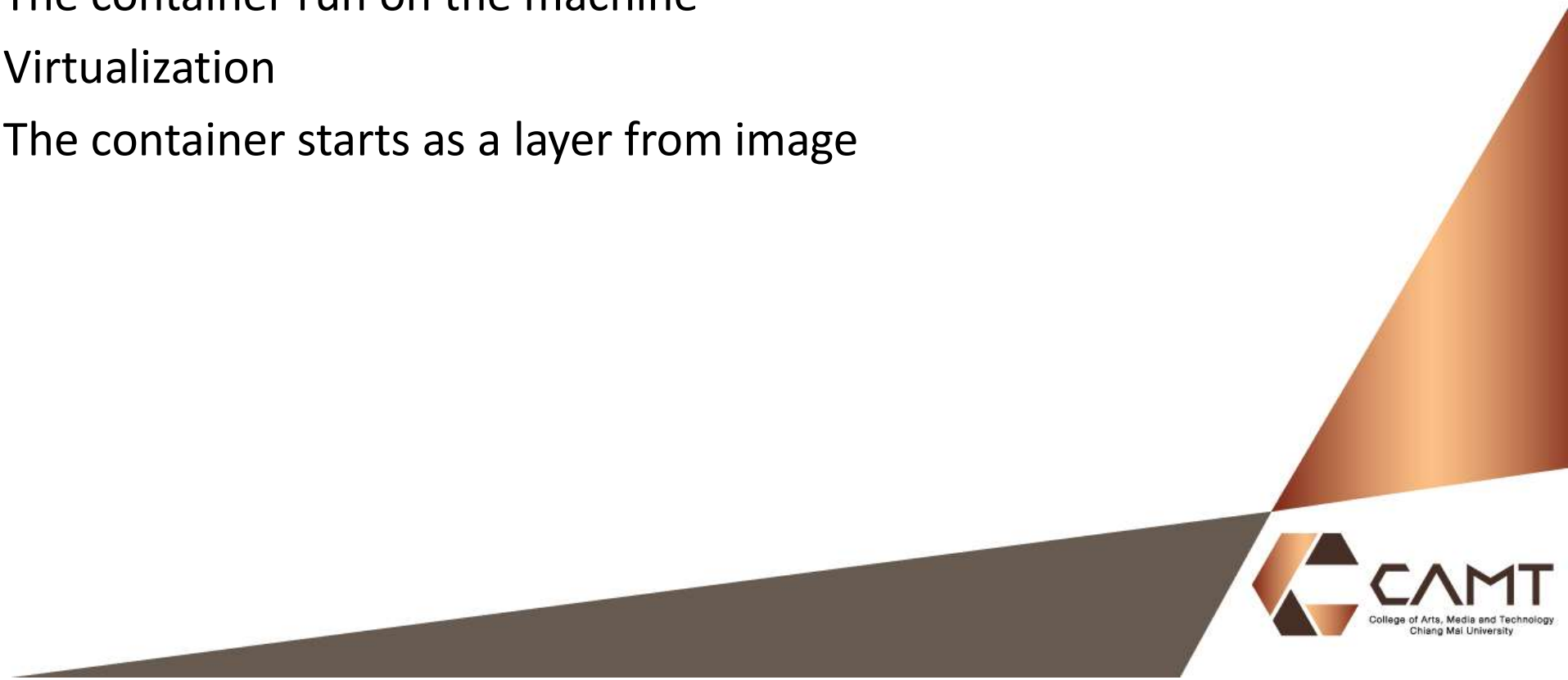
# Container

SE234 Advance Software Development

# Docker

- The container run on the machine
- Virtualization
- The container starts as a layer from image



# Creating an images

- What are these images for?
  - Alpine
  - Nginx
  - mysql

# Customize image

- Put a new layer to the images
- Make the images which can be run in difference computer
  - With the same configuration
- Can be deploy to many computers
  - To run the same set of application

# Images (docker images)

- Pack of deployment application
  - With the operating system
- Set of
  - Web server and html files
  - Application container with its deployment jar file

```
FROM nginx:1.13

COPY ./html/  /usr/share/
EXPOSE 80
```

# Dockerfile example

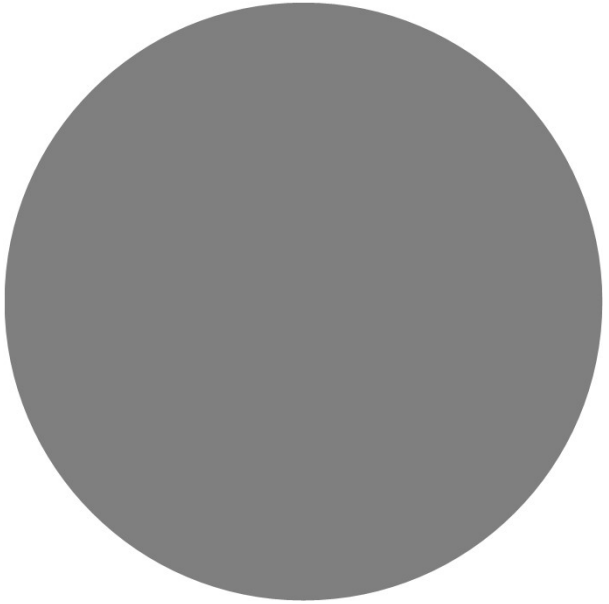
```
FROM nginx:1.13

COPY ./dist /usr/share/nginx/html

COPY ./nginx-custom.conf /etc/nginx/conf.d/default.conf
EXPOSE 80
```

# Dockerfile Example

```
FROM openjdk:8-jdk-alpine
ARG JAR_FILE
COPY ${JAR_FILE} app.jar
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom",
        "-Dspring.profiles.active=dev-server","-jar","app.jar"]
```

# Dockerfile

# Building the images



docker build .

Run at the Dockerfile location
Default Dockerfile will be loaded

```
FROM  nginx:alpine
ADD ./ssl /etc/nginx/certs
ADD ./conf.d /etc/nginx/conf.d
ADD ./password/.htpasswd /etc/nginx/.htpasswd
```

```
λ docker build .
Sending build context to Docker daemon  505.3kB
Step 1/4 : FROM  nginx:alpine
alpine: Pulling from library/nginx
4167d3e14976: Pull complete
bb292c78f105: Pull complete
Digest: sha256:abe5ce652eb78d9c793df34453fddde12bb4d93d9fbf2c363d0992726e4d2cad
Status: Downloaded newer image for nginx:alpine
 ---> 377c0837328f
Step 2/4 : ADD ./ssl /etc/nginx/certs
 ---> 7f45ba6598c0
Step 3/4 : ADD ./conf.d /etc/nginx/conf.d
 ---> bf142d5e878c
Step 4/4 : ADD ./password/.htpasswd /etc/nginx/.htpasswd
 ---> 5fa3ffd6e452
Successfully built 5fa3ffd6e452
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and director
s for sensitive files and directories.
```

```
λ docker images
REPOSITORY                                  TAG                 IMAGE ID            CREATED             SIZE
<none>                                      <none>              5fa3ffd6e452        17 seconds ago      19.7MB
```

- -t “image name”
- Recommended to be [your dockerhub username]/[any name]

# Prevent default name

```
λ docker build -t dto80/reverse_proxy .
Sending build context to Docker daemon  505.3kB
Step 1/4 : FROM  nginx:alpine
 ---> 377c0837328f
Step 2/4 : ADD ./ssl /etc/nginx/certs
 ---> Using cache
 ---> 7f45ba6598c0
Step 3/4 : ADD ./conf.d /etc/nginx/conf.d
 ---> Using cache
 ---> bf142d5e878c
Step 4/4 : ADD ./password/.htpasswd /etc/nginx/.htpasswd
 ---> Using cache
 ---> 5fa3ffd6e452
Successfully built 5fa3ffd6e452
Successfully tagged dto80/reverse_proxy:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host
s for sensitive files and directories.
```

```
λ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
dto80/reverse_proxy                 latest              5fa3ffd6e452        About a minute ago  19.7MB
```

# Push it to the repo

- So the other computer can reach to your repository

```
before_install:
- docker login -u "$DOCKER_USERNAME" -p "$DOCKER_PASSWORD"
```

- Push the image you just build

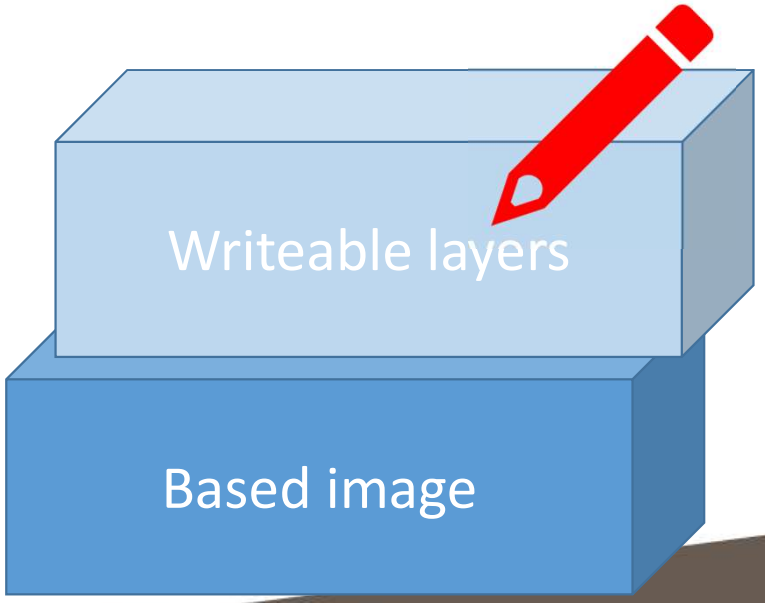
```
- docker push $DOCKER_USERNAME/htmltest
```


https://hub.docker.com/r/dto80/htmltest/

# And then …

# Running the container

- Add a new writable layer
- Maintain all the non-persistence data

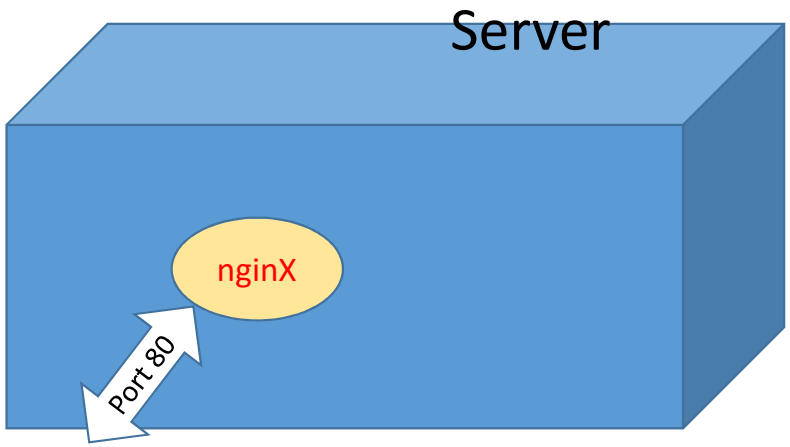


# Docker Run

- Docker run [name of image]
- Change for the local image name
- If do not exist
  - Find in the registry
  - If there are in the registry
    - Download to the local computer
    - Create a new layer
    - Run
- Otherwise
  - failed

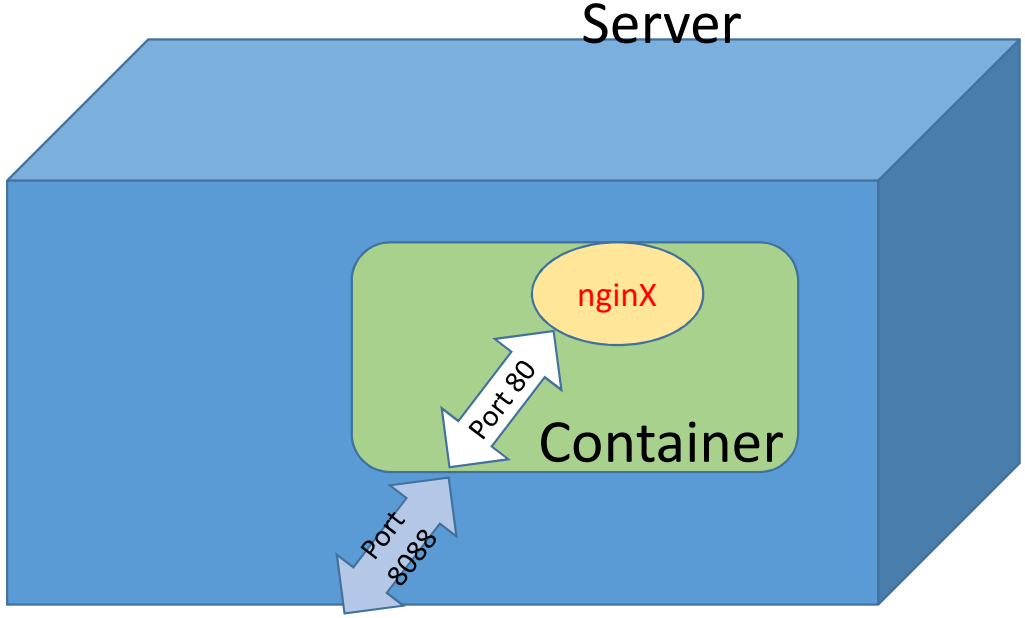
# Do the configuration

- Link to outside
- Other computer can access the component



# Do the Configuration

- docker run -p 8088:80 $DOCKER_USERNAME/htmltest:latest




CAMT
College of Arts, Media and Technology
Chiang Mai University

# Docker process

- Stick with the console
- When Ctrl-C is pressed
- The process is stop
- Use the –detached mode
  - To run the app even we close the console
- docker run -p 8088:80 -d $DOCKER_USERNAME/htmltest:latest

# Docker ps

- To check that how many container is running

```
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS                    NAMES
fc7e620d7091        dto80/htmltest:latest                "nginx -g 'daemon of…"   8 seconds ago       Up 7 seconds        80/tcp                   frosty_newt
e63be9b86bf2        dto80/htmltest:latest                "nginx -g 'daemon of…"   28 hours ago        Up 28 hours         0.0.0.0:8088->80/tcp     htmltest
4a74bf5cd52f        dto80/se234-lab10-backend:latest     "java -Djava.securit…"   3 days ago          Up 3 days           0.0.0.0:8082->8080/tcp   se234lab10b
_backend_1
ec0d3a5a4ea1        dto80/se234-lab10-client:latest      "nginx -g 'daemon of…"   3 days ago          Up 3 days           0.0.0.0:8081->80/tcp     se234lab10b
_frontend_1
ubuntu@ip-172-31-13-145:~$
```

```
NAMES
frosty_newton
htmltest
se234lab10backend_backend_1
se234lab10backend_frontend_
```

# Container name

- Generate automatically
  - If not specified
- Hard to handle automatically
  - As we do not know what it will be

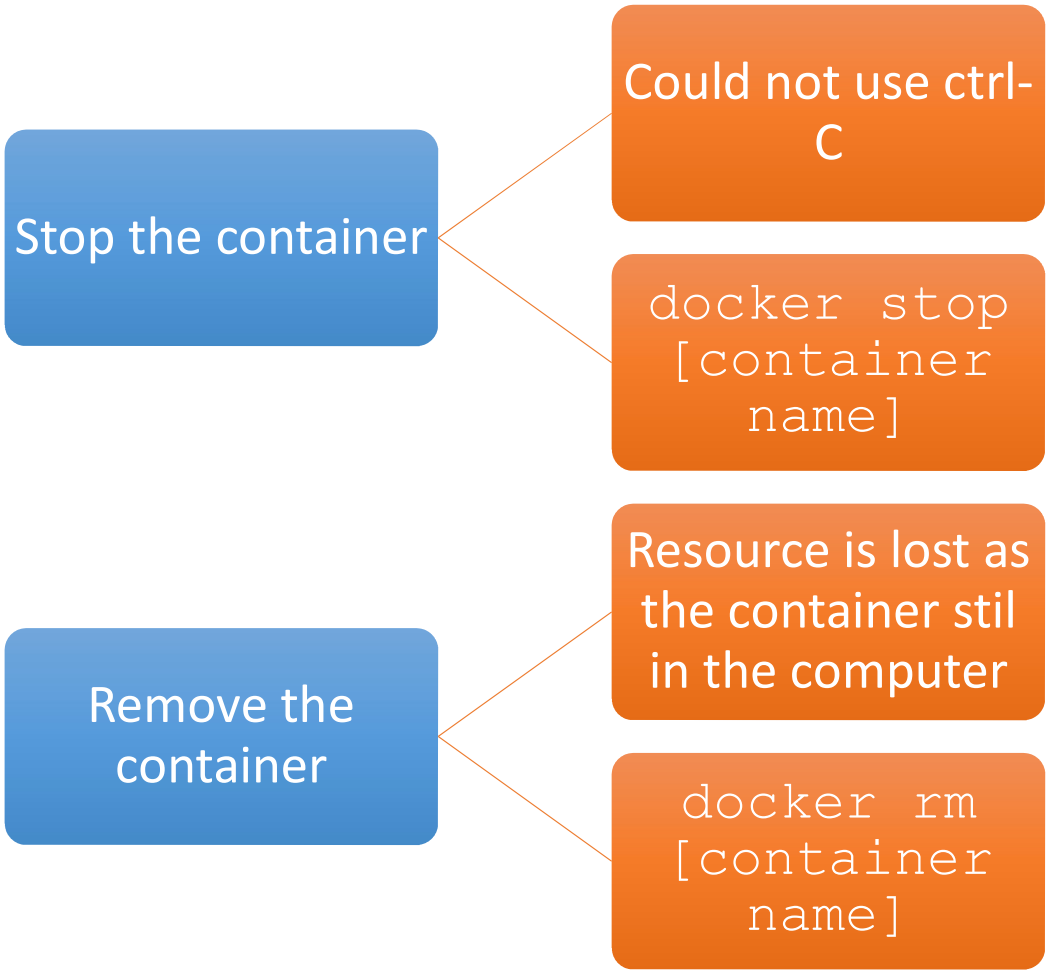
# Docker run --name

- Define the name to be used
- Reference later

```
λ docker run --name aloha dto80/reverse_proxy
```

```
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS               NAMES
573b51616c5e        dto80/reverse_proxy    "nginx -g 'daemon of…"   25 seconds ago      Up 24 seconds       80/tcp              aloha

C:\Users\Dto
```

# As in detached mode



# To remove the container

- docker stop [container name]
- Stop the usage of the docker
  - Can restart, the rewriteable layer is still there
- Container name
  - Name
  - ID or part of ID
    - A hash number

```
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS                    NAMES
fc7e620d7091        dto80/htmltest:latest                "nginx -g 'daemon of…"   8 seconds ago       Up 7 seconds        80/tcp                   frosty_newt
e63be9b86bf2        dto80/htmltest:latest                "nginx -g 'daemon of…"   28 hours ago        Up 28 hours         0.0.0.0:8088->80/tcp     htmltest
4a74bf5cd52f        dto80/se234-lab10-backend:latest     "java -Djava.securit…"   3 days ago          Up 3 days           0.0.0.0:8082->8080/tcp   se234lab10b
_backend_1
ec0d3a5a4ea1        dto80/se234-lab10-client:latest      "nginx -g 'daemon of…"   3 days ago          Up 3 days           0.0.0.0:8081->80/tcp     se234lab10b
_frontend_1
ubuntu@ip-172-31-13-145:~$
```

# Remove container

- When stop
  - The port is still acquired
    - That why you got an error the port is still occupied by some process
- To remove
  - Use command: docker rm [container name or id]

# Docker Machine

- Call the docker command to run on the other computer
- The Target computer should container the docker machine daemon
- The target computer should open the port for the docker command

```
ubuntu@ip-172-31-13-145:~$  sudo ps aux |grep dockerd
ubuntu   15256  0.0  0.0  12944  1012 pts/0    S+   07:33   0:00 grep --color=auto dockerd
root     16155  0.0  4.1 574148 42304 ?        Ssl  Mar20   7:04 /usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2379
```

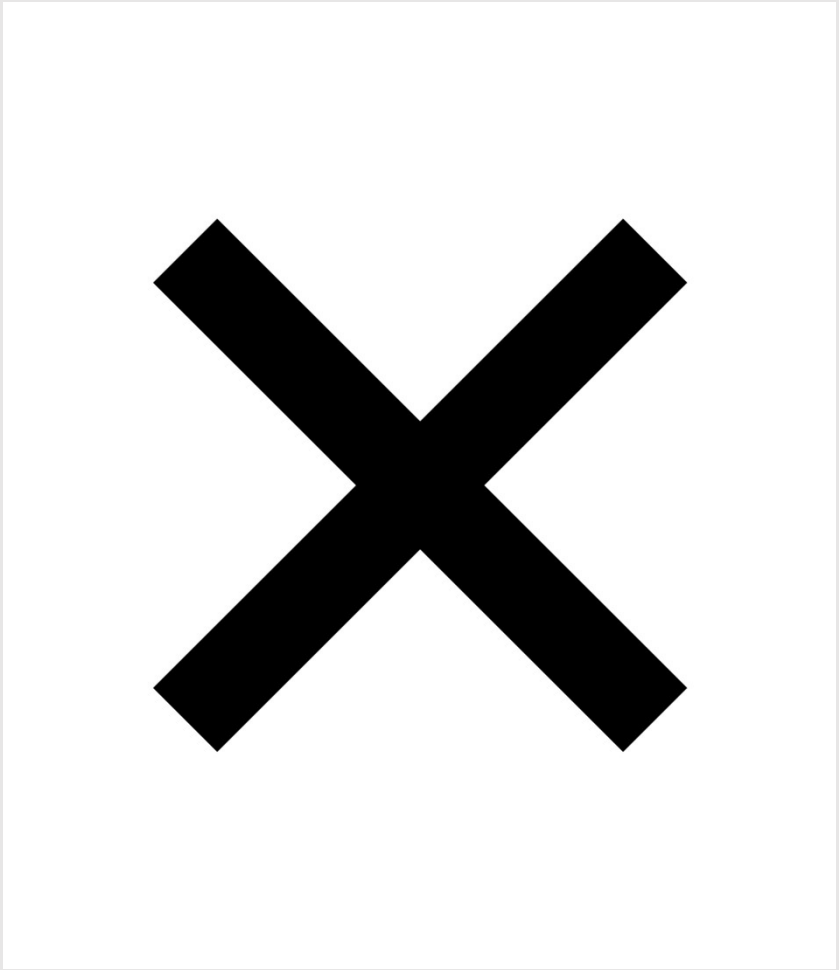
- The DOCKER_HOST variable is defined
  - Use tcp protocol with the ip:port
- More secure technique, check the Docker document

# Multi container application

- The application could not stand alone
- How it can be mapped?

# The application now is separated

- To use difference Third party
- Easy to maintenance
- Focus on the real interest

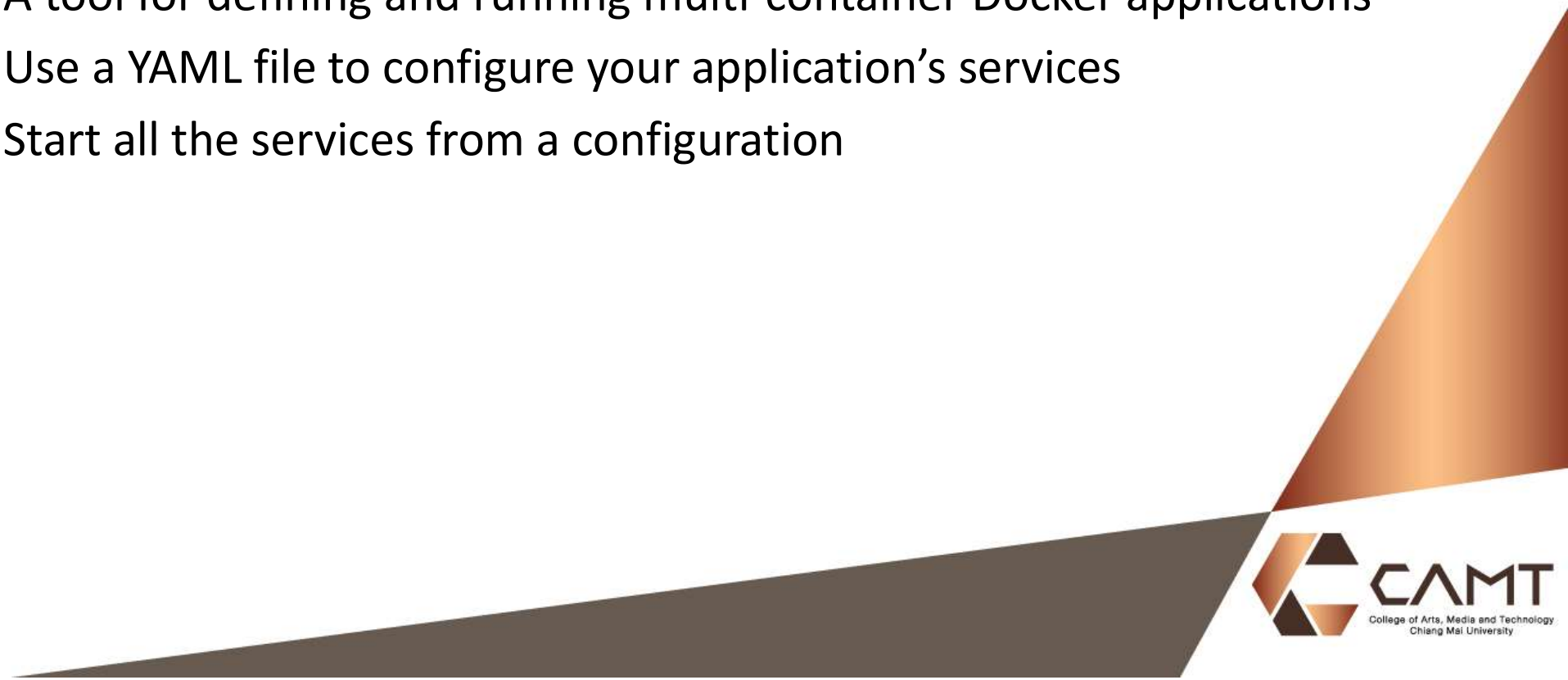


# Using all the components in Docker

Use docker run to run each component manually

# docker-compose

- A tool for defining and running multi-container Docker applications
- Use a YAML file to configure your application’s services
- Start all the services from a configuration



# Basic Configuration file

docker-compose.yml

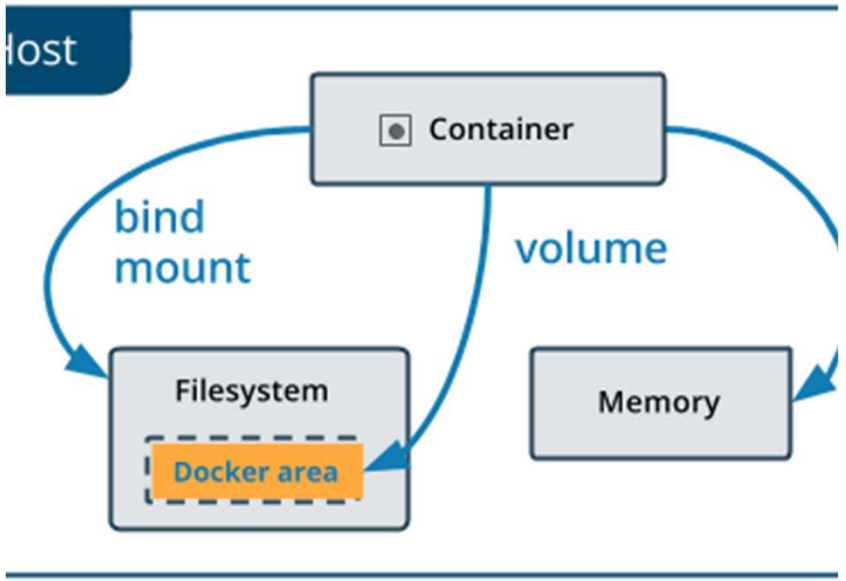
```yaml
version: '3.3'
services:
  backend:
    image: dto80/se234-lab10-backend:latest
    ports:
      - "8082:8080"
  frontend:
    image: dto80/se234-lab10-client:latest
    ports:
      - "8081:80"
```

# Docker compose

- To start both containers
  - `docker-compose up -d`
- To stop both container
  - `docker-compose down`

# Volume mapping

- The rewriteable layer will be removed when we remove the container
- If you used the database application?
- To link the persistence in the container to the hard disk
- The data still there even the container is removed
  - So when start the container again, the data is still in used



# Referred to volume

- Specify directly
  - Using the absolute location of the docker host computer
    /home/ubuntu/name
- Use the docker volume created
  - Docker create the reachable volume in the Docker location
  - Easy to access, easy to reuse the configuration

```
docker volume create my-vol
```

```
volumes:
  mongo-client-volumes:
  kafka_data:
  zookeeper_data:
  data01:
```

# To manage the volume

- List what we have?

```
$ docker volume ls

local               my-vol
```

- Remove the volume

```
$ docker volume rm my-vol
```

# To start container with the volume

```
$ docker run -d \
  --name devtest \
  --mount source=myvol2,target=/app \
  nginx:latest
```

```
$ docker run -d \
  --name devtest \
  -v myvol2:/app \
  nginx:latest
```

- With compose

```
volumes:
  - data01:/usr/share/elasticsearch/data
```

```
volumes:
  - type: volume
    source: mysql_data
    target: /var/lib/mysql
```

```
volumes:
  - type: bind
    source: /home/ubuntu/autopair/logs
    target: /logs
  - "/etc/timezone:/etc/timezone:ro"
  - "/etc/localtime:/etc/localtime:ro"
```

CAMT
College of Arts, Media and Technology
Chiang Mai University

# Q & A


CAMT
College of Arts, Media and Technology
Chiang Mai University