LAB-06
Docker Compose

Olarn Rojanapornpun

Docker Compose

- A tool for defining and running multi-container applications
- Compose makes it easy to manage services, networks, and volumes in a single, comprehensible YAML configuration file
- Create and start all the services with a single command
  - Start, stop, and rebuild services
  - View the status of running services
  - Stream the log output of running services
  - Run a one-off command on a service

https://docs.docker.com/compose/

References

- https://docs.docker.com/reference/cli/docker/compose
- https://docs.docker.com/compose/compose-file/05-services

https://docs.docker.com/reference/cli/docker/compose
https://docs.docker.com/compose/compose-file/05-services

Compose File

- The default path for a Compose file is compose.yaml (preferred) or compose.yml that is placed in the working directory
- Compose also supports docker-compose.yaml and docker-compose.yml for backwards compatibility of earlier versions
- If both files exist, Compose prefers the canonical compose.yaml

Docker Compose Commands


```
# To only build docker image
docker compose build

# To force build Docker Image everytime
docker compose up --build

# To choose custom Docker Compose file
docker compose up -f custom-compose.yml

# To run Docker Compose in background
docker compose up -d

# To stop all containers after docker compose up -d
docker compose stop

# To start all containers after docker compose stop
docker compose start

# To restart all containers
docker compose restart

# To show status of containers
docker compose ps

# To show all containers processes
docker compose top

# To list all images in Docker Compose file
docker compose images

# To stop and clean all docker compose up resources
docker compose down
```

dateproj – compose.yaml

FRONTEND

```
services:
  fe:
    image: dateproj/fe
    build: dateproj-fe
    networks:
      front-tier:
        ipv4_address: 172.16.0.3
        aliases:
          - dateproj-fe
    restart: unless-stopped

networks:
  front-tier:
    ipam:
       config:
         - subnet: "172.16.0.0/24"
```

dateproj – compose.yaml

API

```
services:
  api:
    image: dateproj/api
    build:
      context: dateproj-api
      dockerfile: multi.Dockerfile
    networks:
      front-tier:
        ipv4_address: 172.16.0.4
        aliases:
          - dateproj-api
      back-tier:
        ipv4_address: 172.16.1.4
    restart: unless-stopped

networks:
  front-tier:
    ipam:
       config:
         - subnet: "172.16.0.0/24"
  back-tier:
    ipam:
       config:
         - subnet: "172.16.1.0/24"
```

dateproj – compose.yaml

PROXY

```yaml
services:
  proxy:
    image: nginx:alpine
    configs:
      - source: proxy_config
        target: /etc/nginx/conf.d/default.conf
    depends_on:
      - fe
      - api
    networks:
      front-tier:
        ipv4_address: 172.16.0.2
    ports:
      - "80:80"
    restart: unless-stopped

configs:
  proxy_config:
    file: dateproj-proxy/dateproj.conf
```

dateproj.conf

```
server {
    listen       80 default_server;
    server_name  lvm65999.sit.kmutt.ac.th;

    location /dateproj/ {
        proxy_pass http://dateproj-fe/;
    }

    location /dateproj/api/ {
        proxy_pass http://dateproj-
api:8080/api/;
    }
}
```

Deploy API component


```
mkdir 209lab6; cd 209lab6
git clone https://github.com/olarnr/int210-dateproj.git dateproj

cd dateproj
```

run dateproj-api

```
# create and start api service only
# run in foreground to view messages
docker compose up api
  [CTRL+C]

# list all containers in the compose
# note the container name
docker compose ps -a
docker network ls

# start api container
docker compose start api

# list all containers in the compose
docker compose ps -a

# inspect api container
# note networks; IPAdress/Aliases
docker inspect dateproj-api-1

# stop and remove containers, networks
docker compose down api
docker compose ps -a
docker network ls
```

```
# modify endpoint to /api/dates
nano dateproj-
api/src/main/java/com/example/demo/DemoApplica
tion.java

# create and start api service in background
# note that the image is NOT build again
docker compose up -d api

# build image before starting container
docker compose up api -d –build
docker compose logs

# test that the API has been updated
curl 172.16.0.4:8080/api/dates
```

Deploy FE and PROXY components

run dateproj-fe

```
# create and start fe service only
docker compose up fe
  [CTRL+C]
# start fe container
docker compose start fe
# view processes in containers
docker compose top
# view images of compose
docker compose images
```

run dateproj-proxy

```
# create and start proxy service only
docker compose up proxy
  [CTRL+C]
# start fe container
docker compose start proxy
# view containers
docker compose ps

docker compose down
docker compose images
```

- Open the page on browser
- Note the problem with fetching API
- Try to fix the problem

Experiment with depends_on


```
# start all services
# note the order the services are started
docker compose up -d
docker compose down

# create and start proxy service only
# note the services that are started
docker compose up -d proxy
docker compose down

# modify compose.yaml, comment depends_on
# note the order the services are started
docker compose up -d
docker compose down
```

studentproj – compose.yaml

PROXY

```
services:
  proxy:
    image: nginx:alpine
    configs:
      - source: proxy_config
        target: /etc/nginx/conf.d/default.conf
    depends_on:
      - fe
      - api
    networks:
      front-tier:
        ipv4_address: 172.15.0.2
    ports:
      - "80:80"
    restart: unless-stopped

configs:
  proxy_config:
    file: studentproj-proxy/proxy.conf
```

proxy.conf

```
server {
    listen       80 default_server;
    server_name  lvm65999.sit.kmutt.ac.th;

    location / {
        proxy_pass http://studentproj-fe/;
    }

    location /api/ {
        proxy_pass http://studentproj-
api:8080/api/;
    }
}
```

studentproj – compose.yaml

FRONTEND

```
services:
  fe:
    image: ghcr.io/olarnr/studentproj/fe
    build:
      context: https://github.com/olarnr/int210-studentproj.git#main:studentproj-fe
      dockerfile: multi.Dockerfile
    networks:
      front-tier:
        ipv4_address: 172.15.0.3
        aliases:
          - studentproj-fe
    restart: unless-stopped

networks:
  front-tier:
    ipam:
       config:
         - subnet: "172.15.0.0/24"
```

studentproj – compose.yaml

API

```
services:
  api:
    image: ghcr.io/olarnr/studentproj/api
    build:
      context: https://github.com/olarnr/int210-studentproj.git#main:studentproj-api
      dockerfile: multi.Dockerfile
    environment:
      - mysql_url=studentproj-db
    depends_on:
      db:
        condition: service_healthy
    networks:
      front-tier:
        ipv4_address: 172.15.0.4
        aliases:
          - studentproj-api
      back-tier:
        ipv4_address: 172.15.1.4
    restart: unless-stopped

networks:
  back-tier:
    ipam:
       config:
         - subnet: "172.16.1.0/24"
```

studentproj – compose.yaml

DB

```
services:
  db:
    image: mysql/mysql-server
    env_file: studentproj-db/env.list
    volumes:
      - ./studentproj-db/scripts:/docker-entrypoint-initdb.d
      - db-datadir:/var/lib/mysql
    healthcheck:
      test: ["CMD", "/healthcheck.sh"]
      interval: 2s
      timeout: 2s
      retries: 20
    networks:
      back-tier:
        ipv4_address: 172.15.1.5
        aliases:
          - studentproj-db
    restart: unless-stopped

volumes:
  db-datadir:
```