第 5 章课程辅助资料

为了便于大家更好地理解课程中的知识，学院教研组配套整理了课程所需的辅助资料，供大家参考使用。

1. 指令包

```
# 启动 Docker

# None 模式设置
docker network ls
docker pull alpine:3.17
docker images
docker run -d --network none --name none-network-c   alpine:3.17 tail -f /dev/null
docker container ls
docker network inspect none

# None 使用
docker container ls
docker exec -it none-network-c /bin/sh
/ # ip addr ls
/ # ping 8.8.8.8
- expect to fail
/ # exit

docker network ls

# Bridge Network A 建立
docker network ls
docker network create --driver bridge my-bridge-A
docker network ls
docker network inspect my-bridge-A
- get subnet & gateway ip address

# Bridge Network A - 放入 container 1
docker network ls
docker run -d --network my-bridge-A --name bridge-network-c-001 alpine:3.17 tail -f /dev/null
docker container ls
docker network inspect my-bridge-A
- get container's ip address

docker container ls
docker exec -it bridge-network-c-001 /bin/sh
/ # ip addr ls
/ # ping 172.18.0.1
/ # ping 8.8.8.8
/ # exit
```

```
# Bridge 模式 - 放入 container 2
docker run -d --network my-bridge-A --name bridge-network-c-002 alpine:3.17 tail -f /dev/null
docker container ls
docker network inspect my-bridge-A
- get container's ip address
```

```
docker container ls
docker exec -it bridge-network-c-002 /bin/sh
/ # ip addr ls
/ # ping 172.18.0.1
/ # ping 8.8.8.8
/ # ping 172.18.0.2
/ # exit
```

```
docker container ls
docker network inspect my-bridge-A
```

# Bridge network B 建立

```
docker network ls
docker network create --driver bridge my-bridge-B
docker network ls
docker network inspect my-bridge-B
- get subnet & gateway ip address
```

```
# Bridge network 2 - 放入 container 3
docker run -d --network my-bridge-B --name bridge-network-c-003 alpine:3.17 tail -f /dev/null
docker container ls
docker network inspect my-bridge-B
- get container's ip address
```

```
docker container ls
docker exec -it bridge-network-c-003 /bin/sh
/ # ip addr ls
/ # ping 172.19.0.1
/ # ping 8.8.8.8
/ # ping 172.18.0.2
/ # exit

# Bridge network 1 - 放入 container 3
docker network connect my-bridge-A bridge-network-c-003
docker network inspect my-bridge-A
- get container's ip address
docker network inspect my-bridge-B
- one container sits in two bridge networks

docker container ls
docker exec -it bridge-network-c-003 /bin/sh
/ # ip addr ls
/ # ping 172.18.0.1
/ # ping 172.19.0.1
/ # ping 8.8.8.8
/ # ping 172.18.0.2
/ # ping 172.18.0.3
/ # exit

# 强制清理三指令
docker stop $(docker container ls -a -q)
docker rm $(docker container ls -a -q)
docker rmi -f $(docker images -q)

docker network ls

docker network remove my-bridge-A
docker network remove my-bridge-B
docker network remove my-bridge-C
```

=====

如何判定 ip 是否属于同一个网络空间:

172.17.0.0/8

172.19.0.0

172.19.1.0

172.17.199.199

172.17.0.0/16

172.17.80.1

172.17.100.199

172.17.0.0/24

172.17.0.0/32

```
bridge: 172.17.0.0/16
- gateway: 172.17.0.1

my-bridge-A: 172.21.0.0/16
- gateway: 172.21.0.1
bridge-network-c-001: 172.21.0.2/16
bridge-network-c-002: 172.21.0.3/16
bridge-network-c-003: 172.21.0.4/16

my-bridge-B: 172.22.0.0/16
- gateway: 172.22.0.1
bridge-network-c-003: 172.22.0.2/16
=====
```

# Host 模式（Linux OS only）

```
sudo docker network ls
sudo docker pull nginx:1.23
sudo docker images
sudo docker run -d -p 8090:80 --name c-nginx nginx:1.23
sudo docker container ls
curl http://localhost:80

sudo docker run -d --network host --name host-network-c-nginx nginx:1.23
sudo docker container ls
- expect no port mapping
curl http://localhost:80

sudo docker network ls
sudo docker network inspect host

# 查看 主机本身 ip adress
ip addr ls
- all bridge network gateway ip will be seen
- host ip can be seen here

# 进入 容器 从内部查看两者设置
sudo docker exec -it host-network-c-nginx /bin/sh
/ # cat /etc/os-release
/ # apt update
/ # apt install -y iproute2
/ # ip addr ls
/ # apt install net-tools
/ # netstat -tulpn
/ # exit

sudo docker container ls
curl http://localhost:80
```