REST API

frontend
client side
UI

http req
GET
POST
PUT
DELETE
PATCH
OPTION

backend
server side
API

http response
status code
1XX
2XX
3XX
4XX
5XX

RESTful
API

db

Docker networks

Host OS

docker

existes

nginx
ubuntu

.images

docker hub

localhost
127.0.0.1

ubuntu

grafana

docker images

terminal

3000

3005

container1

3000

ip

grafana

— — docker run -p 3000:3000 --name container1 grafana/grafana— — — — — —

container2

3000

ip

grafana

— — docker run -p 3005:3000 --name container2 grafana/grafana— — — — —

3000

50501

50502

3000

docker ip

ip
container

docker network connect bridge busy5

172.17.0.2
busy2

172.18.0.3
busy5

172.17.0.4

172.17.0.1/16
bridge

172.18.0.1/16
hothaifa

bridge

172.17.0.3
busy3

172.18.0.2
busy4

HOST

44.13.0.5
host os

44.13.0.5
container

Host

127.0.0.1
container

127.0.0.1
container

127.0.0.1
container

null
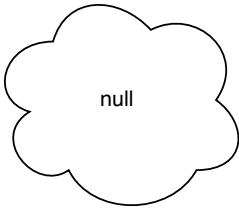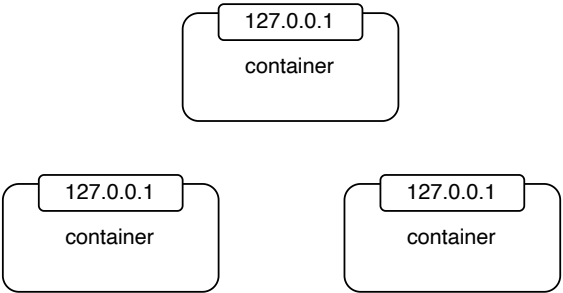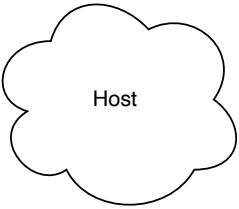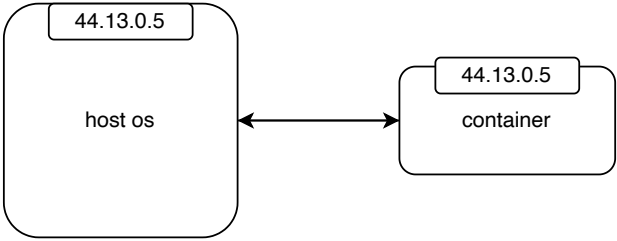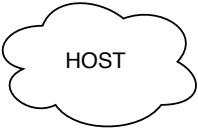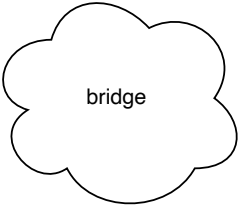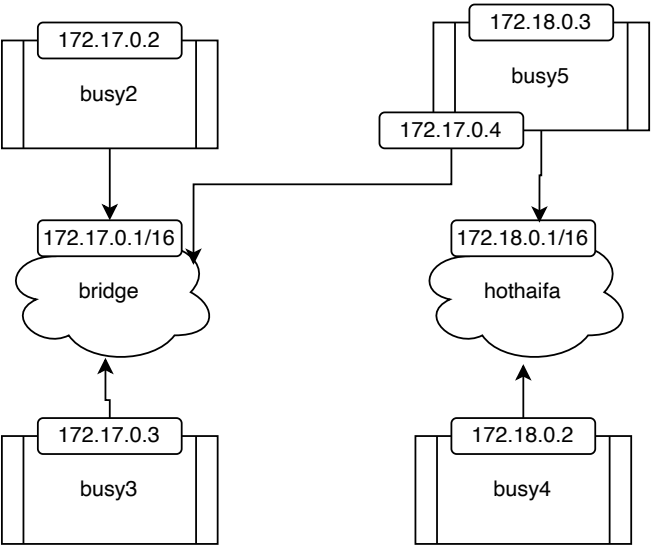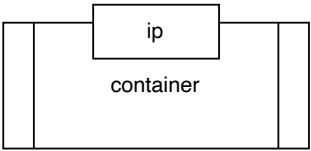
exercise 1

- Create a custom bridge network named "my-network"

- Run two nginx containers on this custom bridge network, naming them web1 and web2

  - what is the ip for web1
  - what is the ip for web2
  - **can we have them in one command ?**

- Test connectivity between the containers by pinging web2 from web1

- Try to curl the nginx service running on web2 from web1 on port 80

- Try to access the nginx service from your host machine using localhost (it should fail without port mapping)

- Remove one container and recreate it with port mapping, then try accessing from your host again

web1

ping
curl 80

web2

my-network

host
localhost 80