Introduction to Advanced Docker Topics

Advanced Docker topics

In Week 1, you learned:

Docker images are created from Dockerfile instructions

How to create Docker containers for basic Django and Flask applications

More advanced applications will rely on: managing ports, volumes, networking, working with more advanced web servers

Managing ports

Server: Application that listens to requests sent to specific network addresses and ports from client applications, then serves responses

Default development ports: Django uses port 8000, and Flask uses port 5000

Containers receive network requests sent to host machine via port mapping/binding

Example: docker run --rm -d -p 8000:8000/tcp hellodjango:latest

Volumes & managing data

Data generated from Docker containers lost once containers are gone

Data volumes stored on Docker host can be mounted to Docker containers

Independent of container lifecycle,
able to preserve container data after container is gone

Volumes can also be used to share data between containers

Volumes & managing data

Storage drivers – used to manage storing image layers and ephemeral container data

Docker supports several different storage drivers

See: https://docs.docker.com/storage/storagedriver

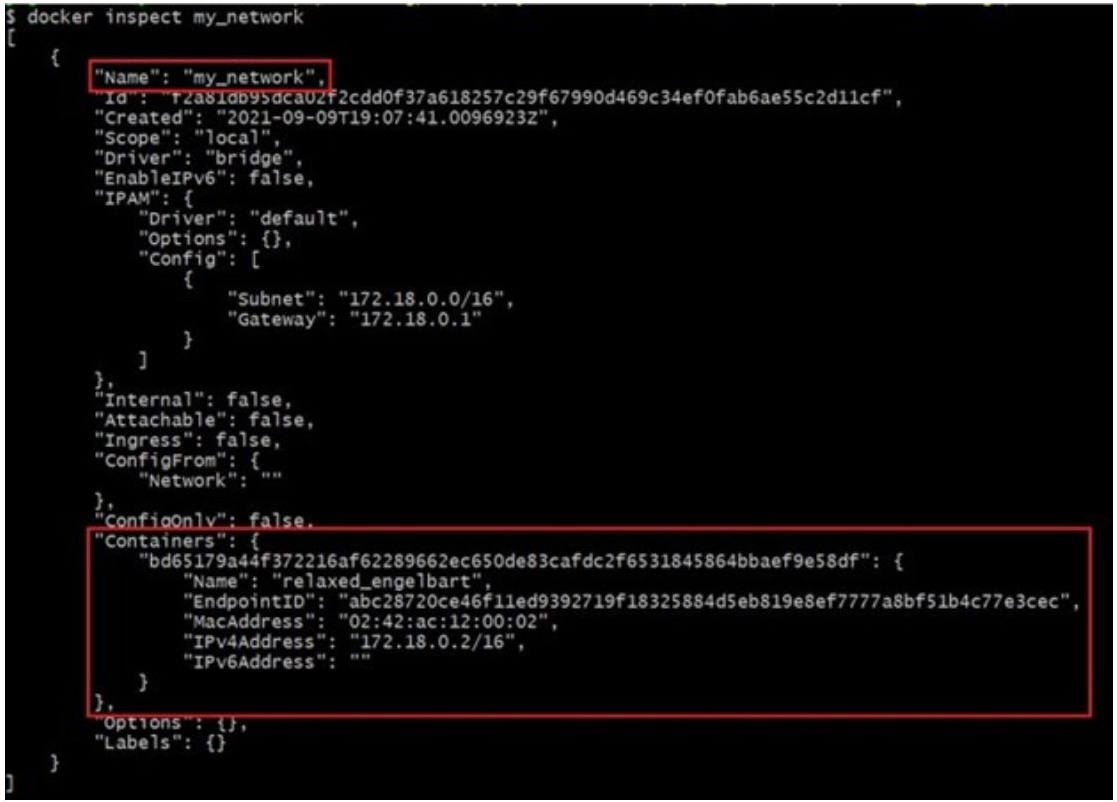
Networking

Docker containers can communicate with each other through Docker networks

Docker automatically handles networking between containers managed together by Docker Compose

We can manually create networks and attach containers to them

Networking


$ docker inspect my_network
[
{
"Name": "my_network",
"Id": "f2a81db95dca02f2cdd0f37a618257c29f67990d469c34ef0fab6ae55c2d11cf",
"Created": "2021-09-09T19:07:41.0096923Z",
"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
"Driver": "default",
"Options": {},
"Config": [
{
"Subnet": "172.18.0.0/16",
"Gateway": "172.18.0.1"
}
]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
"Network": ""
},
"ConfigOnly": false,
"Containers": {
"bd65179a44f372216af62289662ec650de83cafdc2f6531845864bbaef9e58df": {
"Name": "relaxed_engelbart",
"EndpointID": "abc28720ce46f11ed9392719f18325884d5eb819e8ef7777a8bf51b4c77e3cec",
"MacAddress": "02:42:ac:12:00:02",
"IPv4Address": "172.18.0.2/16",
"IPv6Address": ""
}
},
"Options": {},
"Labels": {}
}
]

The NGINX web server

https://nginx.org

Popular lightweight web server

Useful for Dockerized web apps

We will pull NGINX image from Docker Hub