Section 6 - Docker Networking Basics

3 - Docker Network DNS

Overview

Understand how containers communicate using dynamic DNS rather than IP addresses.

Docker Networks: DNS

- The Docker daemon implements an embedded DNS server which provides built-in service discovery for any container created.
- Docker uses the container name as the DNS name.
- DNS is a very important service, because we cannot rely on the IP address of the containers, since they are dynamic.
- After a container is created, we should no rely on the container IP address because it can be changed (e.x. in case the container is restarted due to a failure).
- Static IPs and using IPs for talking to containers is an anti-patern.
- Avoid using IP addresses. Use only container names.

Example

```
# docker network create my_app_net
# docker container run -d --name web_server --network my_app_net nginx
# docker container run --network my_app_net alpine ping -c3  web_server
PING web_server (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: seq=0 ttl=64 time=0.070 ms
64 bytes from 172.18.0.2: seq=1 ttl=64 time=0.196 ms
64 bytes from 172.18.0.2: seq=2 ttl=64 time=0.255 ms
```

Notes:

- Any user-defined bridge network, such as the `my_app_net`, has the DNS service available.
- So, in this example we can use the container name "web_server" to test the DNS resolution.

DNS server not running on the default bridge network

- The DNS service is DISABLE on the default bridge (docker0) network.
- In the default bridge network we can use the `--link` option for name resolution.
- It is always better to use a custom network instead of the `--link` option.
- Later we will see how to use docker-compose to create containers and how in this case custom networks are automatically created.

LAB

- Ref:
- D_S6_L3_Docker_Network_DNS_lab.md