Practica 5 – Docker Networking

Markel Orallo Nogueira

2021

# Docker networking

**Siguiendo con la práctica de Dockercoins:**

**¿Cuál es la red que utilizan los diferentes contenedores?**

version: "3.1"

services:

rng:

image: markel149/rng:latest

networks:

- dockercoins

ports:

- "8001:80"

hasher:

image: markel149/hasher:latest

networks:

- dockercoins

ports:

- "8002:80"

webui:

image: markel149/webui:latest

networks:

- dockercoins

ports:

- "8000:80"

redis:

image: redis

networks:

- dockercoins

worker:

image: markel149/worker:latest

networks:

- dockercoins

networks:

dockercoins:

Analizando el docker-compose la red que utilizan los diferentes contenedores es la red dockercoins.

**Pon en marcha los contenedores(docker-compose up)**

**Texto

Descripción generada automáticamente con confianza media**

**Entra dentro de uno de los contenedores ($ docker container exec -it <Container ID> bash ) y ejecuta el comando ping nombre\_contenedor**

**Texto

Descripción generada automáticamente**

**– Llega al destino? A todos los contenedores?**

Si, llega a todos los contenedores.

**– ¿Qué driver estará utilizando?**

Esta utilizando el driver “bridge”.

**Texto

Descripción generada automáticamente**

Si hacemos docker network inspect dockercoins\_dockercoins

[

{

"Name": "dockercoins\_dockercoins",

"Id": "03bf9102131bd8e9a328fa6e5e8d61fe2a383cfc9d3bf77530c7713ef3b1da45",

"Created": "2021-09-27T20:22:22.268898752Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.20.0.0/16",

"Gateway": "172.20.0.1"

}

]

},

"Internal": false,

"Attachable": true,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"1128ae84ef8bc1ede9855d54015069943c4af90d863687f9a1331e92cb50f374": {

"Name": "dockercoins\_webui\_1",

"EndpointID": "94933092bc605f5e109a58f7f31edc8113f9be845bef2e6bee98ca81b8d2d0dd",

"MacAddress": "02:42:ac:14:00:03",

"IPv4Address": "172.20.0.3/16",

"IPv6Address": ""

},

"52b80343223c201c4a17adae713352054c55696d17cae65a96748c8b3eff3eb3": {

"Name": "dockercoins\_redis\_1",

"EndpointID": "076bd1db64eddd63a2b28d3d685e7672dc080d1ee3113ec8f9783f31914791f7",

"MacAddress": "02:42:ac:14:00:04",

"IPv4Address": "172.20.0.4/16",

"IPv6Address": ""

},

"59efcc7e2e24d9b8d81c85d790aa41d4b3b6ccc53baaec38ae0975687274cbc4": {

"Name": "dockercoins\_hasher\_1",

"EndpointID": "1c9b1d0a62694b5cd296d4a8ec062641b9dddca5887fcdddd787a8b6714f5254",

"MacAddress": "02:42:ac:14:00:05",

"IPv4Address": "172.20.0.5/16",

"IPv6Address": ""

},

"c0061048816a68aca15a4cb7de704ef3662068f1e3121fa0a83e8e4d12c251c0": {

"Name": "dockercoins\_worker\_1",

"EndpointID": "a3832e182afe237160c7a735ac407402c819862f30ad96a88a50834506269a13",

"MacAddress": "02:42:ac:14:00:02",

"IPv4Address": "172.20.0.2/16",

"IPv6Address": ""

},

"c995c8d3bd56e4122cd95b5fb7e123c8979a42a1cd76fa10540f26b51b386b6e": {

"Name": "dockercoins\_rng\_1",

"EndpointID": "350fb7838ebd43183e52f8f4c83043b1fa63d8c1cd099648cae4b8ebaef9777f",

"MacAddress": "02:42:ac:14:00:06",

"IPv4Address": "172.20.0.6/16",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {

"com.docker.compose.network": "dockercoins",

"com.docker.compose.project": "dockercoins",

"com.docker.compose.version": "1.29.2"

}

}

]

Podemos ver en el output del comando anterior que los 5 contenedores que están conectados a la red son los contenedores que hemos levantado con docker-compose:

* Dockercoins\_rng\_1
* Dockercoins\_worker\_1
* Dockercoins\_redis\_1
* Dockercoins\_hasher\_1
* Dockercoins\_webui\_1

**– ¿Hace uso del DNS?**

Si hace uso del DNS. Al ser una red diferente a la default los contenedores puedes llegar de uno a otro a través de DNS. Podemos comprobarlo entrando a uno de ellos y resolviendo el nombre de otro contenedor:

Texto

Descripción generada automáticamente

Como podemos ver nos resuelve la IP 172.20.0.3 para el contenedor dockercions\_webui\_1. Comprobemos que la resolución es correcta:

Texto

Descripción generada automáticamente

Lanzando el comando ip addr dentro del contenedor vemos que la IP de webui es 172.20.0.3.

# Crea la siguiente arquitectura de redes y contenedores

Escala de tiempo

Descripción generada automáticamente

docker network create my\_bridge -d bridge

docker run –name=c1 -d -it centos:7 /bin/bash

docker run --net=my\_bridge --name=c2 -d -it centos:7 /bin/bash

docker run --net=my\_bridge --name=c3 -d -it centos:7 /bin/bash

De esta manera nos quedan dos contenedores (c3 y c2) conectados entre si a través de la red my\_bridge. El contenedor c1 al contrario quedará en la red por defecto de docker (docker0).

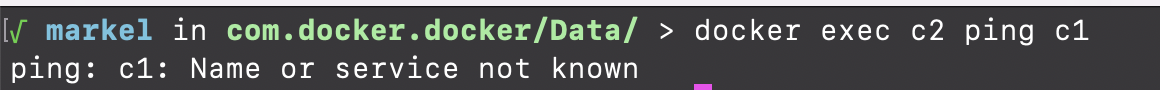
De esta manera al hacer ping de c2 a c3 llegaremos, tanto por nombre como por IP ya que también hace uso del DNS. En cambio, si intentamos hacer ping a c1 no llegaran los paquetes al encontrarse en otra red.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Imagen que contiene Calendario

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente