What is the difference between network_mode and networks using Docker

Asked 2 years, 2 months ago Modified 2 years, 2 months ago Viewed 12k times

8

I was trying to use the network `"bridge"` to join services from different `"docker-compose.yml"` and I was getting the following error:

```
Network-scoped alias is supported only for containers in user defined networks
```

Searching a bit, I came to this reference link:: https://github.com/docker/compose/issues/3012

I read and re-read everything several times, but I couldn't understand the real reason for using one or the other.

Can someone explain to me what the difference is in practice?

Why when you use `network_mode` and run `docker network inspect bridge` all containers correctly linked appear?

Before:

```
services:
  local-db:
    image: mysql

networks:
  default:
    external:
      name: bridge
```

After:

```
local-db:
  image: mysql
  network_mode: bridge
```

docker docker-compose containers

Share Improve this question Follow

asked Dec 23, 2022 at 19:10

José Victor

345 ●1 ●2 ●14

1 Answer

Sorted by: Highest score (default)

▲
6
▼

The top level `networks` section creates networks in compose (or specifies external user created networks) that can be used by various services. Each service then defines a `networks` section to identify one or more user created networks to join. That service gets automatically configured with a network alias for DNS based service discovery. This is the preferred method to setup networking with services, and by default each of these networks is a bridge network.

To skip that entire process, you can use a `network_mode` that changes from the compose managed networking to switch to `host`, `bridge`, `none`, `service:$name`, or `container:$name`. Each of these has unique properties.

- `host` has no network namespacing at all, similar to starting a process outside of a container.
- `bridge` is the legacy bridge network docker always creates, but disables some features like DNS based service discovery.
- `none` is no networking at all, only the loopback interface is defined.
- `service:$name` and `container:$name` are special ways of connecting multiple containers into the same network namespace. The same thing is done by pods in kubernetes, and it can be useful for network troubleshooting or when creating sidecars that require access to localhost. This should only be used as a last resort when building microservices since it breaks the ability to independently scale and deploy the containers.

Share Improve this answer Follow

answered Dec 23, 2022 at 19:37
BMitch
265k ● 50 ● 542 ● 499

And why can't I use "networks" to add the "bridge"? Just because "bridge" is a Docker standard? – José Victor Dec 23, 2022 at 19:46

@JoséVictor the default networks from docker have different properties from user created networks. Compose can't apply its normal settings to containers on those networks. – BMitch Dec 23, 2022 at 19:57

Now I understand, it is a particularity due to the fact that it is standard docker networks. Thanks a lot for your patience and explanation. @BMitch – José Victor Dec 23, 2022 at 20:23

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

docker docker-compose containers

See similar questions with these tags.