What is the difference between ports and expose in docker-compose?

Asked 8 years, 3 months ago Modified 11 months ago Viewed 730k times

1017

What is the difference between `ports` and `expose` options in `docker-compose.yml`?

docker docker-compose

Share Improve this question Follow

edited Oct 21, 2022 at 11:26

Luis Lavaire
701 ● 7 ● 18

asked Nov 25, 2016 at 9:39

Bibek Shrestha
35.1k ● 7 ● 34 ● 35

1 baeldung.com/ops/docker-compose-expose-vs-ports – Mike Lowery May 6, 2023 at 18:23

5 Answers

Sorted by: Highest score (default)

1016

According to the docker-compose reference,

Ports is defined as:

> Expose **ports**. Either specify both ports (HOST:CONTAINER), or just the container port (a random host port will be chosen).

- Ports mentioned in docker-compose.yml will be shared among different services started by the docker-compose.
- Ports will be exposed to the host machine to a random port or a given port.

My `docker-compose.yml` looks like:

```
mysql:
  image: mysql:5.7
  ports:
    - "3306"
```

If I do `docker-compose ps`, it will look like:

```
  Name                     Command               State           Ports
-------------------------------------------------------------------------------
------
  mysql_1       docker-entrypoint.sh mysqld      Up      0.0.0.0:32769-
>3306/tcp
```

Expose is defined as:

> Expose ports without publishing them to the host machine - they'll only be accessible to linked services. Only the internal port can be specified.

Ports are not exposed to host machines, only exposed to other services.

```
mysql:
  image: mysql:5.7
  expose:
    - "3306"
```

If I do `docker-compose ps`, it will look like:

```
  Name                  Command             State    Ports
-------------------------------------------------------------------
 mysql_1      docker-entrypoint.sh mysqld   Up      3306/tcp
```

Edit

In recent versions of Dockerfile, `EXPOSE` typically doesn't have any operational impact anymore, it is just informative. (see also)

Share Improve this answer Follow

edited Feb 14, 2024 at 5:15

rogerdpack

67k • 40 • 284 • 404

answered Nov 25, 2016 at 9:39

Bibek Shrestha

35.1k • 7 • 34 • 35

72 would it be possible eto explain what advantages there are to specifying `expose` in a `docker-compose`? As far as I can tell, you don't need to specify expose to make ports accessible to linked services. – Juicy May 6, 2018 at 11:23

6 Shouldn't it tell that exposed ports will only be available to services in the same docker network (linking is being replaced for most parts)?. – meow Aug 26, 2018 at 11:28

21 @Juicy I guess it's similar to `expose` in Dockerfiles: "The EXPOSE instruction does not actually publish the port. It functions as a type of documentation..." docs.docker.com/engine/reference/builder/#expose – tsauerwein Sep 6, 2018 at 11:47

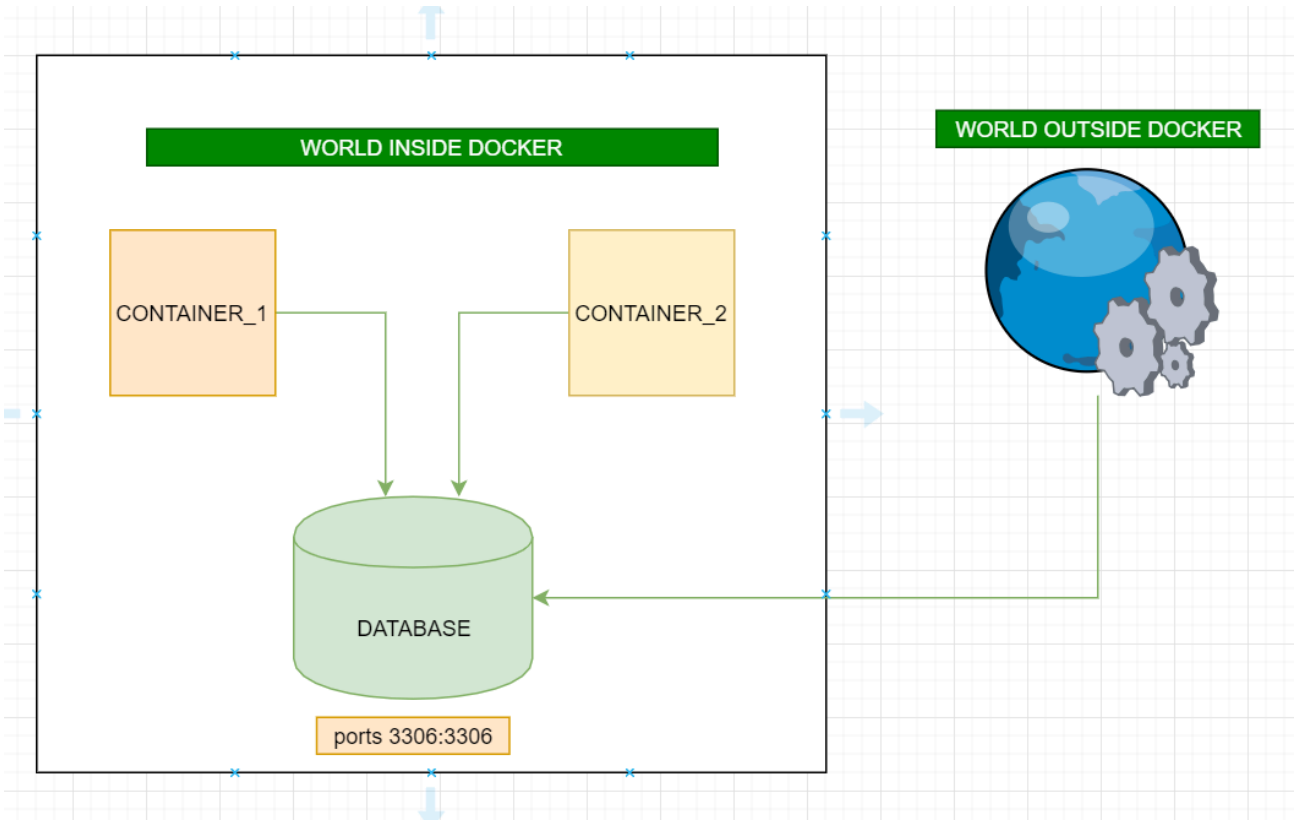
3 Do Ports override any settings at the firewall level? I just noticed that I didn't open up ports for mysql on the firewall, but they were accessible remotely. I had Ports set to "3306:3306" instead of expose.
– TechFanDan May 28, 2019 at 23:06

6 And remember, if you use `docker-compose run`, the port definition in `docker-compose.yml` is **ignored** by default. Either use `docker-compose up` or provide the parameter `--service-ports`
– Juha Untinen Oct 27, 2019 at 13:44

471

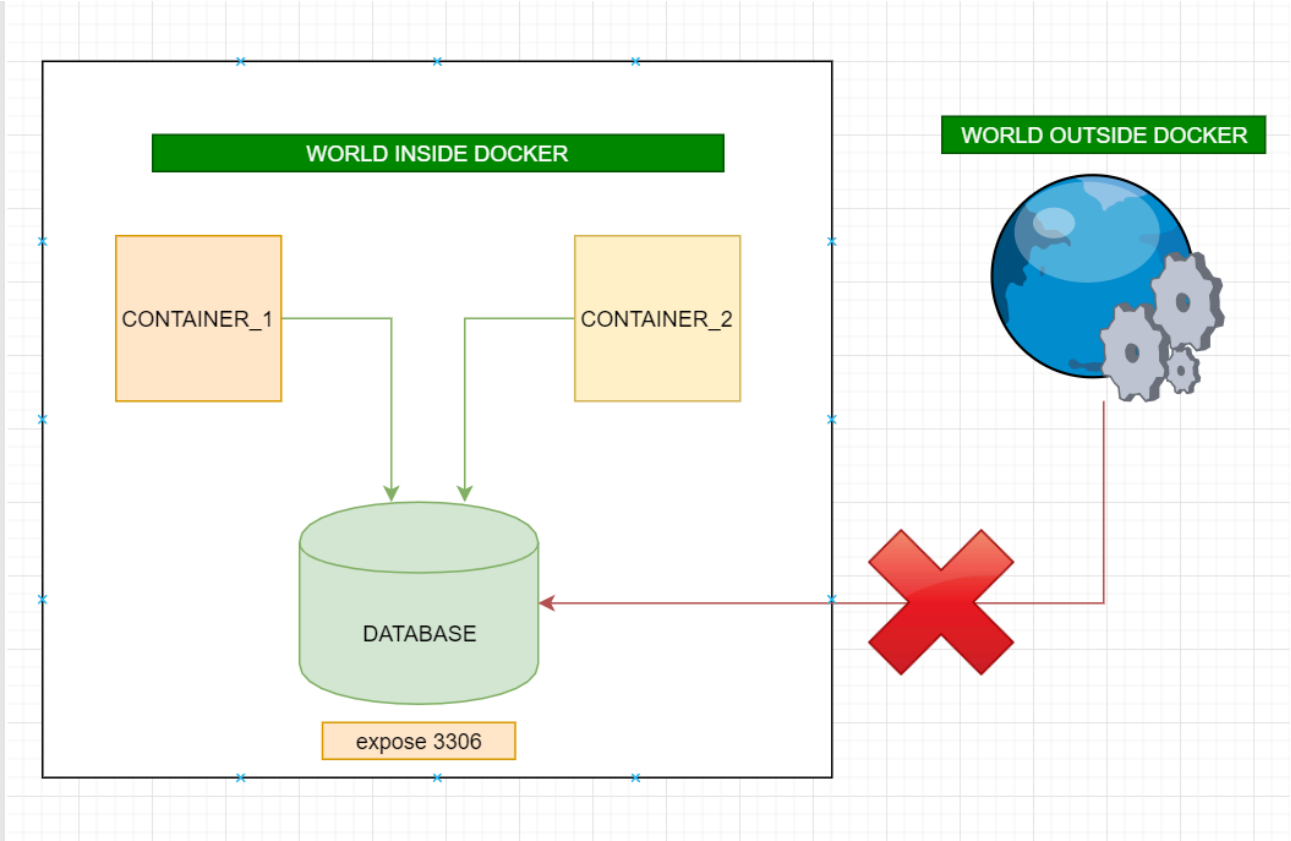
ports:

1. Activates the container to listen for specified port(s) from the world outside of the docker (can be the same host machine or a different machine) AND is also accessible to the world inside Docker.
2. More than one port can be specified (that's why it's **ports** not port).



expose:

1. Activates container to listen for a specific port only from the world inside of docker AND is not accessible to the world outside of Docker.
2. More than one port can be specified.



Share Improve this answer Follow

edited Apr 19, 2024 at 23:06
Matthias Braun
34.4k ● 27 ● 153 ● 176

answered Feb 26, 2019 at 10:58
Mehraj Malik
15.9k ● 18 ● 61 ● 88

8 Note that expose allows multiple ports - docs.docker.com/compose/compose-file/#expose - however you only supply the internal port rather than internal + external – Stuart Moore Apr 24, 2019 at 14:02

1 That's very important part: 'Activates container to listen for a specific port only from the world inside' – Reven Jul 24, 2020 at 10:53

actually, all these days I understood this concept in reverse order thank you!! – Aditya Jul 13, 2023 at 19:40

1 larsks claims this is incorrect stackoverflow.com/a/65785558/3755989 – Johan Jan 11, 2024 at 8:58

When you say *world inside docker*, does it include *all* containers running on the same host (EC2 instance), even those containers outside the specific `docker-compose` file? – Della Jul 8, 2024 at 8:00

99

Ports This section is used to define the mapping between the host server and Docker container.

```
ports:
   - 10005:80
```

It means the application running inside the container is exposed at port 80. But external system/entity cannot access it, so it need to be mapped to host server port.

Note: you have to open the host port 10005 and modify firewall rules to allow external entities to access the application.

They can use

> http://{host IP}:10005

something like this

EXPOSE This is exclusively used to define the port on which application is running inside the docker container.

You can define it in dockerfile as well. Generally, it is good and widely used practice to define EXPOSE inside dockerfile because very rarely anyone run them on other port than default 80 port

Share Improve this answer Follow edited Oct 6, 2018 at 18:42 answered Oct 6, 2018 at 17:24

sorabzone
1,037 ● 7 ● 8

80

Ports

The `ports` section will publish ports on the host. Docker will set up a forward for a specific port from the host network into the container. By default, this is implemented with a userspace proxy process (`docker-proxy`) that listens on the first port, and forwards into the container, which needs to listen on the second point. If the container is not listening on the destination port, you will still see something listening on the host, but get a connection refused if you try to connect to that host port, from the failed forward into your container.

Note, the container must be listening on all network interfaces since this proxy is not running within the container's network namespace and cannot reach 127.0.0.1 inside the container. The IPv4 method for that is to configure your application to listen on `0.0.0.0`.

Also note that published ports do not work in the opposite direction. You cannot connect to a service on the host from the container by publishing a port. Instead you'll find docker errors trying to listen to the already-in-use host port.

Expose

Expose is documentation. It sets metadata on the image, and when running, on the container too. Typically, you configure this in the Dockerfile with the `EXPOSE` instruction, and it serves as documentation for the users running your image, for them to know on which ports by default your

application will be listening. When configured with a compose file, this metadata is only set on the container. You can see the exposed ports when you run a `docker inspect` on the image or container.

There are a few tools that rely on exposed ports. In docker, the `-P` flag will publish all exposed ports onto ephemeral ports on the host. There are also various reverse proxies that will default to using an exposed port when sending traffic to your application if you do not explicitly set the container port.

Other than those external tools, expose has no impact at all on the networking between containers. You only need a common docker network, and connecting to the container port, to access one container from another. If that network is user created (e.g. not the default bridge network named `bridge`), you can use DNS to connect to the other containers.

Share Improve this answer Follow

edited Oct 6, 2022 at 4:48

Pang
10.1k ● 146 ● 86 ● 124

answered Jun 30, 2019 at 15:50

BMitch
265k ● 50 ● 542 ● 499

12

I totally agree with the answers before. I just like to mention that the difference between expose and ports is part of the security concept in docker. It goes hand in hand with the networking of docker. For example:

> Imagine an application with a web front-end and a database back-end. The outside world needs access to the web front-end (perhaps on port 80), but only the back-end itself needs access to the database host and port. Using a user-defined bridge, only the web port needs to be opened, and the database application doesn't need any ports open, since the web front-end can reach it over the user-defined bridge.

This is a common use case when setting up a network architecture in docker. So for example in a default bridge network, not ports are accessible from the outer world. Therefor you can open an ingresspoint with "ports". With using "expose" you define communication within the network. If you want to expose the default ports you don't need to define "expose" in your docker-compose file.

Share Improve this answer Follow

answered Feb 13, 2019 at 8:04

FishingIsLife
2,372 ● 4 ● 37 ● 64

Ask question

Explore related questions

docker docker-compose

See similar questions with these tags.