Tim van Baarsen

238 Followers About Follow

How to connect to the Docker host from inside a Docker container?

Tim van Baarsen 1 day ago · 5 min read

As of Docker version `18.03`, you can use the `host.docker.internal` hostname to connect to your Docker host from inside a Docker container.

This works fine on Docker for Mac and Docker for Windows, but unfortunately, this is not was not supported on Linux until Docker `20.10.0` was released in December 2020.

Starting from version `20.10`, the Docker Engine now also supports communicating with the Docker host via `host.docker.internal` on Linux. Unfortunately, this won't work out of the box on Linux because you need to add the extra `–add-host` run flag:

```
--add-host=host.docker.internal:host-gateway
```

the configu Mac, and Windows.

From the Docker documentation for MacOS and Windows:

> *The host has a changing IP address (or none if you have no network access). We recommend that you connect to the special DNS name* `host.docker.internal` *which resolves to the internal IP address used by the host. This is for development purpose and will not work in a production environment outside of Docker Desktop for Windows / Mac.*
>
> *You can also reach the gateway using* `gateway.docker.internal`.

Docker version

At the time of writing this blog post, the latest stable Docker version is:

```
docker -v
Docker version 20.10.8, build 3967b7d
```

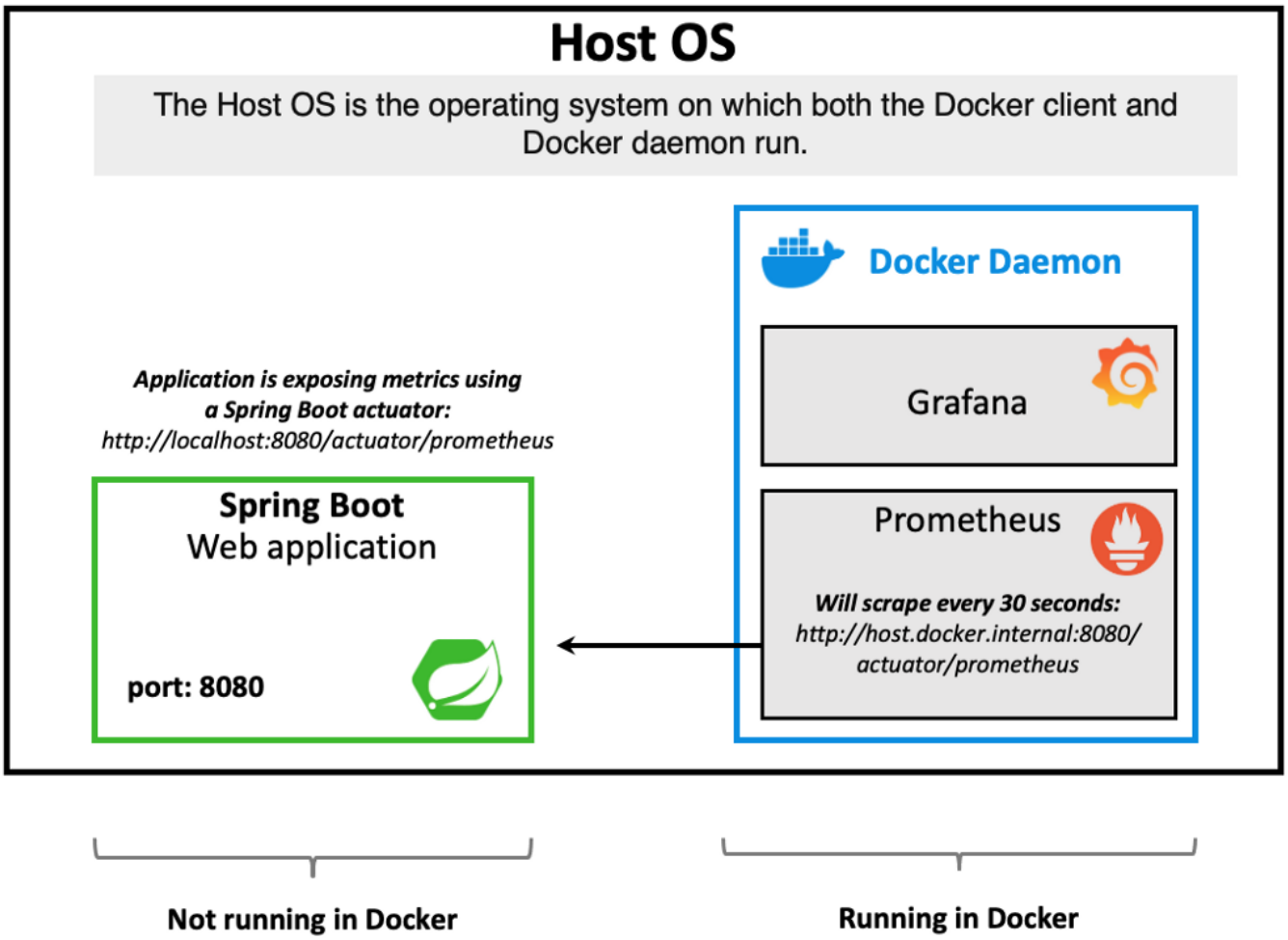
You need to have at least Docker version 20.10.0 installed to make this example work on Linux machines.

Why do I even want to reach my Docker host from inside a Docker container?

Before diving into the issue and solution, it's good to understand a use case; why do I even need to reach the Docker host from inside my running Docker container?

observabilit sed via a Spring Boot actuator endpoint in a way Prometheus can read them. On top of the metrics, I would like to create Grafana dashboards.

For development purposes, I like to run Prometheus and Grafana in Docker while I'm developing my application in my favorite IDE, IntelliJ.

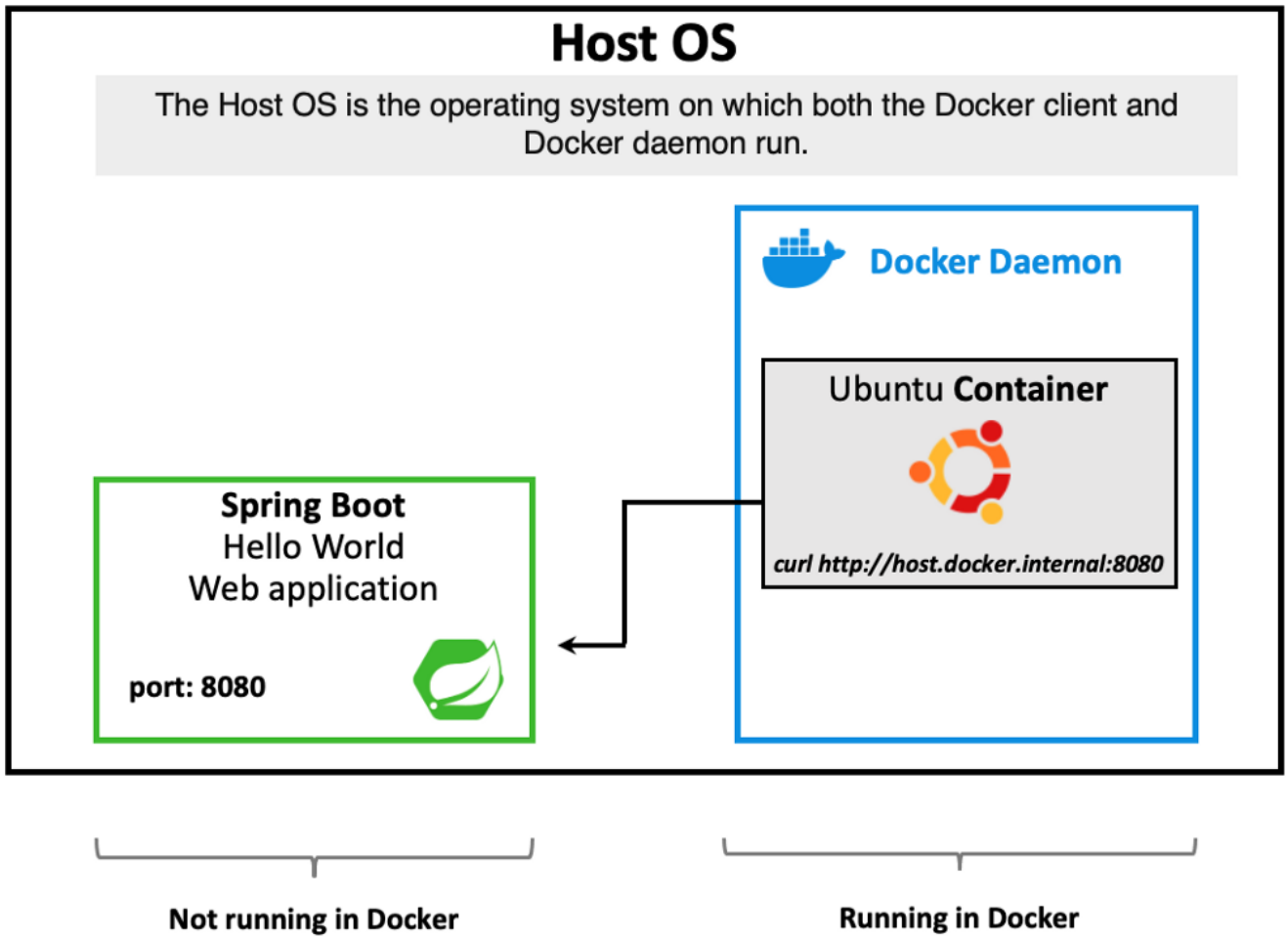


After I implemented my changes, I launch the application from my IDEA, and the application will start on my host.

In Docker, both Prometheus and Grafana are running. Before I can build some nice dashboards, Prometheus needs to scrape the metrics endpoint of my application and will continue to do so on a configured interval. Once again, the Spring Boot application is not running in Docker but on the host itself!

Simple example

To not bother you with a complete Grafana and Prometheus setup but still be able to show the default behavior on Linux, we will use the following setup:



- We run a simple Spring Boot application that exposes a REST endpoint returning a 'Hello World' message.
- We start a Ubuntu container in Docker and execute a REST call to the Spring Boot application running on the host OS(Linux) using curl.

You can clone the example from Github: https://github.com/j-tim/connect-docker-host-from-docker-container-example

Prerequisites

- Docker installed

The Spring Boot Hello World application code (groovy) will:

- expose a REST endpoint
- start at port 8080

```
@RestController
class SpringBootWebApplication {

    @RequestMapping("/")
    String hello() {
        "Hello world container. You are able to reach the Docker host!\n"
    }
}
```

helloWorld.groovy hosted with ♡ by GitHub view raw

First, start the Spring Boot application using the Spring CLI:

```
spring run helloWorld.groovy
```

Start an Ubuntu container:

```
docker run -it ubuntu bash
```

- `it` is short for `--interactive` plus `--tty`
- executing the command take you directly inside the container
- allows you to interact with `/bin/bash` inside the container

inside the U

```
apt update & apt install curl -y
```

Do a call to the Spring Boot application running on the host using `curl`:

```
curl http://host.docker.internal:8080
```

Because we didn't start the container with the extra run flag, we will run into the following issue on Linux (on both Mac and Windows, it will just work fine):

```
root@8024db7c38fe:/# ping host.docker.internal
ping: host.docker.internal: Name or service not known
```

How to make it work?

On Linux, you can't automatically resolve `host.docker.internal`, you need to provide the following run flag when you start the container:

```
--add-host=host.docker.internal:host-gateway
```

In the example of the Ubuntu container:

```
docker run -it --add-host=host.docker.internal:host-gateway ubuntu
bash
```

When you repeat the steps again, you now can reach the Spring Boot application that is running on the host from inside the Ubuntu container:

How about Docker compose?

Most of the time I use Docker compose to start my Docker containers. To get the same behavior, you need to specify the `host.docker.internal:host-gateway` using the extra_hosts parameter.

See `docker-compose.yml` example:

On Mac and Windows, it is possible to use:

- `docker.for.mac.host.internal` (replaces `docker.for.mac.localhost`)
- `docker.for.win.host.internal` (replaces `docker.for.win.localhost`)

Conclusion

- It's easy to connect to the host OS from inside a Docker container
- Avoid the use of OS-specific DNS names flavors like: `docker.for.mac.host.internal`
- In Docker for Mac and Docker for Windows, you can connect to the host out of the box by using the special DNS name: `host.docker.internal`
- For Linux, you need the magic string `host-gateway` to map to the gateway inside the container. This allows you to use the hostname `host.docker.internal` to access the host from inside a container.
- To get a consistent behavior on all platforms (Mac, Windows, and Linux) use `host.docker.internal:host-gateway` in your Docker compose file.
- Play around yourself! Clone my Github repo.

Resources

- Docker Desktop for Windows Networking documentation
- Docker Desktop for Mac Networking Documentation
- Github pull request: Support host.docker.internal in dockerd on Linux
- Docker Engine release notes 20.10.0

Tap the 👏 button if you found this article useful!

Docker Docker Compose Linux Spring Boot