



Docker x Nginx

Using Docker to Run a Simple Nginx Server



Aditya Purwa

Sep 1, 2017 · 2 min read

I just started to learn about docker, mostly because I love automation, and using Docker can help me automate testing and deployment stuff. It just feels good for me to type `docker run abc:def` and see those self writing text on the command line (make me feel like those hacker in movies).

I personally wanted to create a solution for Myriatek's client. So whenever they requested a project, I can use Docker and make it easier for me to test and deploy the project.

I assume that you already know the basics of Docker, and have it installed on your computer.

In this article, we will try to create a container that will run Nginx server. We can use the `nginx:alpine` image because it is small. To download the image, we use the command `docker pull nginx:alpine`.

```
D:\Playground\Docker\simple-nginx
λ docker pull nginx:alpine
alpine: Pulling from library/nginx
019300c8a437: Downloading [=====>] 1.147MB/1.97MB
a3fe4a77433d: Downloading [=====>] 1.08MB/4.774MB
a5443900e7f5: Download complete
0ae275323c0f: Download complete
```

We then create a Dockerfile that will use this image, the Dockerfile simply copy an html file from our computer, into our docker container.

Lets create the `index.html`, a simple hello world is enough.

Now we can build our Dockerfile by running `docker build -t simple-nginx .`, Docker will then generate an image based on our Dockerfile.

Now we can run our image by using `docker run --rm -it -p 8080:80 simple-nginx`. The `--rm` tells docker to remove the container once its stopped, and the `-it` is used to attach interactive TTY, the `-p 8080:80` is used to map the port 80 from the container to our computer port 8080, so if we access `localhost:8080` it means we are accessing the `container.ip:80`. The name `simple-nginx` is the name we defined when we build our Dockerfile above.

Now, open `localhost:8080` and you should see our `index.html` file. Congratulations, you just run a Nginx server on a Docker container.

Our container manages all the dependencies needed to run a Nginx server, this mean that the environment will always be the same wherever we run our docker image, it makes our environment consistent, as long as they are using the same image, there are no reasons for our team member to say that it is not working on their machine (unless their machine got Docker installed wrong, and any other Docker related issues, but it is really something that the Docker's team will help you).

[About](#) [Help](#) [Legal](#)

Get the Medium app

