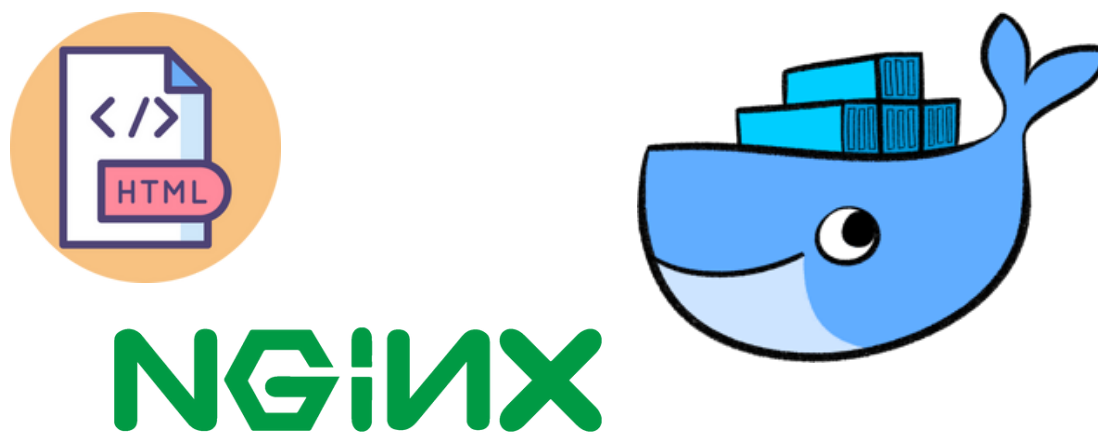


# GET STARTED WITH NGINX ON DOCKER

The idea of this mini-tutorial is to implement a web application with NGINX and Docker, also understand the steps to use the Dockerfile, and create the Docker image and the container.



To make this mini tutorial we have used Docker Engine, Nginx, and HTML.

## 1 Create a directory for our application

Let's create a directory called, "**my-website**", which will contain our web application.



On Linux, you can create this folder with "**mkdir my-website**"



## 2

## Create the website in HTML

Inside "my-website" directory, create an HTML web page: [index.html](#)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Basic Website</title>
  </head>
  <body>
    <h1>Hello Docker Community!</h1>
    
  </body>
</html>
```



Website for our  
Docker  
Community!



## 3

## Create a Dockerfile based on the Nginx image

With "FROM" this Dockerfile indicates to use the Nginx image  
We need to "COPY" our website file "[index.html](#)" into the default public web root for Nginx to make it visible in our browser.



```
FROM nginx:stable
```

```
COPY index.html /usr/share/nginx/html/
```

Nginx is a web  
server that will  
host our static  
website.



A Dockerfile is a text  
document that contains  
all the commands to  
build an image.



## 4

Build the **Dockerfile**

We will build our image with the Dockerfile already defined. The name of our image will be "**my-website**"



```
docker build -t my-website .
```

We use the dot ".", to indicate that our Dockerfile is in the current directory.



## 5

Run the **Container**

Now, let's create the container for our website. Our website will be published in port **8080** of our host. And the name of our container will be "**my-website**".

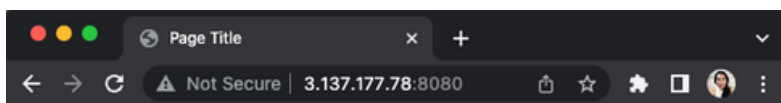


```
docker run --rm --name my-website -d -p 8080:80 my-website
```

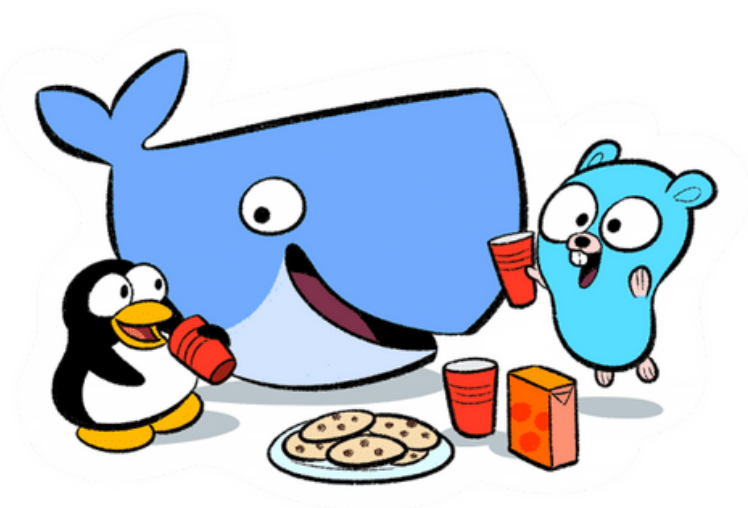
## 6

*Check the website on the browser*

*We open the browser and, see our static webpage in port 8080.*



**Hello Docker Community!**



If you modify your web page, you have to build the Dockerfile and run the container again!

Steps: **4** **5**

