

Dockerfile Commands & Example

Command	Description
ADD	Copies a file from the host system onto the container
CMD	The command that runs when the container starts
ENTRYPOINT	
ENV	Sets an environment variable in the new container
EXPOSE	Opens a port for linked containers
FROM	The base image to use in the build. This is mandatory and must be the first command in the file.
MAINTAINER	An optional value for the maintainer of the script
ONBUILD	A command that is triggered when the image in the Dockerfile is used as a base for another image
RUN	Executes a command and save the result as a new layer
USER	Sets the default user within the container
VOLUME	Creates a shared volume that can be shared among containers or by the host machine
WORKDIR	Set the default working directory for the container

Once you've created a Dockerfile and added all your instructions, you can use it to build an image using the docker build command. The format for this command is:

```
docker build [OPTIONS] PATH | URL | -
```

The build command results in a new image that you can start using docker run, just like any other image. Each line in the Dockerfile will correspond to a layer in the images' commit history.

Example:

Let us create our own nginx image and then create a container and check if we are able to access it over port 80 :

Create a directory:

```
mkdir dirname
```

Login to the directory:

```
cd dirname
```

Create a Dockerfile:

```
vim Dockerfile
```

```
FROM ubuntu:latest
```

```
#
```

```
# Identify the maintainer of an image
```

```
LABEL maintainer="myname@somecompany.com"
```

```
#
```

```
# Update the image to the latest packages
```

```
RUN apt-get update && apt-get upgrade -y
```

```
#
```

```
# Install NGINX to test.
```

```
RUN apt-get install nginx -y
```

```
#
```

```
# Expose port 80
```

```
EXPOSE 80
```

```
#
```

```
# Last is the actual command to start up NGINX within our Container
```

```
CMD ["nginx", "-g", "daemon off;"]
```

Save the file and exit

```
:wq
```

Build the new image using the command `docker build <path>`. Path refers to the directory containing the Dockerfile.

```
docker build -t imagename .
```

Check if image is created :

```
docker images
```

Create a container :

```
docker run -d imagename
```

Check if container has been created:

```
docker ps
```

Login to the container and perform the below steps

```
docker exec -it containername or containerid /bin/sh
```

```
apt-get install iputils-ping
```

```
ping google.com
```

```
exit
```

Get the container ip :

```
docker inspect containername or containerid
```

Access the container ip from docker host:

```
curl container ip
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Welcome to nginx!</title>
```

```
<style>
```

```
  body {
```

```
    width: 35em;
```

```
    margin: 0 auto;
```

```
    font-family: Tahoma, Verdana, Arial, sans-serif;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to nginx!</h1>
```

```
<p>If you see this page, the nginx web server is successfully installed and  
working. Further configuration is required.</p>
```

```
<p>For online documentation and support please refer to
```

```
<a href="http://nginx.org/">nginx.org</a>.<br/>
```

```
Commercial support is available at
```

```
<a href="http://nginx.com/">nginx.com</a>.</p>
```

<p>Thank you for using nginx.</p>

</body>

</html>