# Docker - File

In the earlier chapters, we have seen the various Image files such as Centos which get downloaded from **Docker hub** from which you can spin up containers. An example is again shown below.



If we use the Docker **images** command, we can see the existing images in our system. From the above screenshot, we can see that there are two images: **centos** and **nsenter**.

But Docker also gives you the capability to create your own Docker images, and it can be done with the help of **Docker Files**. A Docker File is a simple text file with instructions on how to build your images.

The following steps explain how you should go about creating a Docker File.

**Step 1** − Create a file called **Docker File** and edit it using **vim**. Please note that the name of the file has to be "Dockerfile" with "D" as capital.



**Step 2** − Build your Docker File using the following instructions.

#This is a sample Image

FROM ubuntu

MAINTAINER demousr@gmail.com

RUN apt-get update

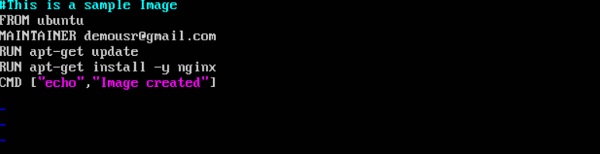
RUN apt-get install –y nginx

CMD [“echo”,”Image created”]

The following points need to be noted about the above file −

* The first line "#This is a sample Image" is a comment. You can add comments to the Docker File with the help of the **#** command
* The next line has to start with the **FROM** keyword. It tells docker, from which base image you want to base your image from. In our example, we are creating an image from the **ubuntu** image.
* The next command is the person who is going to maintain this image. Here you specify the **MAINTAINER** keyword and just mention the email ID.
* The **RUN** command is used to run instructions against the image. In our case, we first update our Ubuntu system and then install the nginx server on our **ubuntu** image.
* The last command is used to display a message to the user.

**Step 3** − Save the file. In the next chapter, we will discuss how to build the image.



# Docker - Building Files

We created our Docker File in the last chapter. It’s now time to build the Docker File. The Docker File can be built with the following command −

docker build

Let’s learn more about this command.

docker build

This method allows the users to build their own Docker images.

Syntax

docker build -t ImageName:TagName dir

Options

* **-t** − is to mention a tag to the image
* **ImageName** − This is the name you want to give to your image.
* **TagName** − This is the tag you want to give to your image.
* **Dir** − The directory where the Docker File is present.

Return Value

None

Example

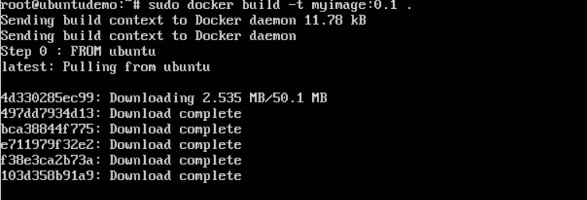
sudo docker build –t myimage:0.1 .

Here, **myimage** is the name we are giving to the Image and **0.1** is the tag number we are giving to our image.

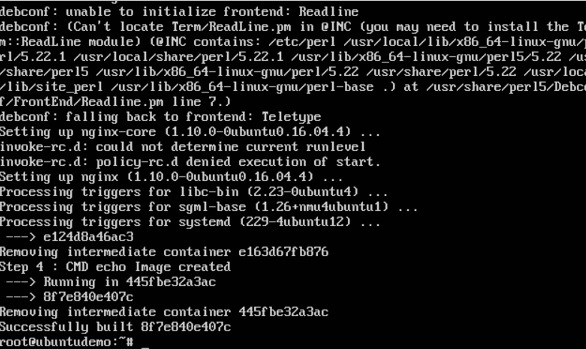
Since the Docker File is in the present working directory, we used "." at the end of the command to signify the present working directory.

Output

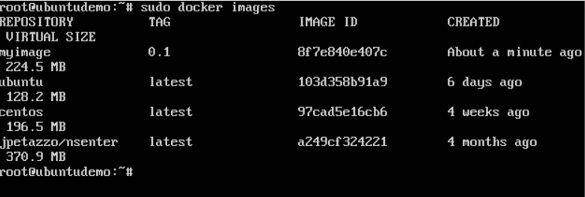
From the output, you will first see that the Ubuntu Image will be downloaded from Docker Hub, because there is no image available locally on the machine.



Finally, when the build is complete, all the necessary commands would have run on the image.



You will then see the successfully built message and the ID of the new Image. When you run the Docker **images command**, you would then be able to see your new image.



You can now build containers from your new Image.

Docker - Instruction Commands

Docker has a host of instruction commands. These are commands that are put in the Docker File. Let’s look at the ones which are available.

CMD Instruction

This command is used to execute a command at runtime when the container is executed.

Syntax

CMD command param1

Options

* **command** − This is the command to run when the container is launched.
* **param1** − This is the parameter entered to the command.

Return Value

The command will execute accordingly.

Example

In our example, we will enter a simple **Hello World** echo in our Docker File and create an image and launch a container from it.

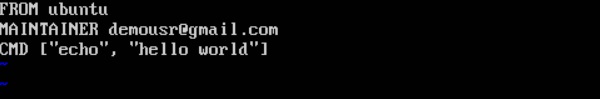
**Step 1** − Build the Docker File with the following commands −

FROM ubuntu

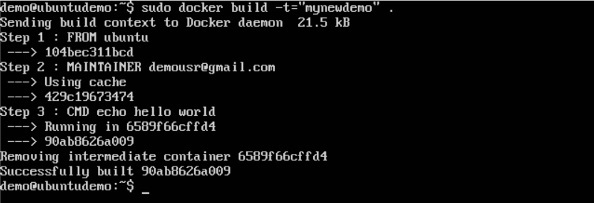
MAINTAINER demousr@gmail.com

CMD [“echo” , “hello world”]

Here, the CMD is just used to print **hello world**.



**Step 2** − Build the image using the Docker **build** command.



**Step 3** − Run a container from the image.



ENTRYPOINT

This command can also be used to execute commands at runtime for the container. But we can be more flexible with the ENTRYPOINT command.

Syntax

ENTRYPOINT command param1

Options

* **command** − This is the command to run when the container is launched.
* **param1** − This is the parameter entered into the command.

Return Value

The command will execute accordingly.

Example

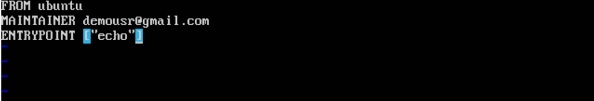
Let’s take a look at an example to understand more about ENTRYPOINT. In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

**Step 1** − Build the Docker File with the following commands −

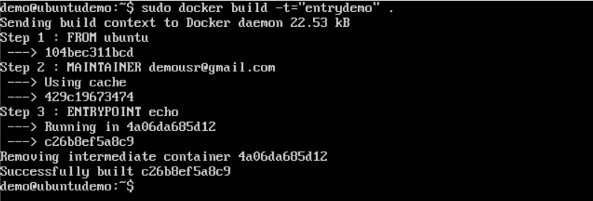
FROM ubuntu

MAINTAINER demousr@gmail.com

ENTRYPOINT [“echo”]



**Step 2** − Build the image using the Docker **build** command.



**Step 3** − Run a container from the image.



ENV

This command is used to set environment variables in the container.

Syntax

ENV key value

Options

* **Key** − This is the key for the environment variable.
* **value** − This is the value for the environment variable.

Return Value

The command will execute accordingly.

Example

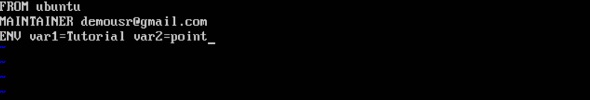
In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

**Step 1** − Build the Docker File with the following commands −

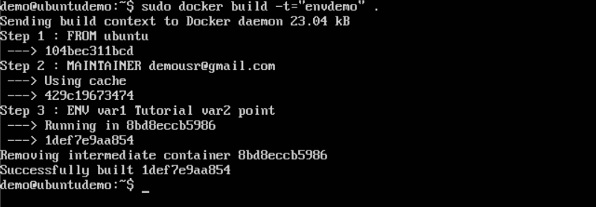
FROM ubuntu

MAINTAINER demousr@gmail.com

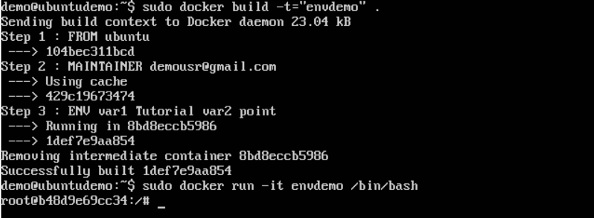
ENV var1=Tutorial var2=point



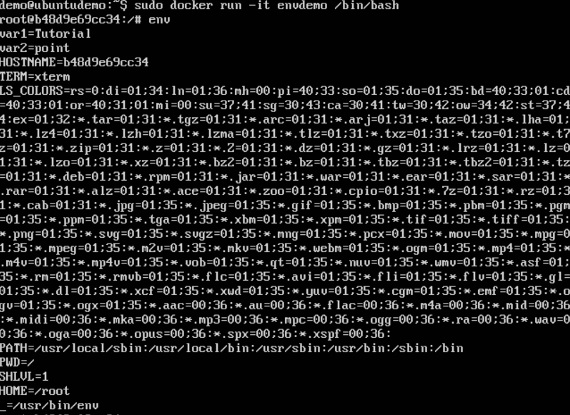
**Step 2** − Build the image using the Docker **build** command.



**Step 3** − Run a container from the image.



**Step 4** − Finally, execute the **env** command to see the environment variables.



WORKDIR

This command is used to set the working directory of the container.

Syntax

WORKDIR dirname

Options

* **dirname** − The new working directory. If the directory does not exist, it will be added.

Return Value

The command will execute accordingly.

Example

In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

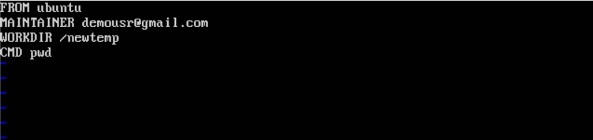
**Step 1** − Build the Docker File with the following commands −

FROM ubuntu

MAINTAINER demousr@gmail.com

WORKDIR /newtemp

CMD pwd



**Step 2** − Build the image using the Docker **build** command.



**Step 3** − Run a container from the image.

