Lesson 02 Demo 02

Creating a Docker Image

Objective: To create a Docker image using a Docker file, which installs and configures a Nginx web server with a custom welcome message

Tools required: Docker

Prerequisites: None

Steps to be followed:

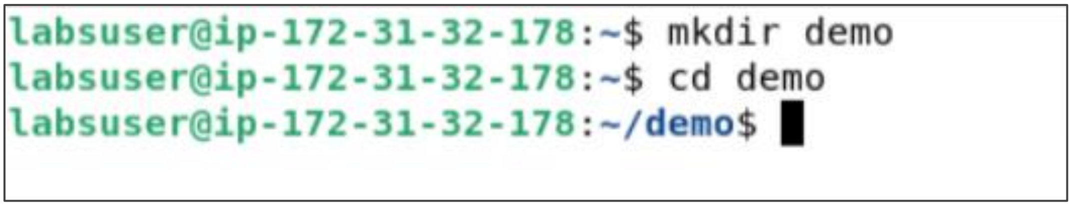
1. Create a Docker image using the Docker file

Step 1: Create a Docker image using the Docker file

1.1 Use the following command to create a directory:

mkdir demo

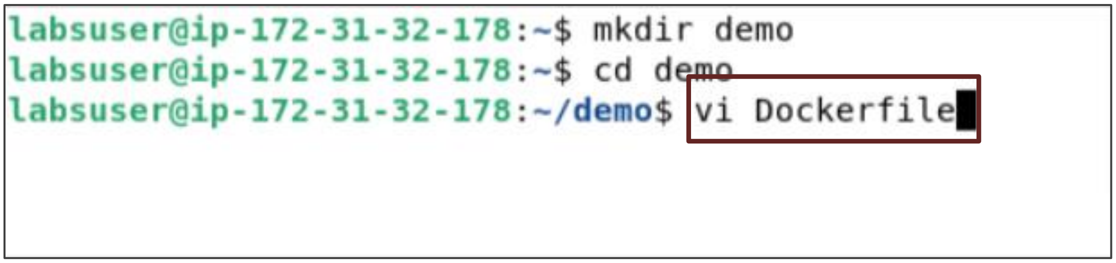
cd demo



```
labsuser@ip-172-31-32-178:~$ mkdir demo
labsuser@ip-172-31-32-178:~$ cd demo
labsuser@ip-172-31-32-178:~/demo$
```

1.2 Use the following command to create the Docker file:

vi Dockerfile



```
labsuser@ip-172-31-32-178:~$ mkdir demo
labsuser@ip-172-31-32-178:~$ cd demo
labsuser@ip-172-31-32-178:~/demo$ vi Dockerfile
```

1.3 Add the following code to the **Dockerfile**:

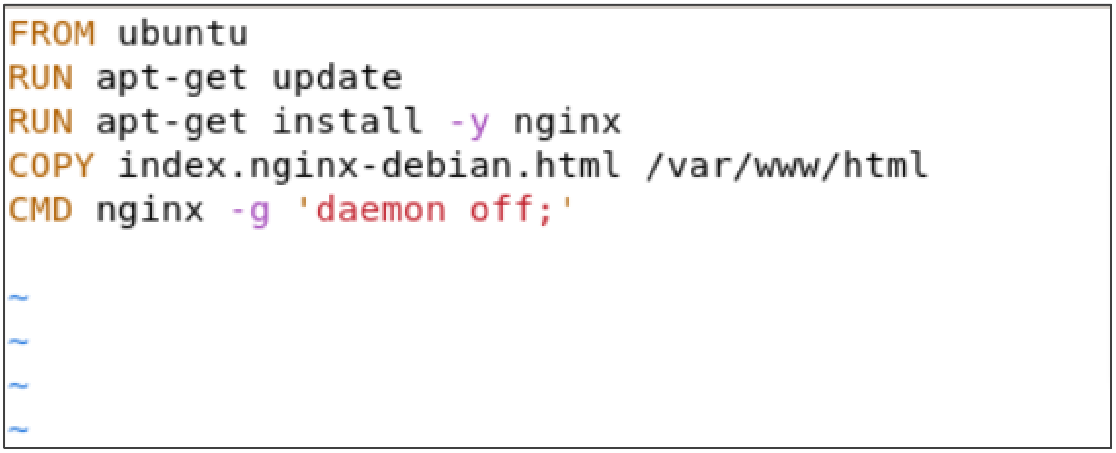
FROM ubuntu

RUN apt-get update

RUN apt-get install -y nginx

COPY index.nginx-debian.html /var/www/html

CMD nginx -g 'daemon off;'

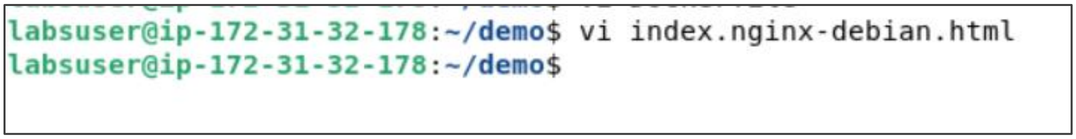


```
FROM ubuntu
RUN apt-get update
RUN apt-get install -y nginx
COPY index.nginx-debian.html /var/www/html
CMD nginx -g 'daemon off;'


~
~
~
~
```

1.4 Use the following command to create another file in the same directory:

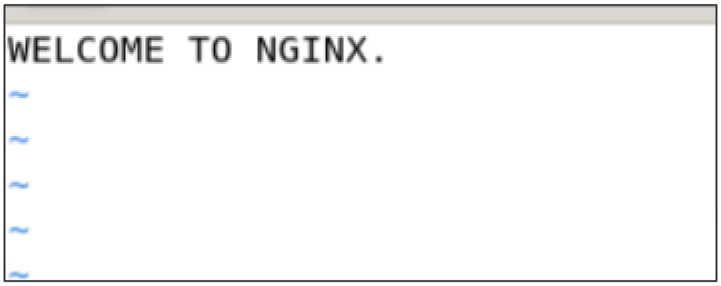
vi index.nginx-debian.html



```
labsuser@ip-172-31-32-178:~/demo$ vi index.nginx-debian.html
labsuser@ip-172-31-32-178:~/demo$
```

1.5 Add the following welcome message to the index file:

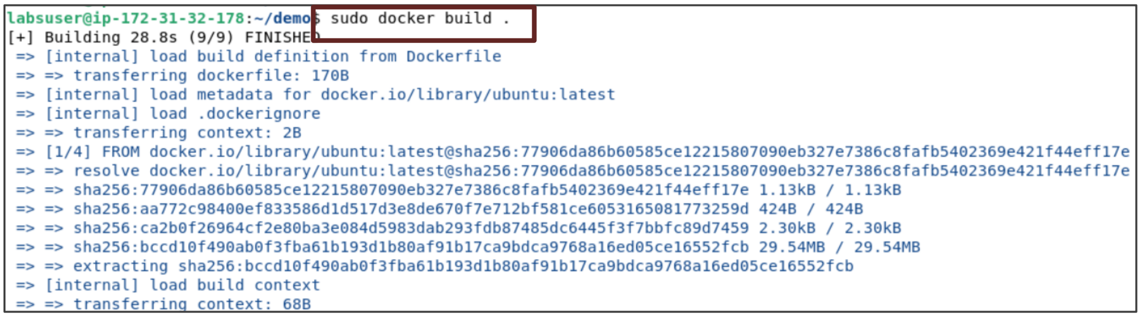
WELCOME TO NGINX.



```
WELCOME TO NGINX.
~
~
~
~
~
```

1.6 Use the following command to execute the Docker file:

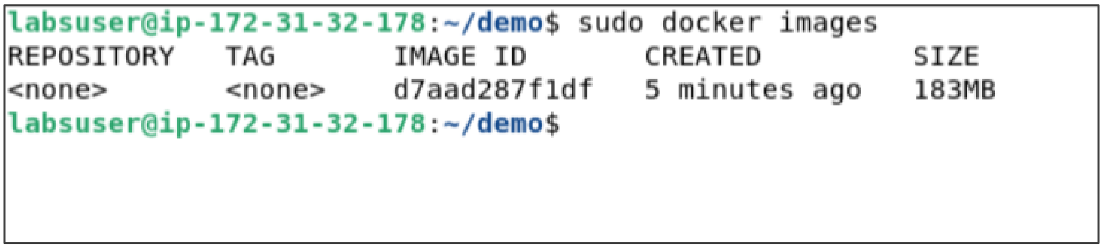
sudo docker build .



```
labsuser@ip-172-31-32-178:~/demo$ sudo docker build .
[+] Building 28.8s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 170B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
 => => resolve docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
 => => sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e 1.13kB / 1.13kB
 => => sha256:aa772c98400ef833586d1d517d3e8de670f7e712bf581ce6053165081773259d 424B / 424B
 => => sha256:ca2b0f26964cf2e80ba3e084d5983dab293fdb87485dc6445f3f7bbfc89d7459 2.30kB / 2.30kB
 => => sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce16552fcb 29.54MB / 29.54MB
 => => extracting sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce16552fcb
 => [internal] load build context
 => => transferring context: 68B
```

1.7 Execute the given command to display the images. This will confirm that the image has been created.

sudo docker images



```
labsuser@ip-172-31-32-178:~/demo$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
<none>       <none>    d7aad287f1df   5 minutes ago   183MB
labsuser@ip-172-31-32-178:~/demo$
```

By following these steps, you have successfully created a Docker image containing a Nginx web server with a custom welcome message using a Docker file. This image can then be used to spin up containers with the configured web server environment.