Lesson 07 Demo 03

Creating a Docker Image

Objective: To generate a Docker image by creating and executing a Dockerfile for better deployment

Tools required: Docker configuration

Pre-requisites: Ubuntu configuration and Docker

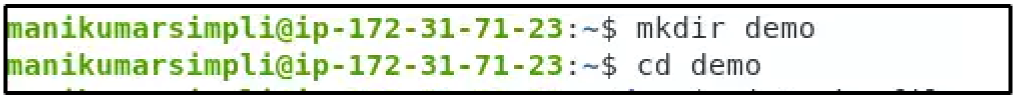
Steps to be followed:

1. Create a Dockerfile
2. Execute the Dockerfile

Step 1: Create a Dockerfile

1.1 Create a directory by running the following commands:

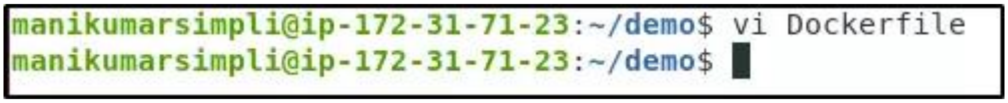
mkdir demo
cd demo



```
manikumarsimpli@ip-172-31-71-23:~$ mkdir demo
manikumarsimpli@ip-172-31-71-23:~$ cd demo
```

1.2 Create the Dockerfile by running the following command:

vi Dockerfile



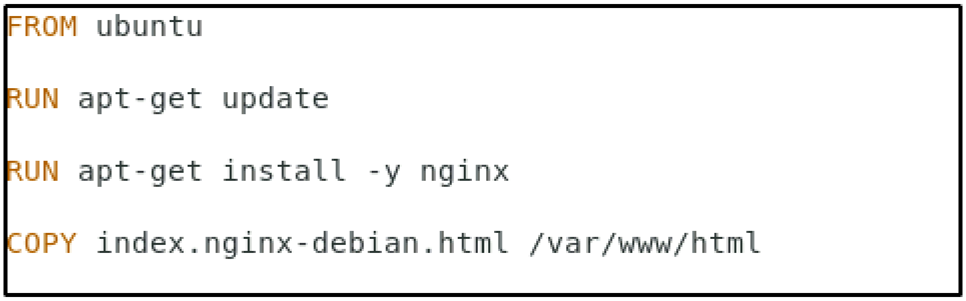
```
manikumarsimpli@ip-172-31-71-23:~/demo$ vi Dockerfile
manikumarsimpli@ip-172-31-71-23:~/demo$
```

1.3 Add the following code snippet 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;'

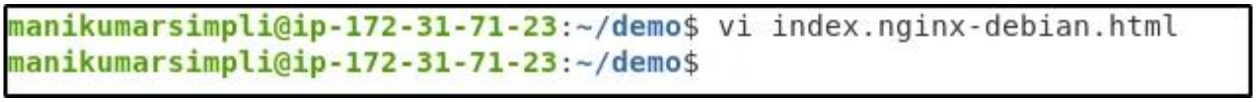


```
FROM ubuntu

RUN apt-get update

RUN apt-get install -y nginx

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

> **Note**: After writing the above code in the *Dockerfile*, press **ESC** button and enter “**:wq”** to save the file and exit the editor.

1.4 Create another file in the same directory

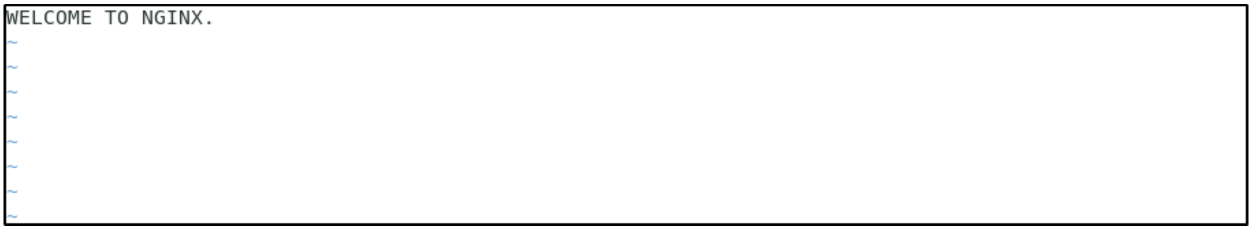
vi index.nginx-debian.html



```
manikumarsimpli@ip-172-31-71-23:~/demo$ vi index.nginx-debian.html
manikumarsimpli@ip-172-31-71-23:~/demo$
```

1.5 Add the following welcome message to the index file:

WELCOME TO NGINX.

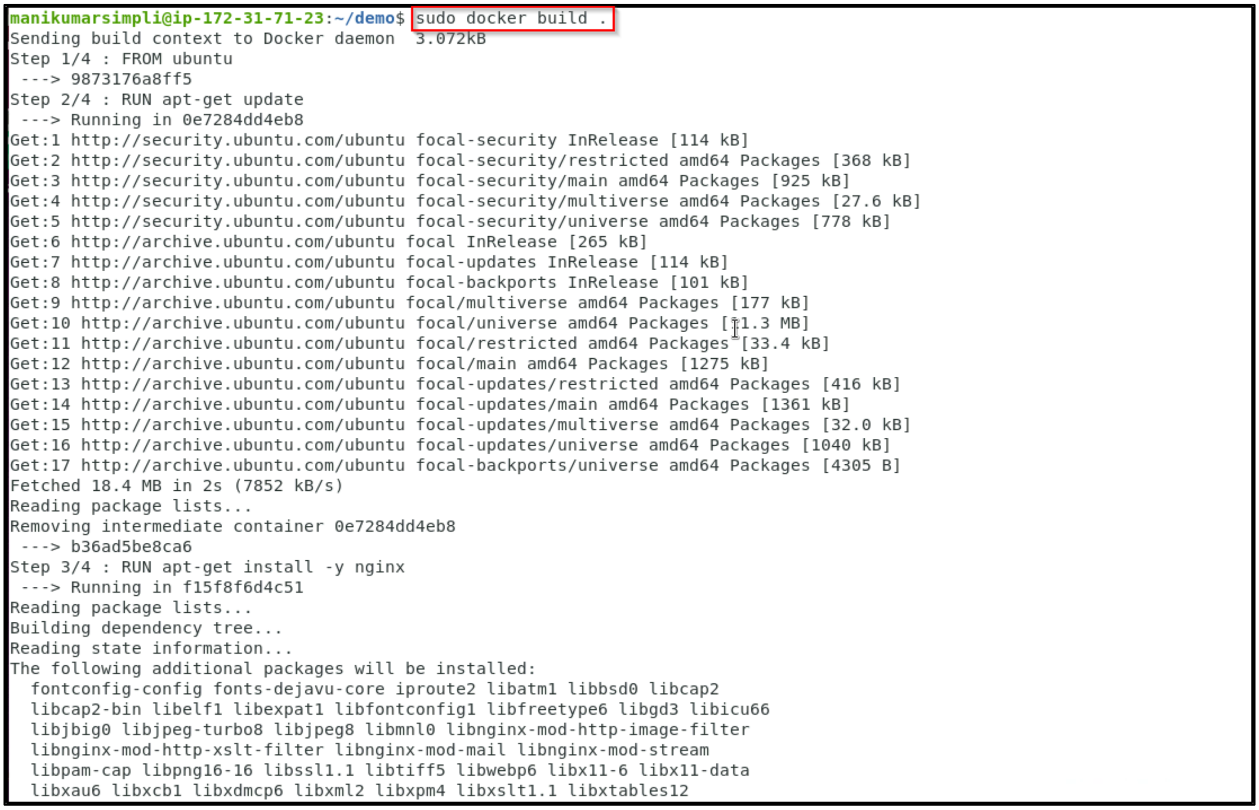


```
WELCOME TO NGINX.
~
~
~
~
~
~
~
~
```

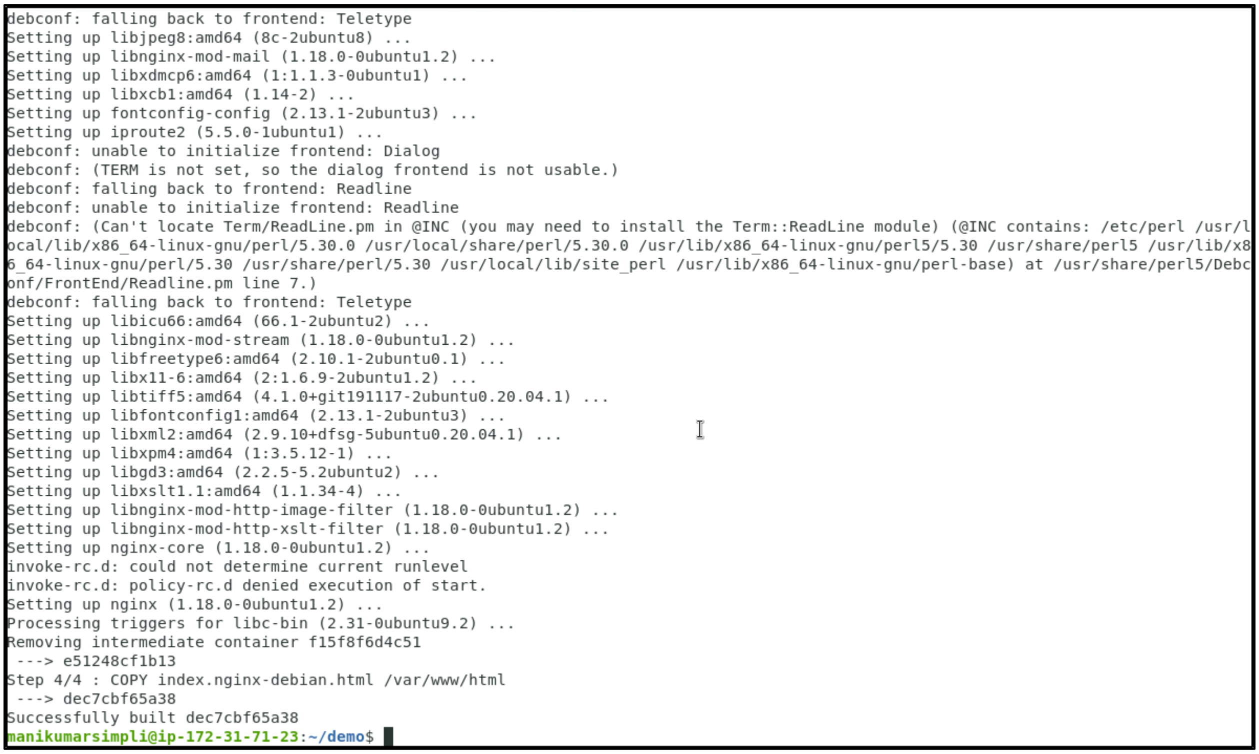
> **Note**: After writing the above code in the *html* file, press **ESC** button and enter **:wq** to save the file and exit the editor.

Step 2: Execute the file

2.1 Execute the Dockerfile (Note that there is a space between **build** and **“.”**.)
sudo docker build



```
manikumarsimpli@ip-172-31-71-23:~/demo$ sudo docker build .
Sending build context to Docker daemon  3.072kB
Step 1/4 : FROM ubuntu
 ---> 9873176a8ff5
Step 2/4 : RUN apt-get update
 ---> Running in 0e7284dd4eb8
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [368 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [925 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.6 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [778 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [416 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1361 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.0 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1040 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [4305 B]
Fetched 18.4 MB in 2s (7852 kB/s)
Reading package lists...
Removing intermediate container 0e7284dd4eb8
 ---> b36ad5be8ca6
Step 3/4 : RUN apt-get install -y nginx
 ---> Running in f15f8f6d4c51
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core iproute2 libatm1 libbsd0 libcap2
  libcap2-bin libelf1 libexpat1 libfontconfig1 libfreetype6 libgd3 libicu66
  libjbig0 libjpeg-turbo8 libjpeg8 libmnl0 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libpam-cap libpng16-16 libssl1.1 libtiff5 libwebp6 libx11-6 libx11-data
  libxau6 libxcb1 libxdmcp6 libxml2 libxpm4 libxslt1.1 libxtables12
```

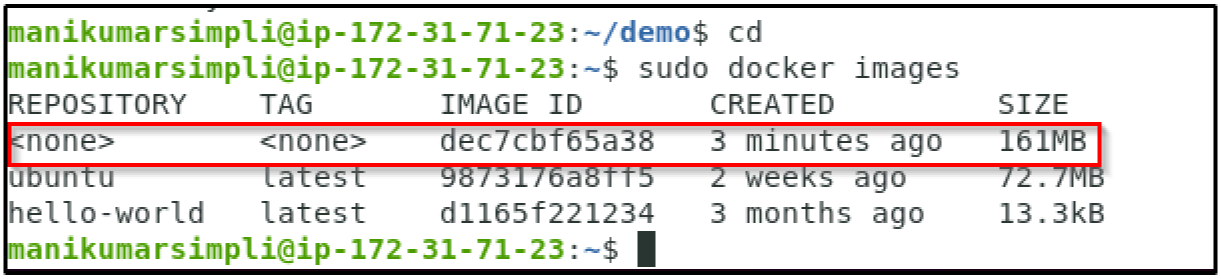


```
debconf: falling back to frontend: Teletype
Setting up libjpeg8:amd64 (8c-2ubuntu8) ...
Setting up libnginx-mod-mail (1.18.0-0ubuntu1.2) ...
Setting up libxdmcp6:amd64 (1:1.1.3-0ubuntu1) ...
Setting up libxcb1:amd64 (1.14-2) ...
Setting up fontconfig-config (2.13.1-2ubuntu3) ...
Setting up iproute2 (5.5.0-1ubuntu1) ...
debconf: unable to initialize frontend: Dialog
debconf: (TERM is not set, so the dialog frontend is not usable.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/l
ocal/lib/x86_64-linux-gnu/perl/5.30.0 /usr/local/share/perl/5.30.0 /usr/lib/x86_64-linux-gnu/perl5/5.30 /usr/share/perl5 /usr/lib/x8
6_64-linux-gnu/perl/5.30 /usr/share/perl/5.30 /usr/local/lib/site_perl /usr/lib/x86_64-linux-gnu/perl-base) at /usr/share/perl5/Debc
onf/FrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
Setting up libicu66:amd64 (66.1-2ubuntu2) ...
Setting up libnginx-mod-stream (1.18.0-0ubuntu1.2) ...
Setting up libfreetype6:amd64 (2.10.1-2ubuntu0.1) ...
Setting up libx11-6:amd64 (2:1.6.9-2ubuntu1.2) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.1) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up libxml2:amd64 (2.9.10+dfsg-5ubuntu0.20.04.1) ...
Setting up libxpm4:amd64 (1:3.5.12-1) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2) ...
Setting up libxslt1.1:amd64 (1.1.34-4) ...
Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.2) ...
Setting up libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.2) ...
Setting up nginx-core (1.18.0-0ubuntu1.2) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Setting up nginx (1.18.0-0ubuntu1.2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Removing intermediate container f15f8f6d4c51
 ---> e51248cf1b13
Step 4/4 : COPY index.nginx-debian.html /var/www/html
 ---> dec7cbf65a38
Successfully built dec7cbf65a38
manikumarsimpli@ip-172-31-71-23:~/demo$
```

2.2 Navigate to the root folder and list the images to check the newly created Docker image

cd

sudo docker images



```
manikumarsimpli@ip-172-31-71-23:~/demo$ cd
manikumarsimpli@ip-172-31-71-23:~$ sudo docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
<none>         <none>    dec7cbf65a38   3 minutes ago    161MB
ubuntu         latest    9873176a8ff5   2 weeks ago      72.7MB
hello-world    latest    d1165f221234   3 months ago     13.3kB
manikumarsimpli@ip-172-31-71-23:~$
```

By following these steps, you have successfully generated a Docker image by creating and executing a Dockerfile for better deployment.