

Module 7: Working with Docker

Demo 1

edureka!

edureka!

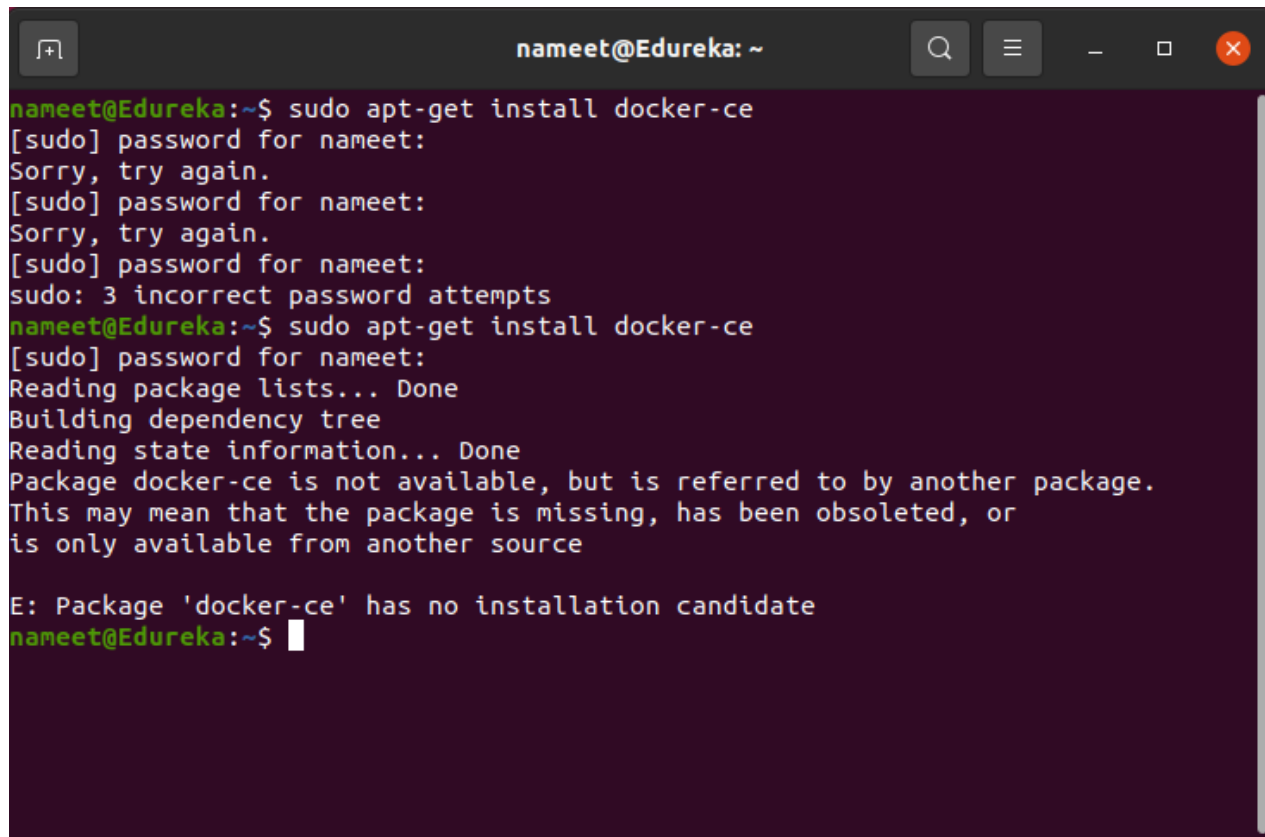
© Brain4ce Education Solutions Pvt. Ltd.

Demo 1: Installing Docker and Working with Docker

Step 1: Open Terminal in Linux

And type: `sudo apt-get install docker-ce`

This cmd checks if you have a prior docker installation



```
nameet@Edureka: ~  
nameet@Edureka:~$ sudo apt-get install docker-ce  
[sudo] password for nameet:  
Sorry, try again.  
[sudo] password for nameet:  
Sorry, try again.  
[sudo] password for nameet:  
sudo: 3 incorrect password attempts  
nameet@Edureka:~$ sudo apt-get install docker-ce  
[sudo] password for nameet:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package docker-ce is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
  
E: Package 'docker-ce' has no installation candidate  
nameet@Edureka:~$
```

If the above cmd doesn't work, then type the following:

`docker install`

`sudo apt install docker.io`

```
nameet@Edureka: ~  
is only available from another source  
E: Package 'docker-ce' has no installation candidate  
nameet@Edureka:~$ docker install  
Command 'docker' not found, but can be installed with:  
  
sudo snap install docker      # version 20.10.11, or  
sudo apt install docker.io    # version 20.10.7-0ubuntu5~20.04.2  
  
See 'snap info docker' for additional versions.  
  
nameet@Edureka:~$ sudo apt install docker.io  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi  
  libgstreamer-plugins-bad1.0-0 libva-wayland2  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan  
Suggested packages:  
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
```

The following cmd checks whether all the dependencies are installed or not

Sudo snap install docker

```
nameet@Edureka: ~  
nameet@Edureka:~$ sudo snap install docker  
docker 20.10.11 from Canonical* installed  
nameet@Edureka:~$
```

Type the cmd to check version

Docker --version

And run a image from pulled from docker hub

sudo docker run hello-world

```
nameet@Edureka: ~  
nameet@Edureka:~$ docker --version  
Docker version 20.10.7, build 20.10.7-0ubuntu5-20.04.2  
nameet@Edureka:~$ sudo docker run hello-world  
[sudo] password for nameet:  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:97a379f4f88575512824f3b352bc03cd75e239179eea0fecc38e597b2209f49a  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
nameet@Edureka:~$
```

To check the images in your system

sudo docker images

```
nameet@Edureka:~$ sudo docker images  
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE  
hello-world     latest     feb5d9fea6a5  4 months ago  13.3kB  
nameet@Edureka:~$
```

To check or display all containers pulled so far

sudo docker ps -a

```
nameet@Edureka:~$ sudo docker ps -a  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES  
63d02f3fdf31   hello-world  "/hello"                6 minutes ago  Exited (0) 6 minutes ago          eager_booth  
nameet@Edureka:~$
```

To display only running container

```
nameet@Edureka:~$ sudo docker ps  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES  
nameet@Edureka:~$
```

To run a docker image

A screenshot of a Linux desktop environment with a dark theme. The top panel shows system icons on the left and date/time (Feb 16 03:21) on the right. A dock on the left contains icons for Firefox, Mail, Files, Music, Videos, and Applications. The main area is a terminal window titled "Terminal". The prompt is "nameet@Edureka: ~". The user enters the command "sudo docker run docker/whalesay cowsay Hello Edureka". The output shows the process of pulling the "docker/whalesay:latest" image from Docker Hub, listing various layers being pulled, and displaying the SHA256 digest. After the pull is complete, the terminal shows the output of the "cowsay" command: "< Hello Edureka >" followed by a ASCII art cow saying "Hello Edureka". The prompt returns to "nameet@Edureka:~\$".

```
nameet@Edureka:~$ sudo docker run docker/whalesay cowsay Hello Edureka
[sudo] password for nameet:
Unable to find image 'docker/whalesay:latest' locally
latest: Pulling from docker/whalesay
Image docker.io/docker/whalesay:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed371d2cf: Pull complete
Digest: sha256:178598e51a26abb958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest

< Hello Edureka >
      \   ^__^
       (oo)\_____
          (__)\       )\/\
              ||----w |
              ||     ||

nameet@Edureka:~$
```

Running java-app

Open Your terminal in the folder where Hello.java file is present or make a directory

Hello.java:

```
public class Hello
```

```
{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
```

Dockerfile:

FROM openjdk:8

```
COPY . /var/www/java
```

WORKDIR /var/www/java

RUN javac Hello.java

```
CMD ["java", "Hello"]
```

Java-app is built into image

```
nameet@Edureka:~/Documents$ mkdir java-app
nameet@Edureka:~/Documents$ cd java-app
nameet@Edureka:~/Documents/java-app$ sudo docker build -t java-app .
[sudo] password for nameet:
```

Run the program:

```
nameet@Edureka:~/Documents/java-app$ sudo docker build -t java-app .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM java:8
--> d23bdf5b1b1b
Step 2/5 : COPY . /var/www/java
--> 1a972c324c61
Step 3/5 : WORKDIR /var/www/java
--> Running in a1ed516b7acd
Removing intermediate container a1ed516b7acd
--> 80303972b0a6
Step 4/5 : RUN javac Hello.java
--> Running in e0a932774972
Removing intermediate container e0a932774972
--> b20c853734d7
Step 5/5 : CMD ["java", "Hello"]
--> Running in 8a7d1819eed
Removing intermediate container 8a7d1819eed
--> 1ea36f44944a
Successfully built 1ea36f44944a
Successfully tagged java-app:latest
nameet@Edureka:~/Documents/java-app$ sudo docker run java-app
Hello World
nameet@Edureka:~/Documents/java-app$
```

edureka!