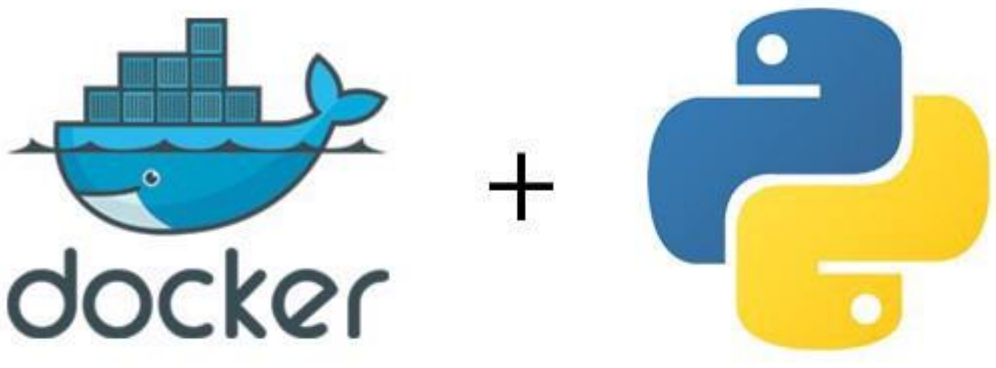
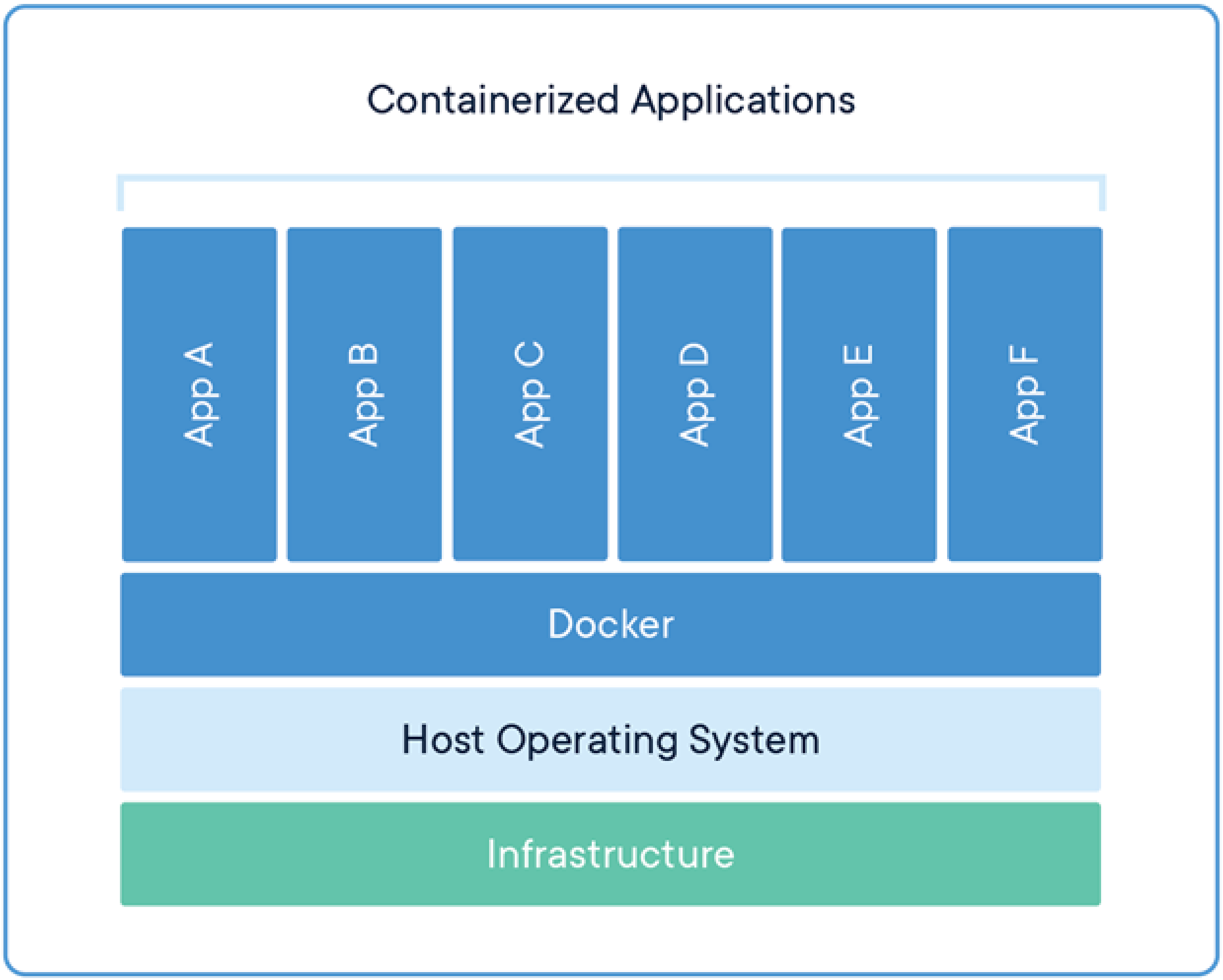


Deployment of Machine Learning Model Inside Docker Container

What is Docker?

A **Docker container** is an open-source software development platform. Its main benefit is to package applications in **containers**, allowing them to be portable to any system running a Linux or Windows operating system (OS). ... While it is a major player in the **container** field, **Docker** is only one form of **container technology**.

NOTE:



In this article, we will replicate a machine learning model into a Docker container and write a Python program to predict the output based on it. For this entire configuration, we will generate a Docker image.

Task Description:

- Create a new container using the CentOS image from Docker Hub.
- On top of the docker container, install the Python software.
- You must train a machine learning model in Container.

NOTE:

1. I have used RHEL 8 for this practical, you can also use any OS.

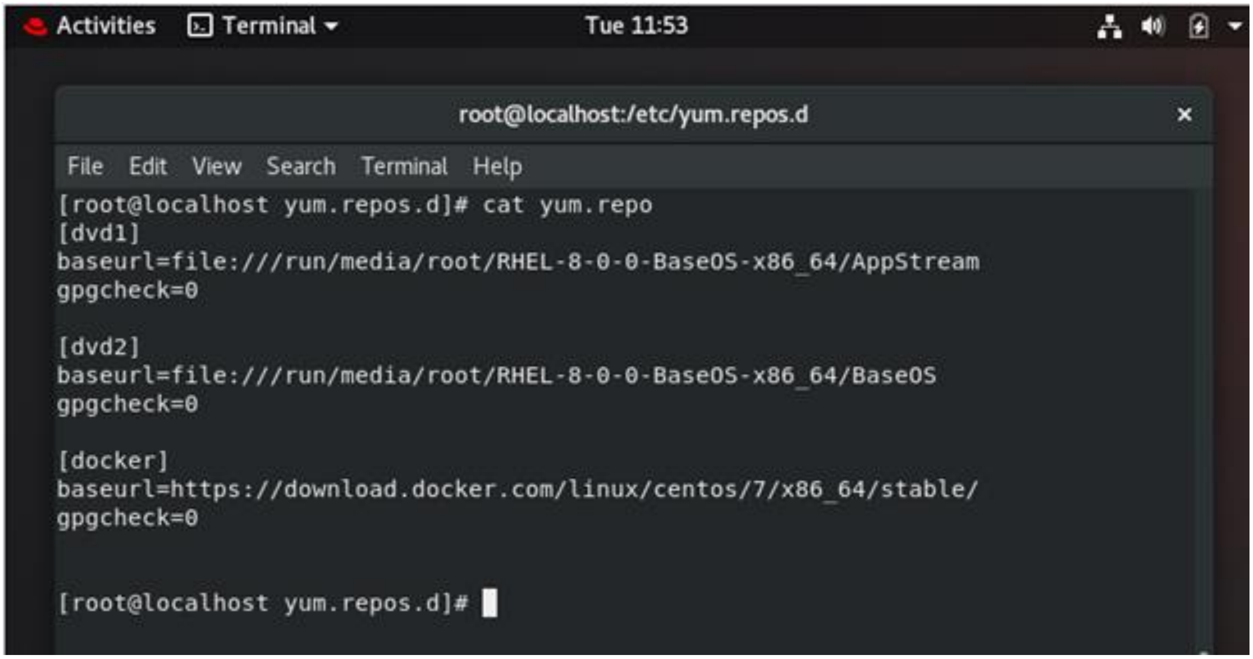
2. You can pull my docker image Anuddeeph/mldocker for reference.

Steps:

1) Install Docker.

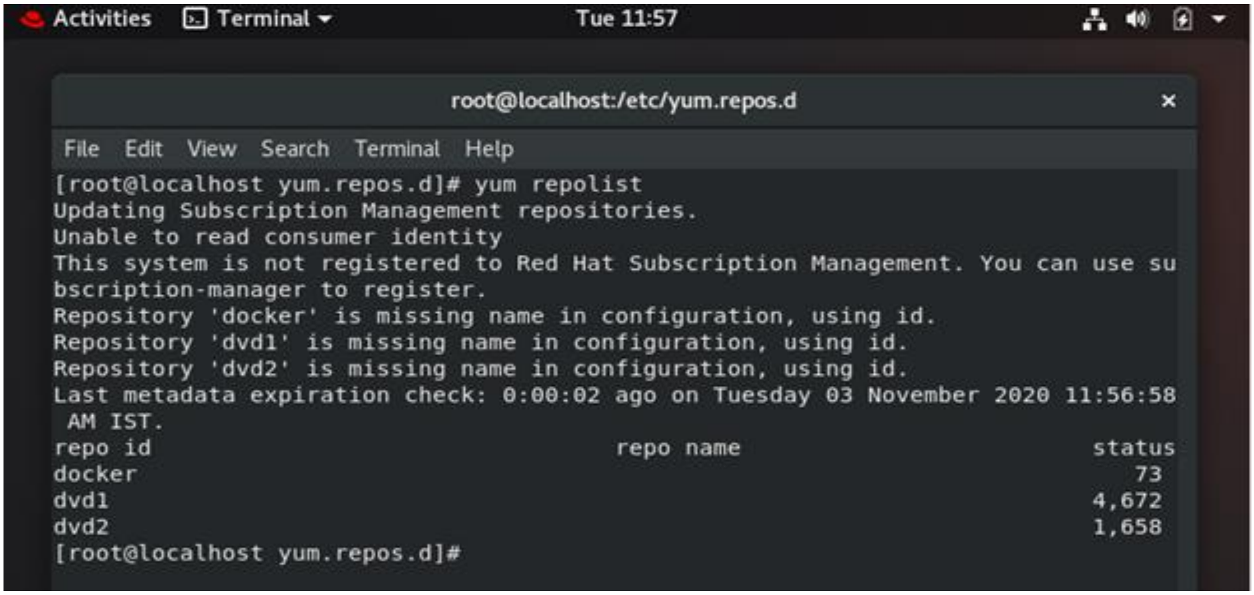
- As I am using Redhat 8, so I need to first configure yum and create docker repo for installing the Docker.

- For this, go to the location, /etc/yum.repos.d and then create a file for configure yum and docker


Activities
Terminal
Tue 11:53
root@localhost:/etc/yum.repos.d
File Edit View Search Terminal Help

```
[root@localhost yum.repos.d]# cat yum.repo
[dvd1]
baseurl=file:///run/media/root/RHEL-8-0-0-BaseOS-x86_64/AppStream
gpgcheck=0

[dvd2]
baseurl=file:///run/media/root/RHEL-8-0-0-BaseOS-x86_64/BaseOS
gpgcheck=0

[docker]
baseurl=https://download.docker.com/linux/centos/7/x86_64/stable/
gpgcheck=0


[root@localhost yum.repos.d]# 
```

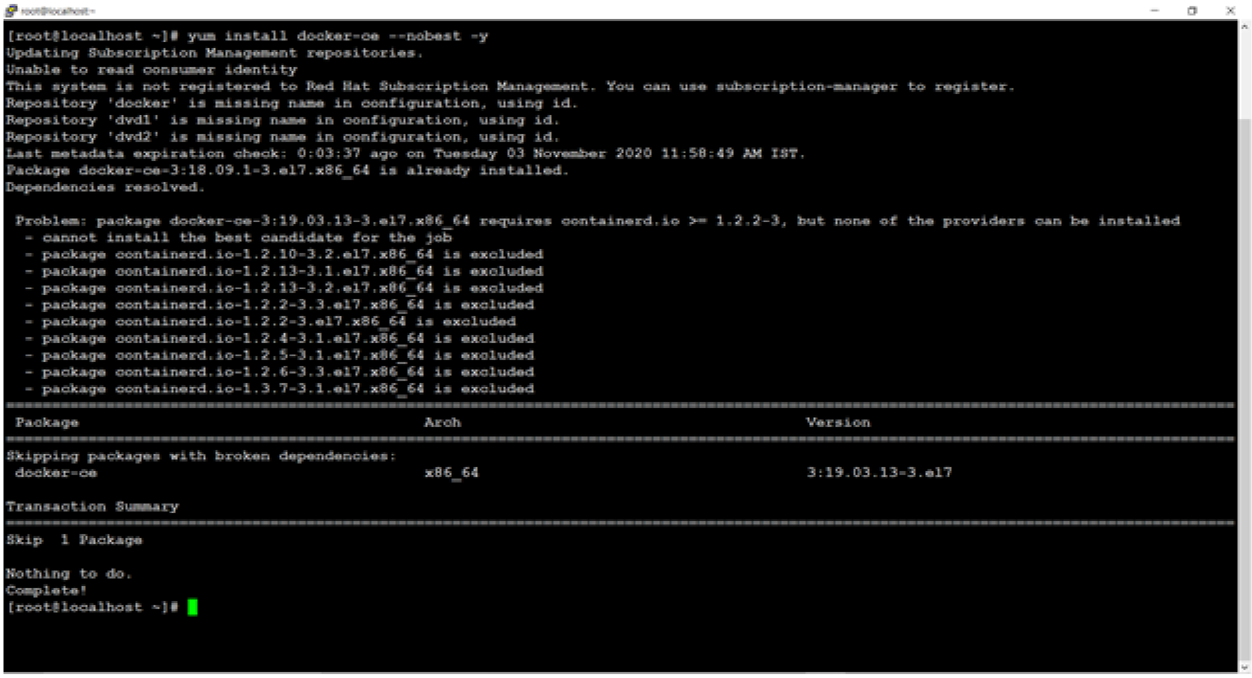
- Check the yum is properly configured or not using,” yum repolist”


Activities
Terminal
Tue 11:57
root@localhost:/etc/yum.repos.d
File Edit View Search Terminal Help

```
[root@localhost yum.repos.d]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use su
bscription-manager to register.
Repository 'docker' is missing name in configuration, using id.
Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
Last metadata expiration check: 0:00:02 ago on Tuesday 03 November 2020 11:56:58
 AM IST.
repo id                               repo name                             status
docker                                                                          73
dvd1                                                                         4,672
dvd2                                                                         1,658
[root@localhost yum.repos.d]#
```

- Its working fine!

- To install Docker-ce software, run: “yum install docker-ce –nobest -y”



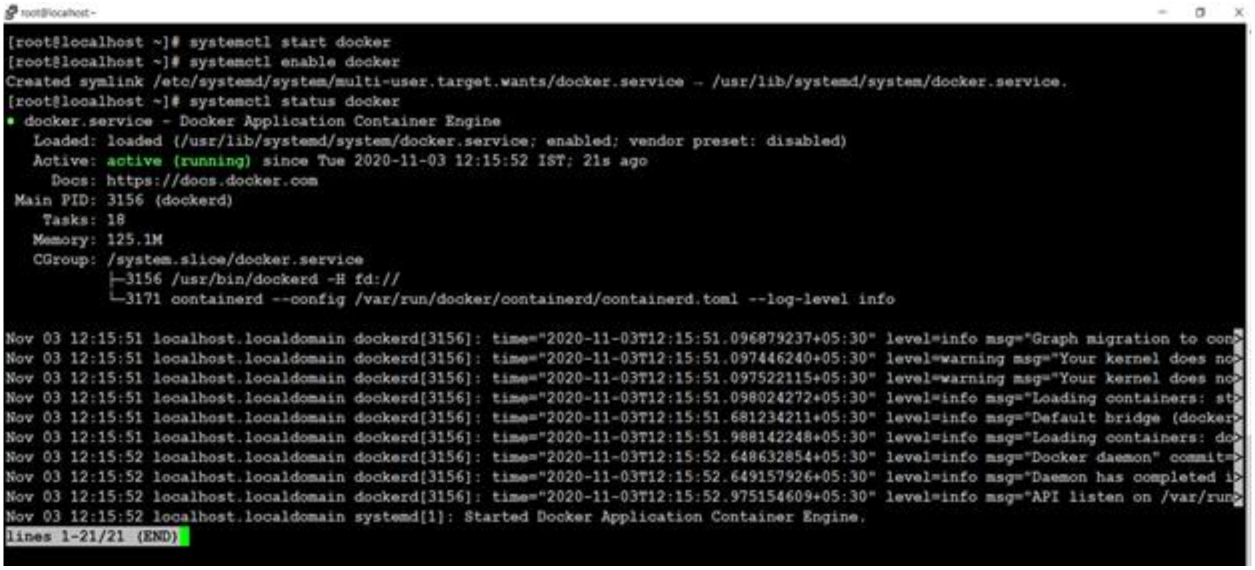
```
[root@localhost ~]# yum install docker-ce --nobest -y
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Repository 'docker' is missing name in configuration, using id.
Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
Last metadata expiration check: 0:03:37 ago on Tuesday 03 November 2020 11:58:49 AM IST.
Package docker-ce-3:18.09.1-3.el7.x86_64 is already installed.
Dependencies resolved.

 Problem: package docker-ce-3:19.03.13-3.el7.x86_64 requires containerd.io >= 1.2.2-3, but none of the providers can be installed
  - cannot install the best candidate for the job
  - package containerd.io-1.2.10-3.2.el7.x86_64 is excluded
  - package containerd.io-1.2.13-3.1.el7.x86_64 is excluded
  - package containerd.io-1.2.13-3.2.el7.x86_64 is excluded
  - package containerd.io-1.2.2-3.3.el7.x86_64 is excluded
  - package containerd.io-1.2.2-3.el7.x86_64 is excluded
  - package containerd.io-1.2.4-3.1.el7.x86_64 is excluded
  - package containerd.io-1.2.5-3.1.el7.x86_64 is excluded
  - package containerd.io-1.2.6-3.3.el7.x86_64 is excluded
  - package containerd.io-1.3.7-3.1.el7.x86_64 is excluded
=======================================================================================================================================
 Package                                      Arch                                      Version
=======================================================================================================================================
Skipping packages with broken dependencies:
 docker-ce                                    x86_64                                    3:19.03.13-3.el7

Transaction Summary
=======================================================================================================================================
Skip  1 Package

Nothing to do.
Complete!
[root@localhost ~]# 
```

- I have already installed docker before, Finally docker installed successfully.

- Start the Docker services using,

  ➔ Systemctl start docker

  ➔ Systemctl enable docker



```
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@localhost ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-11-03 12:15:52 IST; 21s ago
     Docs: https://docs.docker.com
 Main PID: 3156 (dockerd)
    Tasks: 18
   Memory: 125.1M
   CGroup: /system.slice/docker.service
           ├─3156 /usr/bin/dockerd -H fd://
           └─3171 containerd --config /var/run/docker/containerd/containerd.toml --log-level info

Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.096879237+05:30" level=info msg="Graph migration to con>
Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.097446240+05:30" level=warning msg="Your kernel does no>
Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.097522115+05:30" level=warning msg="Your kernel does no>
Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.098024272+05:30" level=info msg="Loading containers: st>
Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.681234211+05:30" level=info msg="Default bridge (docker>
Nov 03 12:15:51 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:51.988142248+05:30" level=info msg="Loading containers: do>
Nov 03 12:15:52 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:52.648632854+05:30" level=info msg="Docker daemon" commit=>
Nov 03 12:15:52 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:52.649157926+05:30" level=info msg="Daemon has completed i>
Nov 03 12:15:52 localhost.localdomain dockerd[3156]: time="2020-11-03T12:15:52.975154609+05:30" level=info msg="API listen on /var/run>
Nov 03 12:15:52 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
lines 1-21/21 (END)
```

- We require ISO file for installing any operating system. In Docker, we have docker images for installing any container.

- Go to **docker hub** and check for images.

2) Dockerfile

- First we need to specify the base image using the **FROM** keyword and then install python3 using the yum installer

```
FROM centos:latest
RUN yum install python3 -y
```

- we need to install the required python libraries using the pip installer and create a workspace for the python program.

```
 RUN pip3 install sklearn

 RUN pip3 install pandas

 RUN mkdir /MLpy_WS
```

- Copy the Salary dataset and the python code for creating the ML model which we had created earlier.

```
COPY Salary_Data.csv /MLpy_WS

COPY salary_pred.py/MLpy_WS
```

- Run the python program whenever the container is launched using this image using the CMD keyword

  ```
  CMD ["python3","/MLpy_WS/salary_pred.py"]
  ```

- The entire code for your reference

  ```
  FROM centos:latest

  RUN yum install python3 -y

  RUN pip3 install sklearn

  RUN pip3 install pandas
  ```

```
RUN pip3 install numpy

RUN mkdir /MLpy_WS

WORKDIR /MLpy_WS

COPY Salary_Data.csv /MLpy_WS
COPY ML_code.py /MLpy_WS

CMD ["python3","/MLpy_WS/ML_code.py"]
```

3) We will create a Simple Linear Regression model from a dataset containing the Years of Experience and Salary

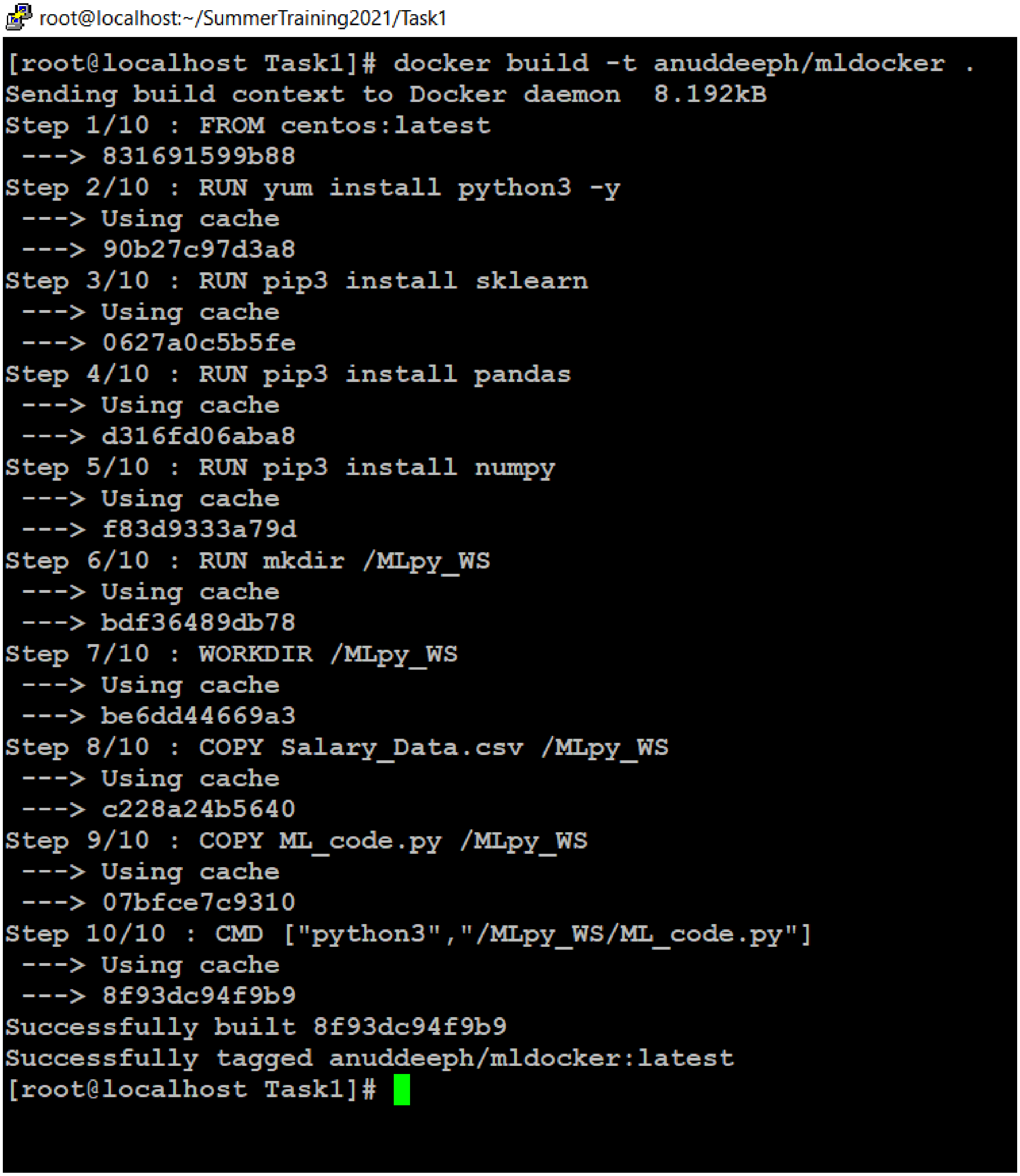
- The code is as follows:

```
import pandas

import numpy as np
```

```
dataset = pandas.read_csv('Salary_Data.csv')


X = dataset[['YearsExperience']]

y = dataset['Salary']


X=X.values.reshape(-1,1)

from sklearn.linear_model import LinearRegression


model = LinearRegression()

model.fit(X,y)


Experience=float(input("Please Enter the Years of
Experience: "))

Pred_Salary=model.predict([[Experience]])

print("You can expect a salary around {}
".format(Pred_Salary) )
```

4) Now, build the image using the **docker build** command

```
docker build -t anuddeeph/mldocker:v1 .
```

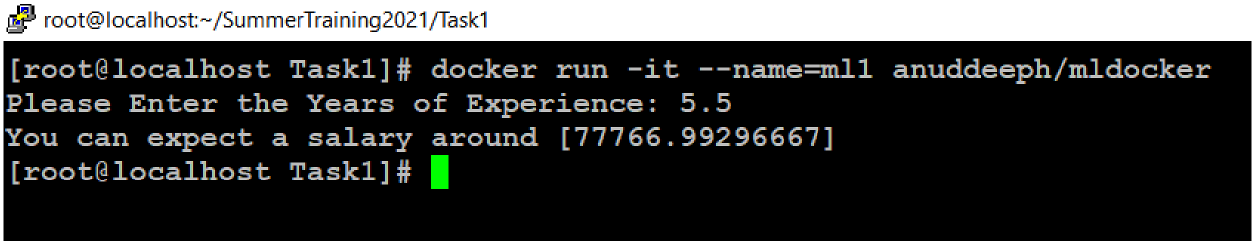


root@localhost:~/SummerTraining2021/Task1

```
[root@localhost Task1]# docker build -t anuddeeph/mldocker .
Sending build context to Docker daemon  8.192kB
Step 1/10 : FROM centos:latest
 ---> 831691599b88
Step 2/10 : RUN yum install python3 -y
 ---> Using cache
 ---> 90b27c97d3a8
Step 3/10 : RUN pip3 install sklearn
 ---> Using cache
 ---> 0627a0c5b5fe
Step 4/10 : RUN pip3 install pandas
 ---> Using cache
 ---> d316fd06aba8
Step 5/10 : RUN pip3 install numpy
 ---> Using cache
 ---> f83d9333a79d
Step 6/10 : RUN mkdir /MLpy_WS
 ---> Using cache
 ---> bdf36489db78
Step 7/10 : WORKDIR /MLpy_WS
 ---> Using cache
 ---> be6dd44669a3
Step 8/10 : COPY Salary_Data.csv /MLpy_WS
 ---> Using cache
 ---> c228a24b5640
Step 9/10 : COPY ML_code.py /MLpy_WS
 ---> Using cache
 ---> 07bfce7c9310
Step 10/10 : CMD ["python3","/MLpy_WS/ML_code.py"]
 ---> Using cache
 ---> 8f93dc94f9b9
Successfully built 8f93dc94f9b9
Successfully tagged anuddeeph/mldocker:latest
[root@localhost Task1]#
```

5) Finally, we can launch the container using this image

- `docker run -it --name ml Anuddeeph/mldocker`

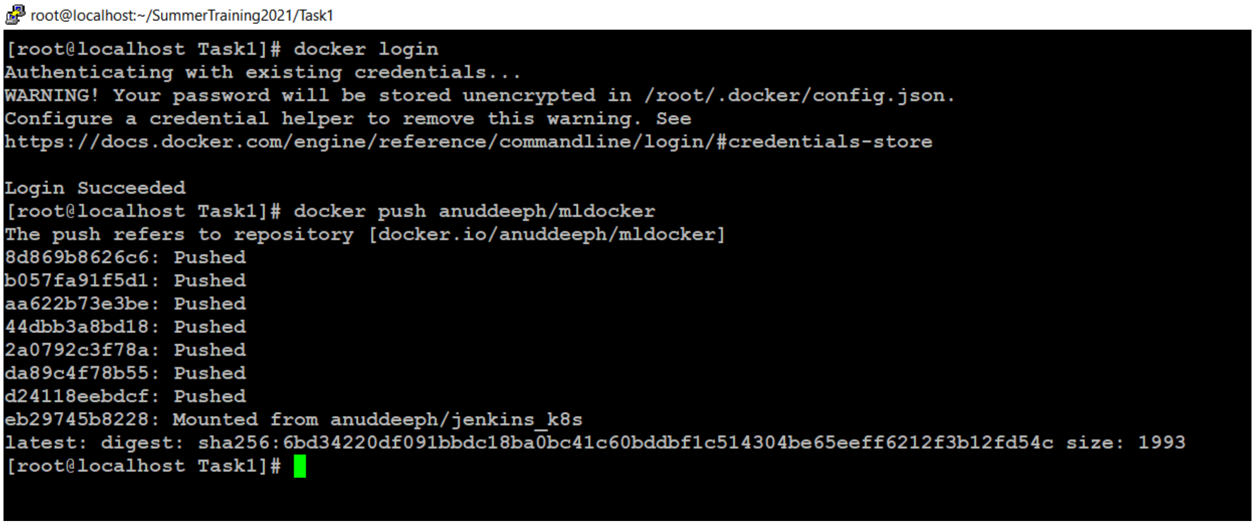


root@localhost:~/SummerTraining2021/Task1

```
[root@localhost Task1]# docker run -it --name=ml1 anuddeeph/mldocker
Please Enter the Years of Experience: 5.5
You can expect a salary around [77766.99296667]
[root@localhost Task1]#
```

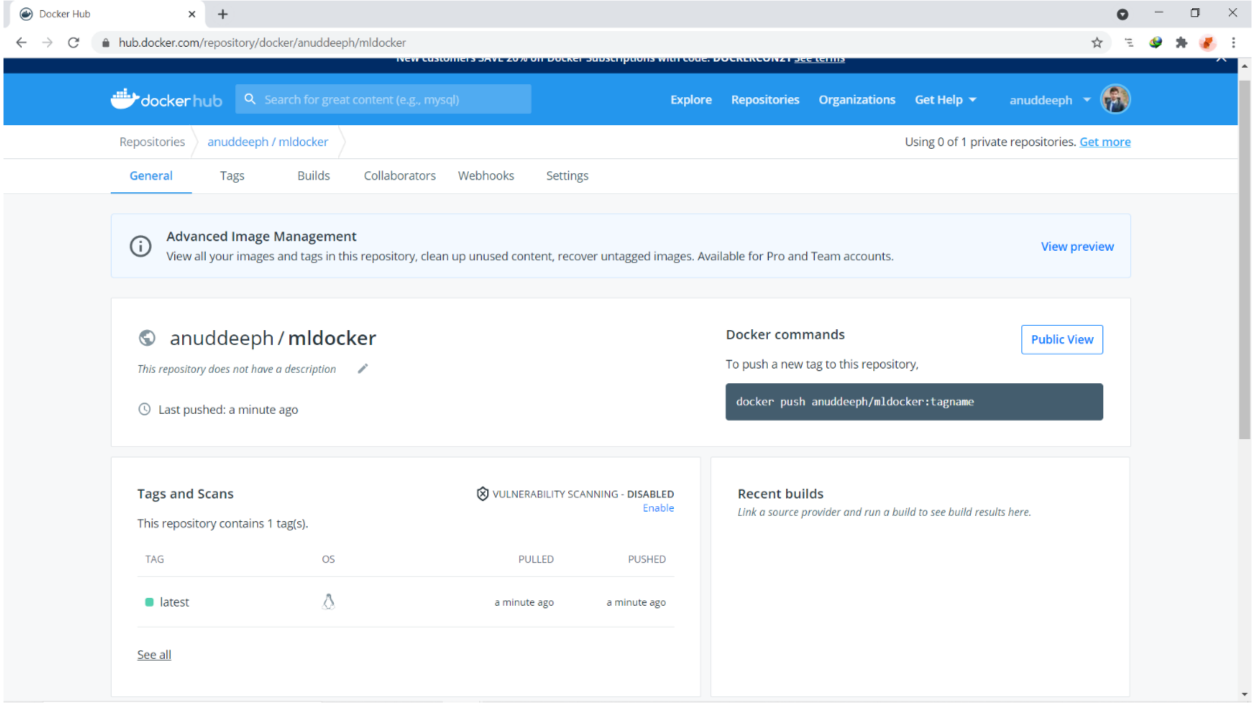
6) `To push to hub.docker.com`

- `docker push Anuddeeph/mldocker`



root@localhost:~/SummerTraining2021/Task1

```
[root@localhost Task1]# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@localhost Task1]# docker push anuddeeph/mldocker
The push refers to repository [docker.io/anuddeeph/mldocker]
8d869b8626c6: Pushed
b057fa91f5d1: Pushed
aa622b73e3be: Pushed
44dbb3a8bd18: Pushed
2a0792c3f78a: Pushed
da89c4f78b55: Pushed
d24118eebdcf: Pushed
eb29745b8228: Mounted from anuddeeph/jenkins_k8s
latest: digest: sha256:6bd34220df091bbdc18ba0bc41c60bddbf1c514304be65eeff6212f3b12fd54c size: 1993
[root@localhost Task1]#
```



Using 0 of 1 private repositories. Get more

Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

anuddeeph / **mldocker**

This repository does not have a description

Last pushed: a minute ago

Docker commands

To push a new tag to this repository,

```
docker push anuddeeph/mldocker:tagname
```

Tags and Scans

VULNERABILITY SCANNING - **DISABLED**

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
🟩 latest		a minute ago	a minute ago

Recent builds

Link a source provider and run a build to see build results here.