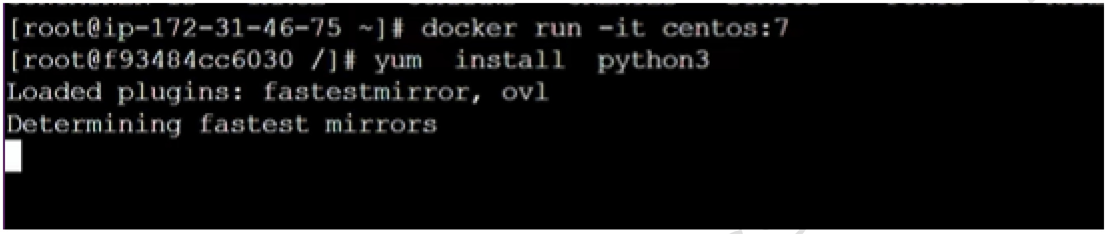
Summary

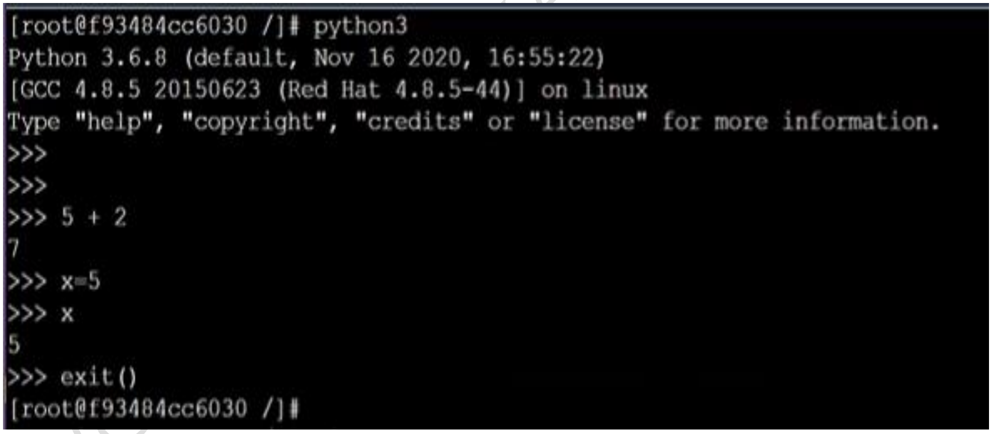
Session No – 11

- For creating a custom image there are two ways. With the commit command & Docker file
- Launching a container and installing python3



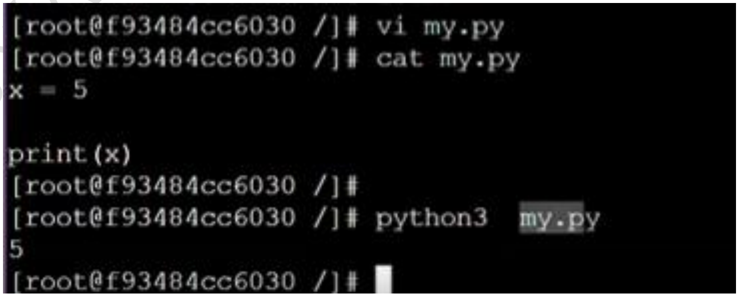
```
[root@ip-172-31-46-75 ~]# docker run -it centos:7
[root@f93484cc6030 /]# yum  install  python3
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
```

- Python gives us two facilities that are we can use a python3 command for a live interpreter and we can create a python code file
  - Python live interpreter



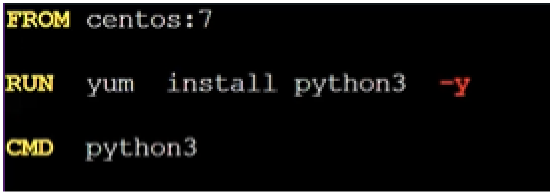
```
[root@f93484cc6030 /]# python3
Python 3.6.8 (default, Nov 16 2020, 16:55:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> 5 + 2
7
>>> x=5
>>> x
5
>>> exit()
[root@f93484cc6030 /]#
```

  - Creating and running python code file



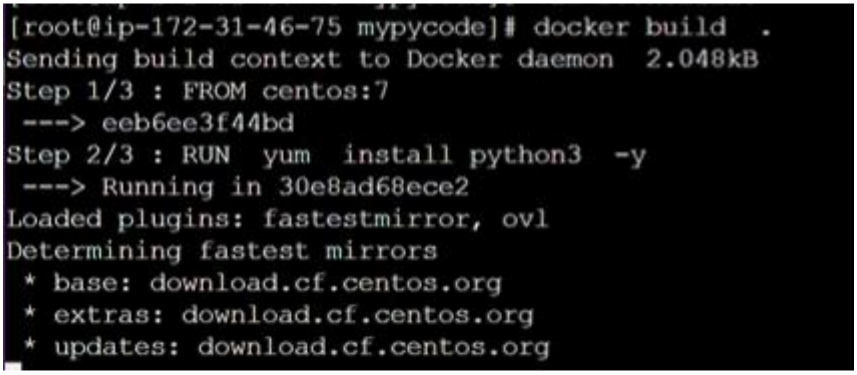
```
[root@f93484cc6030 /]# vi my.py
[root@f93484cc6030 /]# cat my.py
x = 5

print(x)
[root@f93484cc6030 /]#
[root@f93484cc6030 /]# python3  my.py
5
[root@f93484cc6030 /]#
```

- Docker file for python3



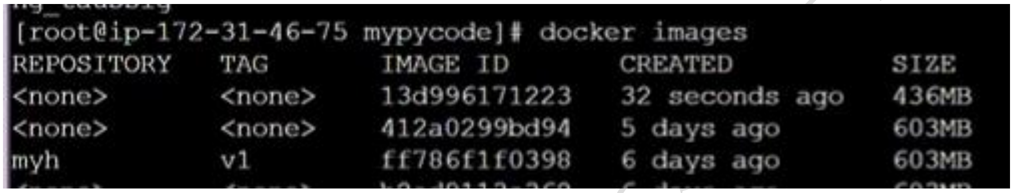
```
FROM centos:7

RUN   yum  install python3  -y

CMD   python3
```

- Building the image



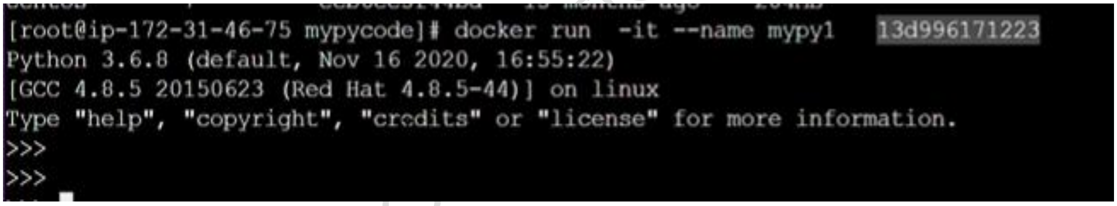
```
[root@ip-172-31-46-75 mypycode]# docker build  .
Sending build context to Docker daemon  2.048kB
Step 1/3 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/3 : RUN  yum  install python3  -y
 ---> Running in 30e8ad68ece2
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: download.cf.centos.org
 * extras: download.cf.centos.org
 * updates: download.cf.centos.org
```

- Like every container has a unique ID similarly every image has a unique id as well



```
[root@ip-172-31-46-75 mypycode]# docker images
REPOSITORY    TAG        IMAGE ID       CREATED          SIZE
<none>        <none>     13d996171223   32 seconds ago   436MB
<none>        <none>     412a0299bd94   5 days ago       603MB
myh           v1         ff786f1f0398   6 days ago       603MB
```

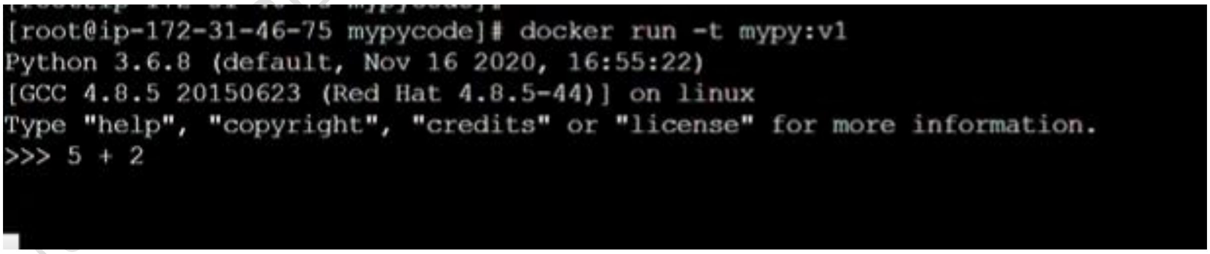
- Launching a container from the image with image id



```
[root@ip-172-31-46-75 mypycode]# docker run  -it --name mypy1   13d996171223
Python 3.6.8 (default, Nov 16 2020, 16:55:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
```

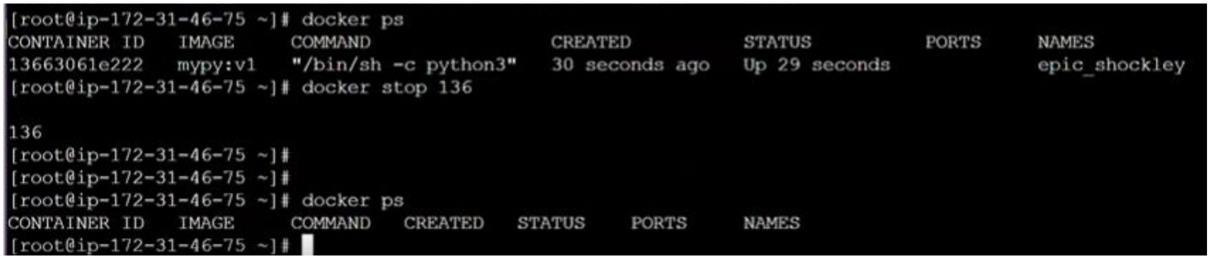
- As soon as we exit from the python program container will automatically stop so the life of the container is the life of the command

- **-t** keyword in the docker run command will give us a non-interactive terminal



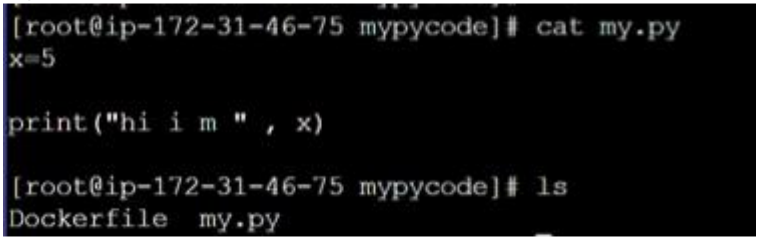
```
[root@ip-172-31-46-75 mypycode]# docker run -t mypy:v1
Python 3.6.8 (default, Nov 16 2020, 16:55:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 5 + 2
```

- Instead of giving the entire id of the container, we can also give initial 3-4 unique characters to the docker commands



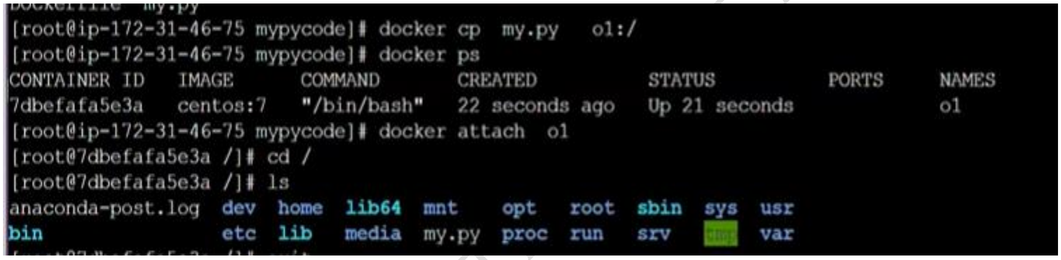
```
[root@ip-172-31-46-75 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
13663061e222   mypy:v1   "/bin/sh -c python3"     30 seconds ago   Up 29 seconds             epic_shockley
[root@ip-172-31-46-75 ~]# docker stop 136

136
[root@ip-172-31-46-75 ~]#
[root@ip-172-31-46-75 ~]#
[root@ip-172-31-46-75 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-46-75 ~]#
```

- If we require a particular file to run in the container we can create an image for this
  - Python file



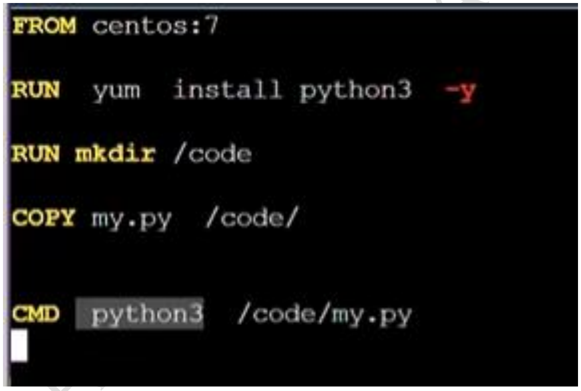
```
[root@ip-172-31-46-75 mypycode]# cat my.py
x=5

print("hi i m " , x)

[root@ip-172-31-46-75 mypycode]# ls
Dockerfile  my.py
```

  - TO copy the program file we have two ways that are with copy command & with the docker file
    - Copy command



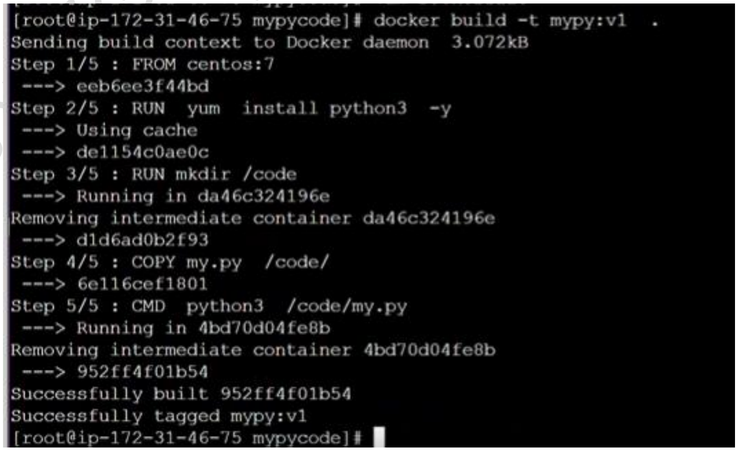
```
[root@ip-172-31-46-75 mypycode]# docker cp  my.py   o1:/
[root@ip-172-31-46-75 mypycode]# docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED          STATUS          PORTS     NAMES
7dbefafa5e3a   centos:7   "/bin/bash"   22 seconds ago   Up 21 seconds             o1
[root@ip-172-31-46-75 mypycode]# docker attach  o1
[root@7dbefafa5e3a /]# cd /
[root@7dbefafa5e3a /]# ls
anaconda-post.log  dev  home  lib64  mnt    opt   root  sbin  sys  usr
bin                etc  lib   media  my.py  proc  run   srv   tmp  var
```

    - Copy with docker file



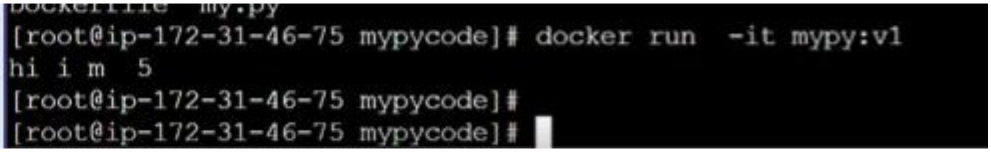
```
FROM centos:7

RUN  yum  install python3  -y

RUN mkdir /code

COPY my.py  /code/


CMD  python3   /code/my.py
```

  - Creating an image from the docker file



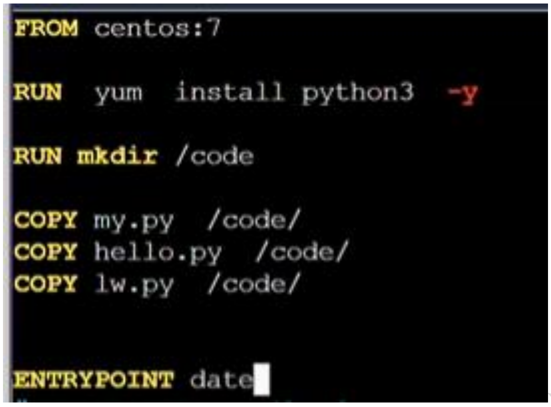
```
[root@ip-172-31-46-75 mypycode]# docker build -t mypy:v1  .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/5 : RUN  yum  install python3  -y
 ---> Using cache
 ---> de1154c0ae0c
Step 3/5 : RUN mkdir /code
 ---> Running in da46c324196e
Removing intermediate container da46c324196e
 ---> d1d6ad0b2f93
Step 4/5 : COPY my.py  /code/
 ---> 6e116cef1801
Step 5/5 : CMD  python3  /code/my.py
 ---> Running in 4bd70d04fe8b
Removing intermediate container 4bd70d04fe8b
 ---> 952ff4f01b54
Successfully built 952ff4f01b54
Successfully tagged mypy:v1
[root@ip-172-31-46-75 mypycode]#
```

- Now as soon as we launch the container from the new image the python program file will run



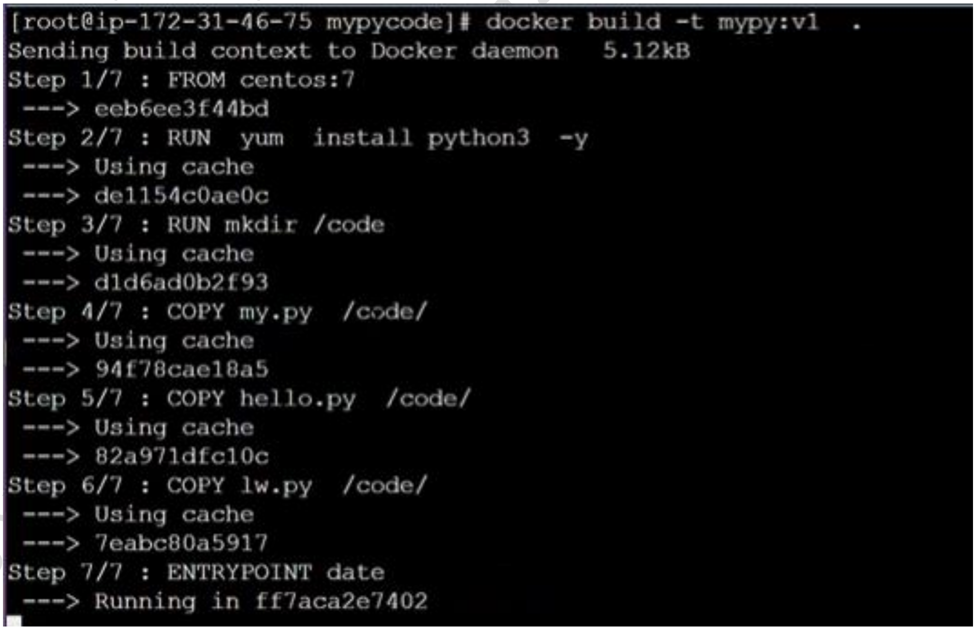
```
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1
hi i m  5
[root@ip-172-31-46-75 mypycode]#
[root@ip-172-31-46-75 mypycode]#
```

- **ENTRYPOINT** is a keyword in the docker file which has the capability to take the argument from the command line
  - Docker file for entry point



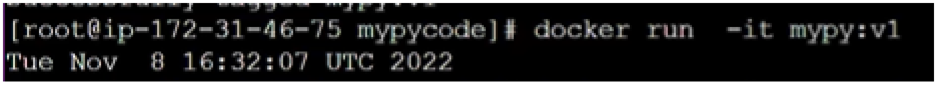
```
FROM centos:7

RUN  yum  install python3  -y

RUN mkdir /code

COPY my.py  /code/
COPY hello.py  /code/
COPY lw.py  /code/


ENTRYPOINT date
```

  - Building the image



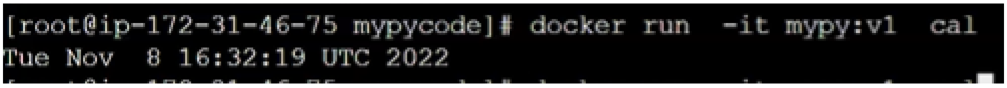
```
[root@ip-172-31-46-75 mypycode]# docker build -t mypy:v1  .
Sending build context to Docker daemon   5.12kB
Step 1/7 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/7 : RUN  yum  install python3  -y
 ---> Using cache
 ---> de1154c0ae0c
Step 3/7 : RUN mkdir /code
 ---> Using cache
 ---> d1d6ad0b2f93
Step 4/7 : COPY my.py  /code/
 ---> Using cache
 ---> 94f78cae18a5
Step 5/7 : COPY hello.py  /code/
 ---> Using cache
 ---> 82a971dfc10c
Step 6/7 : COPY lw.py  /code/
 ---> Using cache
 ---> 7eabc80a5917
Step 7/7 : ENTRYPOINT date
 ---> Running in ff7aca2e7402
```

  - If we launch the container from the image it will run the date command automatically as the container launches



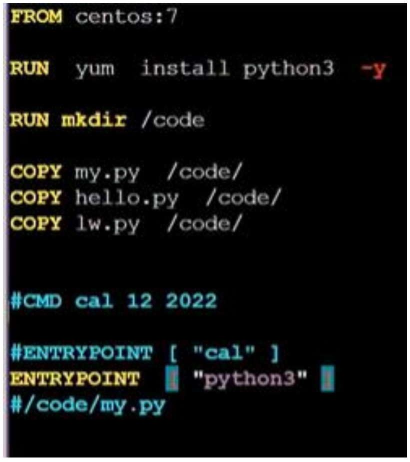
```
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1
Tue Nov  8 16:32:07 UTC 2022
```

  - ENTRYPOINT does not have the capability to change the command while launching the container



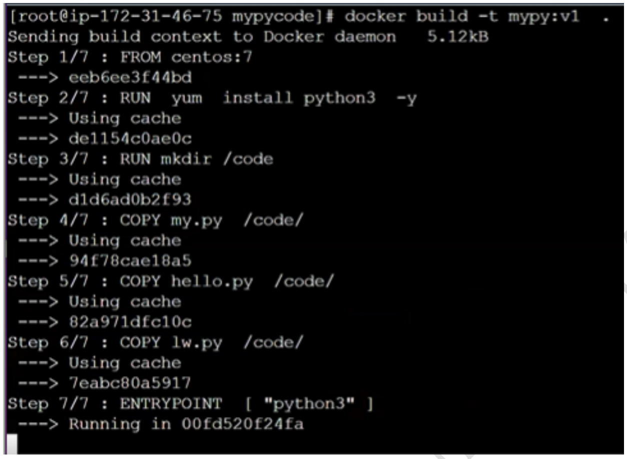
```
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1  cal
Tue Nov  8 16:32:19 UTC 2022
```

- Docker file for python program



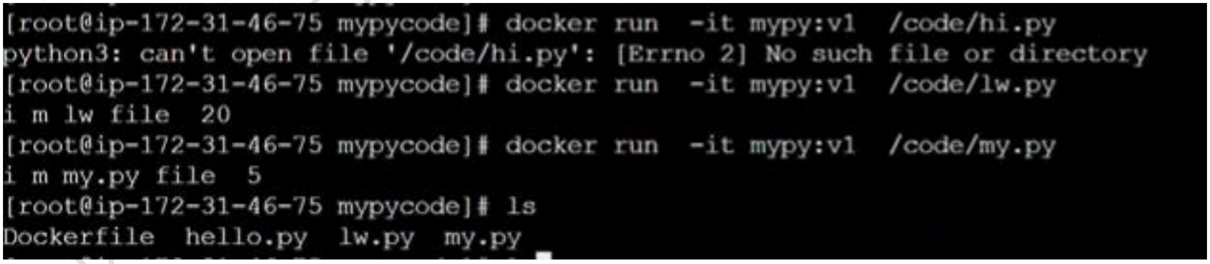
```
FROM centos:7

RUN  yum  install python3  -y

RUN mkdir /code

COPY my.py  /code/
COPY hello.py  /code/
COPY lw.py  /code/


#CMD cal 12 2022

#ENTRYPOINT [ "cal" ]
ENTRYPOINT  [ "python3" ]
#/code/my.py
```

- Building the image from the docker file



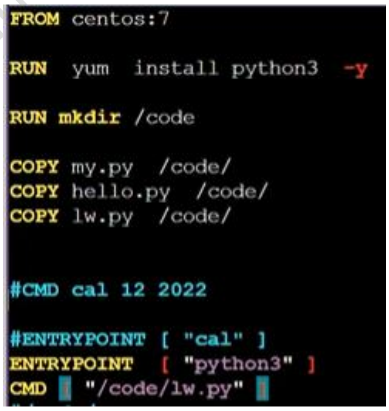
```
[root@ip-172-31-46-75 mypycode]# docker build -t mypy:v1  .
Sending build context to Docker daemon   5.12kB
Step 1/7 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/7 : RUN  yum  install python3  -y
 ---> Using cache
 ---> de1154c0ae0c
Step 3/7 : RUN mkdir /code
 ---> Using cache
 ---> d1d6ad0b2f93
Step 4/7 : COPY my.py  /code/
 ---> Using cache
 ---> 94f78cae18a5
Step 5/7 : COPY hello.py  /code/
 ---> Using cache
 ---> 82a971dfc10c
Step 6/7 : COPY lw.py  /code/
 ---> Using cache
 ---> 7eabc80a5917
Step 7/7 : ENTRYPOINT  [ "python3" ]
 ---> Running in 00fd520f24fa
```

- Launching the container and passing the program file name as arguments



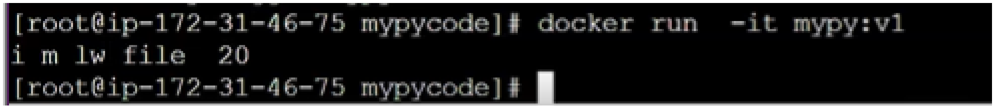
```
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1  /code/hi.py
python3: can't open file '/code/hi.py': [Errno 2] No such file or directory
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1  /code/lw.py
i m lw file  20
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1  /code/my.py
i m my.py file  5
[root@ip-172-31-46-75 mypycode]# ls
Dockerfile  hello.py  lw.py  my.py
```

- If we want to run a default file when no argument is passed then we have to use ENTRYPOINT with CMD



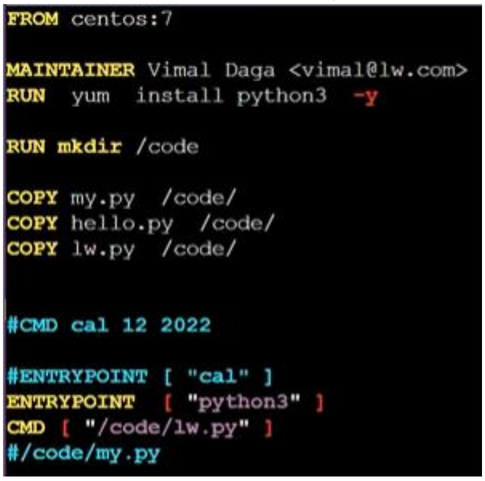
```
FROM centos:7

RUN  yum  install python3  -y

RUN mkdir /code

COPY my.py  /code/
COPY hello.py  /code/
COPY lw.py  /code/


#CMD cal 12 2022

#ENTRYPOINT [ "cal" ]
ENTRYPOINT  [ "python3" ]
CMD [ "/code/lw.py" ]
```

- Now if we don’t pass any argument it will run the default file



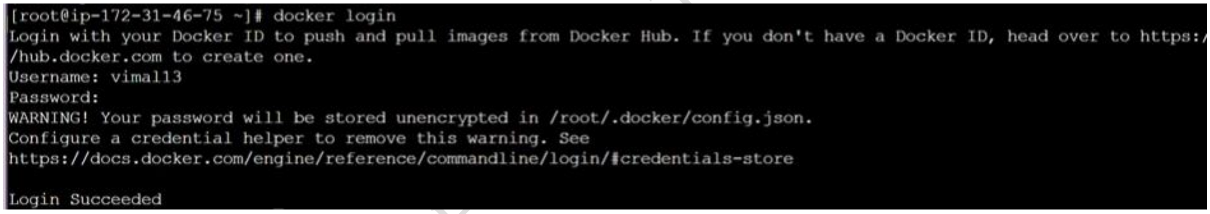
```
[root@ip-172-31-46-75 mypycode]# docker run  -it mypy:v1
i m lw file  20
[root@ip-172-31-46-75 mypycode]#
```

- **MAINTAINER** is a keyword in the docker file to add the author details



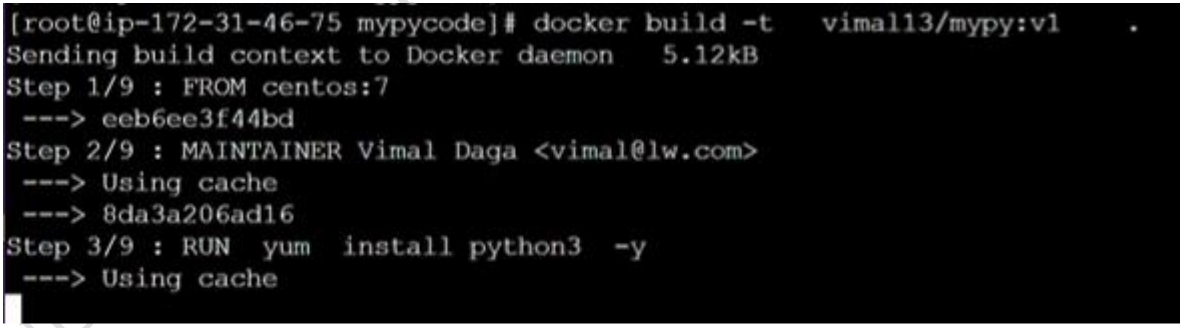
```
FROM centos:7

MAINTAINER Vimal Daga <vimal@lw.com>
RUN  yum  install python3  -y

RUN mkdir /code

COPY my.py  /code/
COPY hello.py  /code/
COPY lw.py  /code/


#CMD cal 12 2022

#ENTRYPOINT [ "cal" ]
ENTRYPOINT  [ "python3" ]
CMD [ "/code/lw.py" ]
#/code/my.py
```

- One of the famous public docker registries is the docker hub
- **docker login** command is used for logging in to the docker registry



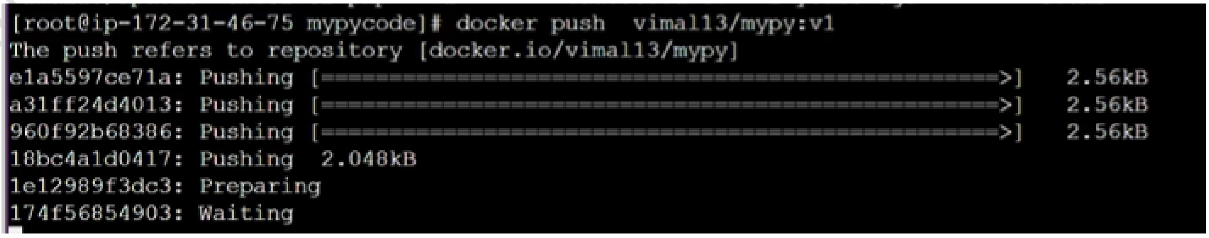
```
[root@ip-172-31-46-75 ~]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https:/
/hub.docker.com to create one.
Username: vimal13
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- Before uploading or pushing the image to the docker hub we need to tag the image with the account name and then the image name



```
[root@ip-172-31-46-75 mypycode]# docker build -t   vimal13/mypy:v1    .
Sending build context to Docker daemon   5.12kB
Step 1/9 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/9 : MAINTAINER Vimal Daga <vimal@lw.com>
 ---> Using cache
 ---> 8da3a206ad16
Step 3/9 : RUN  yum  install python3  -y
 ---> Using cache
```

- **docker push** command is used to push the image to the docker hub.



```
[root@ip-172-31-46-75 mypycode]# docker push  vimal13/mypy:v1
The push refers to repository [docker.io/vimal13/mypy]
e1a5597ce71a: Pushing [==================================================>]   2.56kB
a31ff24d4013: Pushing [==================================================>]   2.56kB
960f92b68386: Pushing [==================================================>]   2.56kB
18bc4a1d0417: Pushing  2.048kB
1e12989f3dc3: Preparing
174f56854903: Waiting
```