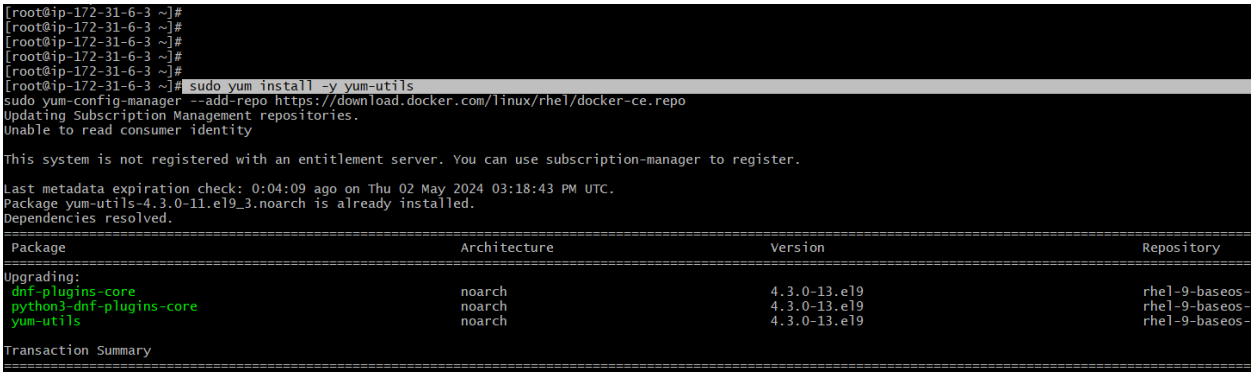
Task -8
Docker basics and installation

Description
1. Research and study the basics of Docker Containerization platform .
2. install docker on your machine
3.build a container from the available docker images.
4.study basic docker commands

Solution

1.Install docker for that go to docker documentation take commands from that

- sudo yum install -y yum-utils
- sudo yum-config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
- sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
- sudo systemctl start docker
- sudo docker run hello-world

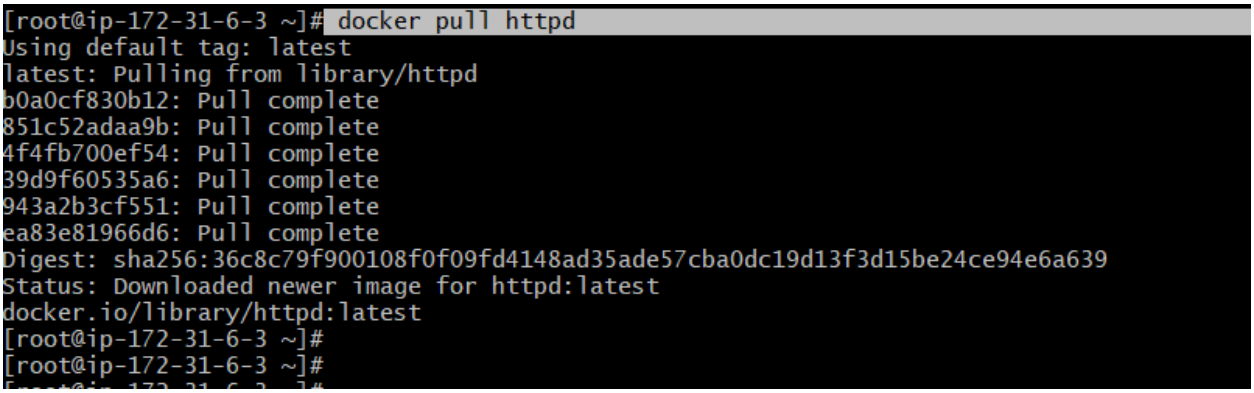


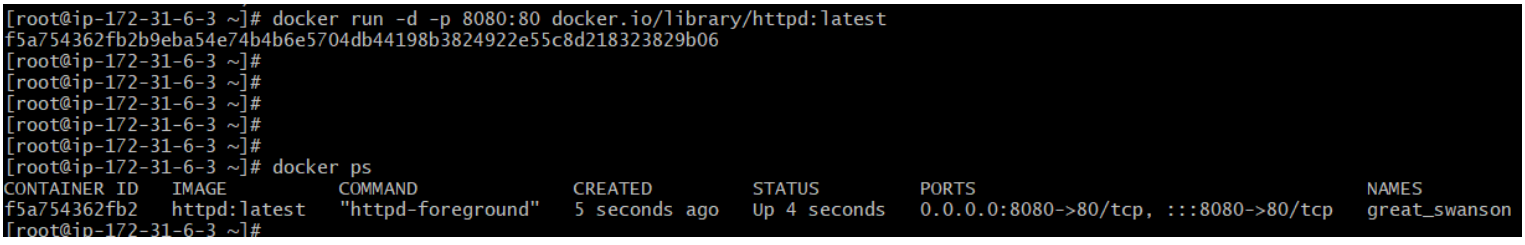
```
Adding repo from: https://download.docker.com/linux/rhel/docker-ce.repo
[root@ip-172-31-6-3 ~]# sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Docker CE Stable - x86_64                                                                  69 kB/s |  12 kB     00:00
Dependencies resolved.
==========================================================================================================================
 Package                        Architecture       Version                    Repository                           Size
==========================================================================================================================
Installing:
 containerd.io                  x86_64             1.6.31-3.1.el9             docker-ce-stable                     34 M
 docker-buildx-plugin           x86_64             0.14.0-1.el9               docker-ce-stable                     13 M
 docker-ce                      x86_64             3:26.1.1-1.el9             docker-ce-stable                     27 M
 docker-ce-cli                  x86_64             1:26.1.1-1.el9             docker-ce-stable                    7.8 M
 docker-compose-plugin          x86_64             2.27.0-1.el9               docker-ce-stable                     13 M
Installing dependencies:
 container-selinux              noarch             3:2.229.0-1.el9_3          rhel-9-appstream-rhui-rpms           58 k
 fuse-common                    x86_64             3.10.2-8.el9               rhel-9-baseos-rhui-rpms             8.6 k
 fuse-overlayfs                 x86_64             1.13-1.el9                 rhel-9-appstream-rhui-rpms           69 k
 fuse3                          x86_64             3.10.2-8.el9               rhel-9-appstream-rhui-rpms           57 k
 fuse3-libs                     x86_64             3.10.2-8.el9               rhel-9-appstream-rhui-rpms           93 k
 iptables-nft                   x86_64             1.8.8-6.el9_1              rhel-9-baseos-rhui-rpms             207 k
 libnftnl                       x86_64             1.2.6-2.el9                rhel-9-baseos-rhui-rpms              85 k
 libslirp                       x86_64             4.4.0-7.el9                rhel-9-appstream-rhui-rpms           72 k
 slirp4netns                    x86_64             1.2.3-1.el9                rhel-9-appstream-rhui-rpms           49 k
Installing weak dependencies:
 docker-ce-rootless-extras      x86_64             26.1.1-1.el9               docker-ce-stable                    4.0 M

Transaction Summary
==========================================================================================================================
Install  15 Packages
```

```
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]# sudo systemctl start docker
[root@ip-172-31-6-3 ~]# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:a26bff933ddc26d5cdf7faa98b4ae1e3ec20c4985e6f87ac0973052224d24302
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/
```

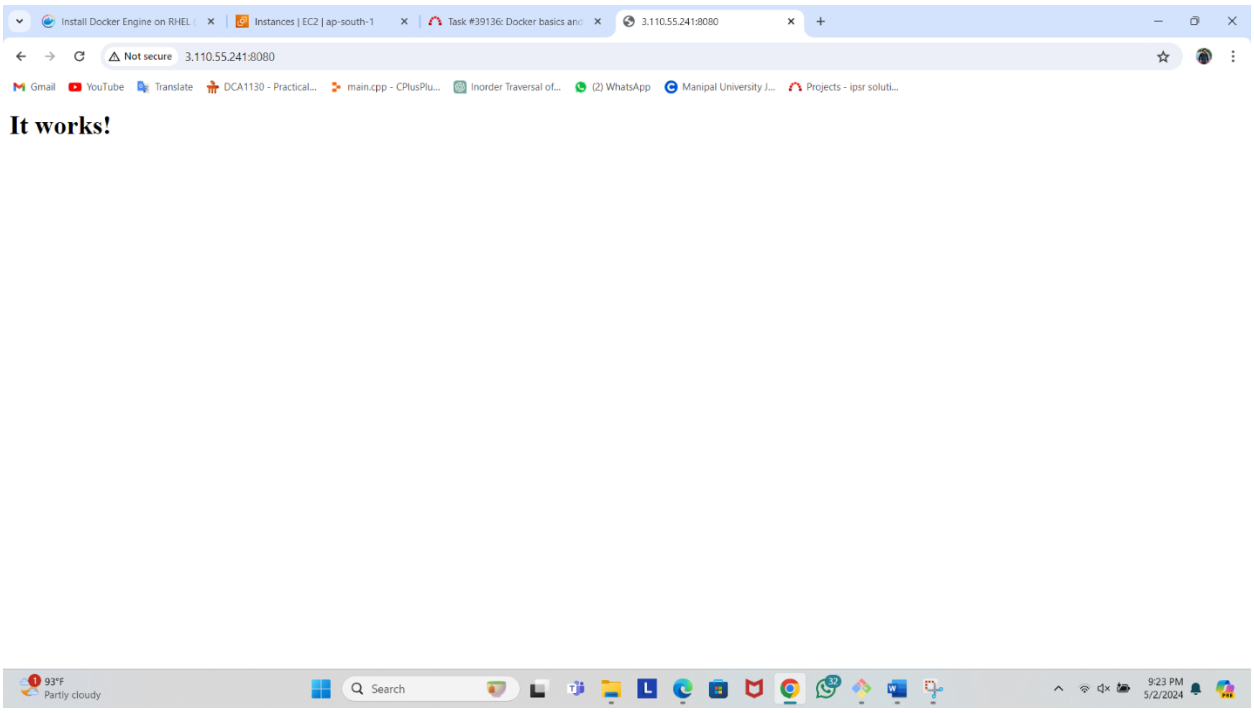
Then pull the image using docker

- docker pull httpd
- docker docker images
- docker run -d -p 8080:80 docker.io/library/httpd:latest
- docker ps





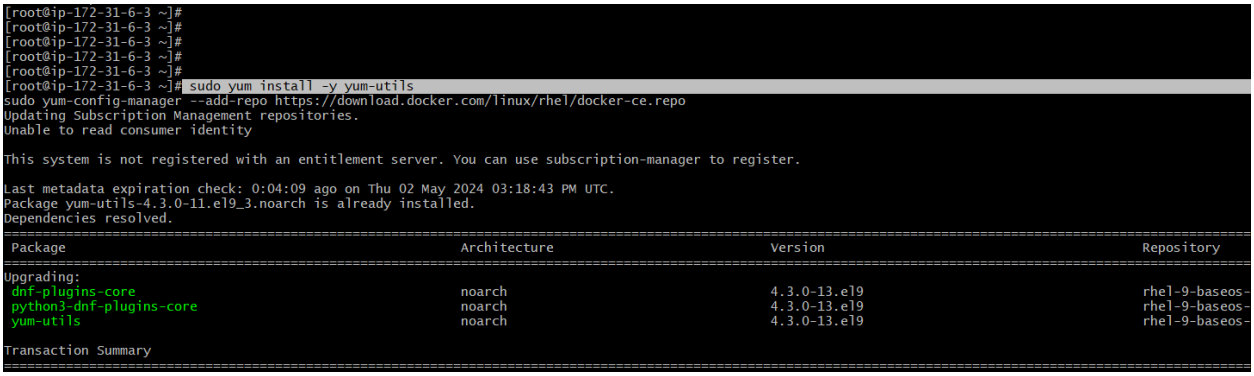
Now we have pulled the image and ready for use now check for result



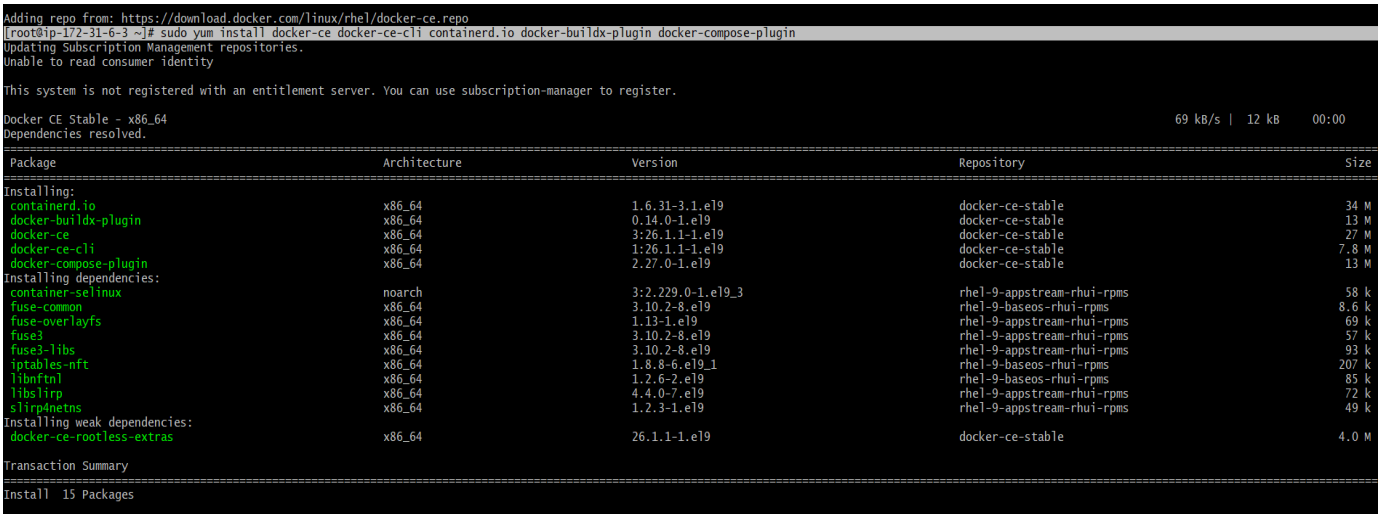
Create a docker image

Install docker

- Sudo yum install -y yum-utils
- sudo yum-config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
- sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
- sudo systemctl start docker
- sudo docker run hello-world



```
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]# sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:04:09 ago on Thu 02 May 2024 03:18:43 PM UTC.
Package yum-utils-4.3.0-11.el9_3.noarch is already installed.
Dependencies resolved.
================================================================================
 Package                 Architecture          Version                Repository
================================================================================
Upgrading:
 dnf-plugins-core        noarch                4.3.0-13.el9           rhel-9-baseos-
 python3-dnf-plugins-core noarch               4.3.0-13.el9           rhel-9-baseos-
 yum-utils               noarch                4.3.0-13.el9           rhel-9-baseos-

Transaction Summary
================================================================================
```



```
Adding repo from: https://download.docker.com/linux/rhel/docker-ce.repo
[root@ip-172-31-6-3 ~]# sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Docker CE Stable - x86_64                                   69 kB/s |  12 kB     00:00
Dependencies resolved.
================================================================================
 Package                 Architecture          Version                Repository                Size
================================================================================
Installing:
 containerd.io           x86_64                1.6.31-3.1.el9         docker-ce-stable          34 M
 docker-buildx-plugin    x86_64                0.14.0-1.el9           docker-ce-stable          13 M
 docker-ce               x86_64                3:26.1.1-1.el9         docker-ce-stable          27 M
 docker-ce-cli           x86_64                1:26.1.1-1.el9         docker-ce-stable         7.8 M
 docker-compose-plugin   x86_64                2.27.0-1.el9           docker-ce-stable          13 M
Installing dependencies:
 container-selinux       noarch                3:2.229.0-1.el9_3      rhel-9-appstream-rhui-rpms 58 k
 fuse-common             x86_64                3.10.2-8.el9           rhel-9-baseos-rhui-rpms  8.6 k
 fuse-overlayfs          x86_64                1.13-1.el9             rhel-9-appstream-rhui-rpms 69 k
 fuse3                   x86_64                3.10.2-8.el9           rhel-9-appstream-rhui-rpms 57 k
 fuse3-libs              x86_64                3.10.2-8.el9           rhel-9-appstream-rhui-rpms 93 k
 iptables-nft            x86_64                1.8.8-6.el9_1          rhel-9-baseos-rhui-rpms 207 k
 libnftnl                x86_64                1.2.6-2.el9            rhel-9-baseos-rhui-rpms  85 k
 libslirp                x86_64                4.4.0-7.el9            rhel-9-appstream-rhui-rpms 72 k
 slirp4netns             x86_64                1.2.3-1.el9            rhel-9-appstream-rhui-rpms 49 k
Installing weak dependencies:
 docker-ce-rootless-extras x86_64              26.1.1-1.el9           docker-ce-stable         4.0 M

Transaction Summary
================================================================================
Install  15 Packages
```

```
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]# sudo systemctl start docker
[root@ip-172-31-6-3 ~]# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:a26bff933ddc26d5cdf7faa98b4ae1e3ec20c4985e6f87ac0973052224d24302
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/
```

Then create a directory

```
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]#
[root@ip-172-31-6-3 ~]# mkdir docker
[root@ip-172-31-6-3 ~]# rm -rf ocker/
[root@ip-172-31-6-3 ~]# ls
docker
[root@ip-172-31-6-3 ~]# cd docker/
[root@ip-172-31-6-3 docker]# touch DOCclient_loop: send disconnect: Connection reset by peer
```

Then change directory to docker and create **index.html** and **Dockerfile**

Index.html - for adding the web content

Dockerfile – it is for adding the executing program for creation of container

```
[root@ip-172-31-6-3 ~]# ls
docker
[root@ip-172-31-6-3 ~]# cd docker/
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]# ls
[root@ip-172-31-6-3 docker]# touch Docker
[root@ip-172-31-6-3 docker]# touch index.html
[root@ip-172-31-6-3 docker]# vim Docker
-bash: vim: command not found
[root@ip-172-31-6-3 docker]# yum install vim
Updating Subscription Management repositories.
Unable to read consumer identity
```

Then add the content in this file

Docker file

root@ip-172-31-6-3:~/docker

```
# Use the RHEL 9 base image
FROM registry.access.redhat.com/ubi8/ubi
# Install Apache HTTP Server
RUN yum install -y httpd
# Copy your index.html file into the container
COPY index.html /var/www/html/index.html
# Expose port 80
EXPOSE 80
# Start Apache in the foreground
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
```

Index.html

```
root@ip-172-31-6-3:~/docker
welcome to my world
~
~
~
~
~
~
~
~
~
```

Then run the file and add the port to container

```
[root@ip-172-31-6-3 docker]# docker build -t docker .
[+] Building 17.3s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 423B
 => [internal] load metadata for registry.access.redhat.com/ubi8/ubi:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM registry.access.redhat.com/ubi8/ubi:latest@sha256:edc34f89cf9c818c2fb28b8ea1780f384db563ce4293dc0ab8e73ec01791e5af
 => => resolve registry.access.redhat.com/ubi8/ubi:latest@sha256:edc34f89cf9c818c2fb28b8ea1780f384db563ce4293dc0ab8e73ec01791e5af
 => => sha256:edc34f89cf9c818c2fb28b8ea1780f384db563ce4293dc0ab8e73ec01791e5af 1.47kB / 1.47kB
 => => sha256:aecf4c902aa445782fb1732386f4fb739f76bae4797c2eb5206ca79c739c6da8 429B / 429B
 => => sha256:c70d72aaebb4563c57b5d3d1e72cdfdac5bdc32ad8255f1cb1365e1850b91d92 6.43kB / 6.43kB
 => => sha256:2efec45cd878dd2784f6aa8338e6f4a788be9a65fd288fe555bc971308f70615 78.74MB / 78.74MB
 => => extracting sha256:2efec45cd878dd2784f6aa8338e6f4a788be9a65fd288fe555bc971308f70615
 => [internal] load build context
 => => transferring context: 119B
 => [2/3] RUN yum install -y httpd
 => [3/3] COPY index.html /var/www/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:7bfe306b2c4091e76d1738c611910157544742a0bafa5eafa51c35c05137dda6
 => => naming to docker.io/library/docker
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]# docker run -d -p 8085:80 docker
85d56c8fc9c06b369a08eb589a8174ce3fec770576b19339dec156447a630365
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
[root@ip-172-31-6-3 docker]#
```

Then check the result