Docker Task - 2

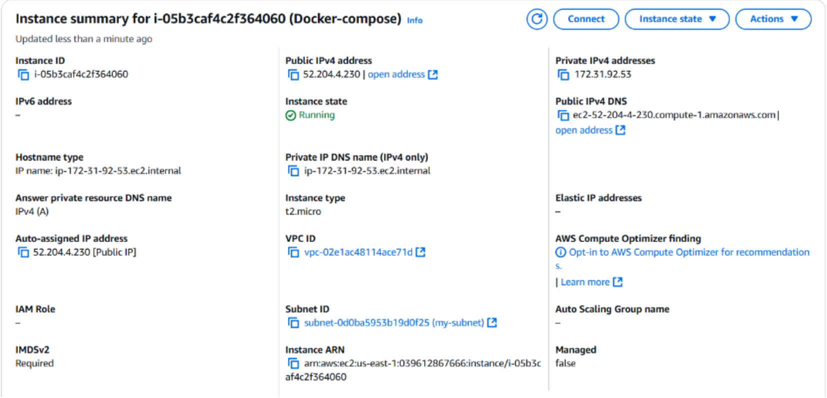
TASKS

Work Flow:

- Create an EC2 instance with the help of AWS Management Console with linux OS of required configuration and also update an security group with port number 80 (http).
- Now, Connect an EC2 instance with an help of Windows Terminal or Gitbash or Vbox.
- To connect an EC2 instance the command is:
    - ssh -i “**key_file**” ec2-user@“**Public_IP_address**”

Key_file --- Key file of the instance with the extension .pem

Public_IP_address --- Public IP address of the instance.



Instance summary for i-05b3caf4c2f364060 (Docker-compose)

Updated less than a minute ago

Instance ID
i-05b3caf4c2f364060

IPv6 address
–

Hostname type
IP name: ip-172-31-92-53.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
52.204.4.230 [Public IP]

IAM Role
–

IMDSv2
Required

Public IPv4 address
52.204.4.230 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-92-53.ec2.internal

Instance type
t2.micro

VPC ID
vpc-02e1ac48114ace71d

Subnet ID
subnet-0d0ba5953b19d0f25 (my-subnet)

Instance ARN
arn:aws:ec2:us-east-1:039612867666:instance/i-05b3caf4c2f364060

Private IPv4 addresses
172.31.92.53

Public IPv4 DNS
ec2-52-204-4-230.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

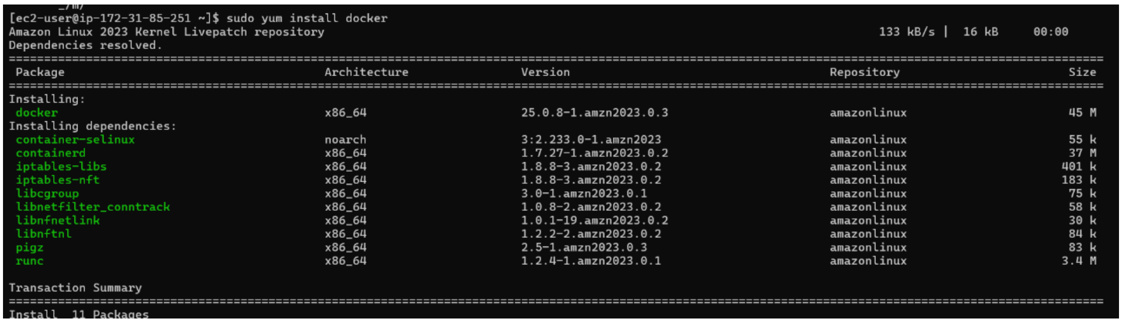
Managed
false

1. **Create a dockerfile, docker-compose file which when executed must display your basic details in the website.**

Step 1: Install an Docker in an EC2 instance

✓ To install an docker in linux machine, the command is:

- **sudo yum install docker**



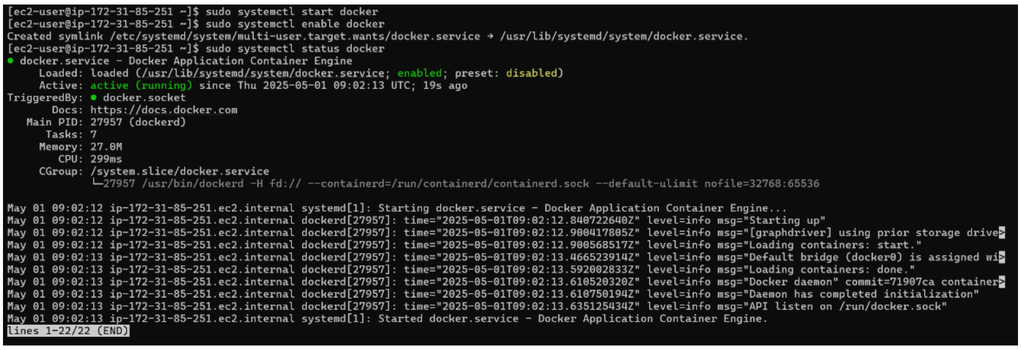
```
[ec2-user@ip-172-31-85-251 ~]$ sudo yum install docker
Amazon Linux 2023 Kernel Livepatch repository                                              133 kB/s |  16 kB     00:00
Dependencies resolved.
=============================================================================================================================
 Package                       Architecture          Version                                   Repository               Size
=============================================================================================================================
Installing:
 docker                        x86_64                25.0.8-1.amzn2023.0.3                     amazonlinux              45 M
Installing dependencies:
 container-selinux             noarch                3:2.233.0-1.amzn2023                      amazonlinux              55 k
 containerd                    x86_64                1.7.27-1.amzn2023.0.2                     amazonlinux              37 M
 iptables-libs                 x86_64                1.8.8-3.amzn2023.0.2                      amazonlinux             401 k
 iptables-nft                  x86_64                1.8.8-3.amzn2023.0.2                      amazonlinux             183 k
 libcgroup                     x86_64                3.0-1.amzn2023.0.1                        amazonlinux              75 k
 libnetfilter_conntrack        x86_64                1.0.8-2.amzn2023.0.2                      amazonlinux              58 k
 libnfnetlink                  x86_64                1.0.1-19.amzn2023.0.2                     amazonlinux              30 k
 libnftnl                      x86_64                1.2.2-2.amzn2023.0.2                      amazonlinux              84 k
 pigz                          x86_64                2.5-1.amzn2023.0.3                        amazonlinux              83 k
 runc                          x86_64                1.2.4-1.amzn2023.0.1                      amazonlinux             3.4 M

Transaction Summary
=============================================================================================================================
Install  11 Packages
```

✓ To start and enable an docker service, The command is:

- **sudo systemctl start docker**
- **sudo systemctl enable docker**

✓ To check the status of the docker service, The command is:

- **sudo systemctl status docker**



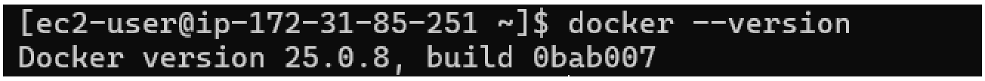
```
[ec2-user@ip-172-31-85-251 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-85-251 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-85-251 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-05-01 09:02:13 UTC; 19s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 27957 (dockerd)
      Tasks: 7
     Memory: 27.0M
        CPU: 299ms
     CGroup: /system.slice/docker.service
             └─27957 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

May 01 09:02:12 ip-172-31-85-251.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
May 01 09:02:12 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:12.840722640Z" level=info msg="Starting up"
May 01 09:02:12 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:12.900417805Z" level=info msg="[graphdriver] using prior storage drive>
May 01 09:02:12 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:12.900568517Z" level=info msg="Loading containers: start."
May 01 09:02:13 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:13.466523914Z" level=info msg="Default bridge (docker0) is assigned wi>
May 01 09:02:13 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:13.592002833Z" level=info msg="Loading containers: done."
May 01 09:02:13 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:13.610520320Z" level=info msg="Docker daemon" commit=71907ca container>
May 01 09:02:13 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:13.610750194Z" level=info msg="Daemon has completed initialization"
May 01 09:02:13 ip-172-31-85-251.ec2.internal dockerd[27957]: time="2025-05-01T09:02:13.635125434Z" level=info msg="API listen on /run/docker.sock"
May 01 09:02:13 ip-172-31-85-251.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)
```

✓ To add an ec2-user to docker group, the command is:

- **sudo usermod -aG docker ec2-user**

✓ To check an version of the docker and to verify an installation, the command is:

- **docker --version**

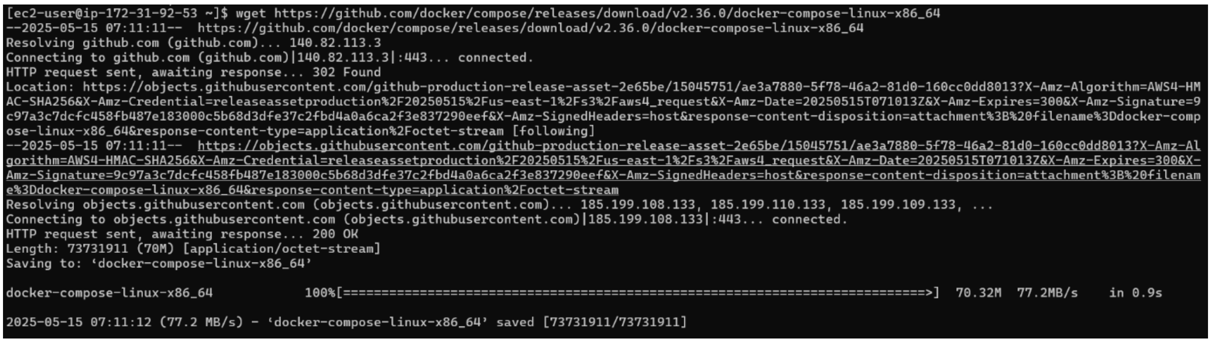


```
[ec2-user@ip-172-31-85-251 ~]$ docker --version
Docker version 25.0.8, build 0bab007
```

Step 2: Install an Docker Compose in an EC2 instance

✓ Now, We have to install an docker-compose file we can get the docker compose file in github, we can copy the link and paste in an below command formate, The command is:

- **wget <docker-compose-file>**



```
[ec2-user@ip-172-31-92-53 ~]$ wget https://github.com/docker/compose/releases/download/v2.36.0/docker-compose-linux-x86_64
--2025-05-15 07:11:11--  https://github.com/docker/compose/releases/download/v2.36.0/docker-compose-linux-x86_64
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/15045751/ae3a7880-5f78-46a2-81d0-160cc0dd8013?X-Amz-Algorithm=AWS4-HM
AC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250515%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250515T071013Z&X-Amz-Expires=300&X-Amz-Signature=9
c97a3c7dcfc458fb487e183000c5b68d3dfe37c2fbd4a0a6ca2f3e837290eef&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Ddocker-comp
ose-linux-x86_64&response-content-type=application%2Foctet-stream [following]
--2025-05-15 07:11:11--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/15045751/ae3a7880-5f78-46a2-81d0-160cc0dd8013?X-Amz-Al
gorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250515%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250515T071013Z&X-Amz-Expires=300&X-
Amz-Signature=9c97a3c7dcfc458fb487e183000c5b68d3dfe37c2fbd4a0a6ca2f3e837290eef&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filenam
e%3Ddocker-compose-linux-x86_64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73731911 (70M) [application/octet-stream]
Saving to: ‘docker-compose-linux-x86_64’

docker-compose-linux-x86_64           100%[==========================================================================>]  70.32M  77.2MB/s    in 0.9s

2025-05-15 07:11:12 (77.2 MB/s) - ‘docker-compose-linux-x86_64’ saved [73731911/73731911]
```

✓ To add some permission to that docker-compose file, The command is:

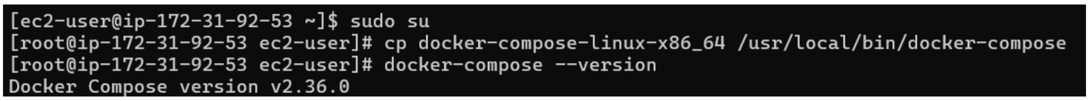
- **Chmod +x <docker-compose-file>**

✓ And now copy that file to the default location of the docker compose file, The command is:

- **Cp <docker-compose-file> /usr/local/bin/<docker-compose-file>**

✓ To check an version of docker-compose file, The command is:

- **docker-compose –version**



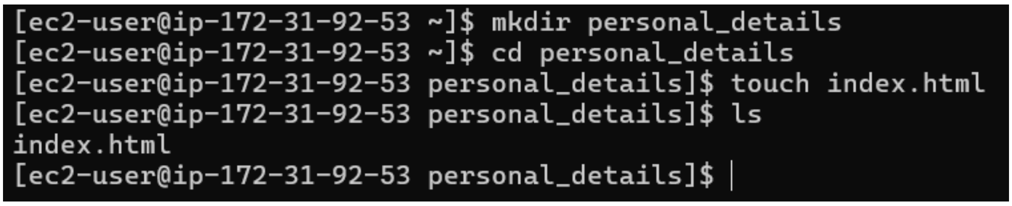
```
[ec2-user@ip-172-31-92-53 ~]$ sudo su
[root@ip-172-31-92-53 ec2-user]# cp docker-compose-linux-x86_64 /usr/local/bin/docker-compose
[root@ip-172-31-92-53 ec2-user]# docker-compose --version
Docker Compose version v2.36.0
```

Step 3: Create an HTML document with an Personal Details

- ✓ Create one directory inside that directory we are going to create an html file, docker file and docker compose file, so to create an one directory, The command is:
    - **mkdir personal_details**

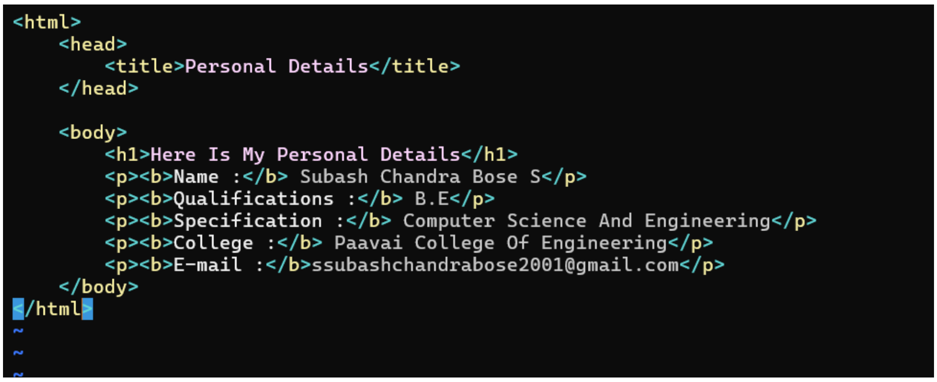
- ✓ To move inside that directory, The command is:
    - **cd personal_details**

- ✓ Now, create an html file which should be named with “index.html”, The command is:
    - **touch index.html**



```
[ec2-user@ip-172-31-92-53 ~]$ mkdir personal_details
[ec2-user@ip-172-31-92-53 ~]$ cd personal_details
[ec2-user@ip-172-31-92-53 personal_details]$ touch index.html
[ec2-user@ip-172-31-92-53 personal_details]$ ls
index.html
[ec2-user@ip-172-31-92-53 personal_details]$ |
```

- ✓ Open that html file and write an html code which should our personal details which as given below and save and close it, The command is:
    - **vi index.html**

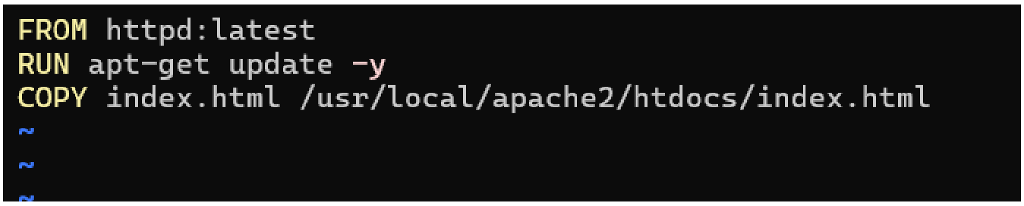


```
<html>
    <head>
        <title>Personal Details</title>
    </head>

    <body>
        <h1>Here Is My Personal Details</h1>
        <p><b>Name :</b> Subash Chandra Bose S</p>
        <p><b>Qualifications :</b> B.E</p>
        <p><b>Specification :</b> Computer Science And Engineering</p>
        <p><b>College :</b> Paavai College Of Engineering</p>
        <p><b>E-mail :</b>ssubashchandrabose2001@gmail.com</p>
    </body>
</html>
~
~
~
```

Step 4: Create an Dockerfile

- ✓ Create one Dockerfile and write an command to perform an required task, The command is:
    - **touch Dockerfile**

- ✓ Open that docker file and write an command to perform an required task which as given below, The command is:
    - **vi Dockerfile**



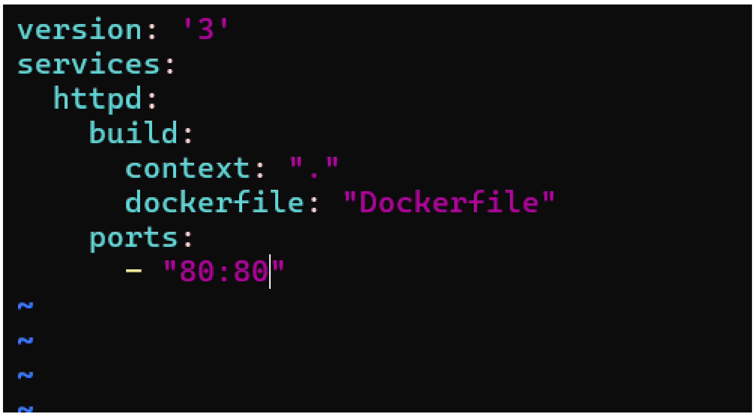
```
FROM httpd:latest
RUN apt-get update -y
COPY index.html /usr/local/apache2/htdocs/index.html
~
~
~
```

COMMAND	EXPLAINATION
FROM httpd:latest	Specifies the **base image** to use for the container.
RUN apt-get update -y	This command **updates the package list** inside a Debian/Ubuntu-based Docker image.
COPY index.html /usr/local/apache2/htdocs/index.html	Copies your local index.html file from the build context (your project folder) into the Apache server's default document root inside the container.

Step 4: Create an Docker-compose file

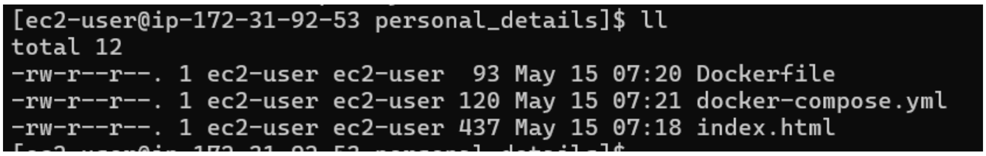
- ✓ Create one Docker compose file and write an command to perform an required task, The command is:
    - **touch docker-compose.yml**

- ✓ Open that docker compose file and write an command to perform an required task which as given below, The command is:
    - **vi docker-compose.yml**



```
version: '3'
services:
  httpd:
    build:
      context: "."
      dockerfile: "Dockerfile"
    ports:
      - "80:80"
~
~
~
~
```

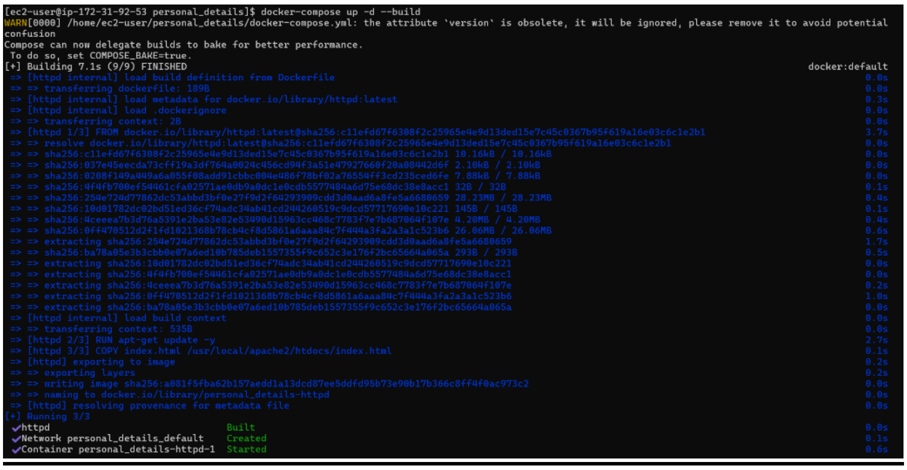
KEYWORDS	EXPLAINATION
services	Tells Docker Compose you’re defining a list of services.
httpd:	Your Apache container.
build:	Tells Docker Compose how to build the image from your local Dockerfile.
context: .	The . means "current directory" (where your Dockerfile and index.html are)
ports:	Maps port 80 on the host (your EC2) to port 80 in the container.



```
[ec2-user@ip-172-31-92-53 personal_details]$ ll
total 12
-rw-r--r--. 1 ec2-user ec2-user  93 May 15 07:20 Dockerfile
-rw-r--r--. 1 ec2-user ec2-user 120 May 15 07:21 docker-compose.yml
-rw-r--r--. 1 ec2-user ec2-user 437 May 15 07:18 index.html
[ec2-user@ip-172-31-92-53 personal_details]$
```

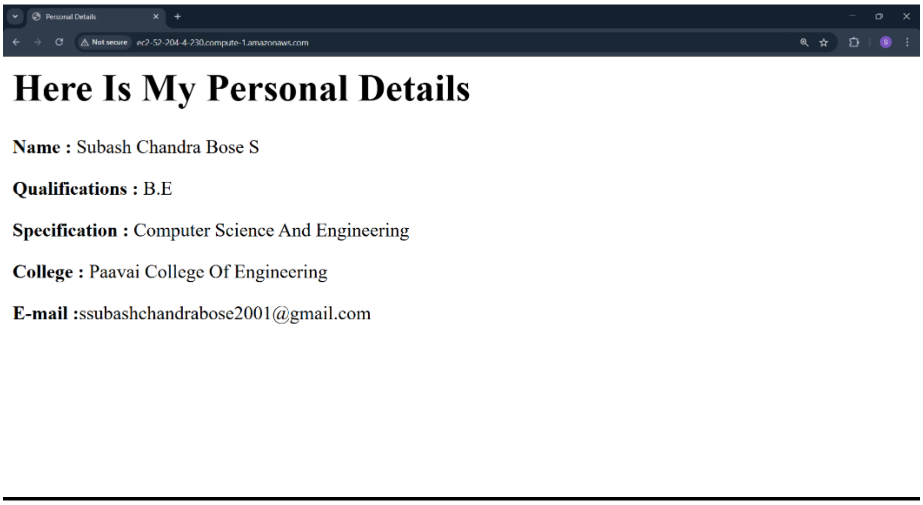
Step 4: Build an docker-compose file and host an web application.

- ✓ Now, after creating all three file such as html file, docker file and docker compose file. Now build and run that docker compose file to build and run the docker compose file, The command is:
  - **docker-compose up -d –build**



```
[ec2-user@ip-172-31-92-53 personal_details]$ docker-compose up -d --build
WARN[0000] /home/ec2-user/personal_details/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential
confusion
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 7.1s (9/9) FINISHED                                                                                                          docker:default
 => [httpd internal] load build definition from Dockerfile                                                                                           0.0s
 => => transferring dockerfile: 189B                                                                                                                 0.0s
 => [httpd internal] load metadata for docker.io/library/httpd:latest                                                                                0.3s
 => [httpd internal] load .dockerignore                                                                                                              0.0s
 => => transferring context: 2B                                                                                                                      0.0s
 => [httpd 1/3] FROM docker.io/library/httpd:latest@sha256:c11efd67f6308f2c25965e4e9d13ded15e7c45c0367b95f619a16e03c6c1e2b1                          3.7s
 => => resolve docker.io/library/httpd:latest@sha256:c11efd67f6308f2c25965e4e9d13ded15e7c45c0367b95f619a16e03c6c1e2b1                                0.0s
 => => sha256:c11efd67f6308f2c25965e4e9d13ded15e7c45c0367b95f619a16e03c6c1e2b1 10.16kB / 10.16kB                                                     0.0s
 => => sha256:037e45eecda73cff19a3df764a0024c456cd94f3a51e47927660f20a08442d6f 2.10kB / 2.10kB                                                       0.0s
 => => sha256:0208f149a449a6a055f08add91cbbc004e486f78bf02a76554ff3cd235ced6fe 7.88kB / 7.88kB                                                       0.0s
 => => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B                                                             0.1s
 => => sha256:254e724d77862dc53abbd3bf0e27f9d2f64293909cdd3d0aad6a8fe5a6680659 28.23MB / 28.23MB                                                     0.4s
 => => sha256:10d01782dc02bd51ed36cf74adc34ab41cd244260519c9dcd57717690e10c221 145B / 145B                                                           0.1s
 => => sha256:4ceeea7b3d76a5391e2ba53e82e53490d15963cc468c7783f7e7b687064f107e 4.20MB / 4.20MB                                                       0.4s
 => => sha256:0ff470512d2f1fd1021368b78cb4cf8d5861a6aaa84c7f444a3fa2a3a1c523b6 26.06MB / 26.06MB                                                     0.6s
 => => extracting sha256:254e724d77862dc53abbd3bf0e27f9d2f64293909cdd3d0aad6a8fe5a6680659                                                            1.7s
 => => sha256:ba78a05e3b3cbb0e07a6ed10b785deb1557355f9c652c3e176f2bc65664a065a 293B / 293B                                                           0.5s
 => => extracting sha256:10d01782dc02bd51ed36cf74adc34ab41cd244260519c9dcd57717690e10c221                                                            0.0s
 => => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1                                                            0.0s
 => => extracting sha256:4ceeea7b3d76a5391e2ba53e82e53490d15963cc468c7783f7e7b687064f107e                                                            0.2s
 => => extracting sha256:0ff470512d2f1fd1021368b78cb4cf8d5861a6aaa84c7f444a3fa2a3a1c523b6                                                            1.0s
 => => extracting sha256:ba78a05e3b3cbb0e07a6ed10b785deb1557355f9c652c3e176f2bc65664a065a                                                            0.0s
 => [httpd internal] load build context                                                                                                              0.0s
 => => transferring context: 535B                                                                                                                    0.0s
 => [httpd 2/3] RUN apt-get update -y                                                                                                                2.7s
 => [httpd 3/3] COPY index.html /usr/local/apache2/htdocs/index.html                                                                                 0.1s
 => [httpd] exporting to image                                                                                                                       0.2s
 => => exporting layers                                                                                                                              0.2s
 => => writing image sha256:a081f5fba62b157aedd1a13dcd87ee5ddfd95b73e90b17b366c8ff4f0ac973c2                                                         0.0s
 => => naming to docker.io/library/personal_details-httpd                                                                                            0.0s
 => [httpd] resolving provenance for metadata file                                                                                                   0.0s
[+] Running 3/3
 ✔httpd                              Built                                                                                                           0.0s
 ✔Network personal_details_default   Created                                                                                                         0.1s
 ✔Container personal_details-httpd-1 Started                                                                                                         0.6s
```

- ✓ Once the build is gets succeed , go to aws console and copy the instance Public DNS and paste it into the browser, where you can see your personal details website will be hosted .



ec2-52-204-4-230.compute-1.amazonaws.com

Here Is My Personal Details

Name : Subash Chandra Bose S

Qualifications : B.E

Specification : Computer Science And Engineering

College : Paavai College Of Engineering

E-mail :ssubashchandrabose2001@gmail.com

************** **TASK COMPLETED** **************