Deploying a simple web application(apache) in Docker-Compose

Docker-Compose:

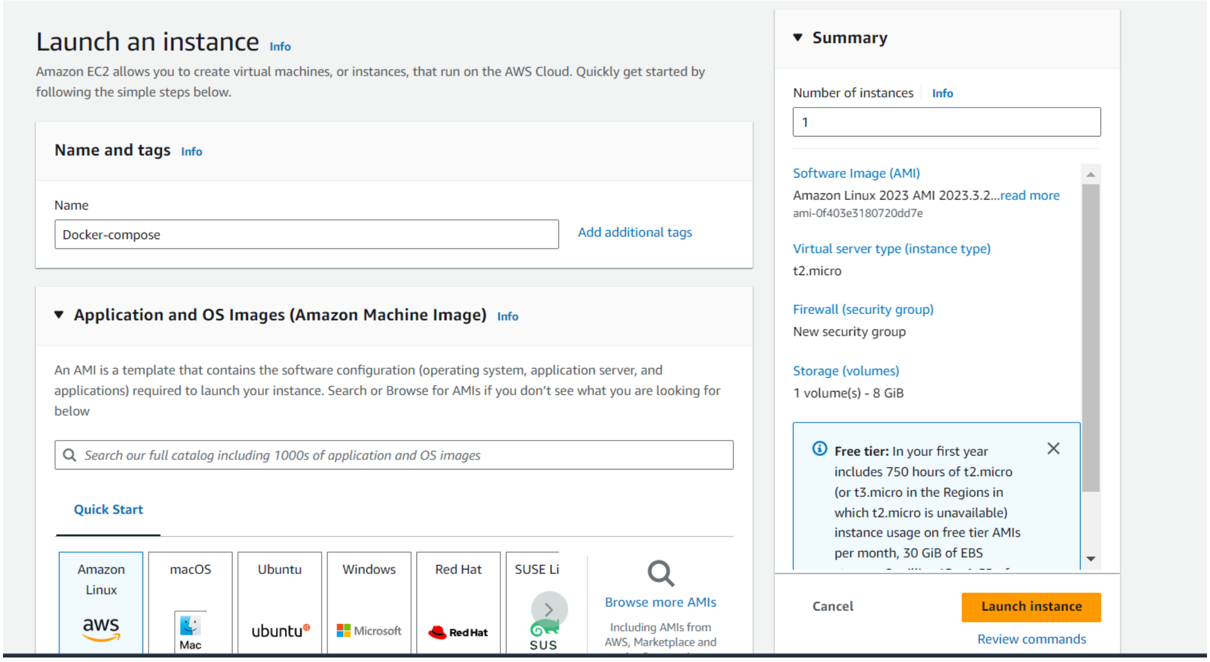
Docker-compose at a time to run the multiple Docker files, reduce the time and build to the images.

➔ First, we have to create security groups for Docker-compose

SSH--------------------22---------------------myip----This for admin purpose

All TCP--------------------- 0 - 65535--------------------Anywhere (0.0.0.0) ----- This is for End-user purpose

➔ First, we have to login to the Console ------Click on ec2 Dashboard----click on create instance -----after that we can pass name of an instance, ami and storage.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Docker-compose

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

▼ Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0f403e3180720dd7e

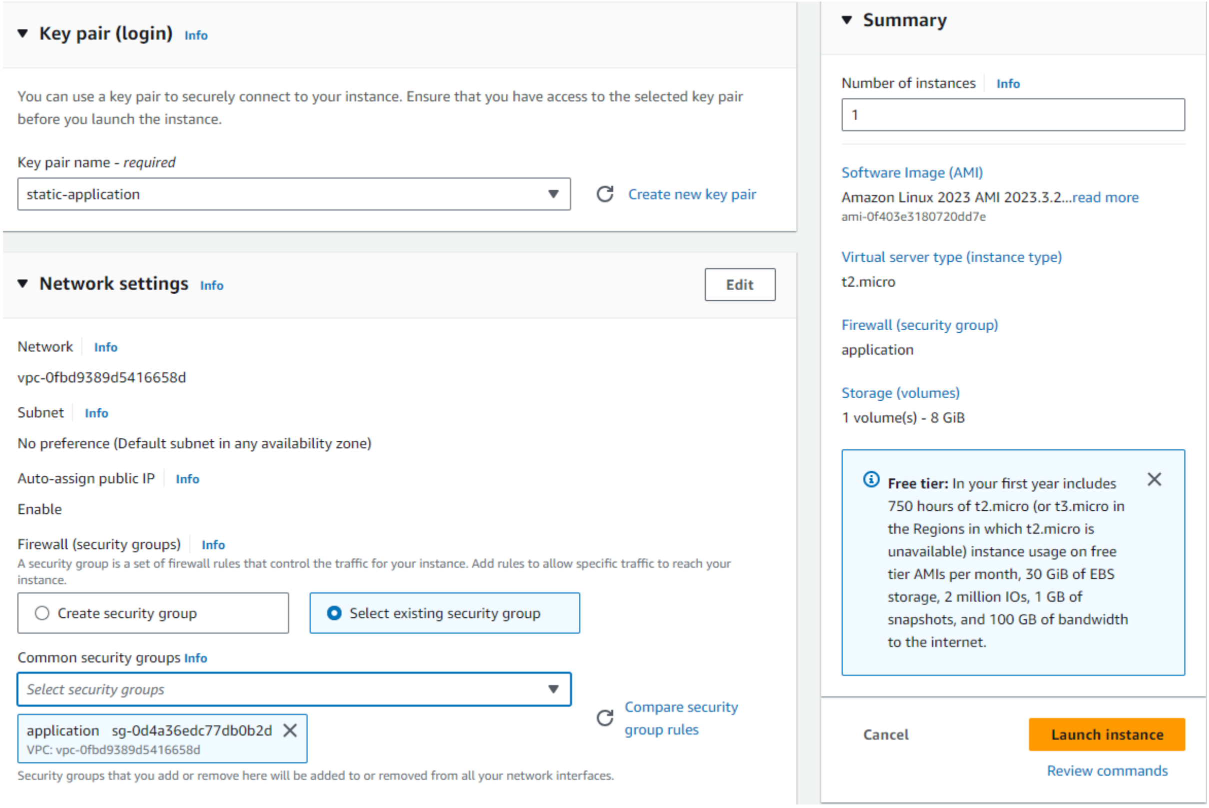
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS

➔ After we click on existing key-pair and ----->click on existing security_group.



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

static-application ▼

Create new key pair

▼ **Network settings** Info

Edit

Network | Info

vpc-0fbd9389d5416658d

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups Info

Select security groups ▼

application sg-0d4a36edc77db0b2d ✕
VPC: vpc-0fbd9389d5416658d

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ **Summary**

Number of instances | Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0f403e3180720dd7e

Virtual server type (instance type)
t2.micro

Firewall (security group)
application

Storage (volumes)
1 volume(s) - 8 GiB

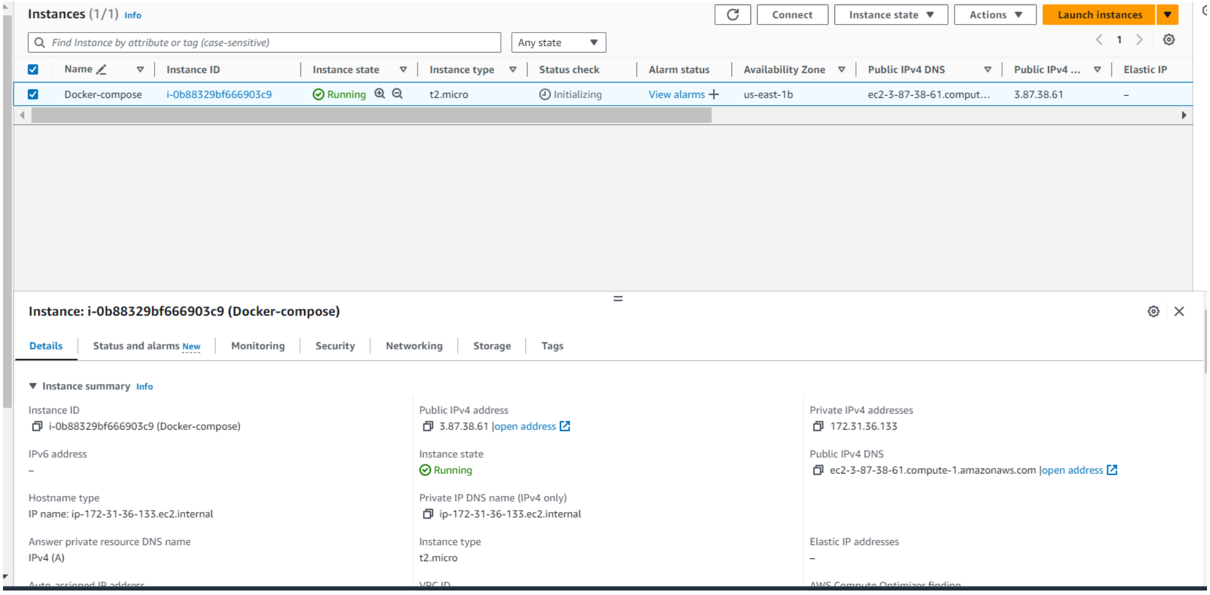
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

➔ After that click on create instance -----The instance is Created



Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive) Any state ▼

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Docker-compose	i-0b88329bf666903c9	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-3-87-38-61.comput...	3.87.38.61	–

Instance: i-0b88329bf666903c9 (Docker-compose)

▼ Instance summary Info

Instance ID
i-0b88329bf666903c9 (Docker-compose)

Public IPv4 address
3.87.38.61 |open address

Private IPv4 addresses
172.31.36.133

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-3-87-38-61.compute-1.amazonaws.com |open address

Hostname type
IP name: ip-172-31-36-133.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-36-133.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

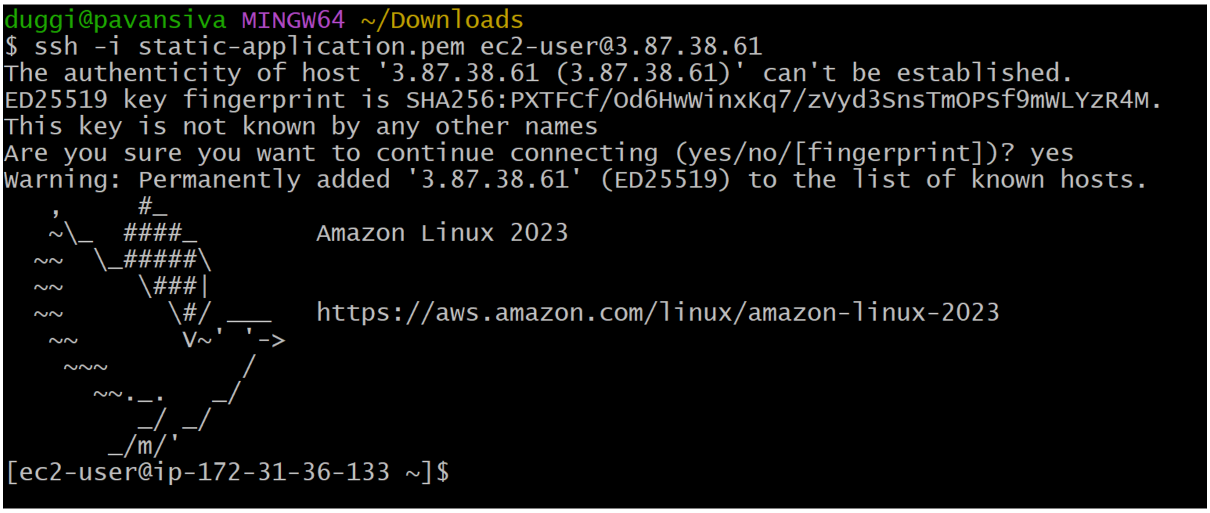
Auto-assigned IP address

VPC ID

AWS Compute Optimizer finding

➔ After we connected to the server by using command.

ssh -I <your.pem file> user-name@Public_ip



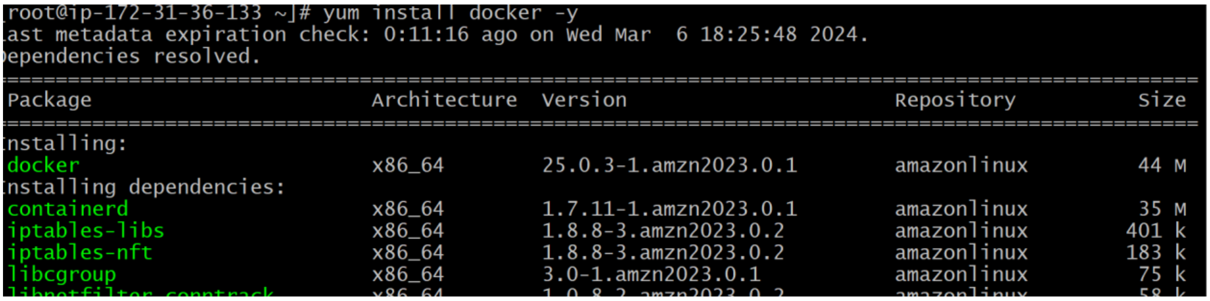
```
duggi@pavansiva MINGW64 ~/Downloads
$ ssh -i static-application.pem ec2-user@3.87.38.61
The authenticity of host '3.87.38.61 (3.87.38.61)' can't be established.
ED25519 key fingerprint is SHA256:PXTFCf/Od6HwWinxKq7/zVyd3SnsTmOPSf9mWLYzR4M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.87.38.61' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-36-133 ~]$
```

➔ Then we connect to the root user using the below command

Sudo su –

➔ Then we can install Docker Install by using below commands

yum install docker -y

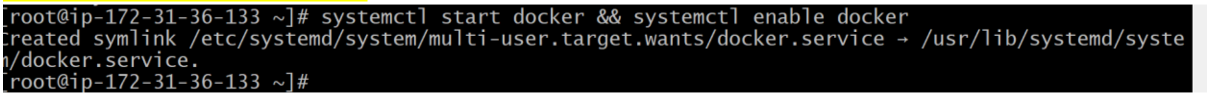


```
root@ip-172-31-36-133 ~]# yum install docker -y
ast metadata expiration check: 0:11:16 ago on Wed Mar  6 18:25:48 2024.
ependencies resolved.
================================================================================================
 Package                  Architecture  Version                       Repository         Size
================================================================================================
nstalling:
 docker                   x86_64        25.0.3-1.amzn2023.0.1         amazonlinux        44 M
nstalling dependencies:
 containerd               x86_64        1.7.11-1.amzn2023.0.1         amazonlinux        35 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
```

➔ Then we can start and enable the Docker by using below commands

systemctl start docker

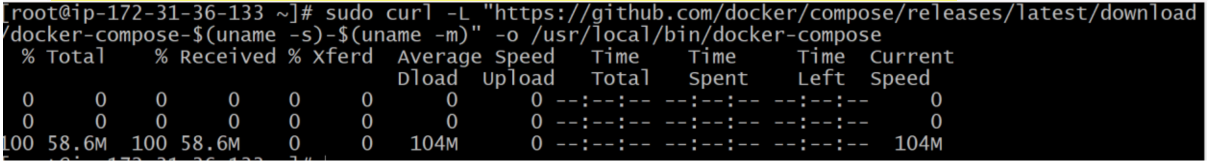
systemctl enable docker



```
root@ip-172-31-36-133 ~]# systemctl start docker && systemctl enable docker
reated symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/syste
/docker.service.
root@ip-172-31-36-133 ~]#
```

➔ After we can install to the Docker-Compose by using below commands

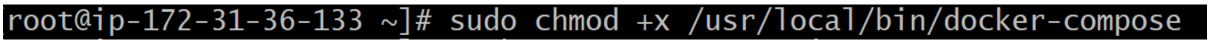
```
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```



```
root@ip-172-31-36-133 ~]# sudo curl -L "https://github.com/docker/compose/releases/latest/download
/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 58.6M  100 58.6M    0     0   104M      0 --:--:-- --:--:-- --:--:--  104M
```

➔ Next we set the Permission.

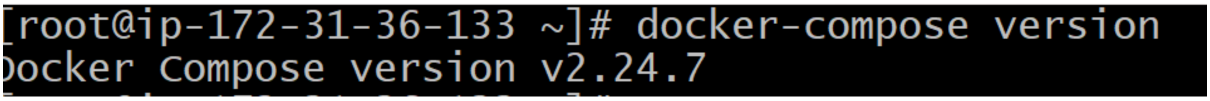
```
sudo chmod +x /usr/local/bin/docker-compose
```



```
root@ip-172-31-36-133 ~]# sudo chmod +x /usr/local/bin/docker-compose
```

➔ Next we run the below command

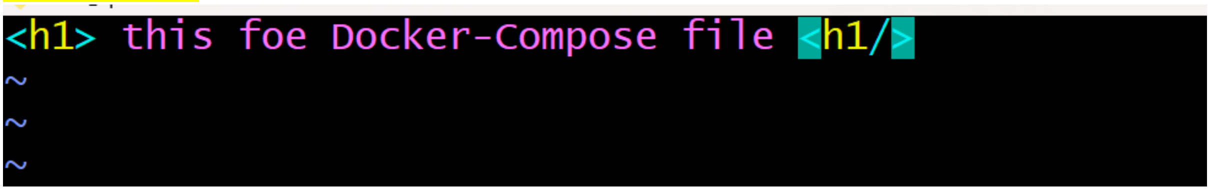
Docker-compose



```
[root@ip-172-31-36-133 ~]# docker-compose version
Docker Compose version v2.24.7
```

➔ After we write one sample index.html file by using below command.

vi index.html



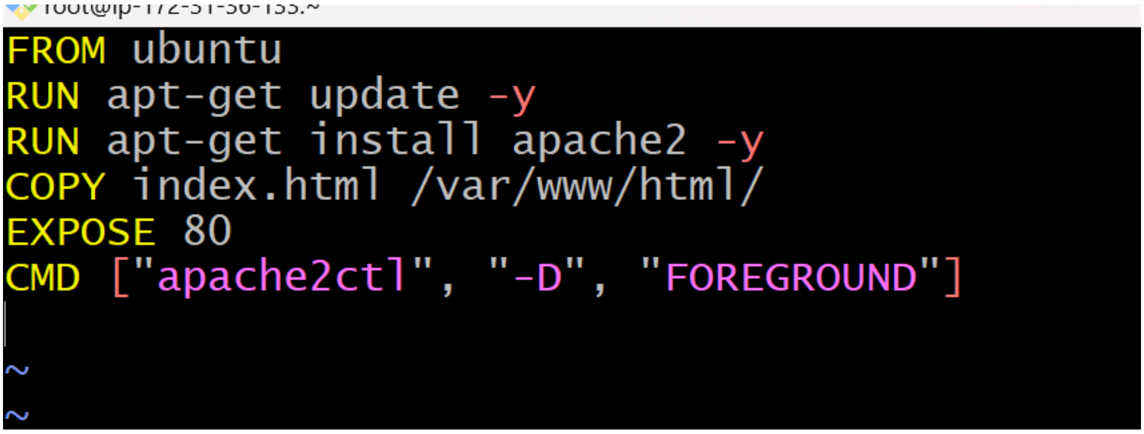
```
<h1> this foe Docker-Compose file <h1/>
~
~
~
```

And save the file and using command --→ :wq!

➔ After we write one sample Apache Docker file .

Vi dockerfile

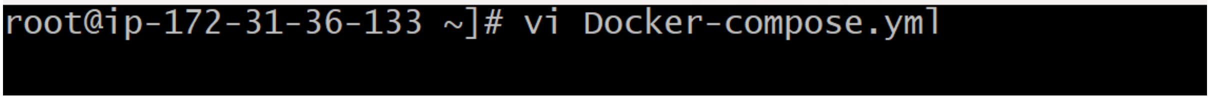
```
FROM ubuntu

RUN apt-get update -y

RUN apt-get install apache2 -y

COPY index.html /var/www/html/

EXPOSE 80

CMD ["apache2ctl", "-D", "FOREGROUND"]
```



```
FROM ubuntu
RUN apt-get update -y
RUN apt-get install apache2 -y
COPY index.html /var/www/html/
EXPOSE 80
CMD ["apache2ctl", "-D", "FOREGROUND"]

~
~
```

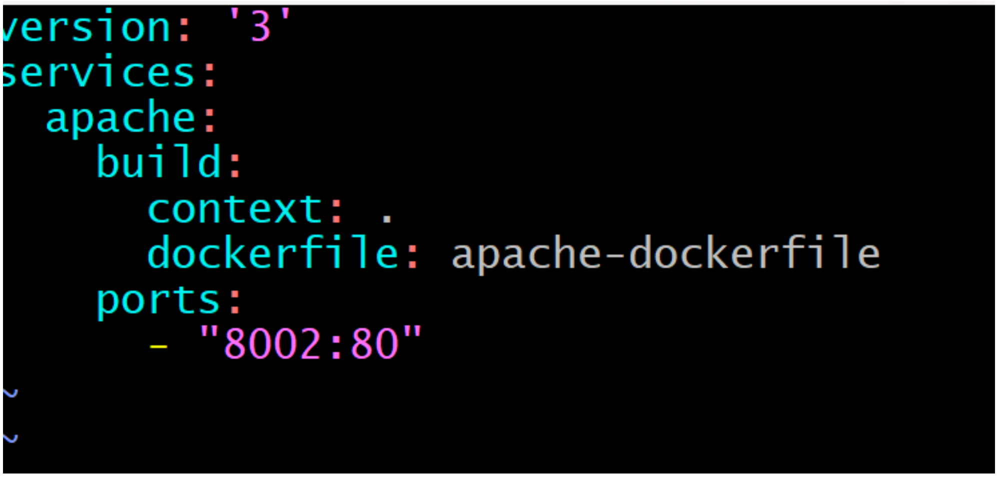
Ater we save the file **:wq!** Command

- ➔ After we write one Docker-Compose file by using below command.
  Vi docker-compose yml



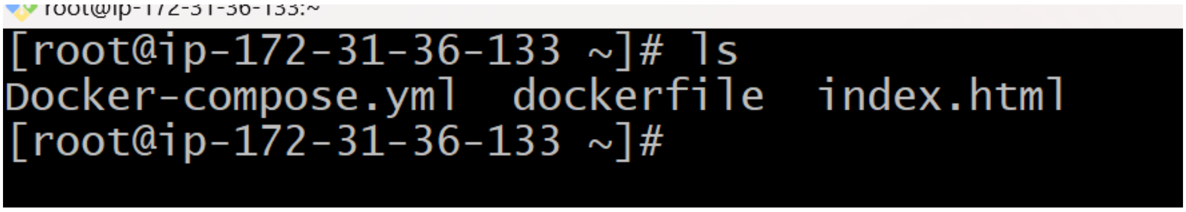
```
root@ip-172-31-36-133 ~]# vi Docker-compose.yml
```

```
version: '3'
services:
  apache:
    build:
      context: .
      dockerfile: apache-dockerfile
    ports:
      - "8002:80"
```



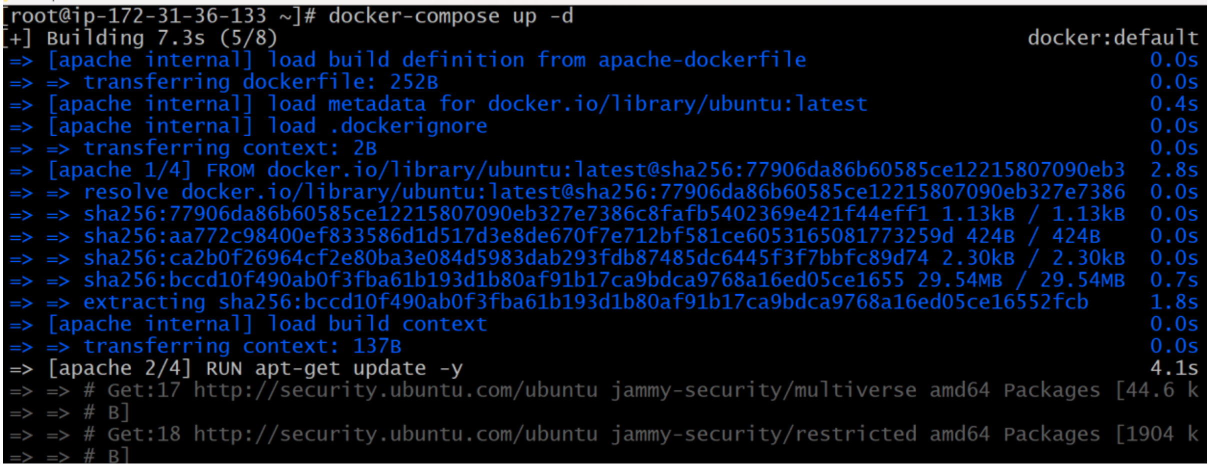
```
ersion: '3'
ervices:
  apache:
    build:
      context: .
      dockerfile: apache-dockerfile
    ports:
      - "8002:80"
~
~
```

And save the file :wq! Command



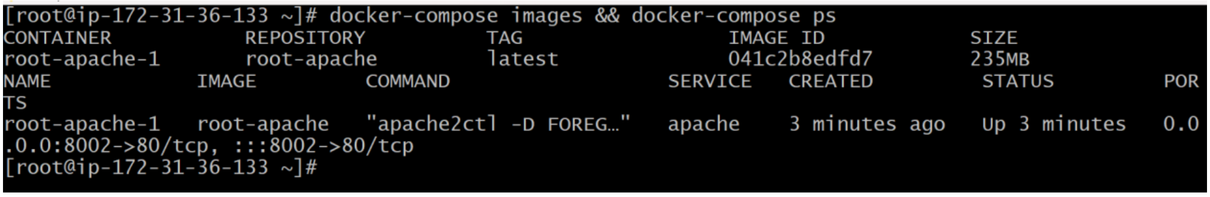
```
root@ip-172-31-36-133:~
[root@ip-172-31-36-133 ~]# ls
Docker-compose.yml  dockerfile  index.html
[root@ip-172-31-36-133 ~]#
```

- ➔ After we run the below commands
  Docker-compose up -d



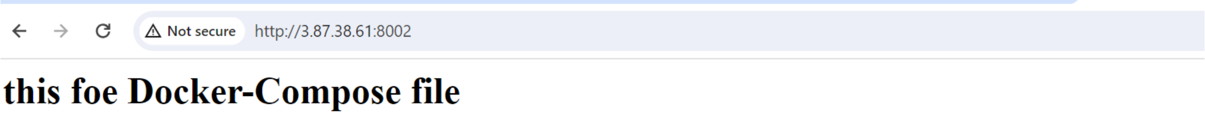
```
[root@ip-172-31-36-133 ~]# docker-compose up -d
[+] Building 7.3s (5/8)                                                                        docker:default
 => [apache internal] load build definition from apache-dockerfile                                       0.0s
 => => transferring dockerfile: 252B                                                                     0.0s
 => [apache internal] load metadata for docker.io/library/ubuntu:latest                                  0.4s
 => [apache internal] load .dockerignore                                                                 0.0s
 => => transferring context: 2B                                                                          0.0s
 => [apache 1/4] FROM docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb3  2.8s
 => => resolve docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386  0.0s
 => => sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff1 1.13kB / 1.13kB  0.0s
 => => sha256:aa772c98400ef833586d1d517d3e8de670f7e712bf581ce6053165081773259d 424B / 424B    0.0s
 => => sha256:ca2b0f26964cf2e80ba3e084d5983dab293fdb87485dc6445f3f7bbfc89d74 2.30kB / 2.30kB  0.0s
 => => sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce1655 29.54MB / 29.54MB  0.7s
 => => extracting sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce16552fcb     1.8s
 => [apache internal] load build context                                                                 0.0s
 => => transferring context: 137B                                                                        0.0s
 => [apache 2/4] RUN apt-get update -y                                                                   4.1s
 => => # Get:17 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.6 k
 => => # B]
 => => # Get:18 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1904 k
 => => # B]
```

- After we run this below commands .
  Docker-compose images---list images command
  Docker-compose ps -----list containers command



```
[root@ip-172-31-36-133 ~]# docker-compose images && docker-compose ps
CONTAINER           REPOSITORY          TAG                 IMAGE ID            SIZE
root-apache-1       root-apache         latest              041c2b8edfd7        235MB
NAME             IMAGE          COMMAND                   SERVICE    CREATED          STATUS          POR
TS
root-apache-1    root-apache    "apache2ctl -D FOREG…"    apache     3 minutes ago    Up 3 minutes    0.0
.0.0:8002->80/tcp, :::8002->80/tcp
[root@ip-172-31-36-133 ~]#
```

- After to access the page by using public Ip: port number. shown below



http://3.87.38.61:8002

this foe Docker-Compose file