Install Docker on EC2 instance on AWS

1. Launch Linux server on AWS

☑	My SSH Server	i-00cd09d56b3119b5b	Running	t2.micro	Initializing	View alarms +	us-east-2b

2. Connect to Linux server

```
The authenticity of host 'ec2-18-116-48-244.us-east-2.compute.amazonaws.com (18.116.48.244)' can't be established.
ED25519 key fingerprint is SHA256:/J24Yo5+Pm78Z1F/IOoQDWTm9XGa4G/vb/O2KMYi93U.
This host key is known by the following other names/addresses:
    C:\Users\jilan/.ssh/known_hosts:2: ec2-3-145-139-53.us-east-2.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-116-48-244.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.
  Version 2023.5.20240916:
  Version 2023.5.20241001:
  Version 2023.6.20241010:
  Version 2023.6.20241028:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Sep 15 05:32:03 2024 from 94.98.236.76
[ec2-user@ip-172-31-22-49 ~]$
```

3. Install Docker on AWS

```
[ec2-user@ip-172-31-22-49 ~]$ sudo yum install -y docker
Last metadata expiration check: 0:08:03 ago on Thu Oct 31 20:54:23 2024.
Dependencies resolved.
================================================================================
 Package                  Architecture   Version                     Repository     Size
================================================================================
Installing:
 docker                   x86_64         25.0.6-1.amzn2023.0.2       amazonlinux    44 M
Installing dependencies:
 containerd               x86_64         1.7.20-1.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
 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         1.1.13-1.amzn2023.0.1       amazonlinux   3.2 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm          7.4 MB/s | 401 kB   00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm           5.2 MB/s | 183 kB   00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm                2.4 MB/s |  75 kB   00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.9 MB/s |  58 kB   00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm          774 kB/s |  30 kB   00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm               4.2 MB/s |  84 kB   00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                     2.9 MB/s |  83 kB   00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm                   17 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm             22 MB/s |  35 MB   00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm                24 MB/s |  44 MB   00:01
--------------------------------------------------------------------------------
Total                                                           44 MB/s |  84 MB   00:01
```

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64          docker-25.0.6-1.amzn2023.0.2.x86_64          iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64         libcgroup-3.0-1.amzn2023.0.1.x86_64          libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64        libnftnl-1.2.2-2.amzn2023.0.2.x86_64         pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
```

4. Start docker

```
[ec2-user@ip-172-31-22-49 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-22-49 ~]$
```

5. Added user in Sudo

```
[ec2-user@ip-172-31-22-49 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-22-49 ~]$
```

6. Check docker status

```
[ec2-user@ip-172-31-22-49 ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[ec2-user@ip-172-31-22-49 ~]$
```

7. Created 2 files index.html , dockerfile

Index.html file

```
[ec2-user@ip-172-31-22-49 dockerfolder]$ cat index.html
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello Docker</title>
</head>
<body>
    <h1>Hello, Docker Its is my Website!</h1>
    <p>This is a simple static page served by Nginx.</p>
</body>
</html>
```

```
[ec2-user@ip-172-31-22-49 dockerfolder]$ cat dockerfile
FROM nginx:alpine
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

8. Build images and run

9. Docker image to verify

```
[ec2-user@ip-172-31-22-49 dockerfolder]$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
ec2image     latest    5f984790b4ef   2 minutes ago    47MB
[ec2-user@ip-172-31-22-49 dockerfolder]$
```

10. Verify container is running

```
[ec2-user@ip-172-31-22-49 dockerfolder]$ docker run -d -p 3000:80 ec2image
4a09355d806af8c96fcc0b34a21f52d82c86eca4e227fdf3825097ba5d7e33de
[ec2-user@ip-172-31-22-49 dockerfolder]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS                                       NAMES
4a09355d806a   ec2image   "/docker-entrypoint.…"   4 seconds ago   Up 3 seconds   0.0.0.0:3000->80/tcp, :::3000->80/tcp       unruffled_shamir
[ec2-user@ip-172-31-22-49 dockerfolder]$
```

11. Create volume in Docker

```
[ec2-user@ip-172-31-22-49 ~]$  docker volume create my-vol
my-vol
[ec2-user@ip-172-31-22-49 ~]$ docker volume ls
DRIVER    VOLUME NAME
local     my-vol
```

12. Docker network interface created

```
[ec2-user@ip-172-31-22-49 /]$ docker network create -d bridge my-net
f88893eed982ec9e4df3c629ea79e2d9c935154436f46f2567575f099821b979
```