root@NASSCHNGY-LP215:/mnt/c/Users/Saravanan Sivasankar/Desktop/SELECT/PERSONAL# ssh -i docker1.pem ec2-user@3.110.148.24

The authenticity of host '3.110.148.24 (3.110.148.24)' can't be established.

ED25519 key fingerprint is SHA256:zTypbb3/kZhLpaNolnEVgLnUJSdbiLf09QqbSHdx51w.

This host key is known by the following other names/addresses:

~/.ssh/known\_hosts:24: [hashed name]

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '3.110.148.24' (ED25519) to the list of known hosts.

A newer release of "Amazon Linux" is available.

Version 2023.9.20250929:

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: Sat Sep 27 06:43:25 2025 from 49.37.218.160

[ec2-user@ip-172-31-45-2 ~]$ pwd

/home/ec2-user

[ec2-user@ip-172-31-45-2 ~]$ docker version

Client:

Version: 25.0.8

API version: 1.44

Go version: go1.24.6

Git commit: 0bab007

Built: Thu Jul 31 00:00:00 2025

OS/Arch: linux/amd64

Context: default

permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/version": dial unix /var/run/docker.sock: connect: permission denied

[ec2-user@ip-172-31-45-2 ~]$ sudo su

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker version

Client:

Version: 25.0.8

API version: 1.44

Go version: go1.24.6

Git commit: 0bab007

Built: Thu Jul 31 00:00:00 2025

OS/Arch: linux/amd64

Context: default

Server:

Engine:

Version: 25.0.8

API version: 1.44 (minimum version 1.24)

Go version: go1.24.6

Git commit: 71907ca

Built: Thu Jul 31 00:00:00 2025

OS/Arch: linux/amd64

Experimental: false

containerd:

Version: 2.0.6

GitCommit: 991cc3363c290ffd074e069f2b3034c7286ecbe0

runc:

Version: 1.2.6

GitCommit: 6c52b3fc541fb26fe8c374d5f58112a0a5dbda66

docker-init:

Version: 0.19.0

GitCommit: de40ad0

[root@ip-172-31-45-2 ec2-user]# docker info

Client:

Version: 25.0.8

Context: default

Debug Mode: false

Plugins:

buildx: Docker Buildx (Docker Inc.)

Version: 0.12.1

Path: /usr/libexec/docker/cli-plugins/docker-buildx

Server:

Containers: 1

Running: 0

Paused: 0

Stopped: 1

Images: 1

Server Version: 25.0.8

Storage Driver: overlay2

Backing Filesystem: xfs

Supports d\_type: true

Using metacopy: false

Native Overlay Diff: true

userxattr: false

Logging Driver: json-file

Cgroup Driver: systemd

Cgroup Version: 2

Plugins:

Volume: local

Network: bridge host ipvlan macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog

Swarm: inactive

Runtimes: runc io.containerd.runc.v2

Default Runtime: runc

Init Binary: docker-init

containerd version: 991cc3363c290ffd074e069f2b3034c7286ecbe0

runc version: 6c52b3fc541fb26fe8c374d5f58112a0a5dbda66

init version: de40ad0

Security Options:

seccomp

Profile: builtin

cgroupns

Kernel Version: 6.1.150-174.273.amzn2023.x86\_64

Operating System: Amazon Linux 2023.8.20250915

OSType: linux

Architecture: x86\_64

CPUs: 2

Total Memory: 904.8MiB

Name: ip-172-31-45-2.ap-south-1.compute.internal

ID: 7ea44e17-df0b-4d35-a092-02fbb9d1e75e

Docker Root Dir: /var/lib/docker

Debug Mode: false

Username: yuvarani1996

Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

[root@ip-172-31-45-2 ec2-user]# docker run hello-world

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

17eec7bbc9d7: Pull complete

Digest: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31

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/

[root@ip-172-31-45-2 ec2-user]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 41f689c20910 7 weeks ago 192MB

hello-world latest 1b44b5a3e06a 7 weeks ago 10.1kB

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker run -d -p 8080:80 --name mynginx nginx

c76590b39b08e6af969d9b9e17b39ecb2b799d1613f1e85aa212da1a6aadceb3

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c76590b39b08 nginx "/docker-entrypoint.…" 5 seconds ago Up 5 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp mynginx

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c76590b39b08 nginx "/docker-entrypoint.…" 17 seconds ago Up 17 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp mynginx

[root@ip-172-31-45-2 ec2-user]# docker stop mynginx

mynginx

[root@ip-172-31-45-2 ec2-user]# docker rm mynginx

mynginx

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker volume create myvolume

myvolume

[root@ip-172-31-45-2 ec2-user]# docker volume ls

DRIVER VOLUME NAME

local myvolume

[root@ip-172-31-45-2 ec2-user]# docker run -d -v myvolume:/data --name voltest nginx

daf3e7b4a765adfd9cd96427c812ebb2818bb2e4b6a27c693f75a4ed6a50f78b

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]#

[root@ip-172-31-45-2 ec2-user]# docker network ls

NETWORK ID NAME DRIVER SCOPE

54b8c8f100ee bridge bridge local

55197aa6d760 host host local

502c3426a288 none null local

[root@ip-172-31-45-2 ec2-user]# docker network inspect bridge

[

{

"Name": "bridge",

"Id": "54b8c8f100eebd3907c9eb6c216771c948336c9b191f52100a1ae7430c97bb70",

"Created": "2025-10-02T14:12:52.965549161Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.17.0.0/16",

"Gateway": "172.17.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"daf3e7b4a765adfd9cd96427c812ebb2818bb2e4b6a27c693f75a4ed6a50f78b": {

"Name": "voltest",

"EndpointID": "fe0f497b059d4a5e6864aa37882d4b58b71879051f2f152433bb6a605de999d2",

"MacAddress": "02:42:ac:11:00:02",

"IPv4Address": "172.17.0.2/16",

"IPv6Address": ""

}

},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

[root@ip-172-31-45-2 ec2-user]# docker network create mynet

8a693939d64463b8f2fdc57bac0b762957af5801726abc291162339027114679

[root@ip-172-31-45-2 ec2-user]# docker run -d --name container1 --network mynet nginx

01429ca238540fbf05ba12c027687ec4c1b28516fd2e5b9b08a95966b63215f9

[root@ip-172-31-45-2 ec2-user]# docker run -d --name container2 --network mynet nginx

ab5211f575807a4a84af64c8553b763fbd1f80fae29416c50694e283946b2226

[root@ip-172-31-45-2 ec2-user]#