

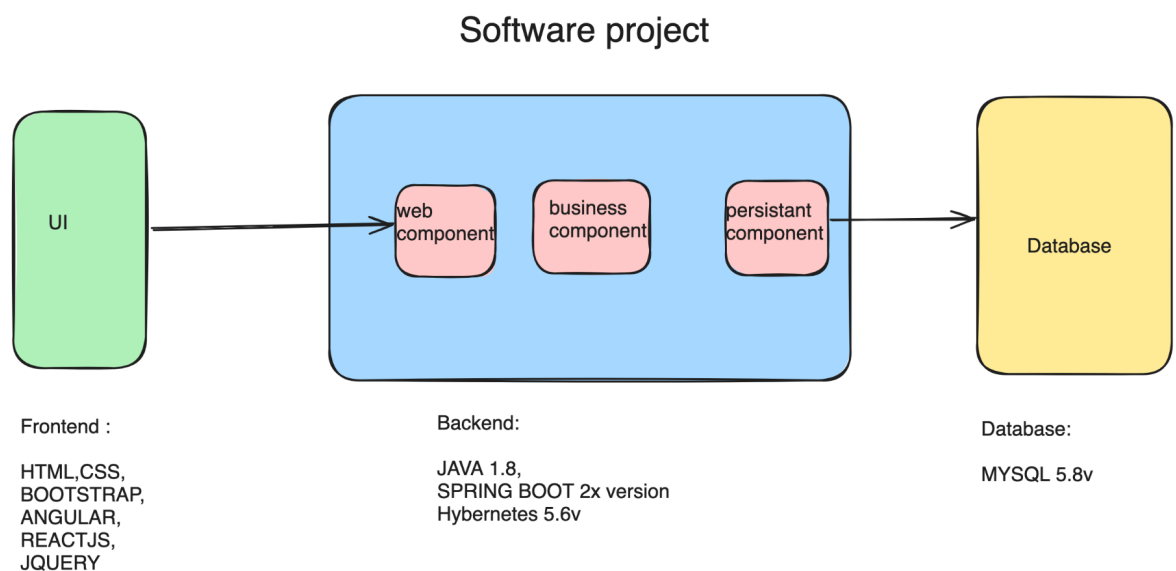
Day1:

As a devops engineer will do Application Deployment. Will come to application deployment but before that we need to understand what is software project

What is a software project ?

Collection of programs is called software project

- 1) Frontend components (user interface logic)
- 2) Backend components (business logic)
- 3) Database components (Persistence logic)



So first we need to setup the environment :

Installing required softwares for developers to start coding.

Developers start the coding and store the code in github (central repository to store code).

Devops engineers comes to the picture :

As a devops engineer you need to deploy the application to multiple environments before deploying to production.

Types of environments:

1. Dev environment (Developer environment)
2. SIT environment (system integration testing environment)
3. UAT environment (user acceptance testing environment - client testing)
4. Preprod environment (before production environment)
5. Prod environment (live running and public accessing application)

To run our application or deploy we need to take care of all the softwares and all the environments.

To Install Frontend, backend , database components into different environments is a complex work and a tedious task (difficult task)

UI : HTML,CSS,BOOTSTRAP,ANGULAR,REACTJS,JQUERY

Backend: JAVA 1.6, SPRING BOOT 2x version , Hybernates 5.6v

Database: MYSQL 5.8v

All the softwares and its version should be the same , suppose if there is a change in version compatibility issues may arise , environment dependency issues will occur.

Here comes the **containerization**.

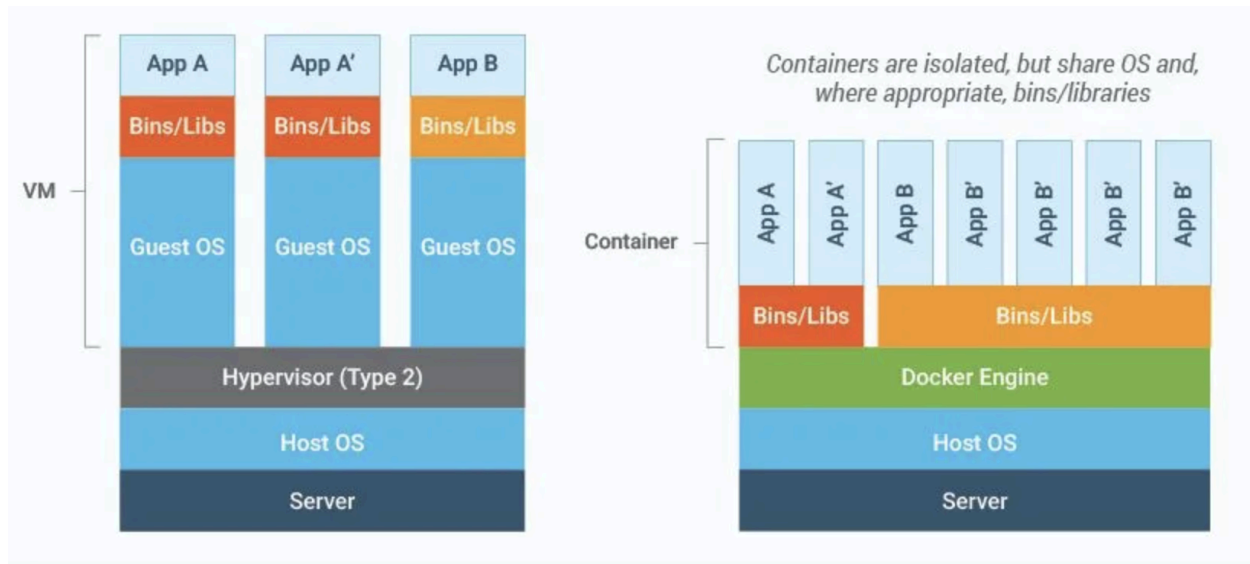
Before going deeper . will understand few basics

Skype is an application software (to run it consumes few resources -> ram,cpu,disk)

systems on a H



need to pack



Comparison Factor	Virtualization	Containerization
Speed	Slower startup and resource allocation.	Faster startup and resource allocation.
Resources	High resource overhead.	Minimal resource overhead.
Security & Isolation	Strong isolation between VMs.	Less isolation, but improved with features.
Portability	Limited portability between platforms.	Highly portable across different platforms.
Operating System	Multiple OS instances.	Shared OS kernel.
Application Lifecycle	Longer deployment and scaling.	Quick deployment and scaling.

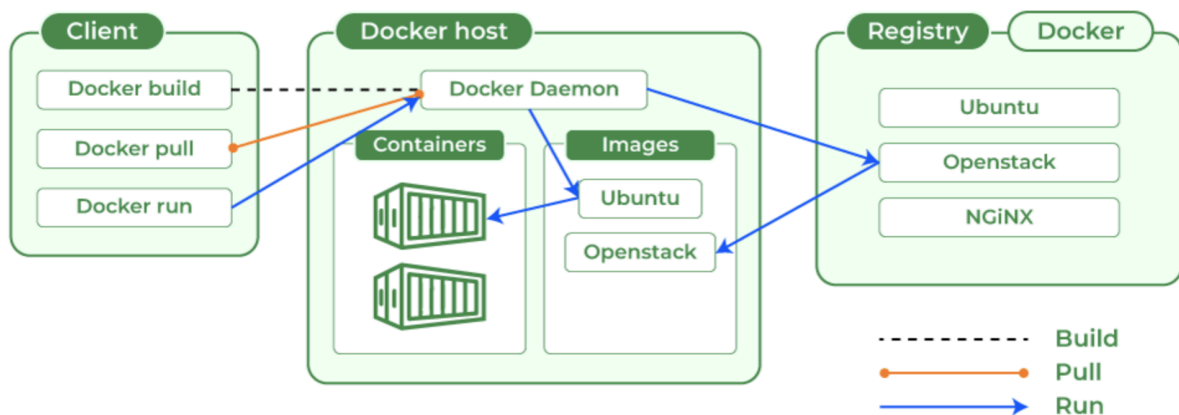
What is Docker ?

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called [containers](#) that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

Running Docker on AWS provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.



Explain docker architecture ?



Docker makes use of a client-server architecture. The Docker client talks with the docker daemon which helps in building, running, and distributing the docker containers. The Docker client runs with the daemon on the same system or we can connect the Docker client with the Docker daemon remotely. With the help of REST API over a UNIX socket or a network, the docker client and daemon interact with each other.

What is Docker Daemon?

Docker daemon manages all the services by communicating with other daemons. It manages docker objects such as images, containers, networks, and volumes with the help of the API requests of Docker.

Docker Client

With the help of the docker client, the docker users can interact with the docker. The docker command uses the Docker API. The Docker client can communicate with multiple daemons. When a docker client runs any docker command on the docker terminal then the terminal sends instructions to the daemon. The Docker daemon gets those instructions from the docker client withinside the shape of the command and REST API's request.

The main objective of the docker client is to provide a way to direct the pull of images from the docker registry and run them on the docker host. The common commands which are used by clients are **docker build**, **docker pull**, and **docker run**.

Docker Host

A Docker host is a type of machine that is responsible for running more than one container. It comprises the Docker daemon, Images, Containers, Networks, and Storage.

Docker Registry

All the docker images are stored in the docker registry. There is a public registry which is known as a [docker hub](#) that can be used by anyone. We can run our private registry also. With the help of **docker run** or **docker pull** commands, we can pull the required images from our configured registry. Images are pushed into the configured registry with the help of the **docker push** command.

INSTALLATION OF DOCKER IN EC2:

How to install docker to make hands dirty (playground environment)

Create an instance in a Ec2:

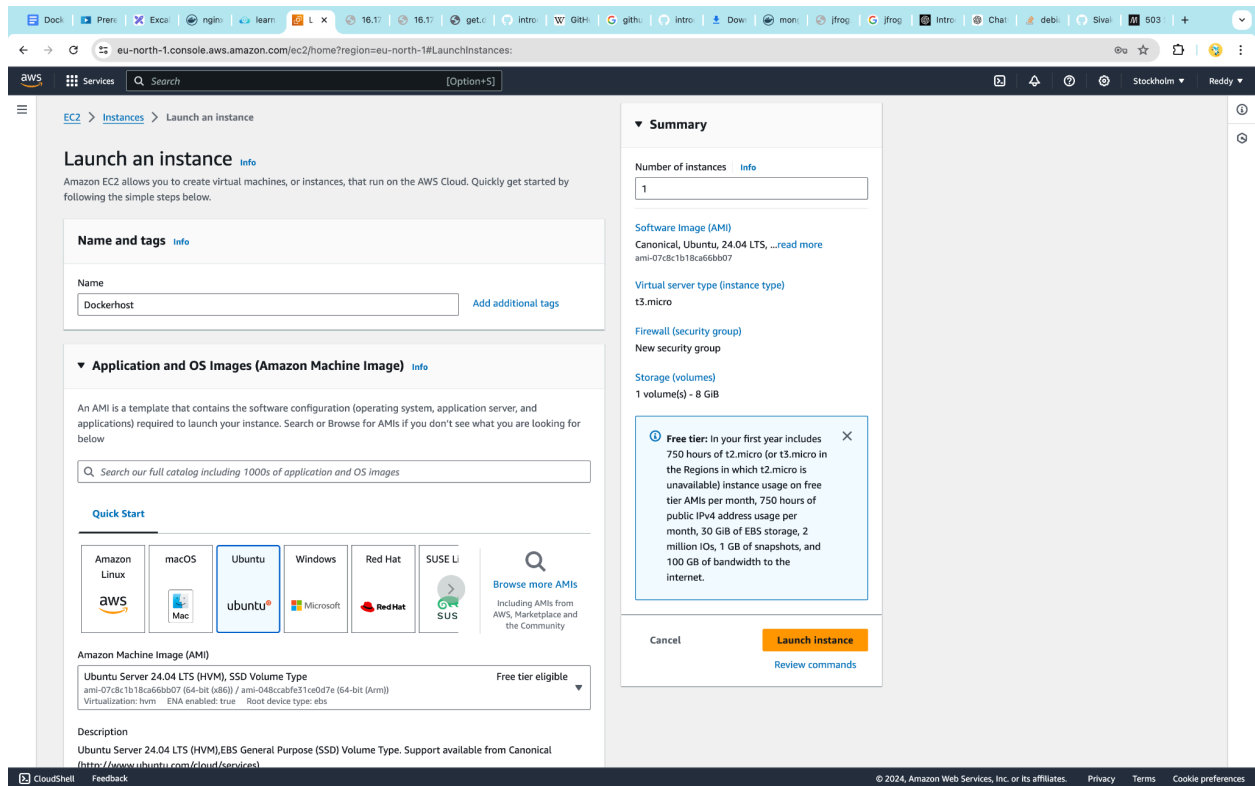
Follow the steps

The screenshot displays the AWS Management Console interface. At the top, a green banner indicates the successful termination of instance `i-0c452851e10718fb3`. Below this, the 'Instances' page shows a list of EC2 instances. The instance `docker host1` (ID: `i-0c452851e10718fb3`) is highlighted, showing it is in the 'Shutting-down' state. The instance details panel for `docker host1` is open, displaying the following information:

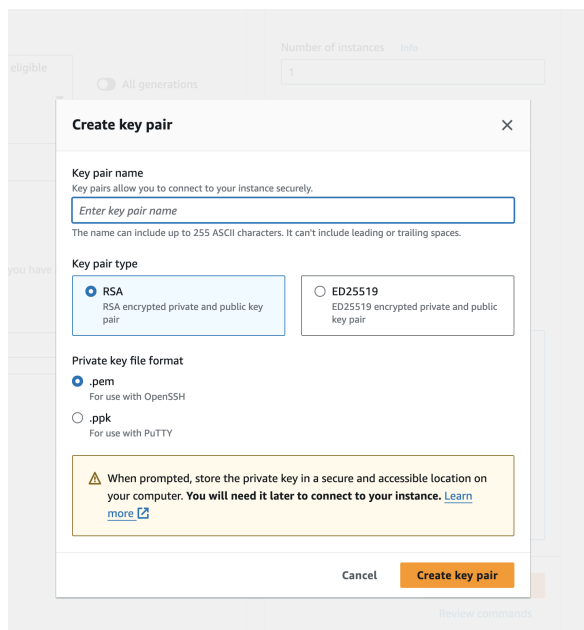
- Instance summary:** Instance ID `i-0c452851e10718fb3` (docker host1), IP6 address `-`, Hostname type `IP name: ip-172-31-45-89.eu-north-1.compute.internal`, Answer private resource DNS name `IPV4 (A)`, Auto-assigned IP address `16.170.248.186 [Public IP]`.
- Public IPv4 address:** `16.170.248.186` | [open address](#)
- Private IPv4 addresses:** `172.31.45.89`
- Public IPv4 DNS:** `ec2-16-170-248-186.eu-north-1.compute.amazonaws.com` | [open address](#)
- Private IP DNS name (IPv4 only):** `ip-172-31-45-89.eu-north-1.compute.internal`
- Instance type:** `t3.micro`
- VPC ID:** `vpc-00cee3be89041810b` | [open address](#)
- Elastic IP addresses:** `-`
- AWS Compute Optimizer finding:** [Opt-in to AWS Compute Optimizer for recommendations. | Learn more](#)

The bottom of the console shows the footer with copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Click on Launch instance on right top.



Selecting OS and hardware resources required for our instance.



A key pair is a combination of a public key that is used to encrypt data and a private key that is used to decrypt data.

VS Services Search [Option+S]

Network [Info](#)
vpc-00cee3be89041810b

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Additional charges apply when outside of free tier allowance

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

We'll create a new security group called 'launch-wizard-8' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

▼ Configure storage [Info](#) [Advanced](#)

1x 8 GIB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

▼ Summary

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)
ami-07c8c1b18ca66bb07

Virtual server type (instance type)
t3.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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#) [Review commands](#)

Security group select anywhere for practise and ssh (22 port for admin login) and http(80 for browser).

Click on Launch instance.

Successfully initiated termination of i-0c452851e10718fb3

Instances (1/6) [Info](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

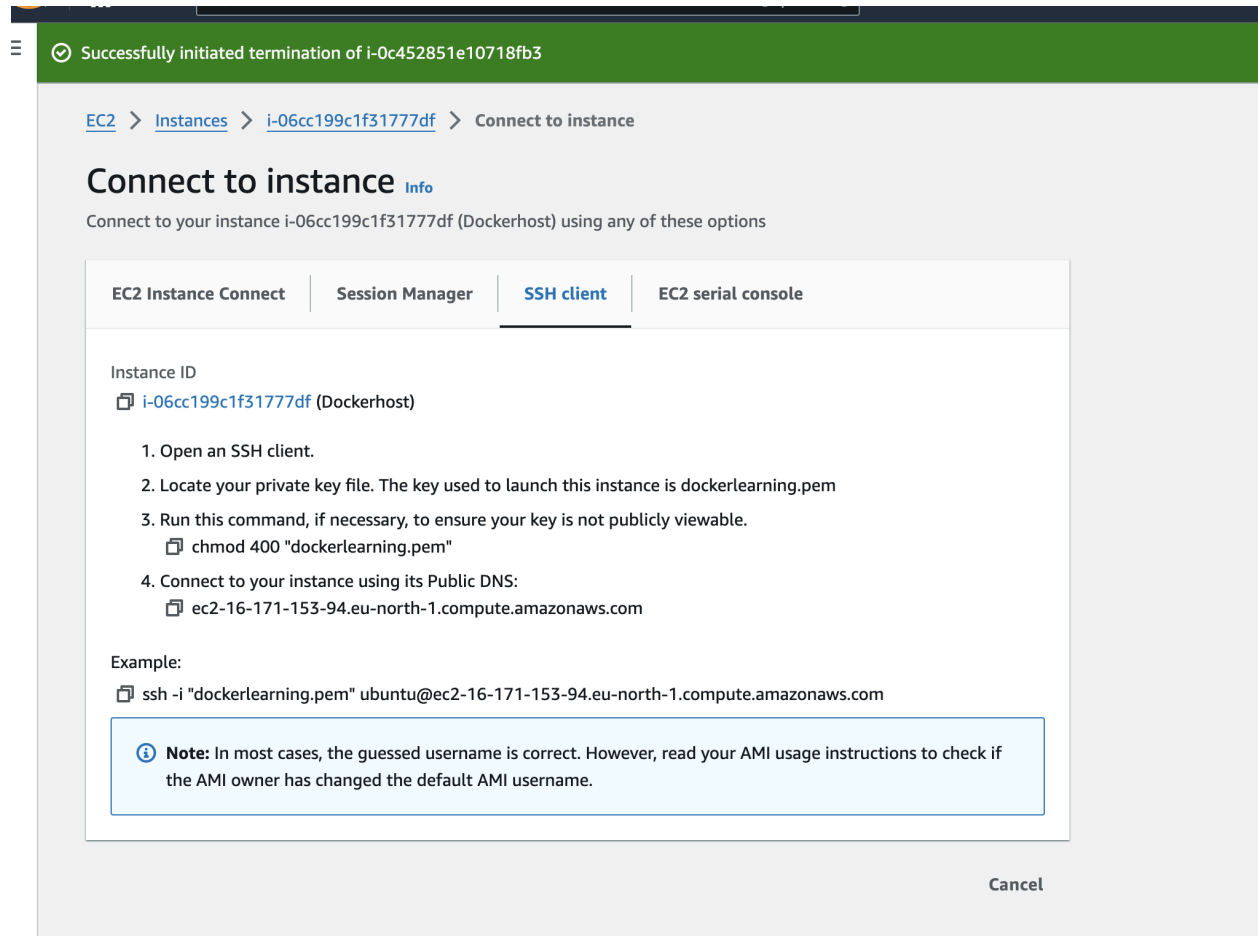
Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	new	i-01f280d09a730a952	Stopped	t3.medium	-	View alarms	eu-north-1a	-	-
<input type="checkbox"/>	mario	i-0f81d144144173ad6	Stopped	t3.micro	-	View alarms	eu-north-1b	-	-
<input type="checkbox"/>	CI-CD	i-0f33a466b9332bc1e	Stopped	t3.large	-	View alarms	eu-north-1b	-	-
<input type="checkbox"/>	server	i-02cd1f9cc7af0bb35	Stopped	t3.large	-	View alarms	eu-north-1b	-	-
<input type="checkbox"/>	docker host1	i-0c452851e10718fb3	Terminated	t3.micro	-	View alarms	eu-north-1b	-	-
<input checked="" type="checkbox"/>	Dockerhost	i-06cc199c1f31777df	Running	t3.micro	2/2 checks passed	View alarms	eu-north-1b	ec2-16-171-153-94.eu-...	16.171.153.94

i-06cc199c1f31777df (Dockerhost)

My vm is up and running.

Click on connect.



My key name is dockerlearning.pem

Provide required permissions `chmod 400 dockerlearning.pem`

```
[192:Downloads n.sivakumar$ ls -lrt dockerlearning.pem
-rw-r--r--@ 1 n.sivakumar 1493847943 1678 Jul 8 07:26 dockerlearning.pem
[192:Downloads n.sivakumar$ chmod 400 dockerlearning.pem
[192:Downloads n.sivakumar$ ls -lrt dockerlearning.pem
-r-----@ 1 n.sivakumar 1493847943 1678 Jul 8 07:26 dockerlearning.pem
192:Downloads n.sivakumar$ █
```

Chmod -> change mode for more details (

<https://www.geeksforgeeks.org/chmod-command-linux/>)

***What is the Secure Shell (SSH) protocol?

The Secure Shell (SSH) protocol is a method for securely sending commands to a computer over an unsecured network. SSH uses cryptography to authenticate and encrypt connections between devices.

```
ssh -i "dockerlearning.pem" ubuntu@ec2-16-171-153-94.eu-north-1.compute.amazonaws.com
```

-i -> identity

Privatekey -> dockerlearning.pem

Public DNS or IP ubuntu@ec2-16-171-153-94.eu-north-1.compute.amazonaws.com

```
192:Downloads n.sivakumar$ ls -lrt dockerlearning.pem
-rw-r--r--@ 1 n.sivakumar 1493847943 1678 Jul 8 07:26 dockerlearning.pem
192:Downloads n.sivakumar$ chmod 400 dockerlearning.pem
192:Downloads n.sivakumar$ ls -lrt dockerlearning.pem
-r-----@ 1 n.sivakumar 1493847943 1678 Jul 8 07:26 dockerlearning.pem
192:Downloads n.sivakumar$ ssh -i "dockerlearning.pem" ubuntu@ec2-16-171-153-94.eu-north-1.compute.amazonaws.com
Initialization failed; error getting cert! Post "https://service.authn-prod.fkcloud.in/cli/getToken?client_id=jumphost": dial tcp: lookup service.authn-prod.fkcloud.in: no such host
d.in: no such hostError: Post "https://service.authn-prod.fkcloud.in/cli/getToken?client_id=jumphost": dial tcp: lookup service.authn-prod.fkcloud.in: no such host
t
The authenticity of host 'ec2-16-171-153-94.eu-north-1.compute.amazonaws.com (16.171.153.94)' can't be established.
ED25519 key fingerprint is SHA256:hKT8Kny1jJ6qWVB0Ukj1a5HMP6W1rqoygIbi9/+A570.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-16-171-153-94.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Jul 8 02:16:10 UTC 2024

System load: 0.16      Temperature:      -273.1 C
Usage of /:  22.9% of 6.71GB  Processes:       108
Memory usage: 33%      Users logged in: 0
Swap usage:  0%        IPv4 address for ens5: 172.31.47.54

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-47-54:~$ sudo su
root@ip-172-31-47-54:/home/ubuntu#
```

Sudo su -> switch to sudo (Sudo is a command in Linux that allows users to run commands with privileges that only root user have. It helps users to do tasks with administrative power)

I am installing docker , I need root privileges.

<https://get.docker.com/>

download the script

```
# $ curl -fsSL https://get.docker.com -o install-docker.sh
```

run the script either as root, or using sudo to perform the installation.

```
# $ sudo sh install-docker.sh
```

```

ubuntu@ip-172-31-47-54:~$ sudo su
[root@ip-172-31-47-54:/home/ubuntu# curl -fsSL https://get.docker.com -o install-docker.sh
[root@ip-172-31-47-54:/home/ubuntu# sudo sh install-docker.sh
# Executing docker install script, commit: 6d9743e9656cc56f699a64800b098d5ea5a60020
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certific
Scanning processes...
Scanning linux images...
+ sh -c install -m 0755 -d /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker
+ sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce docker-ce-cli contain
plugin >/dev/null
Scanning processes...
Scanning linux images...
+ sh -c docker version
Client: Docker Engine - Community
Version:          27.0.3

```

Docker is installed

docker --version

Docker version 27.0.3, build 7d4bcd8

#docker images (to list the images)

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

Day2:

