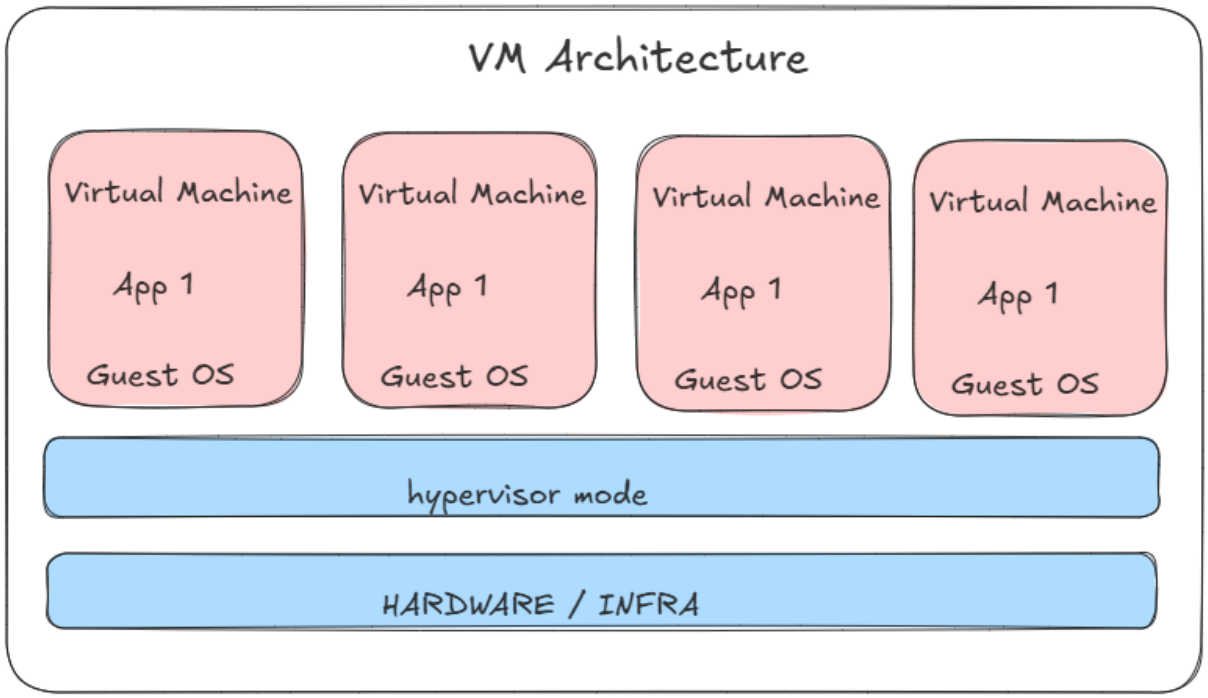
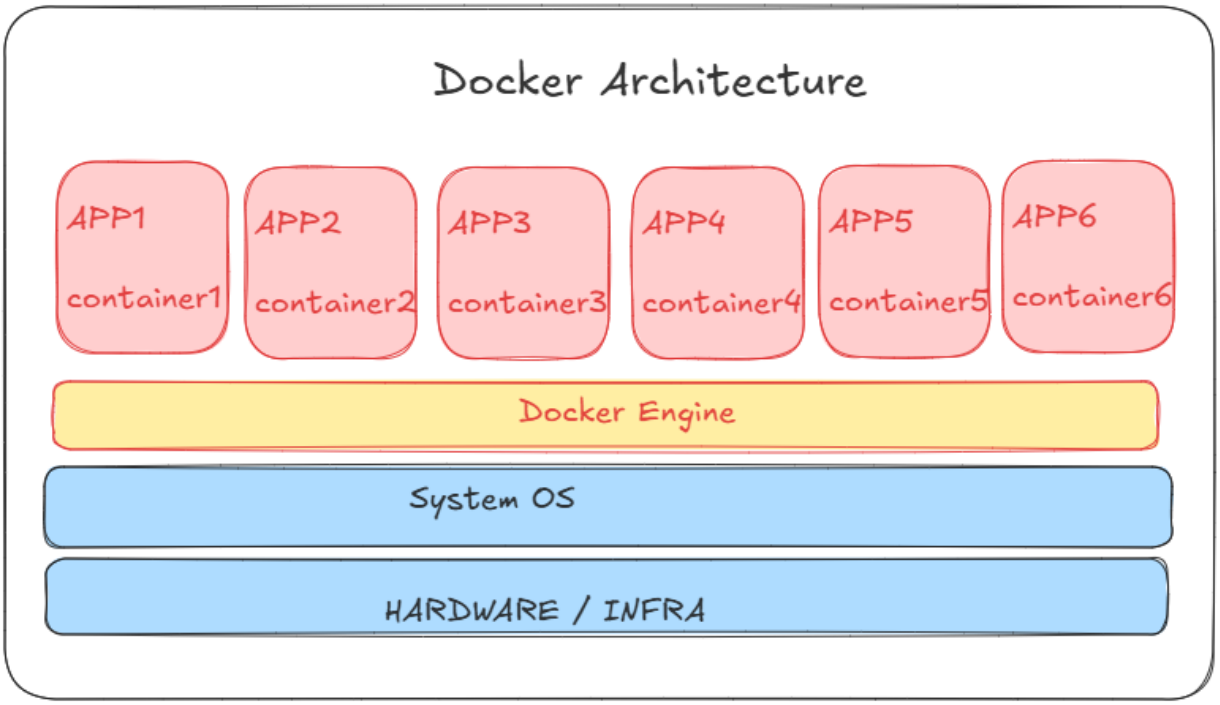
Docker

Virtual Machine Architecture.



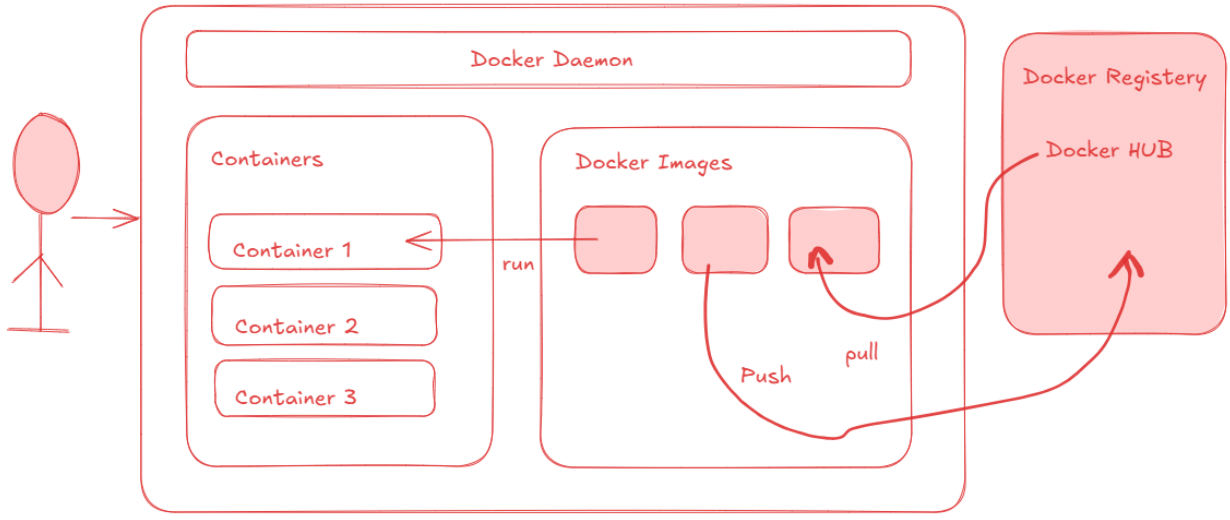
Runs on hypervisor mode with its own OS.
Requires a full OS for each VM.
Using More resources, heavy weighted (more memory, CPU and storage)
starting time: slow
performance: less due full OS overhead
scaling is also slow due to resources.
Security: its providing better isolation.
Use Cases: for better isolation, when you are using multiple OS or old/legacy application

Docker:



Architecture: Using host OS kernel
OS: shares the OS, it uses light weight images
Resources: light weight resources and also give less overhead
Start Up time: fast (seconds)
Isolation: Process-level isolation
Security: we need some additional configuration
Use Cases: CI/CD pipelines, better microservices

How Docker Works?

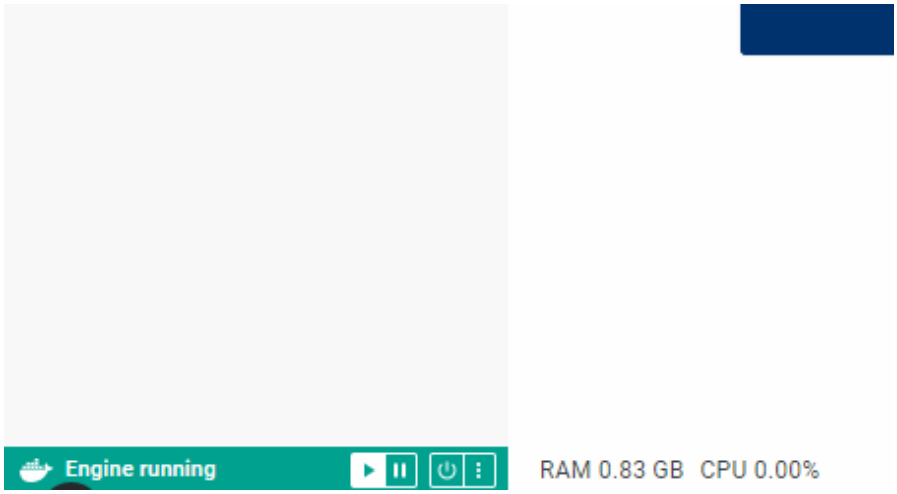


Install Docker using Docker Desktop

then verify the docker version

```
C:\Users\NEW>docker --version
Docker version 26.1.4, build 5650f9b
```

Start the docker desktop, wait for engine to start.



Verify the engine is running.

```
C:\Users\NEW>docker version
Client:
 Version:           26.1.4
 API version:       1.45
 Go version:        go1.21.11
 Git commit:        5650f9b
 Built:             Wed Jun  5 11:29:54 2024
 OS/Arch:           windows/amd64
 Context:           desktop-linux

Server: Docker Desktop 4.31.1 (153621)
 Engine:
  Version:          26.1.4
  API version:      1.45 (minimum version 1.24)
  Go version:       go1.21.11
  Git commit:       de5c9cf
  Built:            Wed Jun  5 11:29:22 2024
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.33
  GitCommit:        d2d58213f83a351ca8f528a95fbd145
 runc:
  Version:          1.1.12
  GitCommit:        v1.1.12-0-g51d5e94
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Now we can verify docker is running perfectly as we can see both client and server details.

To pull the image

```
C:\Users\NEW>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
2c0a233485c3: Pull complete
21577e00f2ba: Pull complete
c294da35c13e: Pull complete
facc8f3107c1: Pull complete
de4342aa4ad8: Pull complete
4643f1cf56c2: Pull complete
139aca660b47: Pull complete
b10e1082570e: Pull complete
26313a3e0799: Pull complete
d43055c38217: Pull complete
Digest: sha256:45f5ae20cfe1d6e6c43684dfffef17db1e1e8dc9bf7133ceaafb25c16b10f31b
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

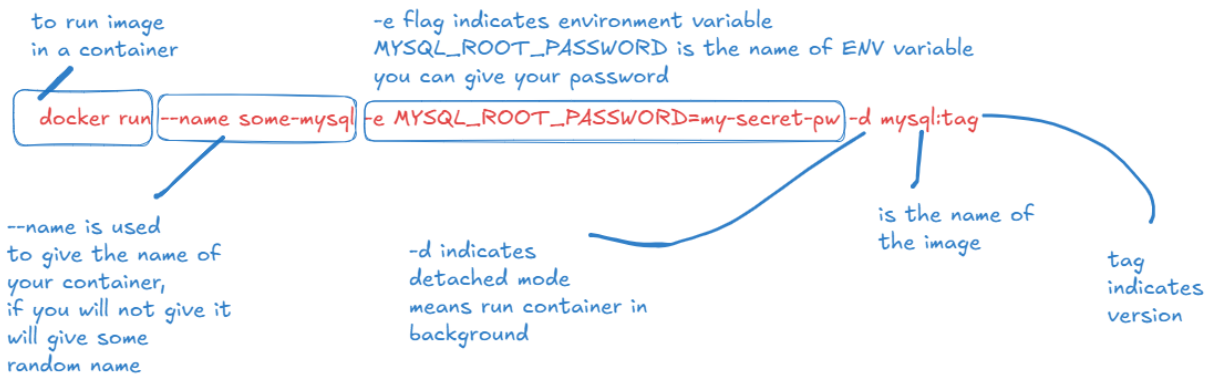
To verify Now:

```
C:\Users\NEW>docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
mysql        latest    a52cba19e8cc   9 days ago   797MB
```

If I have not pulled any image but still I want to run it directly. So it will pull the image first and then it will run it.

```
C:\Users\NEW>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:d715f14f9eca81473d9112df50457893aa4d099adeb4729f679006bf5ea12407
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/
```

Let's Say we want run MySQL image in a container and we want use mysql Database as well.




```
C:\Users\NEW>docker run --name mysqlserver -e MYSQL_ROOT_PASSWORD=123456 -d mysql
4149ae1029e67c2adb9fcc21da78b08d6e5284c66016fb2fb52892e01837e1e4
```

If you are getting ID its a container ID. Let's verify container Status

```
C:\Users\NEW>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                 NAMES
4149ae1029e6   mysql     "docker-entrypoint.s…"   56 seconds ago   Up 54 seconds   3306/tcp, 33060/tcp   mysqlserver
```

Let's Get into this container and use mysql database.

```
C:\Users\NEW>docker exec -it mysqlserver mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database pw;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| pw                 |
| sys                |
+--------------------+
5 rows in set (0.05 sec)
```

you can exit using the exit command.

Let's understand How to create Docker File

Let's understand some basic terms which we can use to create Dockerfile
Whenever you create docker file keep that in your mind like docker file doesn't have any extention.

FROM:

Specify the BASE Image, which you want to use like ubuntu or any other OS
FROM ubuntu:20.04

RUN:

Executes the command in a new layer on the top of current image
RUN apt-get update && apt-get install python3

EXPOSE:

informs docker to start the container which listens to specific port no
EXPOSE 5000

ADD:

copy files, directories from some remote locations
ADD config.tar /etc/config

COPY:

Copy local files
COPY index.html ./app

CMD:

provides defaults for an executing containers
CMD ["python3", "logger.py"]

ENTRYPOINT:

configure a container that will run as executable
ENTRYPOINT ["npm","run","dev"]

WORKDIR:

set a working directory in a project
WORKDIR /app

Let's Create one folder named portfolio and create index.html in that location. Add below code.

```html
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sonam Soni</title>
</head>
<body>
    <h1>Sonam Soni</h1><hr>
    <!-- p*4>lorem -->
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Optio explicabo i
    <p>Rerum non corporis quaerat porro a ipsa blanditiis molestiae! Tenetur poss
    <p>Modi ex cumque totam debitis quo hic numquam, voluptas voluptatum doloribu
    <p>Eius, recusandae? Nemo veniam temporibus blanditiis modi minima fugit quod
    <h2>SKILLS</h2><hr>
    <ul>
        <!-- li*5>lorem5 -->
        <li>Lorem ipsum dolor sit amet.</li>
        <li>Officia voluptate repellat error quaerat!</li>
        <li>Dolorem numquam impedit suscipit. Maxime?</li>
        <li>Earum repudiandae cumque voluptatibus aliquam!</li>
        <li>Eveniet fuga excepturi provident unde?</li>
    </ul>
</body>
```

Create Docker file (Dockerfile)

```
FROM nginx:alpine

COPY index.html /usr/share/nginx/html

EXPOSE 80
```

Execute Below Commands

1. Build an Image: docker build -t portfolio .
2. Run Image in a container: docker run --name my-container -d -p 80:80 portfolio
3. here first port indicates host port and 2nd port is docker container port.
4. verify the running container: docker ps (if it is up then check localhost in browser and verify)
5. Stop the container: docker stop my-container
6. Remove Container: docker rm my-container

Docker Image push to Docker Hub:

```
D:\PhysicsWalla\Devops OCT Batch\oct-git\01-02-Docker>docker login
Authenticating with existing credentials...
Login Succeeded

D:\PhysicsWalla\Devops OCT Batch\oct-git\01-02-Docker>docker tag portfolio sonamsoni/portfolio

D:\PhysicsWalla\Devops OCT Batch\oct-git\01-02-Docker>docker push sonamsoni/portfolio
Using default tag: latest
The push refers to repository [docker.io/sonamsoni/portfolio]
66415d53c518: Pushed
5a760029e979: Mounted from library/nginx
23625999797d: Mounted from library/nginx
9aa22afcf27f: Mounted from library/nginx
59a5cb944b91: Mounted from library/nginx
598e577f3a23: Mounted from library/nginx
fd5f65a144ef: Mounted from library/nginx
a8903c9578e9: Mounted from library/nginx
ce5a8cde9eee: Mounted from library/nginx
latest: digest: sha256:f59e9462ae05e6325b67af5e376486a0452e59f178f04c730b584d1d06b5357c size: 2196
```