MINDSONAI

ML-OPS

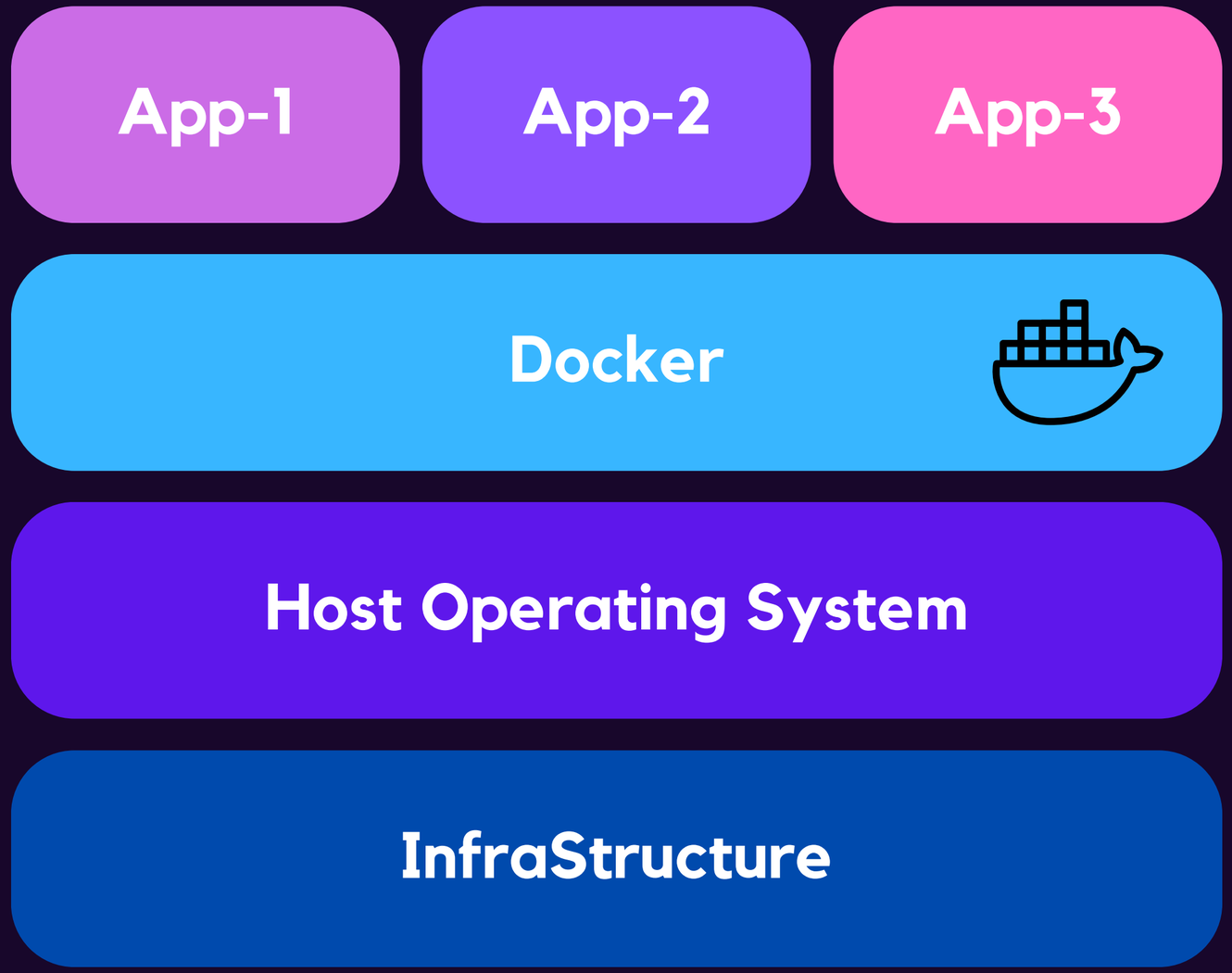
CONTAINERS

linkedin.com/rishabhio

ML-OPs and LLM-OPs Landscape in 2024

This slide is just a FYI and allows you to explore more tools in the ml-ops and llm-ops domain

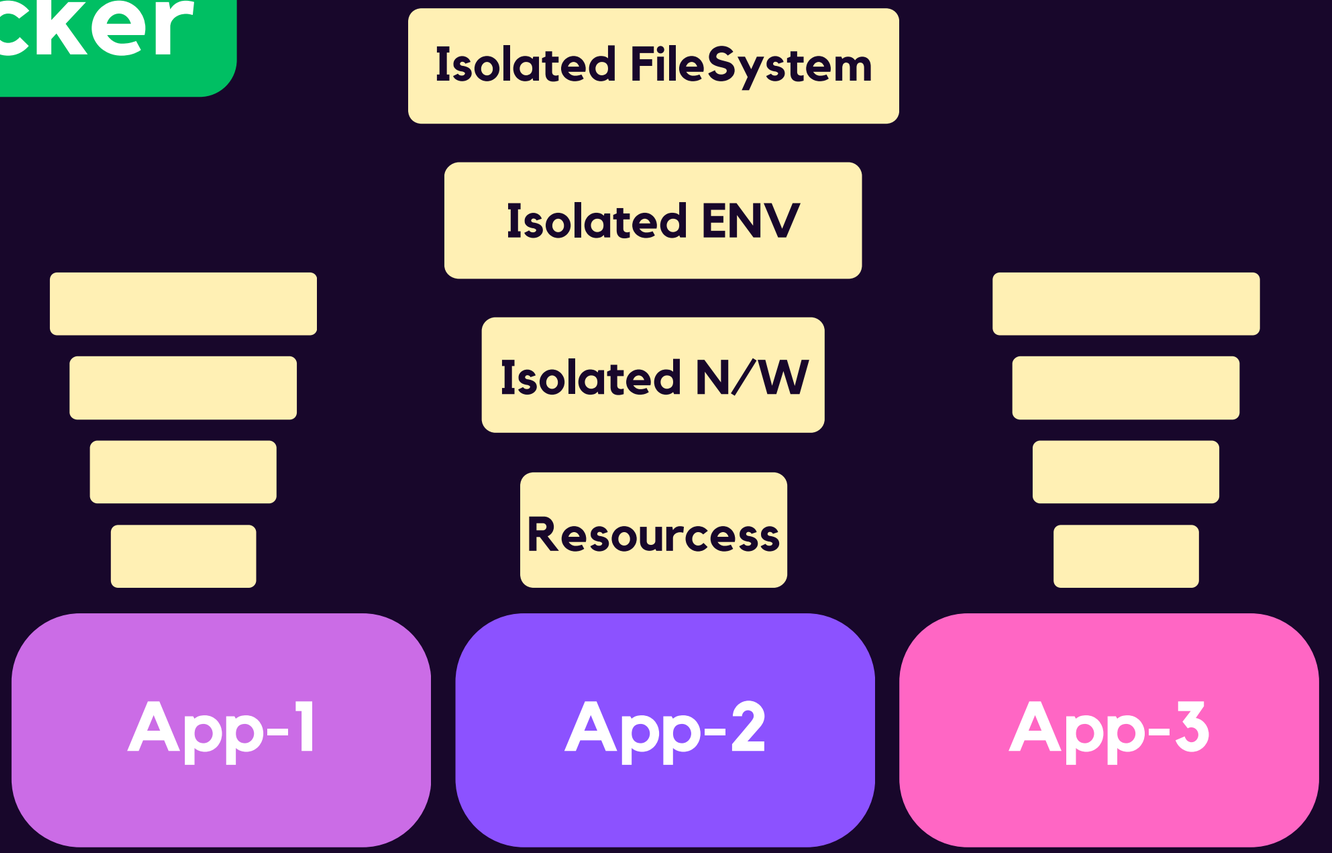
Introduction to Docker



Apps Run as Containers

Think of Containers as isolated pieces of software which appear to have their own OS, FileSystem, Network etc without any interference from other containers.

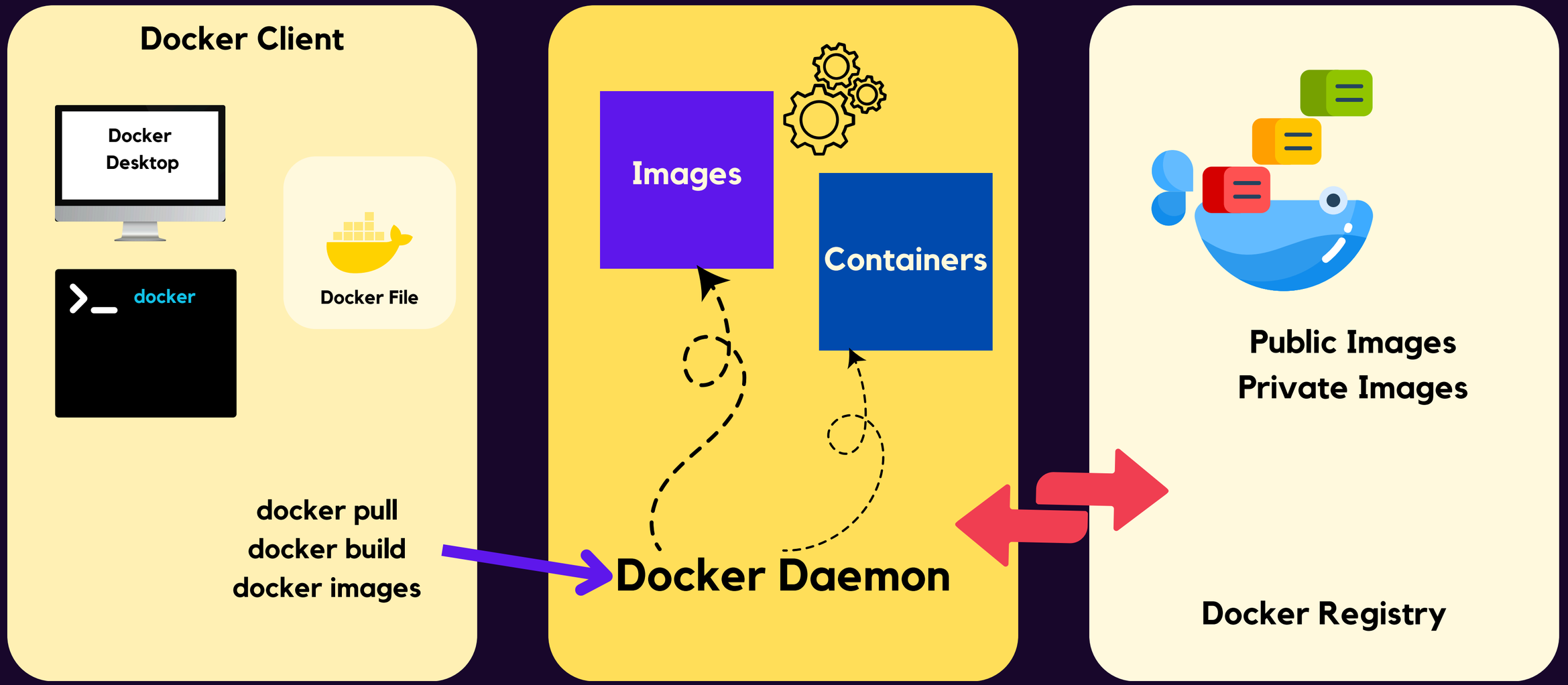
What is Docker



Written in Golang

Docker

Docker Architecture



Installing Docker

Docker Desktop is available for all the major operating systems.

Docker Version on your Machine

```
docker --version
```

Output may be different depending upon your version.

```
Docker version 24.0.6, build ed223bc
```

Docker "Asking for Help"

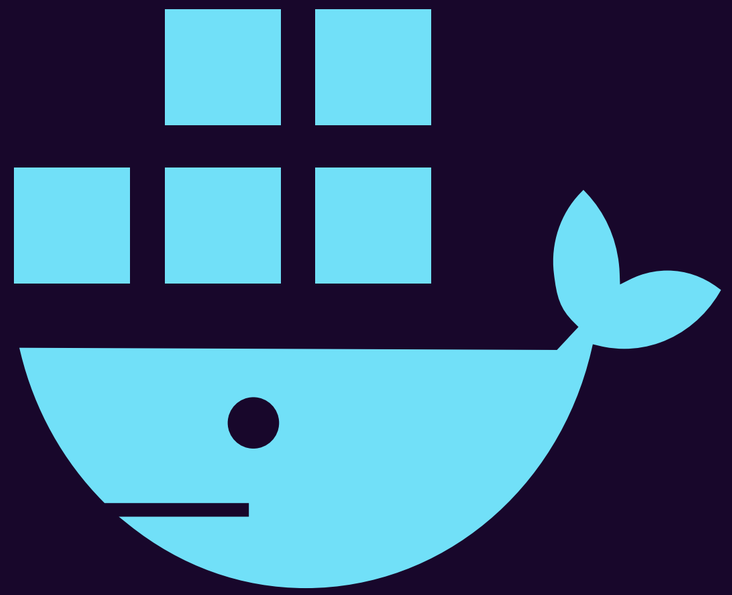
```
docker help
```

```
Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  pull        Download an image from a registry
  push        Upload an image to a registry
```

Output will contain all the commands supported by docker.

Docker Images

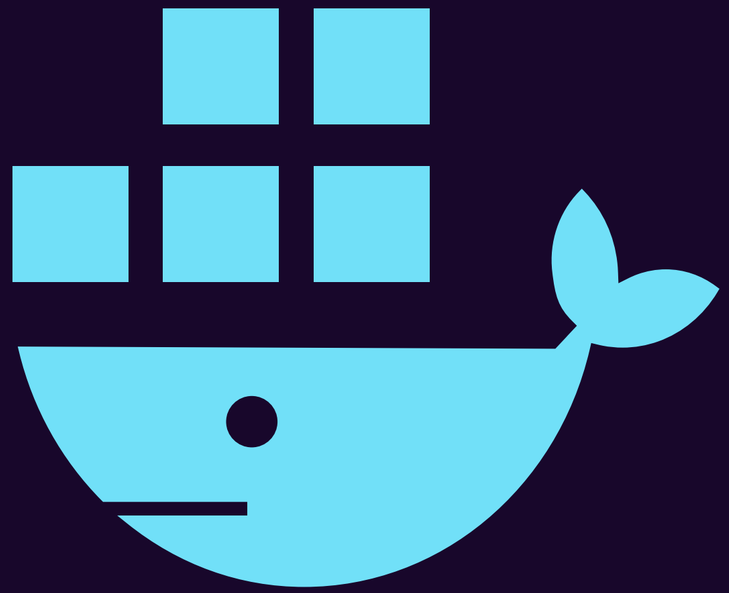
A Docker image is a lightweight, standalone, and executable package that contains everything needed to run a software application



More on Docker Images

Layered Filesystem

Self Contained Package



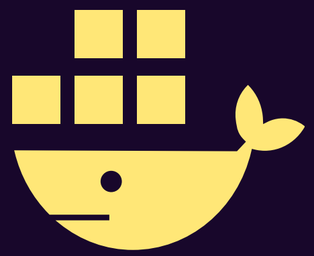
Versioned and Tagged

Immutable and Read-only

Registry

Self-contained Package

A Docker image encapsulates all the components and dependencies required to run a software application.

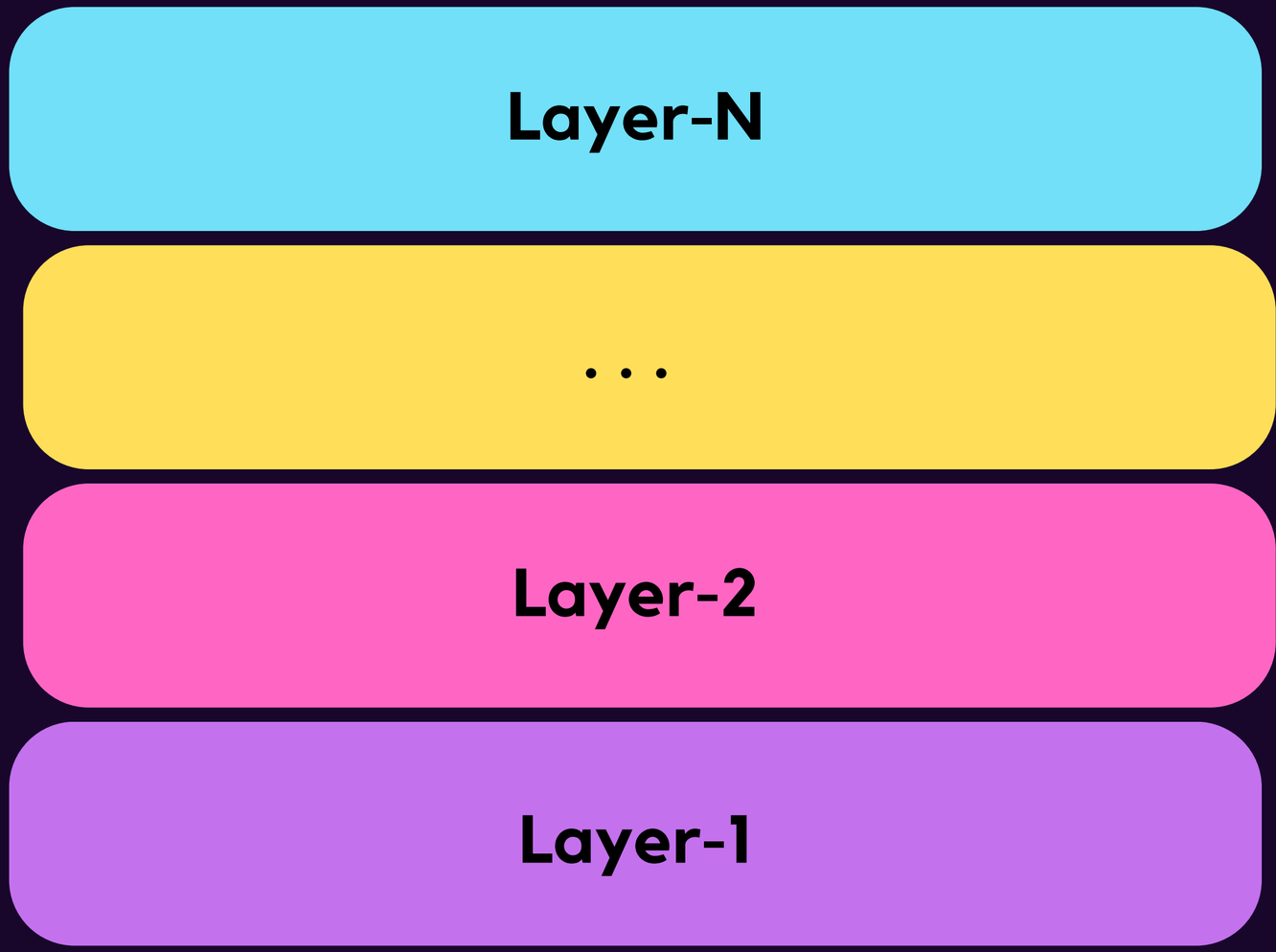


- Configuration Files
- Application Code
- Runtime Environment
- Dependencies
- System Libraries

Immutable and Read-only

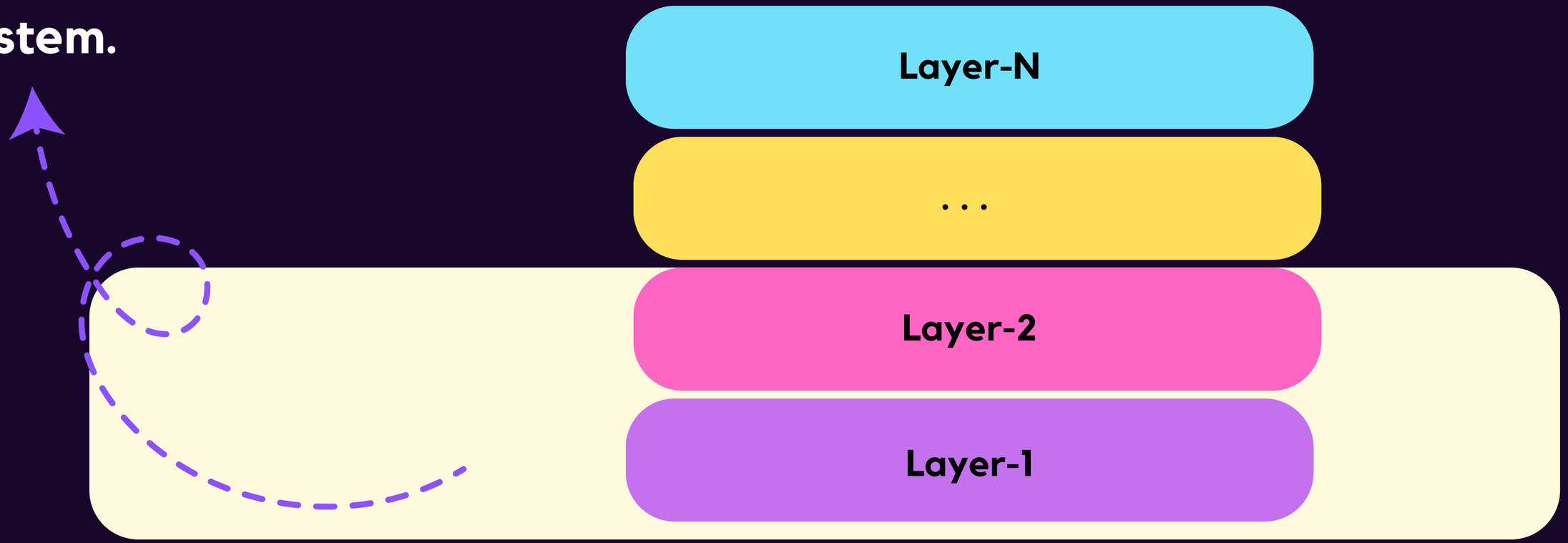
Docker images are composed of multiple layers, with each layer representing a set of filesystem changes.

Layer are immutable and read-only, making Docker images easy to cache, distribute, and share across different environments.



Layered System

Common Layers can be easily reused across images because of a Layered File System.



Versioned and Tagged

Docker images can be versioned and tagged with a unique identifier, such as a version number or a custom label.

This allows users to track and manage different versions of an image and ensures reproducibility and consistency in the deployment process.

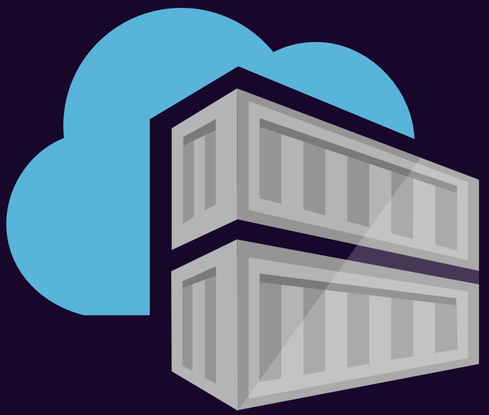
Docker Registry

Docker images can be stored and distributed using Docker registries.

Registries are repositories for storing and sharing Docker images.

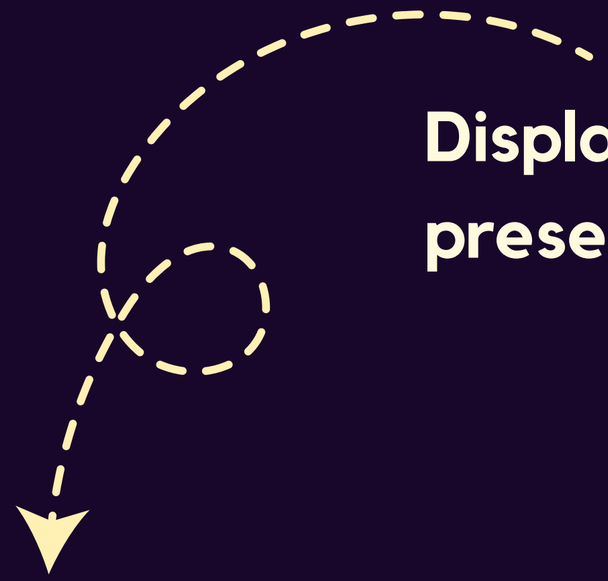
Docker Hub is the official public registry provided by Docker.

private registries can also be set up for internal use within organizations.



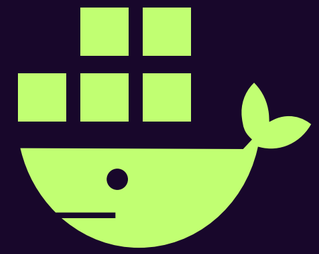
Docker Images Commands

```
docker images
```



Displays all the images present on the System.

```
REPOSITORY      TAG      IMAGE ID      CREATED       SIZE
dpage/pgadmin4  latest   8caab9185d05  3 weeks ago   488MB
postgres        latest   d4ffc32b30ba  2 months ago  453MB
```



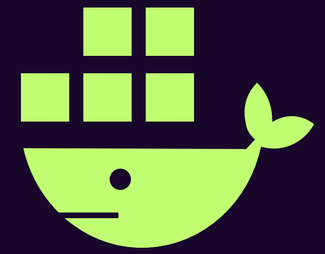
Docker Images Commands

docker pull <image_name>

Downloads a Docker image from a registry (e.g., Docker Hub) to your local machine

docker pull alpine

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dpage/pgadmin4	latest	8caab9185d05	3 weeks ago	488MB
postgres	latest	d4ffc32b30ba	2 months ago	453MB



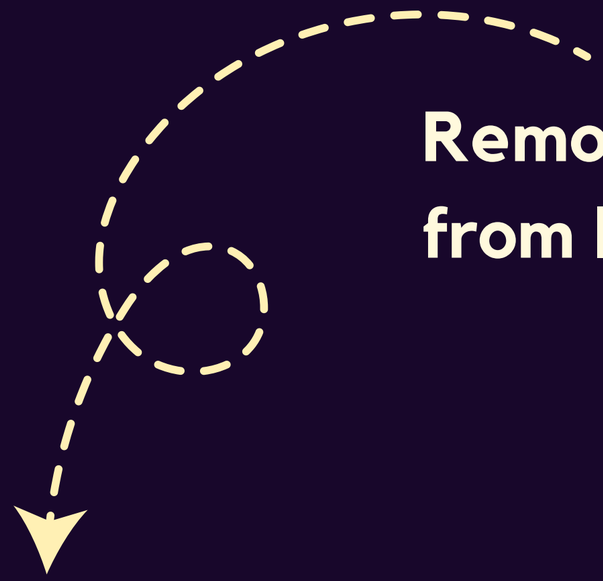
Docker Images Commands

Downloads a Docker image from a registry (e.g., Docker Hub) with specified name and tags

```
docker pull <image_name>:tags
```

Docker Remove Image

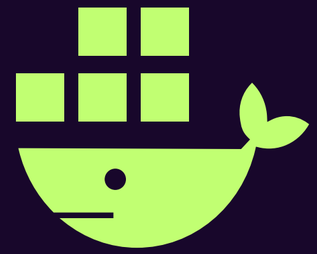
docker rmi <image_name>



Removes a Docker image from local machine.

docker rmi alpine

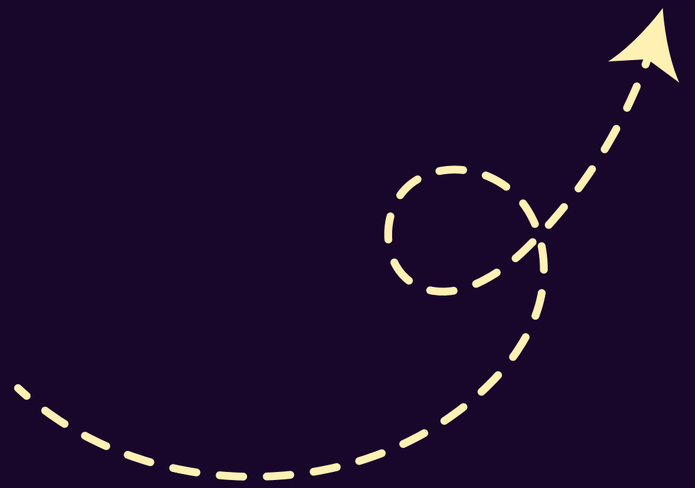
```
docker rmi alpine
Untagged: alpine:latest
Untagged: alpine@sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Deleted: sha256:ace17d5d883e9ea5a21138d0608d60aa2376c68f616c55b0b7e73fba6d8556a3
```



Docker Build Image

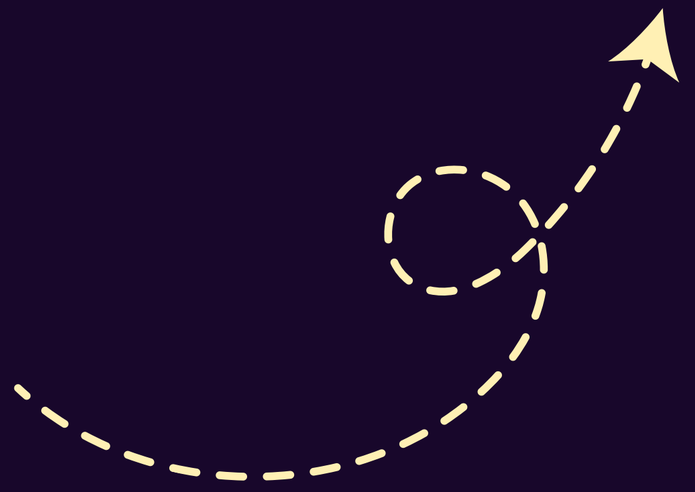
```
docker build -t <image_name> <path_to_Dockerfile>
```

More on Dockerfile in the upcoming slides. . . . !



Docker Tag Image

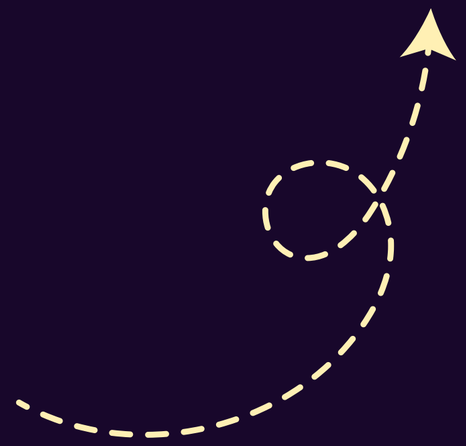
```
docker tag <source_image> <target_image>
```



Adds a tag to an existing Docker image, allowing it to be referenced by an alternate name or version.

Docker Inspect Image

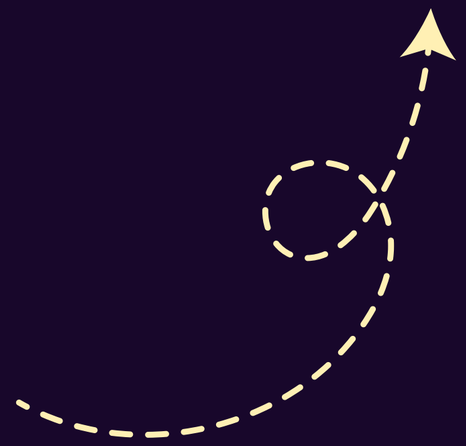
```
docker image inspect <image_name>
```



Displays detailed information about a specific Docker image.

Docker Push Image

docker push <image_name>



Pushes a Docker image from your local machine to a remote registry (e.g. Docker Hub), making it available for others to pull and use

Quiz Time

Raise Your Hand to Answer

Q --- 001

What is a Docker image?

a) A running instance of a Docker container
b) A lightweight, standalone, and executable package that contains everything needed to run an application
c) A virtual machine
d) A Docker registry

Q --- 002

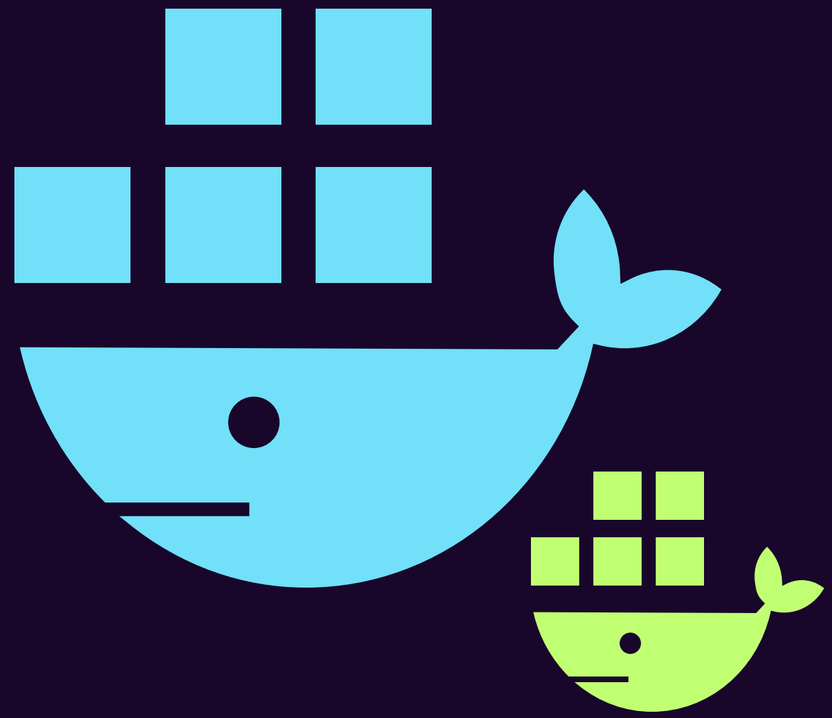
Which command is used to pull a Docker image from a Docker registry?

a) docker create
b) docker build
c) docker run
d) docker pull

Docker Containers

A Docker container is a lightweight, runnable instance of a Docker image.

Container is created when we run the docker image.

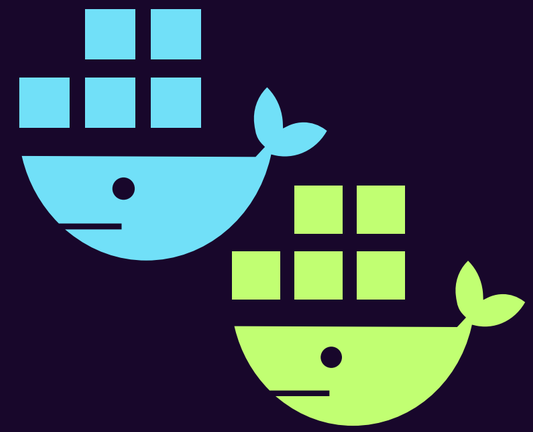


Managing Container Life Cycle

Run a Container:

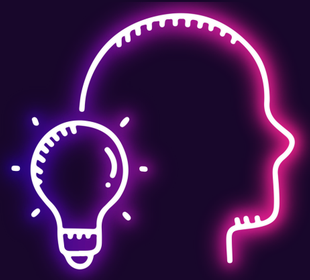
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

e.g.
docker run ubuntu



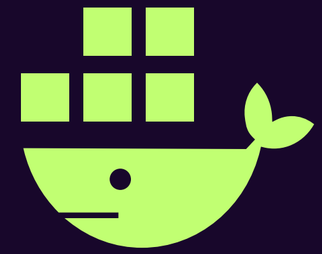
** Note

If you run docker run ubuntu without any additional flags or commands, it will start a container from the Ubuntu image and then immediately exit because there is no ongoing process keeping the container running.



Docker Run Command

```
fourofour@fourofour ~ % docker run -p 10000:8888 quay.io/jupyter/scipy-notebook:2024-03-14
Unable to find image 'quay.io/jupyter/scipy-notebook:2024-03-14' locally
2024-03-14: Pulling from jupyter/scipy-notebook
71dca2167f9f: Pulling fs layer
5513dd08abe9: Pulling fs layer
577fdfe66eda: Pull complete
bfc3e272cc26: Pull complete
2ed046979028: Pull complete
d5fba403e6f9: Pull complete
c55129b7cde6: Waiting
4f4fb700ef54: Download complete
0ed461ac4fe0: Waiting
```



Docker Containers

```
# Managing Containers:
docker run -d --name my_container my_image:latest
docker start my_container
docker stop my_container
docker restart my_container
docker pause my_container
docker unpause my_container
docker rm my_container
docker ps
docker stats my_container
```

Docker Containers

```
# Managing Container Logs:
docker logs my_container
docker log -f my_container

# Inspecting Containers:
docker inspect my_container

# Executing Commands Inside Containers:
docker exec -it my_container bash

# Managing Container Resources:
docker update --memory 512m my_container
```

Docker Containers

```
# Miscellaneous:
docker kill my_container
docker wait my_container
docker rename old_container new_container
docker top my_container
```

Docker Best Practices

More Docker Commands for Reference

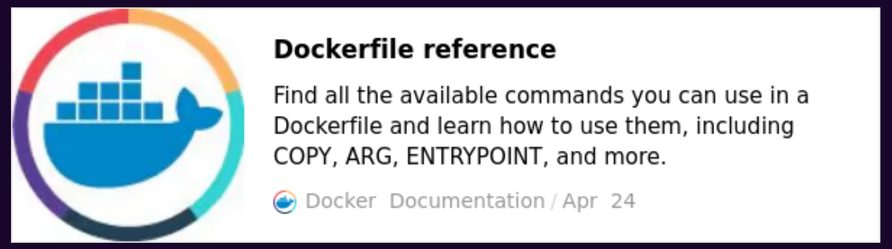
https://docs.docker.com/reference/cli/docker/container/

Task

Task

Explore all of the remaining commands generating at least one example from each of those.

Creating New Docker Images



Reference for all the Dockerfile related commands.

Creating a docker image from file

```
# Use the official Python base image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed dependencies specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Run app.py when the container launches
CMD ["python", "app.py"]
```