Atharva Gawali

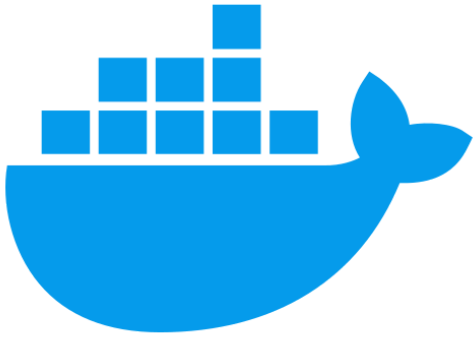
Understanding Docker : Simplifying Application Deployment.

What is Docker?

Docker is an open-source platform for developing, shipping, and running applications in containers. Containers are lightweight, standalone, and executable units that include everything needed to run an application:

✅ Code
✅ Dependencies
✅ Libraries
✅ Configuration

Unlike virtual machines, containers share the host system's kernel, making them faster, more efficient, and portable across various environments.



Key Features of Docker :

1. **Containerization**: Packages applications and their dependencies into lightweight, portable containers.
2. **Portability**: Containers run consistently across different environments (development, testing, production).
3. **Automation**: Simplifies deployment workflows with Dockerfiles and CI/CD pipelines.
4. **Efficiency**: Shares the host OS kernel, making it lightweight and faster than virtual machines.

Core Components of Docker :

1. **Docker Engine:** The runtime that builds and run containers.
2. **Docker Hub:** A registry to find and share container images.
3. **Dockerfile:** A script that defines how to build an image.
4. **Images:** Immutable snapshots used to create containers.
5. **Containers:** Running instances of Docker images.
6. **Docker Swarm:** Native clustering and orchestration tool.

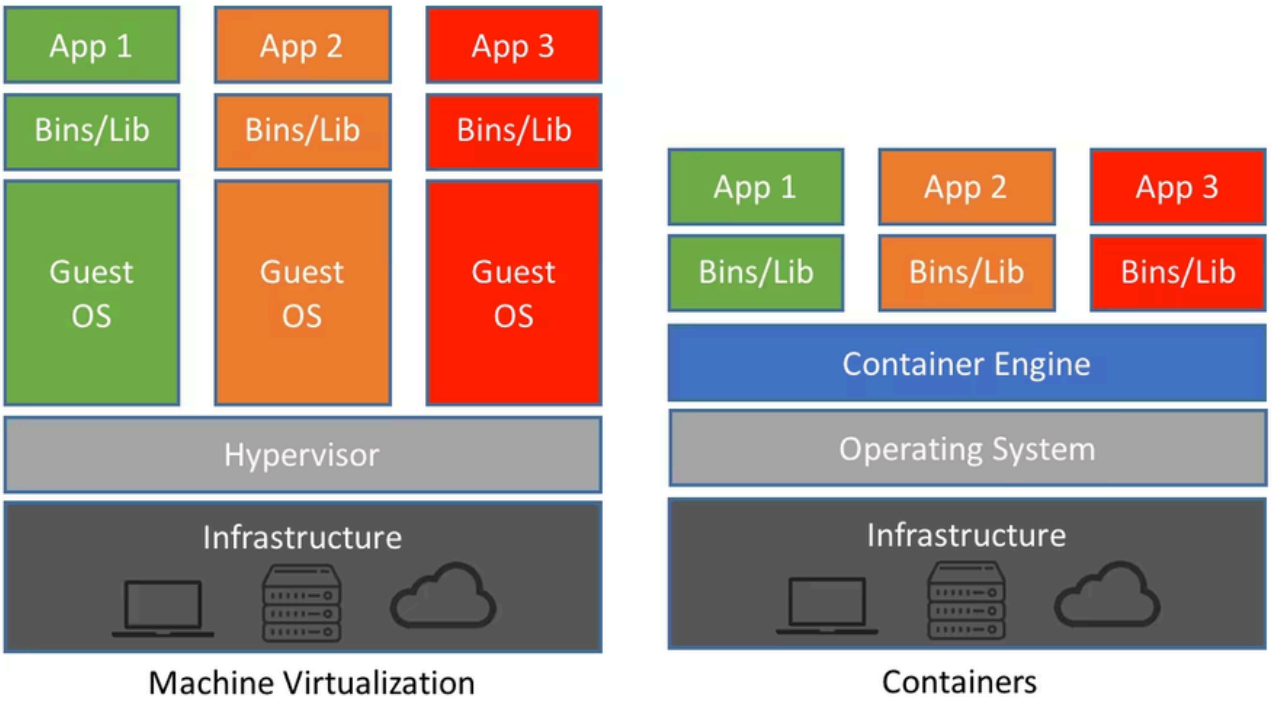
What is Docker Hub?

Docker Hub is a cloud-based registry service provided by Docker, which allows users to store, share, and manage Docker container images. It's essentially a central repository where developers can upload their Docker images for easy access and sharing with others. It hosts both public and private images, enabling developers to find pre-built container images for popular software, or to share their own custom-built images.

What is Container?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

What is Difference Between Container and Virtual Machine?



Virtual machine :
- Each Virtual Machine has its own Guest OS.
-Requires more CPU, RAM, Storage.
-Larger in size (includes OS, Libraries, and app).

Containers :
-Containers share the Host OS.
-Uses fewer resources as there's no Guest OS.
-Lightweight in size (includes only app & dependencies)

What are the Dependencies in Docker?

In Docker, dependencies are the external libraries, tools, or services that an application needs to function properly. These dependencies are often specified in a Dockerfile or through Docker Compose, ensuring that the required components are available when the container is built or run.

Atharva Gawali

Setting up Docker on an Ubuntu server hosted on AWS EC2 using a user data script.

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install -y \
ca-certificates \
curl \
gnupg \
lsb-release
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
sudo systemctl start docker
sudo systemctl enable docker
docker --version
```

Or Else Visit Docker Documentation for Step by Step Installation. 👇
https://docs.docker.com/engine/install/ubuntu/

Atharva Gawali

Some Basic Docker Commands :

1. **Verify the Version of docker installed on your machine.**
   $docker version
2. **Verify Docker Engine Setup and details.**
   $docker info
3. **Download a Docker image from a registry.**
   $docker pull <image name>
4. **Run a Container from Specified image.**
   $docker run <image name>
5. **List all the running Containers.**
   $docker container ls or $docker ps.
6. **List all Containers. (running and stopped)**
   $docker container ls -a or $docker ps -a.
7. **Stop the running Container.**
   $docker container stop <container_id/name>.
8. **Remove Stopped Container.**
   $docker rm <container_id/name>
9. **Check logs of the specific container**
   $docker container logs <container_id/name>
10. **Open Interactive terminal inside running container.**
    $docker exec -it <container_id> /bin/bash.