# DOCKER

Docker is an open-source platform designed to automate the deployment, scaling, and management of applications using containerization. Containers are lightweight, portable, and isolated environments that bundle an application and its dependencies, enabling it to run consistently across different computing environments.

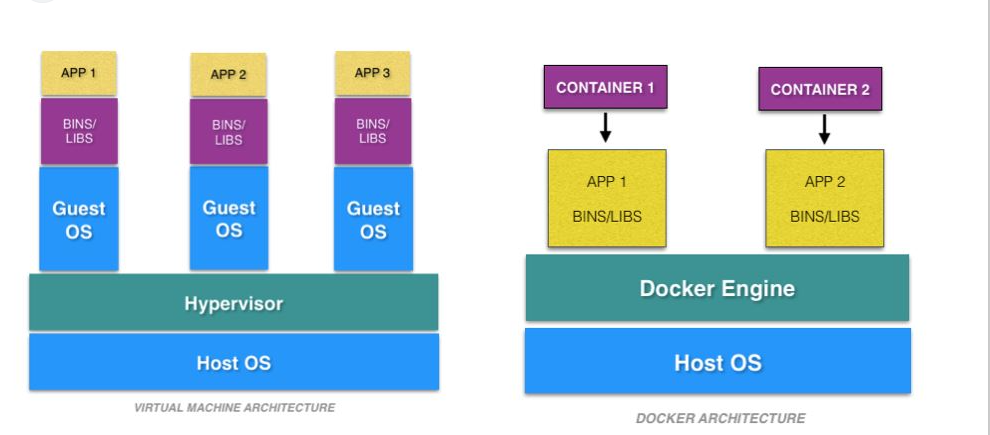
### Key Features of Docker:

1. **Containerization:** Docker allows applications to run in isolated containers.
2. **Lightweight:** Docker containers share the host operating system's kernel, making them faster and more efficient.
3. **Portability:** A containerized application can run on any system with Docker installed, eliminating environment-specific issues.
4. **Image-Based Deployment:** Applications are packaged as Docker images, which are reusable, shareable, and version-controlled.
5. **Scalability:** Docker works well with orchestration tools like Kubernetes, allowing for scalable and efficient deployment of microservices.

**Why Docker Over Traditional Methods?**

Without Docker, developers spend time configuring environments, debugging environment-specific issues, and managing dependencies manually. Docker eliminates these inefficiencies, allowing teams to focus on building and delivering high-quality software faster.

**Docker and Virtual Machine Differences at Architecture Level**



Install Docker on windows using below link:-

[**https://docs.docker.com/desktop/setup/install/windows-install/**](https://docs.docker.com/desktop/setup/install/windows-install/)

Install Docker on Linux using below link:-

[**https://docs.docker.com/desktop/setup/install/linux/**](https://docs.docker.com/desktop/setup/install/linux/)

**Note:-** **Install Docker with recommended setting after finish restart the system.**

SomeDocker commands: -

**1-docker build(**Creates a Docker image from a Dockerfile and related files in the specified context.**)**

🡪docker build -t <image\_name>:<tag> <path>

#docker build -t my-app:1.0 .

**2-docker images(**Lists all Docker images available on your system**)**

#docker images

**3-docker run**(Creates and starts a new container from a specified image)

#docker run -d -p 8080:80 --name my-webserver nginx

**4-docker start**(Restarts an existing, stopped container)

#docker start my-webserver

**5-Displays all currently running containers on the system**.

#docker ps

**6-Displays logs of a running or stopped container.**

#docker logs -f my-webserver

**7- docker stop**(Gracefully stops a running container by sending the SIGTERM signal, allowing processes to terminate cleanly)

docker stop <container\_name\_or\_id>

**8- docker kill**(Forcefully stops a running container by sending the SIGKILL signal, terminating processes immediately)

docker kill <container\_name\_or\_id>

**9- docker rm**(Removes a stopped container from the system, deleting its metadata and resources)

docker rm <container\_name\_or\_id>

**10- Remove Image**(Deletes a specific image from the local system)

docker rmi <image\_name\_or\_id>