Container Vs Virtual Machine:

-> Containers and Virtual machines are two types of virtualization technologies that share many similarities.

-> Virtualization is a process that allows a single resource, such as RAM, CPU, Disk or Networking, to be virtualized and represented as multiple resources.

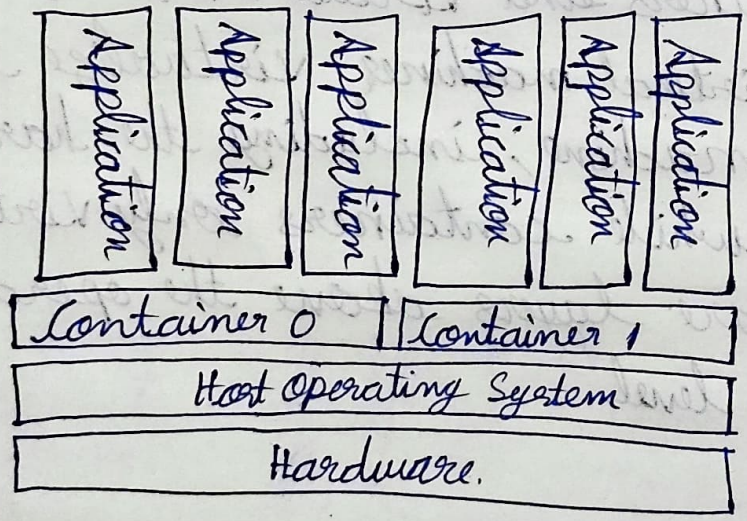
-> However, the main difference between containers and virtual machines is that virtual machines virtualize the entire machine, including the hardware layer, while containers only virtualize software layers above the operating system level.

Containers:-

-> Containers are software packages that are lightweight and self contained, and they comprise all the necessary dependencies to run an application.

-> The dependencies include all external third party packages, system libraries and other operating system-level applications.

-> These dependencies are organized in stack levels that are higher than the operating system.



Advantages

-> It has fast Iteration speed.
-> Can be quickly modified & updated
-> public repositories with pre-made containers (eg:- Docker Hub).

Disadvantages

-> Vulnerabilities in one container may affect other.
-> Public containers may have security risk.

Example:-

-> Docker: Docker is the most widely used container runtime that offers Docker Hub,
A public repository of containerized applications that can be easily deployed to a local Docker runtime.

Virtual Machines

-> Virtualize the entire hardware, including CPU, disk and networking, creating isolated environments with guest OS.

-> Full isolation & security from other VMs.

-> Advantages:- fully system snapshots, manual configuration & isolated environments.

-> Disadvantages:- Slow iteration, large storage footprint.

Example:

-> VirtualBox
-> VMware, QEMU

Docker

→ Docker is a PaaS (Platform as a service) tool uses OS-level virtualization to distribute software as containers.

→ These containers are self-contained, lightweight, operate in isolation from each other and bundle their own software, libraries and configurations.

→ They can communicate with each other through well-defined channels.

→ Unlike virtual machines, all containers share a single operating system kernel, which results in lower resource consumption.

Docker vs Virtual Machine:

→ In Docker the packages, application, dependencies are without a guest OS, and they use the host OS kernel, which makes it lightweight and fast.

→ In Virtual Machine it includes a full OS, making them resource-heavy and slower to boot.

Key Terminologies

1) Docker Image: A read-only template containing application code, libraries and dependencies used to create Docker containers.

2) Docker Container: A lightweight, isolated runtime environment created from a Docker image that runs the application.

3) Dockerfile: A script of instructions that define how to build a Docker image, specifying the base image, app code, and configuration.

4) Docker Engine:- Docker engine is the software that enables the creation and management of Docker containers. It consists of 3 main components:

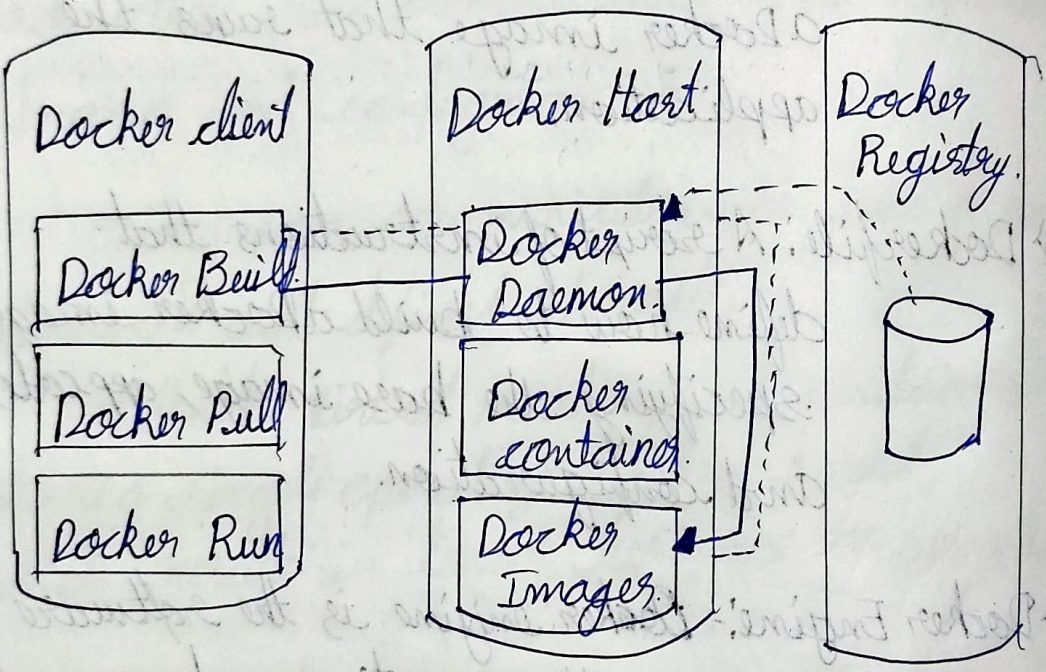
4.1) Docker Daemon:- Server-side component that manages Docker image, containers, networks and storage.

4.2) REST API: Allows external program to communicate with Docker Daemon.

4.3) Docker CLI: A command-line interface for interacting with Docker Engine.

5) Docker Hub:- A cloud-based repository for storing, sharing and managing Docker images, offering both public and private options.

Docker Architecture:-



Docker Components:

→ Docker implements a client-server model where the Docker client communicates with the Docker daemon to create, manage and distribute containers.

1) Docker client:-

The interface for users to interact with Docker via commands like "docker run," "Docker pull", "docker build".

It sends these instructions to the Docker Daemon.

2) Docker Host:-

The machine (physical or virtual) where docker is installed. It runs the Docker daemon, containers, images and networking components.

3) Docker registry:-

A storage system for Docker images. Public registries like Docker Hub or private registries allow users to store, share and pull images.

4) Docker objects:-

Docker fundamental entities - such as images, containers, networks and volumes - that are created and managed in the Docker environment.

5) Docker storage:-

Provides persistent storage options for containers. Includes data volumes and mounts, enabling containers to store and access data outside their lifecycle.

6) Docker Networking:-

Facilitates communication between containers. Types include bridge (default), host, Overlay, None, and Macvlan for container connectivity and isolation.

Advantages of Docker:

→ Fast container creation and deployment.
→ Consistent, portable environment for applications.
→ Improved security and resource efficiency compared to VMs.

Docker Features:

→ Open-Source, lightweight and fast development cycle.
→ Scalable, secure & efficient in managing applications.
→ Enables easy application delivery & deployment.

Docker Hub:

→ A repository for storing, managing & sharing Docker images.

→ Supports public and private images, automation and security checks.

Live VM migration steps and Performance:-

→ Live VM migration allows moving a virtual machine (VM) from one physical machine to another with minimal downtime.

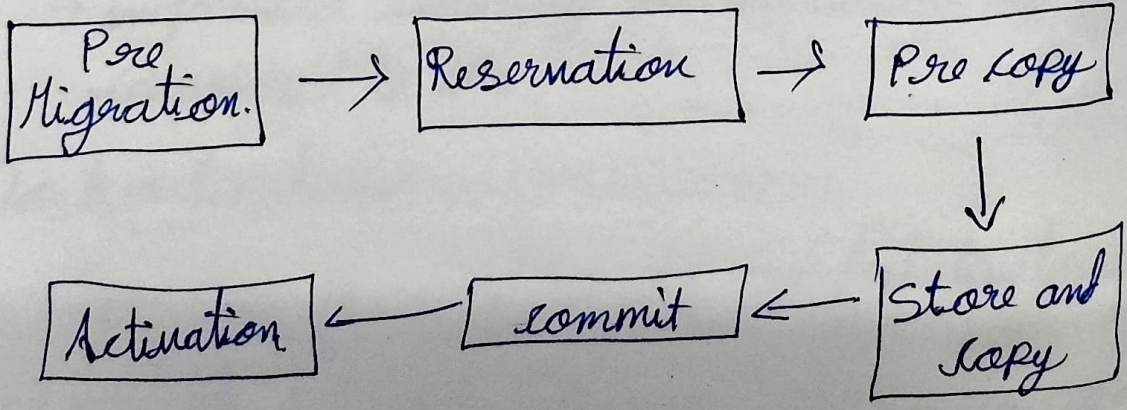
→ It's widely used in cloud platforms, high-performance computing, and computational grids for resource allocation and fault tolerance.

→ In the event of a VM failure, another VM running with the same guest OS can replace it on a different node.

→ During migration, the VM state file is copied from the storage area to the host machine.

→ A VM can be in one of four states, including an inactive state, an active state, a paused state, and a suspended state.

Live Migration steps:-



Migration of Resources:

→ Since clusters have a high initial cost of ownership which includes space, power conditioning and cooling equipment.

→ When one system migrates to another physical node, consider the following issues.

- Memory Migration
- File System Migration
- Network Migration.

Memory Migration:

Transfers the memory state from one physical host to another. Efficiency depends on the guest OS's workload characteristics. Techniques like "internet Suspend-Resume" (ISR) minimize transmission overhead by caching only changed files, but ISR is not ideal for live migration due to high downtime.

File System Migration:

Ensuring a consistent, location-independent view of the file system across hosts. Options include assigning VMs their virtual disks or global file system accessible by all hosts without copying files.

Network Migration:

Maintains network connections during migration by assigning virtual IP and MAC addresses to the VM. The new host maintains communication using these addresses, ensuring remote systems can locate the migrated VM without interruptions.

Dynamic Deployment of Virtual clusters:-

Various virtual cluster systems (eg, cellular Disco, VIOCND) enable dynamic resource allocation and adaptation for enhanced performance in distributed computing environments.

AWS

Amazon web services (AWS) is a cloud computing platform providing on-demand infrastructure and services like computing power, storage and databases. It enables businesses to scale, innovate and pay only for what they use. AWS is widely adopted across industries due to its flexibility, security and cost-efficiency.

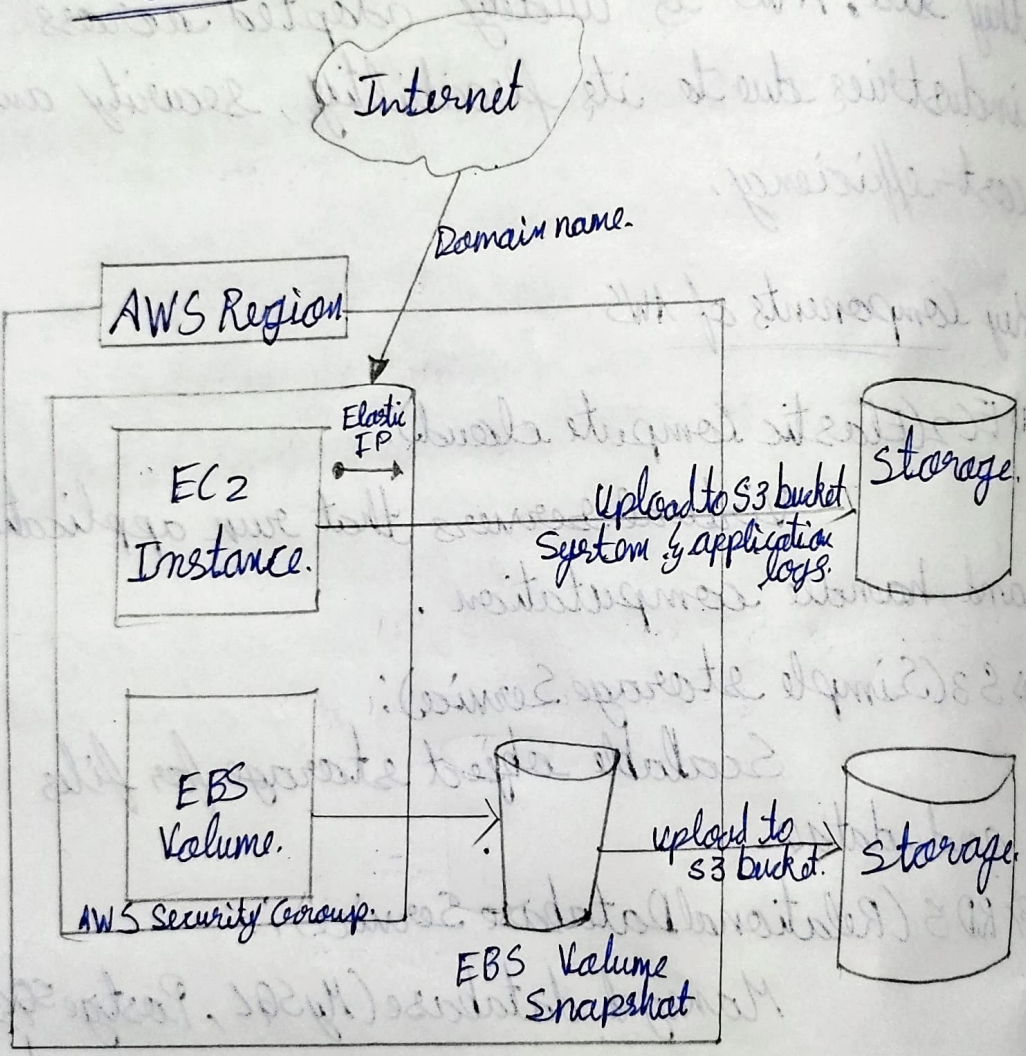
Key Components of AWS

1) EC2 (Elastic Compute cloud):
   Virtual servers that run applications and handle computation.
2) S3 (Simple storage Service):
   Scalable object storage for files and data.
3) RDS (Relational Database Service):
   Managed database (MySQL, PostgreSQL, etc.)
4) Lambda: Serverless compute service that executes code in response to events
5) VPC (Virtual Private cloud): Isolated network environment within AWS for enhanced security.

6) Elastic load Balancing: Automatically distributes traffic to ensure application availability

7) IAM (Identity and access Management): Controls permissions and access to AWS services.

AWS architecture:-



AWS architecture is designed for:-

- Scalability: Services like EC2 and S3 scale automatically with demand.
- High Availability: Multiple data centers (Availability Zones) across regions ensure uptime

and reliability.

- Security: Encryption, VPCs and IAM control access and data protection.
- Fault Tolerance
- Elasticity.

Important of AWS Architecture:-

→ Global Reach: AWS operates in multiple regions, allowing businesses to deploy apps closer to users, reducing latency.

→ Cost Efficiency: Pay-as-you-go pricing allows businesses to avoid upfront infrastructure costs.

→ Innovation: AWS offers cutting-edge services (AI, ML, IoT) to accelerate development and innovation.

→ Security:- Built-in security features with compliance to industry standards.

Uses of AWS:-

→ Web hosting
→ Big Data Analytics
→ Machine learnings.
→ Backup and Disaster Recovery.
→ IoT applications.

Advantages of AWS

-> Scalability: easily scalable based on demand.
-> Cost-effective:
-> Wide Range of Services.
-> Reliability
-> Security

Disadvantages of AWS

-> Complex pricing
-> Learning Curve.
-> Data transfer cost
-> Vendor lock-in
  ↳ ~~Highly~~ heavy reliance on AWS services may make it hard to migrate to another cloud provider.