5.3. CLOUD OPERATING SYSTEM (CLOUD OS)

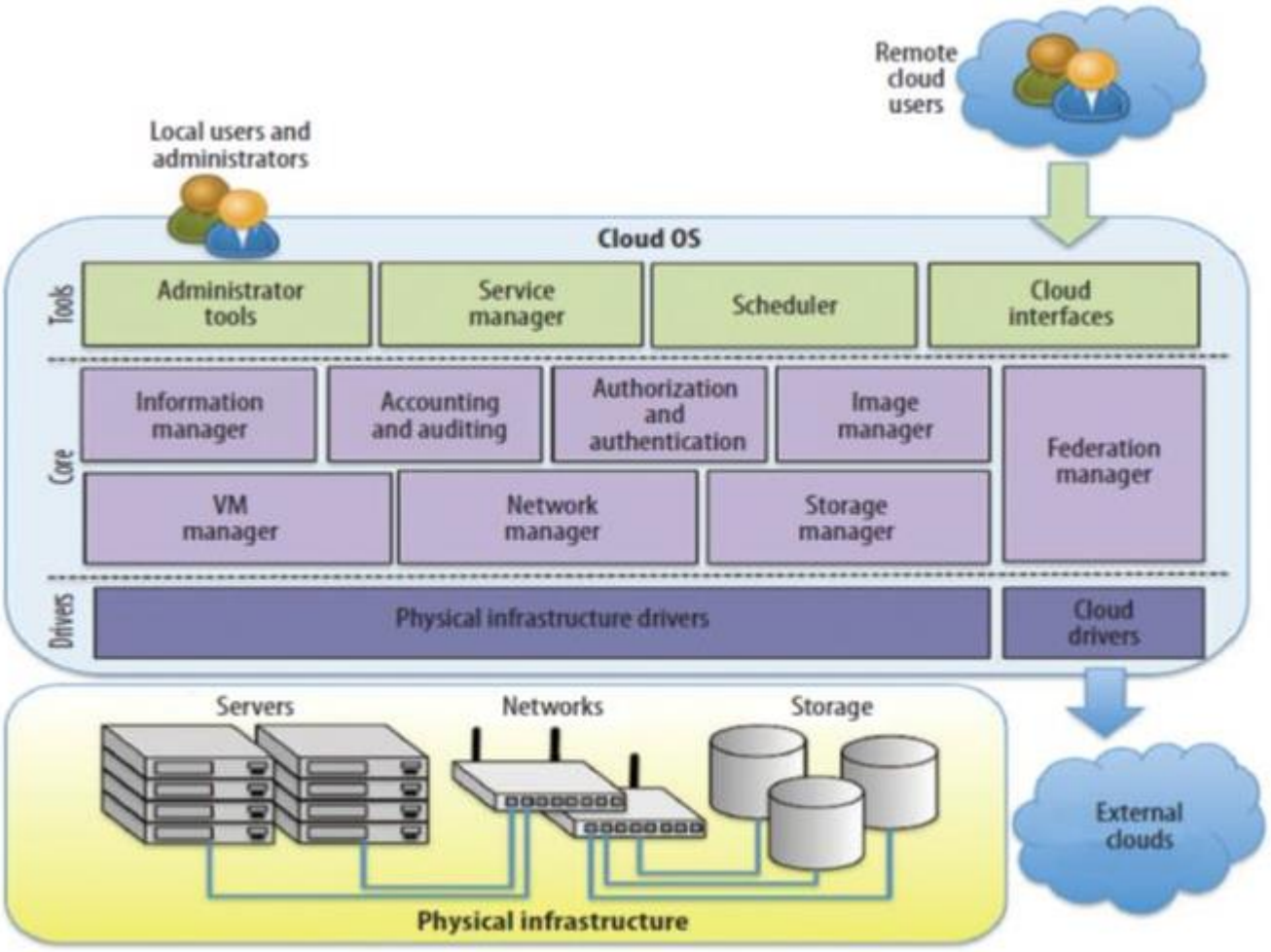
Introduction:

- A **Cloud Operating System (Cloud OS)** is a specialized operating system designed to manage and control cloud computing environments.
- It enables users to access computing resources such as servers, storage, and applications over the internet rather than relying on local systems.
- It acts as an interface between users and cloud infrastructure.

Definition:

A Cloud OS is an operating system that manages virtualized resources in a distributed cloud environment and provides services on-demand through the internet.

Cloud OS Architecture:



Key Aspects of Cloud OS:

1. Virtualization:

- Enables creation of multiple virtual machines on a single physical system.
- Uses hypervisors.
- Provides isolation and flexibility.

2. Resource Management:

- Efficient allocation of CPU, memory, storage, and bandwidth.
- Dynamically adjusts resources based on demand.

3. Scalability:

- Allows expansion or reduction of resources.
- Types:
  - Vertical Scaling (increase system capacity)
  - Horizontal Scaling (add more systems)

4. Multi-tenancy:

- Multiple users share the same infrastructure.
- Ensures data isolation and privacy.

5. Distributed Computing:

- Workloads are distributed across multiple systems.
- Improves speed and performance.

6. Fault Tolerance:

- System continues to operate even if a component fails.
- Uses redundancy and backup mechanisms.

7. Security:

- Provides:
  - Authentication
  - Authorization
  - Encryption
- Protects user data in shared environments.

8. Service Models Support:

Cloud OS supports different cloud services:

- **IaaS** – Infrastructure as a Service
- **PaaS** – Platform as a Service
- **SaaS** – Software as a Service

9. Automation and Orchestration:

- Automates deployment and management.
- Uses orchestration tools for efficient operation.

10. Network Management:

- Handles virtual networking, routing, and IP allocation.
- Ensures communication between cloud resources.

Advantages of Cloud OS:

- High scalability
- Cost-effective (pay-as-you-go)
- Remote accessibility
- High availability and reliability
- Efficient resource utilization

Disadvantages of Cloud OS:

- Security concerns
- Internet dependency
- Latency issues
- Limited control over infrastructure

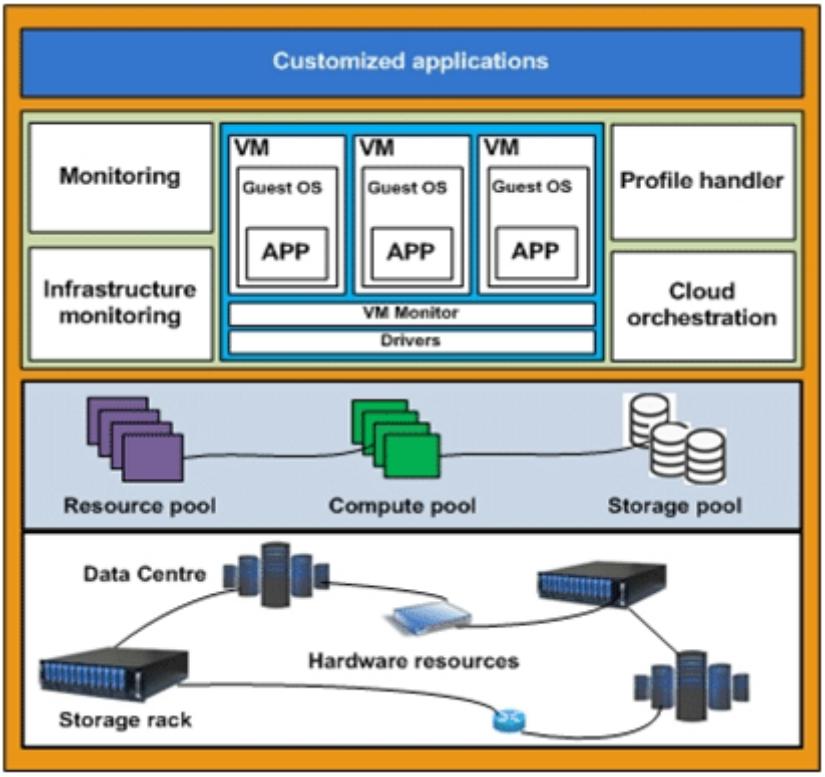
Examples of Cloud OS:

- Microsoft Azure
- Amazon Web Services
- Google Cloud Platform

5.3.1. RESOURCE ORCHESTRATION

Introduction:

- **Resource Orchestration** is a key aspect of a Cloud OS that involves the **automated arrangement, coordination, and management of computing resources** such as CPU, memory, storage, and network to efficiently run applications and services in the cloud.



Definition:

Resource orchestration is the process of **automatically managing, allocating,** and **coordinating cloud resources** to **ensure optimal performance, scalability, and efficient utilization**.

Key Functions of Resource Orchestration:

1. Resource Allocation:

- Assigns required resources to applications dynamically.
- Prevents underutilization and overloading.

2. Scheduling:

- Determines when and where tasks should run.
- Ensures efficient execution of workloads.

3. Load Balancing:

- Distributes workload across multiple systems.
- Avoids system bottlenecks and improves performance.

4. Auto Scaling:

- Automatically increases or decreases resources based on demand.
- Maintains system performance during peak load.

5. Monitoring:

- Continuously tracks system performance and resource usage.
- Helps in decision-making for scaling and allocation.

6. Provisioning:

- Automatically creates and configures resources (VMs, containers).
- Reduces manual intervention.

7. Fault Management:

- Detects failures and recovers resources.
- Ensures high availability.

Types of Resource Orchestration:

1. Static Orchestration:

- Resources are pre-allocated.
- Less flexible.

2. Dynamic Orchestration:

- Resources are allocated in real-time based on demand.
- Highly flexible and efficient.

Tools for Resource Orchestration:

- Kubernetes – Manages containerized applications.
- Docker Swarm – Simple container orchestration.
- Apache Mesos – Distributed resource management.

Advantages:

- Efficient resource utilization.
- Reduced operational cost.
- Improved scalability.
- Faster deployment.
- High system reliability.

Challenges:

- Complexity in management.
- Security risks.
- Dependency on automation tools.
- Requires proper monitoring.

Example:

In a cloud-based e-commerce application:

- During high traffic, orchestration automatically:
  - Adds more servers.
  - Distributes load.
  - Ensures smooth user experience.

5.3.2. VM MIGRATION

Introduction:

- **Virtual Machine (VM) Migration** is a process in cloud computing where a running or stopped virtual machine is moved from one physical host to another without affecting its functionality.
- It is widely used in Cloud OS for load balancing, maintenance, and fault tolerance.

Definition:

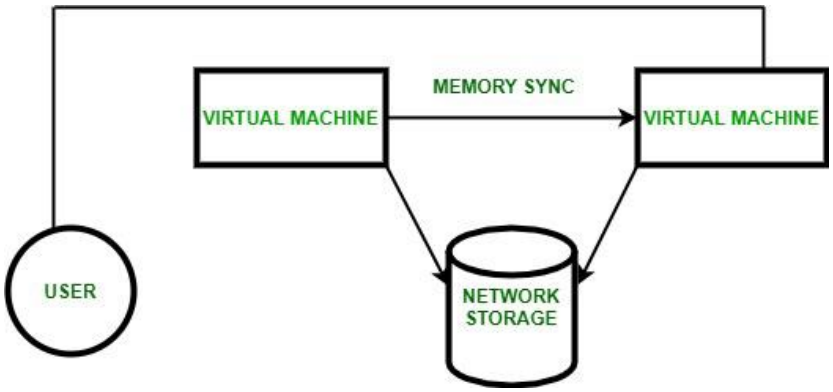
VM Migration is the process of transferring a virtual machine from one physical server to another, either with or without downtime, while preserving its state and execution.

Types of VM Migration:

1. Live Migration (Hot Migration):

- VM is migrated while it is running.
- No or minimal downtime.

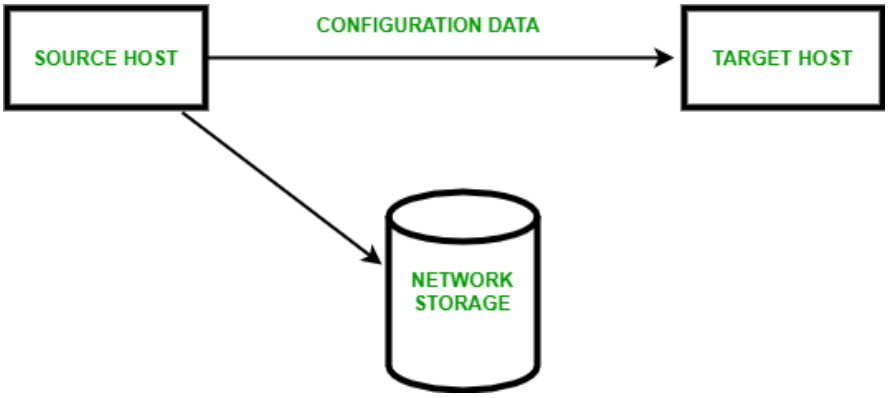
- Memory state is transferred continuously.



☞ Used in real-time applications.

2. Cold Migration (Offline Migration):

- VM is powered off before migration.
- Entire state is moved.
- Causes downtime.



3. Warm Migration:

- Hybrid approach.
- VM is paused temporarily during migration.

Steps in VM Migration:

1. **Pre-Migration:**
   - Select source and destination hosts.
   - Check resource availability.
2. **Reservation:**
   - Allocate resources in destination.
3. **Memory Transfer:**
   - Transfer VM memory pages to destination.
4. **Iteration Phase:**
   - Continuously copy modified pages.
5. **Stop and Copy:**
   - Pause VM and transfer remaining data.

6. **Activation:**
    - Resume VM at destination host.

Techniques of VM Migration:

1. Pre-Copy Migration:

- Memory pages copied before stopping VM.
- Most commonly used.

2. Post-Copy Migration:

- VM is moved first, then memory is fetched.
- Faster but riskier.

Advantages:

- Load balancing across servers.
- Fault tolerance and disaster recovery.
- Hardware maintenance without downtime.
- Energy efficiency (server consolidation).

Disadvantages:

- Network overhead.
- Migration delay.
- Security risks.
- Performance degradation during migration.

Applications:

- Data centers.
- Cloud platforms like:
    - VMware vMotion
    - Microsoft Hyper-V
    - KVM

5.3.3. AUTOSCALING

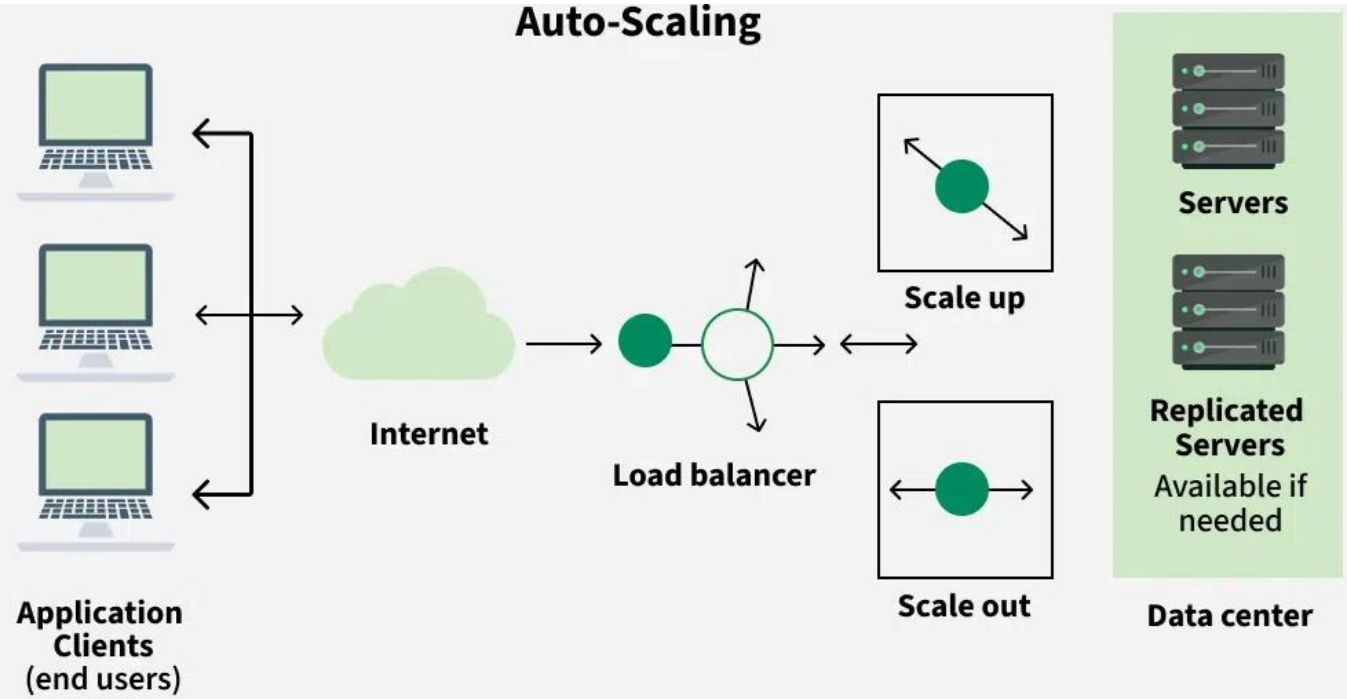
Introduction:

- **Auto Scaling** is a key feature of a Cloud OS that automatically adjusts computing resources based on workload demand.
- It ensures that applications run efficiently without manual intervention by increasing or decreasing resources dynamically.

Definition:

Auto Scaling is the process of automatically adding or removing computing resources (VMs, containers, storage) based on real-time demand to maintain performance and cost efficiency.

Diagram of Auto Scaling:



Types of Auto Scaling:

1. Vertical Scaling (Scale Up/Down):

- Increases or decreases resources (CPU, RAM) of a single machine.
- Simple but has hardware limits.

2. Horizontal Scaling (Scale Out/In):

- Adds or removes multiple machines.
- Highly flexible and widely used in cloud environments.

3. Reactive Scaling:

- Triggered based on real-time metrics (CPU usage, traffic).

4. Predictive Scaling:

- Uses historical data to predict future demand.

Components of Auto Scaling:

1. Monitoring System:

- Tracks metrics like CPU usage, memory, and network traffic.

2. Scaling Policies:

- Defines rules (e.g., CPU > 70% → add instance).

3. Load Balancer:

- Distributes traffic across available resources.

4. Scaling Engine:

- Executes scaling decisions automatically.

Working of Auto Scaling:

1. Monitor system performance continuously.
2. Compare metrics with predefined thresholds.
3. Trigger scaling policy.
4. Add or remove resources.
5. Load balancer redistributes traffic.

Advantages:

- High availability.
- Cost efficiency (pay only for used resources).
- Improved performance.
- Automatic resource management.
- Handles traffic spikes effectively.

Disadvantages:

- Configuration complexity.
- Delay in scaling (latency).
- Risk of over-scaling or under-scaling.
- Dependency on accurate metrics.

Real-Time Examples:

- Amazon EC2 Auto Scaling.
- Microsoft Azure Autoscale.
- Google Compute Engine Autoscaler.

Applications:

- E-commerce websites (during sales).
- Streaming platforms.
- Online gaming systems.
- Banking and financial services.