

## UNIT-II

### What is Server Virtualization?

Server virtualization is a technology that allows a single physical server to be divided into multiple isolated virtual servers (also called virtual machines or VMs). Each VM operates independently with its own operating system (OS), applications, and allocated resources (CPU, memory, storage, networking), while sharing the underlying physical hardware. This abstraction is managed by a software layer called a hypervisor .

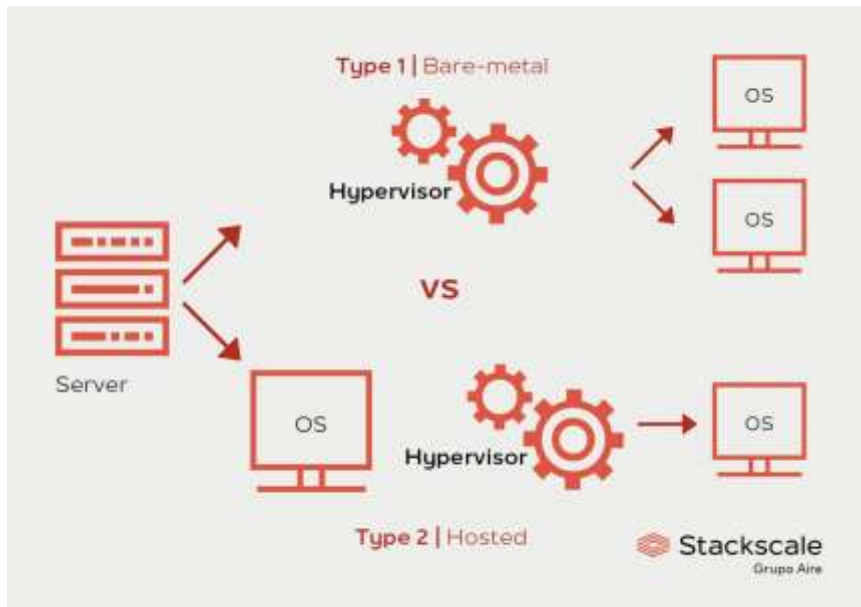
The core idea dates back to mainframes in the 1960s-1970s (e.g., IBM's CP/CMS), but it became widespread in x86 servers in the 2000s due to rising hardware power, energy costs, and underutilized servers (often running at 10-20% capacity).

#### Key Concepts

- Virtual Machine (VM) : A software-based emulation of a physical computer. It includes a guest OS and runs as if it were on dedicated hardware.
- Hypervisor : The software that creates, runs, and manages VMs. It allocates physical resources dynamically and ensures isolation between VMs.
- Host : The physical server running the hypervisor and VMs.
- Guest : The OS and applications inside each VM.

#### Types of Hypervisors

Hypervisors are classified into two main types



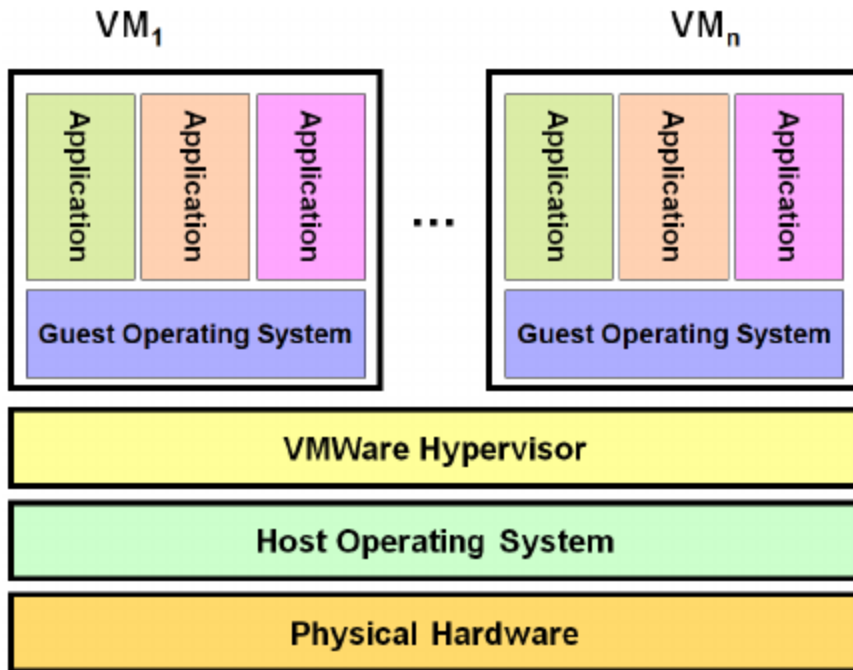
\*Type 1 (Bare-Metal) : Runs directly on hardware for better performance and security. Ideal for production/data centers.

- Type 2 (Hosted) : Runs on top of a host OS. Easier for testing/desktops but with more overhead.

#### #### Types of Server Virtualization

1. Full Virtualization : Hypervisor fully emulates hardware; guest OS runs unmodified (no awareness of virtualization).
2. Paravirtualization : Guest OS is modified for better performance by directly communicating with hypervisor (e.g., via special drivers).
3. OS-Level Virtualization (Containerization) : No hypervisor; multiple isolated instances share the host OS kernel. Lightweight (e.g., Docker, Kubernetes). Often distinguished from traditional VM-based virtualization.

Popular Technologies and Hypervisors (as of 2025)



- VMware vSphere/ESXi (Broadcom): Dominant enterprise Type 1; rich features, but licensing costs.
- Microsoft Hyper-V : Built into Windows Server (enhanced in 2025 with more memory/logical processors, better security).
- KVM (Kernel-based Virtual Machine) : Open-source Type 1 in Linux; basis for Proxmox, Red Hat OpenShift Virtualization.
- Proxmox VE : Open-source platform based on KVM; popular for home labs/SMBs.
- Xen/XCP-ng : Open-source Type 1; used in clouds like Oracle.
- Others: Nutanix AHV, Citrix Hypervisor.

Trends in 2025 include hybrid cloud integration, AI/ML workload support (e.g., GPU passthrough), edge computing, and combining VMs with containers (e.g., via KubeVirt).

### Benefits

- Resource Efficiency : Servers utilize 70-80%+ capacity vs. 10-20% on physical.

- Cost Savings : Fewer servers, less power/cooling/space.
- Flexibility : Easy provisioning, migration (live migration), snapshots, disaster recovery.
- Isolation & Security : VMs are sandboxed.
- Scalability : Foundation for cloud computing.

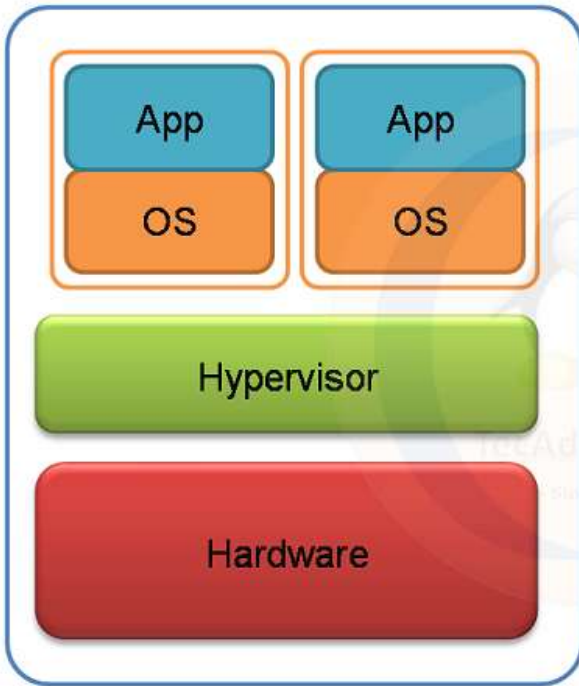
### Challenges

- Performance overhead (minimal with modern hardware-assisted virtualization like Intel VT-x/AMD-V).
- Single point of failure (hypervisor/host).
- Licensing complexity.
- Management overhead for large deployments.

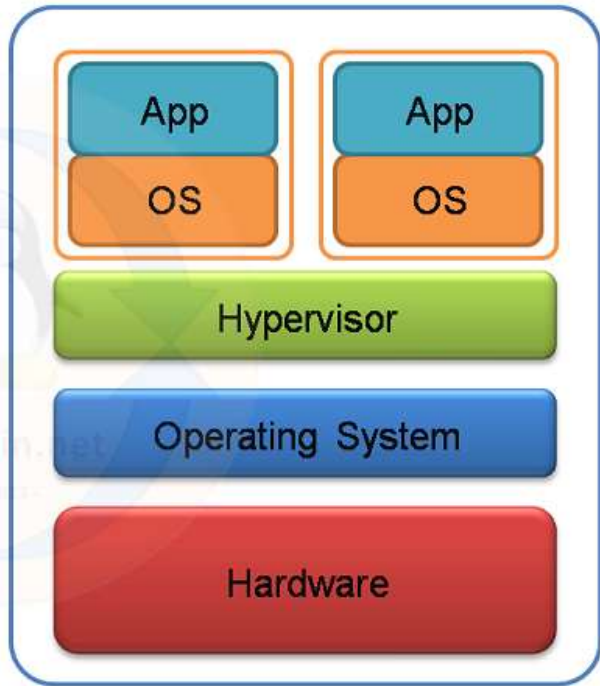
Server virtualization remains foundational to modern IT, enabling cloud, HCI (hyper-converged infrastructure), and efficient data centers.

### Hypervisors: Type 1 (Bare-Metal) vs. Type 2 (Hosted)

Hypervisors are the core software layer in server virtualization that abstracts physical hardware to create and manage virtual machines (VMs).



Type 1 (Bare Metal)  
Virtualization



Type 2 (Hosted)  
Virtualization

Criteria	Type 1 hypervisor	Type 2 hypervisor
AKA	Bare-metal or Native	Hosted
Definition	Runs directly on the system with VMs running on them	Runs on a conventional Operating System
Virtualization	Hardware Virtualization	OS Virtualization
Operation	Guest OS and applications run on the hypervisor	Runs as an application on the host OS
Scalability	Better Scalability	Not so much, because of its reliance on the underlying OS.
Setup/Installation	Simple, as long as you have the necessary hardware support	Lot simpler setup, as you already have an Operating System.
System Independence	Has direct access to hardware along with virtual machines it hosts	Are not allowed to directly access the host hardware and its resources
Speed	Faster	Slower because of the system's dependency
Performance	Higher-performance as there's no middle layer	Comparatively has reduced performance rate as it runs with extra overhead
Security	More Secure	Less Secure, as any problem in the base operating system affects the entire system including the protected Hypervisor
Examples	<ul style="list-style-type: none"> <li>• VMware ESXi</li> <li>• Microsoft Hyper-V</li> <li>• Citrix XenServer</li> </ul>	<ul style="list-style-type: none"> <li>• VMware Workstation Player</li> <li>• Microsoft Virtual PC</li> <li>• Sun's VirtualBox</li> </ul>

**Type 1 (Bare-Metal)** : Runs directly on the physical hardware with no underlying host OS. It has direct access to CPU, memory, and I/O, resulting in higher performance, better security (smaller attack surface), and lower latency. Commonly used in enterprise data centers and production environments.

- Examples: VMware ESXi, Microsoft Hyper-V (in standalone mode), KVM, Xen, Proxmox.

- Pros: Efficient resource utilization, strong isolation.

- Cons: Requires dedicated hardware; management often via remote tools.

**Type 2 (Hosted)** : Runs as an application on top of a conventional host operating system (e.g., Windows, Linux, macOS). The host OS manages hardware, and the hypervisor translates VM requests through it.

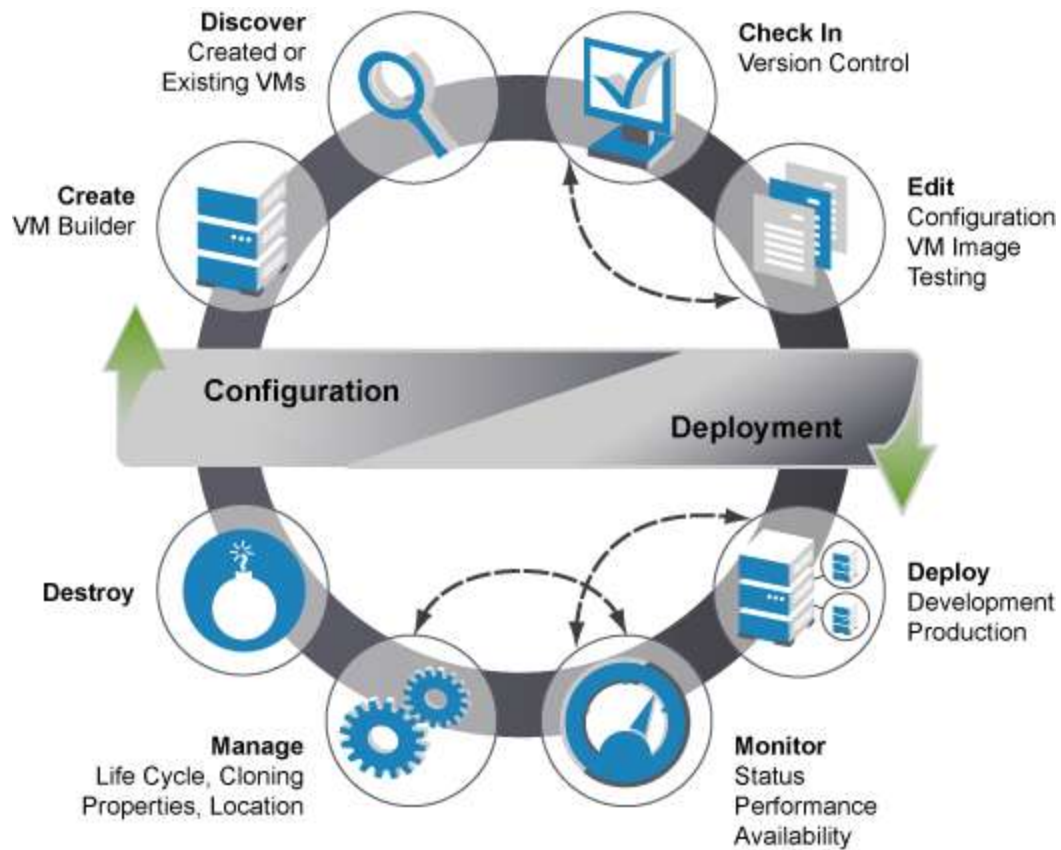
- Examples: VMware Workstation, Oracle VirtualBox, Parallels Desktop.

- Pros: Easier setup, better hardware compatibility (uses host drivers), suitable for desktops, development, and testing.
- Cons: Performance overhead (due to extra OS layer), potential security risks from host OS vulnerabilities.

In practice, Type 1 dominates enterprise server virtualization, while Type 2 is popular for personal or lab use.

### ### Virtual Machine Lifecycle Management

The VM lifecycle encompasses all stages from creation to decommissioning. Management tools (e.g., vCenter for VMware, Hyper-V Manager, or open-source like oVirt/Proxmox) automate these processes for efficiency, scalability, and high availability.



Key stages:

1. **Creation/Provisioning** : Define VM specs (CPU, RAM, storage, network). Use templates or clones for rapid deployment. Options include manual creation, automated scripting (e.g., Terraform), or orchestration tools.
2. **Startup/Powering On** : Boot the guest OS. May include power-on scripts or integration with configuration management (e.g., Ansible).

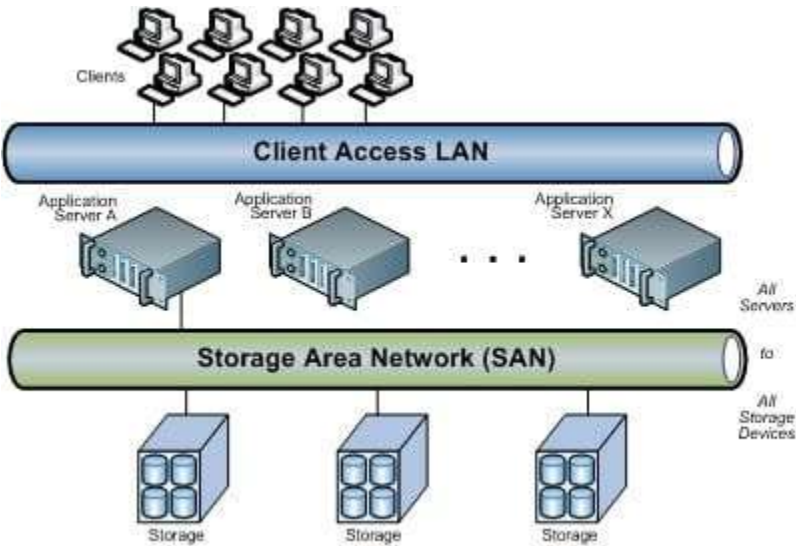
3. Operation/Running : Monitor performance, apply updates, scale resources (hot-add CPU/RAM in supported hypervisors), and handle workloads.
4. Migration : Move VMs between hosts without downtime (live migration, e.g., vMotion in VMware) or with brief pause (cold migration). Useful for load balancing, maintenance, or disaster recovery.
5. Suspension/Snapshot : Pause VM state (suspend) or capture point-in-time copies (snapshots) for backups/testing.
6. Shutdown/Restart : Graceful power-off or reboot.
7. Decommissioning/Deletion : Remove VM and reclaim resources when no longer needed.

Advanced features include high availability (automatic restart on host failure), fault tolerance (real-time mirroring), and integration with container orchestration (e.g., via KubeVirt).

### ### Storage Virtualization: SAN, NAS, and Object Storage

Storage virtualization abstracts physical storage to provide flexible, shared pools for VMs and applications. In virtualized environments, it enables features like thin provisioning, snapshots, and live migration.

Aspect	SAN (Storage Area Network)	NAS (Network Attached Storage)	Object Storage
Access Level	Block-level (raw storage blocks, appears as local disk)	File-level (shared files/folders via protocols)	Object-level (flat namespace with metadata)
Protocols	Fibre Channel (FC), iSCSI, FCoE	NFS, SMB/CIFS, AFP	HTTP/S3 API, RESTful
Architecture	Dedicated high-speed network (often fibre optic)	Ethernet-based appliance	Distributed, scalable (often cloud-native)



## Virtual Storage Pools and Thin Provisioning

Virtual storage pools aggregate physical storage resources (from local disks, SAN, NAS, or HCI) into a logical pool that can be dynamically allocated to VMs. This abstraction enables features like snapshots, cloning, and migration in virtualized environments.

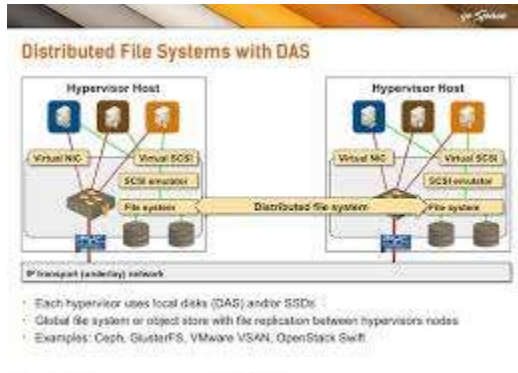
**Thin provisioning** is a key technique within storage pools that allocates storage on-demand rather than upfront.

- **How it works:** A virtual disk appears to have its full provisioned size (e.g., 100 GB), but physical space is consumed only as data is written. Unused space remains available in the pool for other VMs (overcommitment).
- **Vs. Thick/Fixed Provisioning:** Thick allocates all space immediately (better initial performance, no overcommitment risk).
- **Benefits:** Higher utilization (up to 3x better), cost savings (defer hardware purchases), easier provisioning.
- **Risks/Challenges:** Overcommitment can lead to out-of-space conditions if not monitored; potential performance overhead from metadata management and fragmentation.
- **Best Practices:** Monitor usage thresholds, use alerts/auto-extension (e.g., in LVM thin pools or VMware), separate high-I/O workloads.
- **Examples:** VMware vSAN/vSphere datastores, Windows Storage Spaces, DataCore SANsymphony, Azure Local (as of 2025).

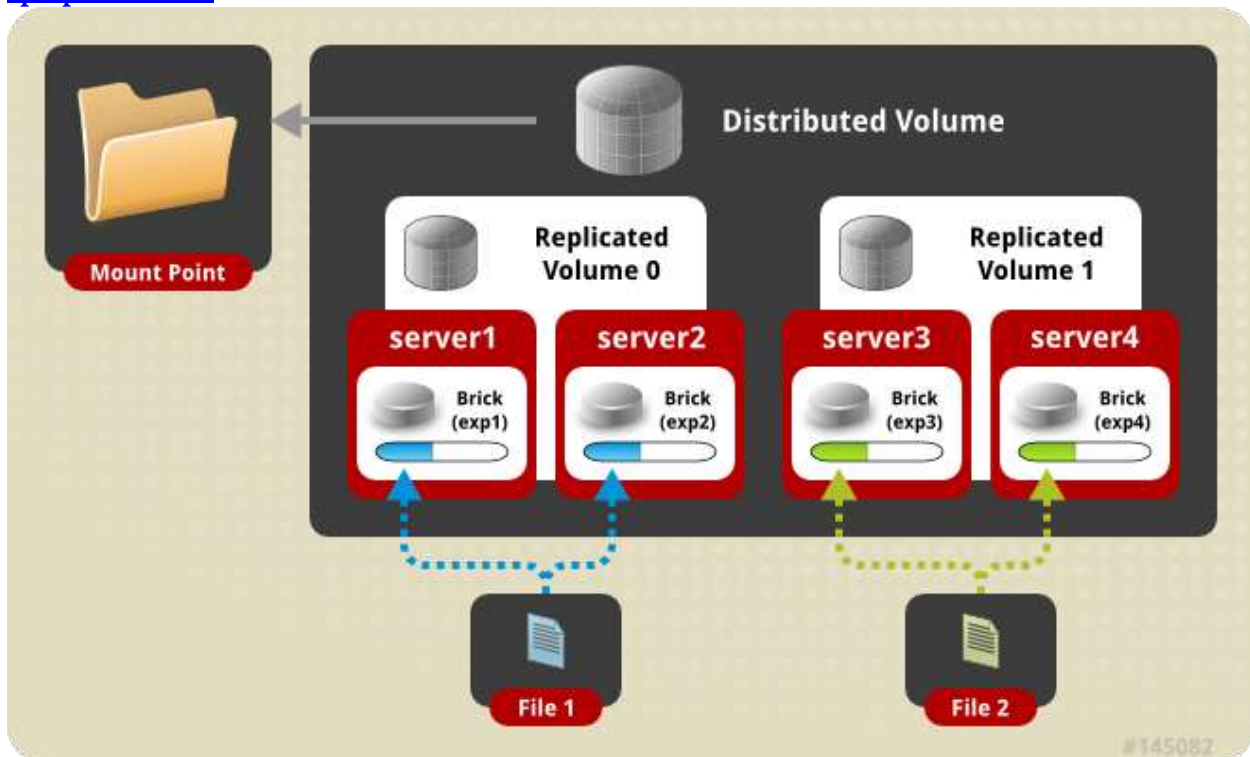
Thin provisioning is foundational in software-defined storage (SDS) and hyper-converged infrastructure (HCI).

## Distributed File Systems in Virtualized Environments

Distributed file systems (DFS) spread data across multiple nodes for scalability, redundancy, and performance in virtualized setups, often replacing traditional SAN/NAS.



[ipSPACE.net](http://ipSPACE.net)



Common options:

- **VMware vSAN:** Hyperconverged, tightly integrated with vSphere/ESXi. Pools local disks into shared datastores; policy-based (e.g., RAID-1/5/6 equivalents). Ideal for VMware-centric environments.

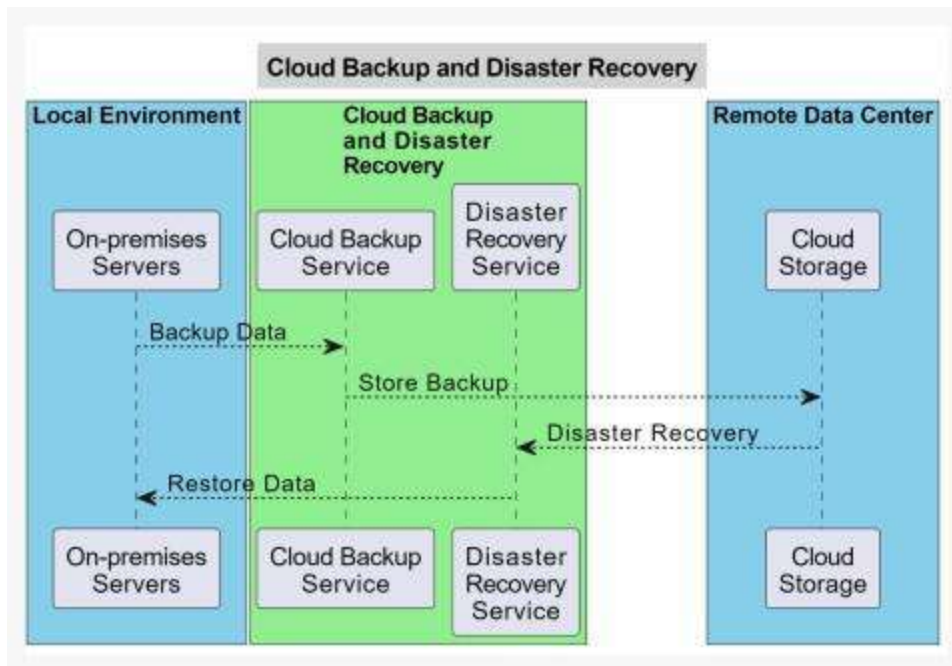
- **Ceph:** Open-source, unified (block/object/file) on commodity hardware. Highly scalable (petabyte+), self-healing via CRUSH algorithm; used with Proxmox, OpenStack, Kubernetes.
- **Others:** GlusterFS (file-focused, simpler), Nutanix Files.

Aspect	VMware vSAN	Ceph
<b>Integration</b>	Native to vSphere (kernel-level)	Independent (works with any hypervisor)
<b>Scalability</b>	Good for mid-large VMware clusters	Massive horizontal scale
<b>Redundancy</b>	Storage policies (FTT, erasure coding)	Replication/erasure coding
<b>Cost</b>	Proprietary licensing	Open-source (free)
<b>Use Cases</b>	VM-centric, seamless management	Multi-protocol, cloud-native, flexible

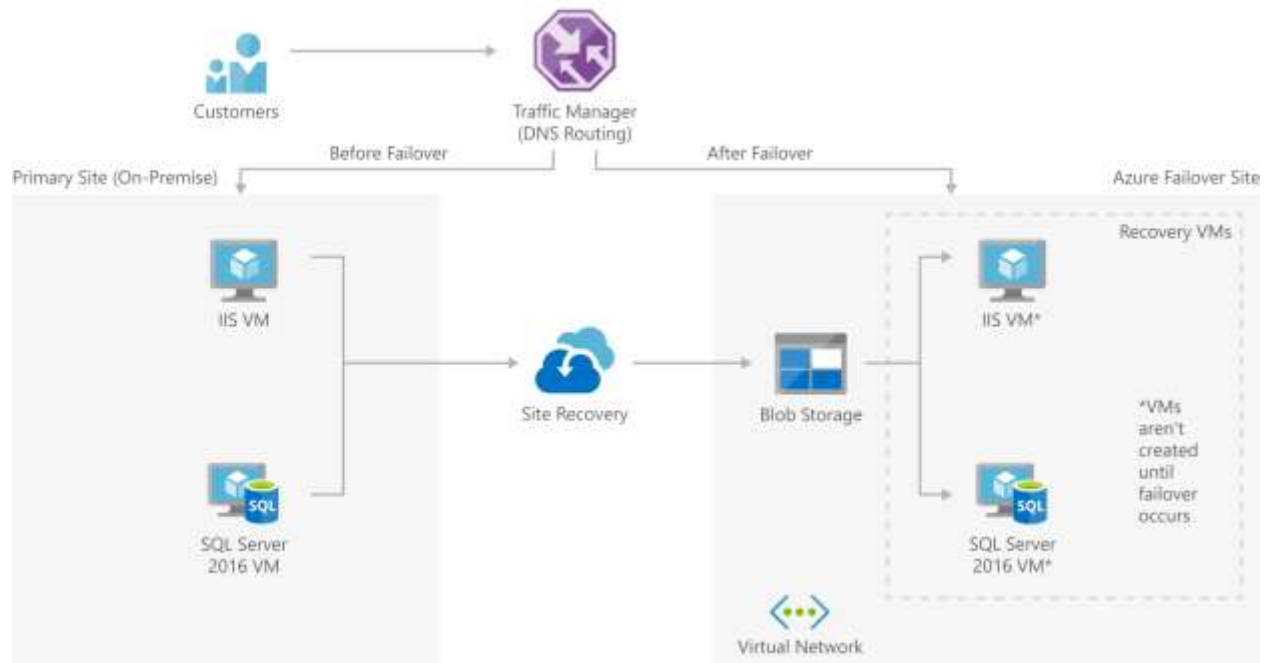
Trends (2025): HCI dominance with AI-driven optimization; Ceph gaining in open ecosystems post-Broadcom VMware changes.

## Backup, Replication, and Disaster Recovery for Virtualized Storage

Virtualized environments leverage image-level backups, snapshots, and replication for efficient protection.



[geeksforgeeks.org](http://geeksforgeeks.org)



[cloudian.com](http://cloudian.com)

- **Backup:** Image-level (via hypervisor APIs like vSphere VADP); incremental with CBT (changed block tracking). Tools: Veeam Backup & Replication (dominant for VMware/Hyper-V).

- **Replication:** Creates ready-to-run VM copies (replicas) on secondary sites. Low RTO (minutes); synchronous/asynchronous.
- **Disaster Recovery (DR):** Failover to replicas; orchestration for multi-VM consistency. Features like Instant VM Recovery (run from backup), SureBackup (verification).

Key Veeam features (2025):

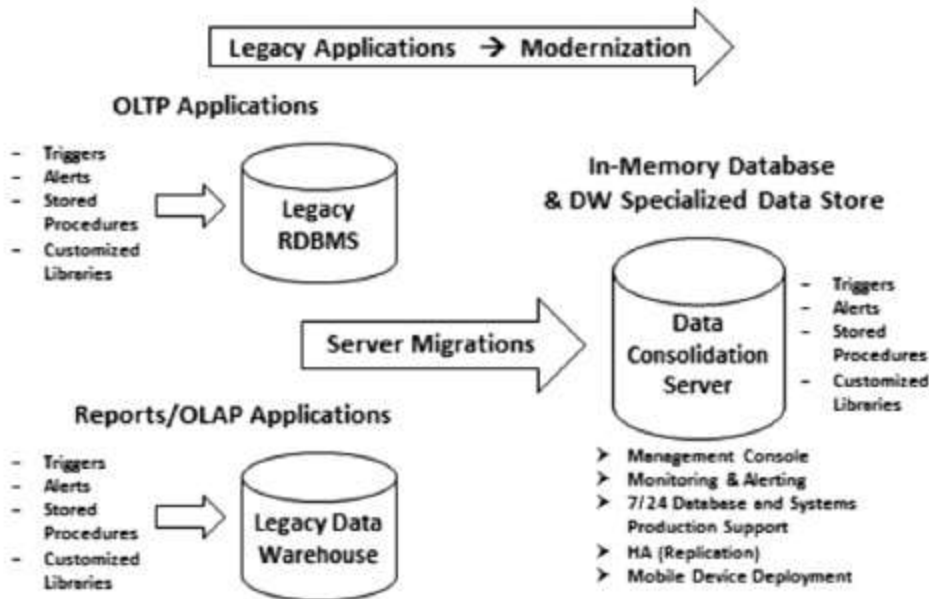
- Instant Recovery: Mount/boot from backup directly.
- CDP (Continuous Data Protection): Near-zero RPO.
- Immutable repositories for ransomware protection.

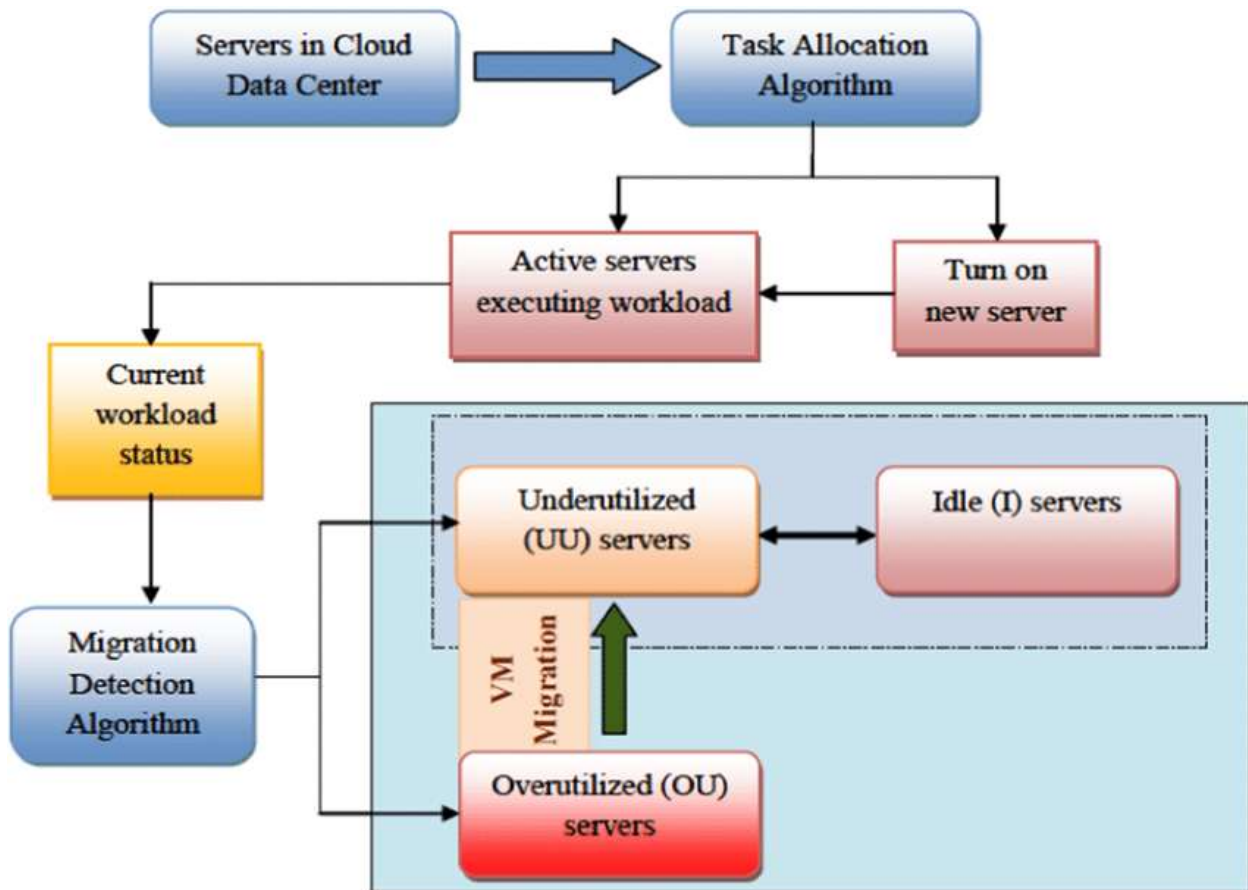
Strategies: 3-2-1 rule (3 copies, 2 media, 1 offsite); integrate with storage snapshots for faster backups.

### **Case Studies on Server and Storage Consolidation**

Server/storage consolidation via virtualization reduces hardware footprint, energy, and costs while improving utilization.

[journals.sagepub.com](https://journals.sagepub.com)





[researchgate.net](https://www.researchgate.net)

- **Hyper-V Migration (2024):** Organization switched from VMware to Windows Server 2025 Hyper-V cluster. Results: 35% virtualization cost reduction, 25% faster VM provisioning, enhanced DR via Azure integration. Leveraged existing licenses amid Broadcom pricing changes.
- **General Trends (2024-2025):** Reports show 80% energy savings from consolidation; multi-cloud adoption (89% organizations). Examples include reducing racks via HCI (vSAN/Nutanix), achieving 70-80% utilization vs. 10-20% on physical.
- **Research/Enterprise:** Studies highlight consolidation ratios of 10-20:1; benefits in cloud onboarding (e.g., network providers via virtualization).