Week 6: Storage Paradigms in Cloud Computing

NT524 — Cloud Architecture and Security

PhD. Nguyen Ngoc Tu

October 15, 2025

Thuật ngữ: storage paradigms = *mô hình lưu trữ*; cloud architecture = *kiến trúc đám mây*

Outline

Learning Objectives

- Motivate storage choices for **cost optimization** and **resilience**.
- Understand **block**, **object**, **file** storage models.
- Apply **data redundancy**: replication, erasure coding, RAID.
- Design for **performance** & **durability** with lifecycle policies.
- Hands-on: OpenStack, AWS, Azure, GCP; backup patterns & validation.

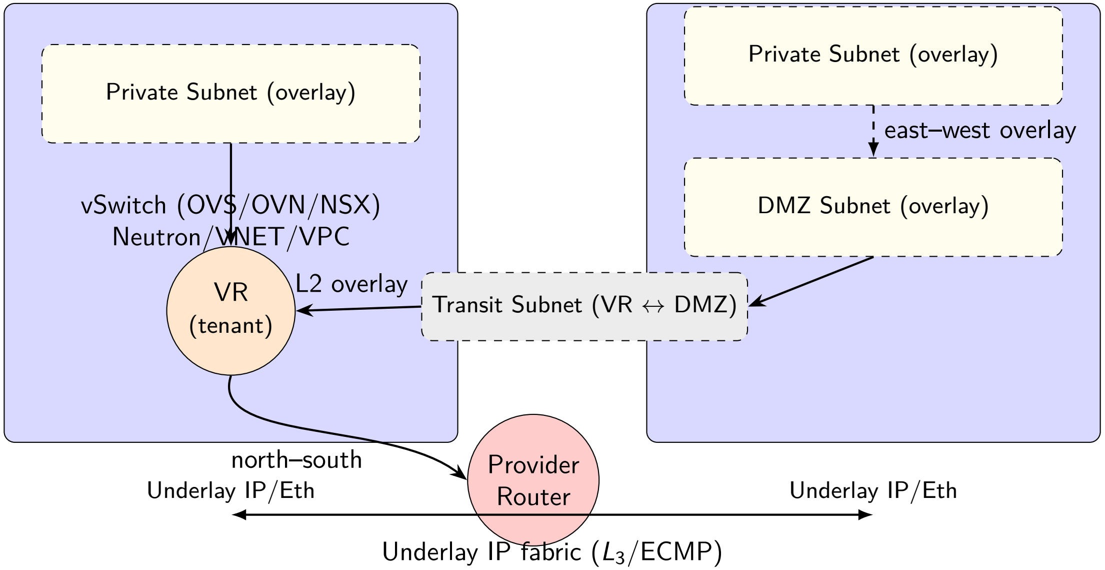
Thuật ngữ: resilience = *khả năng chịu lỗi*; redundancy = *dư thừa*; lifecycle policy = *chính sách vòng đời*

Motivation

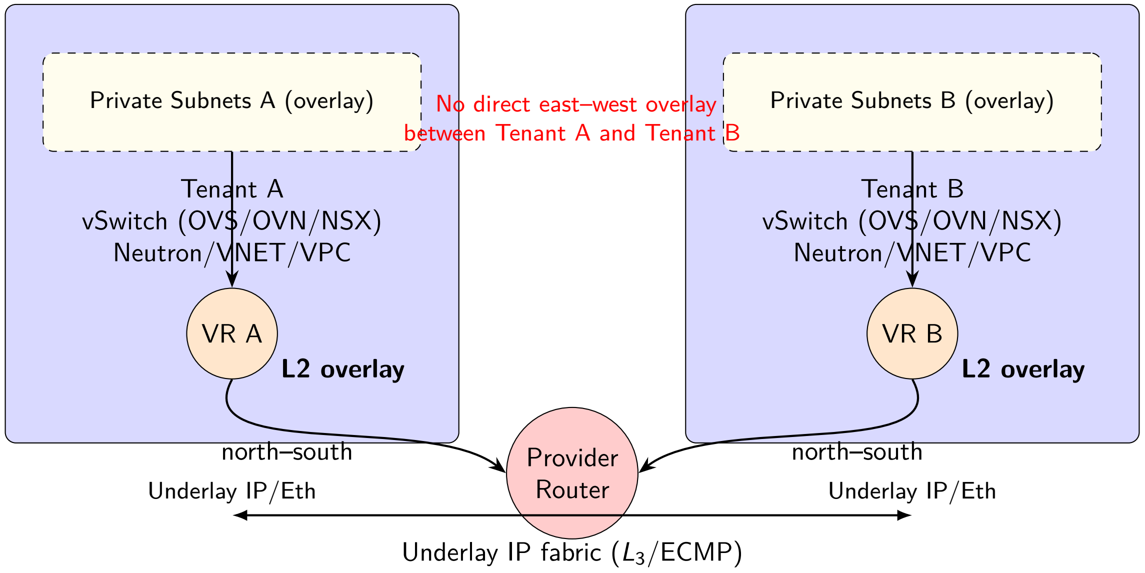
- **Cost:** proper tiering, compression/dedupe, lifecycle (hot/warm/cold).
- **Resilience:** RPO/RTO, multi-AZ/region, tested backups & restores.
- **Fit:** DB $\rightarrow$ block; media/backup $\rightarrow$ object; shared homes $\rightarrow$ file.

Thuật ngữ: RPO/RTO = *mục tiêu điểm/khôi phục*; multi-AZ = *đa vùng sẵn sàng*

Big Picture: Underlay vs. Overlay (Single tenant with Transit Subnet)



Big Picture: Underlay vs. Overlay (Multi-Tenant Isolation)



Overlay–Underlay Mapping (Concepts)

- **Underlay:**
  - Physical IP network (switches, routers, fabric).
  - Provides basic **IP reachability** between nodes.
  - Examples: Data center leaf–spine fabric, WAN backbone.
- **Overlay:**
  - Logical network built *on top of* underlay.
  - Encapsulation: **VXLAN**, **GENEVE**, NVGRE.
  - Creates isolated tenant networks with their own CIDR.
  - Decouples tenant addressing from physical topology.
- **Underlay $\leftrightarrow$ Overlay Mapping:**
  - Overlay packets are encapsulated inside underlay IP packets.
  - **VNI (Virtual Network Identifier, 24 bits)** $\leftrightarrow$ Underlay UDP port + IP path.
  - Example: Tenant subnet 10.0.1.0/24 $\rightarrow$ VXLAN VNI 5001 $\rightarrow$ transported over underlay IP fabric.

Thuật ngữ: Underlay = *Mạng vật lý*; Overlay = *mạng ảo*; Mapping = *ánh xạ mạng vật lý - mạng ảo*

Isolation — the Rule of Three

1. **Separate overlay IDs:** every tenant owns a unique **VNI** and its own logical switch/router (no shared L2/L3).
2. **Default deny across tenants:** no A↔B datapath in the vSwitch; only intra-tenant (same VNI) is reachable.
3. **Edge mediation if needed:** any A↔B must hairpin at the **provider edge/gateway** (FW/NAT/VRF policy).

Thuật ngữ: VNI = *định danh mạng ảo*; default deny = *mặc định chặn*; hairpin qua edge = *đi vòng qua cổng biên*

VXLAN Packet Header Syntax

Outer Ethernet 14 B	Outer IP IPv4 20 B / IPv6 40 B	UDP 8 B, dport=4789	VXLAN 8 B	Original Ethernet + Payload

Underlay (dùng để forward):
– **L2:** Outer Ethernet — *dst/src MAC*
– **L3:** Outer IP — *src/dst IP* $(VTEP_{src} \to VTEP_{dst})$
– **L4:** UDP — *src port (entropy), dport=4789*
→ **vSwitch/VTEP endpoints** thực hiện *encapsulation (add)* và *decapsulation (remove)* các header ngoài.

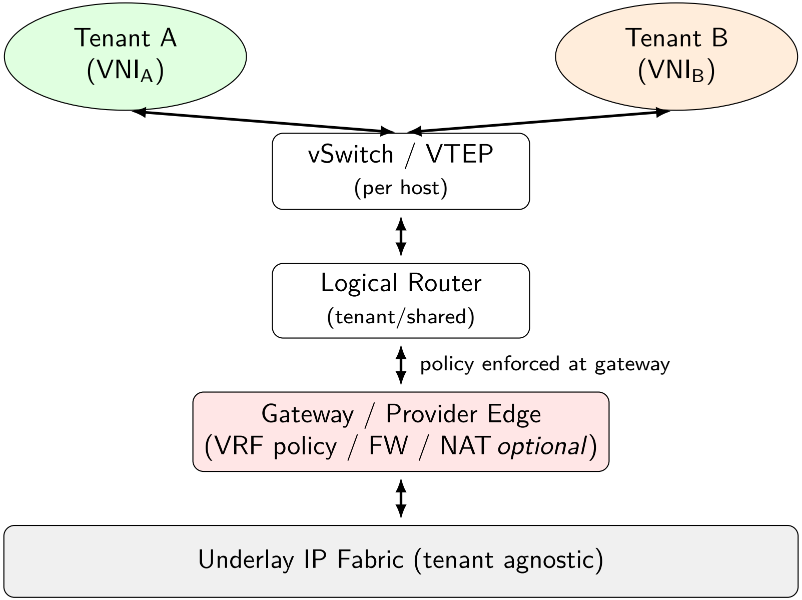
VXLAN (8 Bytes):

0–7:	Flags (I=1)
8–31:	Reserved
32–55:	**VNI (24 bits)**
56–63:	Reserved

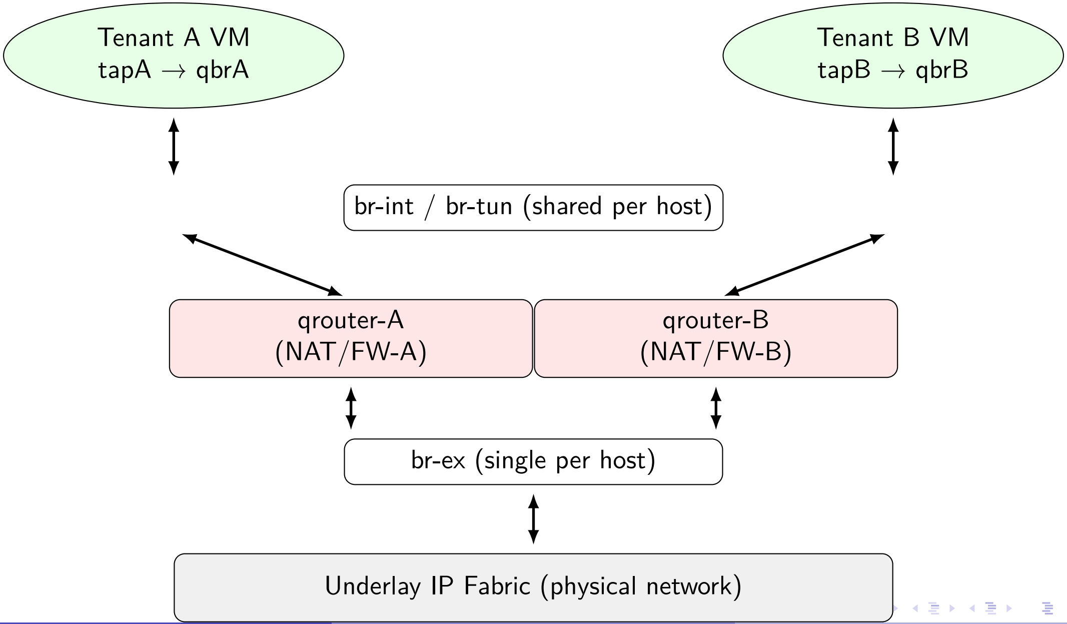
Overlay (bảo toàn bên trong):
– Inner Ethernet — *dst/src MAC*
– Inner IP — *src/dst IP*
– Inner TCP/UDP + Payload

Thuật ngữ: chỉ *bọc thêm* header ngoài (8B, có **VNI 24-bit**); `I=1`. Inner (original) TCP/IP *không mất*.

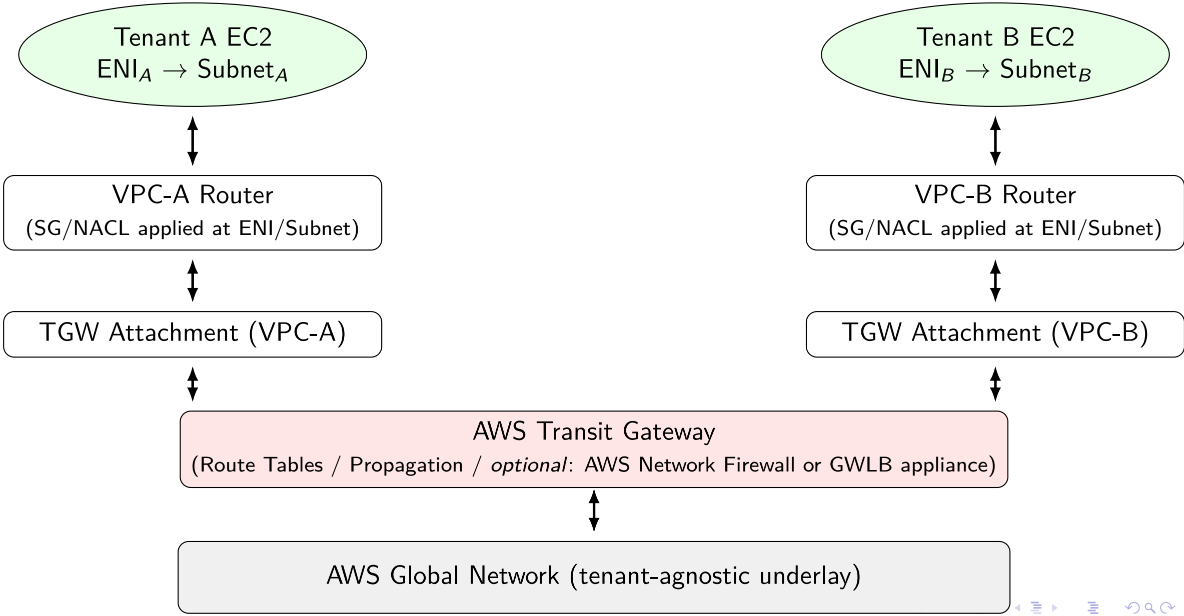
The only allowed inter-tenant path (A $\leftrightarrow$ B via edge)



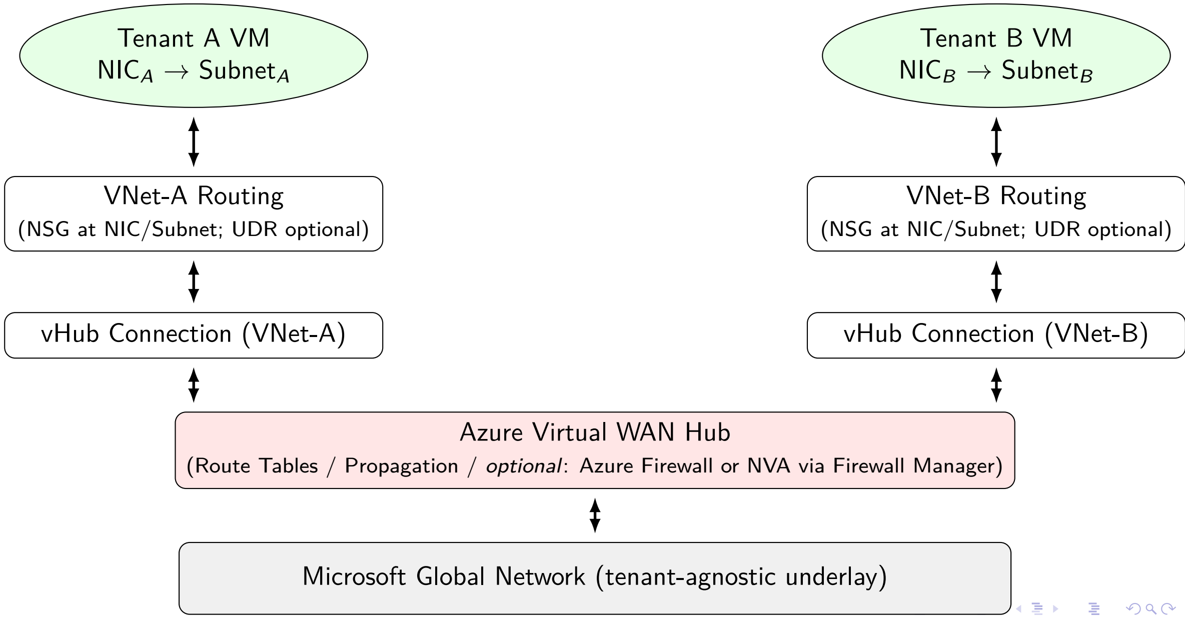
Inter-tenant path in OpenStack (A $\leftrightarrow$ B via L3 routers)



Inter-tenant path in AWS (A $\leftrightarrow$ B via Transit Gateway)


Tenant A EC2
ENI$_A$ $\rightarrow$ Subnet$_A$
Tenant B EC2
ENI$_B$ $\rightarrow$ Subnet$_B$
VPC-A Router
(SG/NACL applied at ENI/Subnet)
VPC-B Router
(SG/NACL applied at ENI/Subnet)
TGW Attachment (VPC-A)
TGW Attachment (VPC-B)
AWS Transit Gateway
(Route Tables / Propagation / *optional*: AWS Network Firewall or GWLB appliance)
AWS Global Network (tenant-agnostic underlay)

Inter-tenant path in Azure (A $\leftrightarrow$ B via vWAN Hub)


Tenant A VM
NIC$_A$ $\rightarrow$ Subnet$_A$
Tenant B VM
NIC$_B$ $\rightarrow$ Subnet$_B$
VNet-A Routing
(NSG at NIC/Subnet; UDR optional)
VNet-B Routing
(NSG at NIC/Subnet; UDR optional)
vHub Connection (VNet-A)
vHub Connection (VNet-B)
Azure Virtual WAN Hub
(Route Tables / Propagation / *optional*: Azure Firewall or NVA via Firewall Manager)
Microsoft Global Network (tenant-agnostic underlay)

Storage Taxonomy

Model	Bare-metal	Cloud cases	Use cases
Block	Local SSD / NVMe namespace / SAN LUN	VM-attached block volume (iSCSI / NVMe-oF); low latency; byte-addressable	Transactional DBs, boot volumes, VM swap disks
Object	Object array / archival system (tape library, DVD archive)-> immutability data model	Key–value object store; flat namespace; HTTP(S) API; rich metadata; *non-POSIX, not mounted directly*	Backups, media, analytics logs, data lakes
File	NAS filer / shared folder / network drive	POSIX semantics via NFS/SMB; multi-client mounts; metadata ops may bottleneck	Home directories, content repositories, legacy apps

Thuật ngữ: iSCSI/NVMe-oF = *giao thức lưu trữ qua mạng*; POSIX = *chuẩn giao diện hệ thống tệp*; NAS (Network-Attached Storage) = *thiết bị lưu trữ gắn qua mạng*.
Lưu ý: Object *không* mount trực tiếp; nếu dùng FUSE (s3fs/gcsfuse) thì *không đầy đủ POSIX*.

Storage Access Layers: File vs Block vs Object

- **File Storage:**
  - Access via **file system API** (open, read, write, close).
  - Provides **hierarchical directories**, file names, and POSIX semantics.
  - Used for user home folders, shared drives (NFS, SMB).
  - **Example:** Accessing files in a NAS share or network drive.
- **Block Storage:**
  - Access via **block device driver** (read/write sectors or blocks).
  - OS formats the volume with a **filesystem (ext4, NTFS)** inside.
  - Behaves as a raw disk; high IOPS and low latency.
  - **Example:** A virtual volume attached to a VM (like `/dev/vdb` or `C:`), even if stored as a `.vhdx` file container.
- **Object Storage:**
  - Access via **HTTP(S) API** (GET, PUT, DELETE); no direct filesystem mount.
  - Stores **immutable objects** = data + metadata + unique ID.
  - Flat namespace (no real folders), designed for scalability and durability.
  - **Example:** Uploading a file to Amazon S3 or Ceph via REST API.

Thuật ngữ: File: truy cập qua API của hệ tệp (thư mục, tên file); Block: truy cập qua driver thiết bị khối (sector, volume, filesystem bên trong); Object: truy cập qua API HTTP (object bất biến, không mount trực

Block Storage Fundamentals

- **Concept:** Data stored in fixed-size blocks (e.g., 512 B, 4 KB) with logical addressing, similar to disk sectors.
- **Model:** *Mutable data model* — blocks can be updated in place (read/write), enabling low-latency transactional workloads.
- **Architecture:** Volume service → backend (LVM, Ceph RBD [*RADOS Block Device*], NVMe-oF, iSCSI, SAN). Volumes attach to hypervisors or VMs **like virtual disks** (.vhdx, .vmdk, or .qcow2). Cloud analogues: *EBS, Azure Managed Disk, Cinder Volume.*
- **Attachment:** A block volume is typically attached to a **single host**. For **multi-attach**, a clustered filesystem (e.g., GFS2, OCFS2, VMFS on ESXi) or **read-only shared** mode is required to prevent data corruption.
- **Performance:** Defined by **IOPS** (operations per second), **throughput** (MB/s), and **latency**. Influenced by **queue depth**, **block size**, **caching policy**, and **tiering strategy**. NVMe and NVMe-oF now dominate due to microsecond-scale latency.
- **Use cases:** Databases, VM boot disks, container storage (PV in Kubernetes), snapshots and clones for rapid provisioning.

Thuật ngữ: clustered filesystem – *hệ tệp cụm*; snapshot – *bản chụp nhanh*; clone – *bản sao nhanh*.

Block Storage — Control Plane (Who Decides)

- **Lifecycle orchestration:** API → scheduler → backend driver.
  - OpenStack: `cinder-api`, `cinder-scheduler`, `cinder-volume`
  - Cloud parallels: EBS control plane, Azure Disk RP, GCP PD manager
- **What it handles:** provision, attach/detach, resize, migrate, snapshot policy, quotas.
- **Replication & consistency:**
  - *Synchronous* (zero RPO; higher latency) vs *asynchronous* (non-zero RPO; geo DR)
  - Quorum/journaled replication across failure domains (e.g., Ceph, ONTAP, EBS Multi-AZ)
- **Encryption & keys:**
  - Per-volume keys via KMS: Barbican (OpenStack), AWS KMS, Azure Key Vault, GCP Cloud KMS
  - Data-at-rest; attach path negotiates keys/policies
- **Governance & telemetry:** quotas, RBAC, audit; metrics/alarms via Ceilometer/CloudWatch/Azure Metrics.

Thuật ngữ: control plane = *mặt phẳng điều khiển*; RPO = *mục tiêu điểm khôi phục*

Block Storage — Data Plane (Who Moves Bytes)

- **I/O path:** Guest VM $\rightarrow$ hypervisor (QEMU/KVM/ESXi) $\rightarrow$ protocol (iSCSI / NVMe-oF / RBD) $\rightarrow$ storage nodes/backends.
- **Ceph example:** librbd $\leftrightarrow$ RADOS (striping, replication/EC).
- **Performance levers:** IOPS, throughput, latency; queue depth, block size, caching, tiering.
- **Snapshots & clones:** CoW or RoW mapping (fast, space-efficient); ideal for gold images/instant restores.
- **QoS & policy in data path:** IOPS/MBps limits, latency targets; placement/tiering hints.
- **Security in flight:** TLS/IPsec for iSCSI/NVMe-oF where required; at-rest encryption handled by backend.
- **Modern note (2025):** NVMe/NVMe-oF widely preferred for microsecond-scale latency and offload support.

Thuật ngữ: data plane = *mặt phẳng dữ liệu*; CoW/RoW = *ghi sao chép/ghi chuyển hướng*

Performance Tuning & Reliability (Compact)

- **IOPS/Throughput:** tune *queue depth* (blk-mq/NVMe multi-queue), align *block size* to stripe, enable request merge for random; multi-path (iSCSI) / ANA (NVMe-oF); NUMA/IRQ pinning.
- **Latency (p95/p99):** cut underlay hops, verify PMTU; prefer NVMe-oF (RoCE/TCP) for µs paths; watch GC/compaction & write amplification.
- **Caching/Tiers:** host cache *read-through*/*write-back* (BBU/journal); SSD/NVMe hot tier + HDD cold tier; FS opts: `noatime`, scheduled TRIM.
- **Reliability:** spread across failure domains (disk→host→rack/AZ); RAID10/3×replica = low-latency, high space; EC(k+m) = space-efficient, higher CPU/latency, slower rebuild; sync (zero RPO) vs async (geo DR).
- **QoS/Multi-tenancy:** token-bucket min/max + **burst credits**; enforce at hypervisor/backend; fairness via cgroup `io.cost`/weights; optional DSCP for storage flows.
- **Observability/Tests:** watch queue depth, device latency, iowait, rebuild; set SLOs on p99; use `fio` with realistic `iodepth`, `numjobs`, R/W mix.

Thuật ngữ: PMTU = *MTU hiệu dụng*; ANA = *Asymmetric Namespace Access*; EC = *mã xoá*; burst credits = *tín dụng bùng nổ*.

Provider Implementations (Well-known Cloud Examples)

Provider	**Tiers / Types**	**Key features & notes**
AWS EBS	gp3, io1/io2, *io2 Block Express*	AZ-scoped volumes; provisioned IOPS/throughput (gp3 decouples); Multi-Attach (io1/io2 on Nitro); snapshots to S3; encryption via KMS.
Azure Disks	Standard HDD/SSD, Premium SSD, *Premium SSD v2*, Ultra Disk	Managed disks; zonal/regional options; Disk Encryption Sets; shared disks for clustering; bursting on select SKUs.
GCP Persistent Disk	Standard, Balanced, SSD; *Hyperdisk (Extreme/Throughput)*	Live resize; regional PD; multi-writer (RWX) for select types; encryption by default / CMEK.
OpenStack Cinder	LVM, Ceph RBD, NetApp ONTAP, Dell EMC (PowerFlex/PowerStore), etc.	API/scheduler/driver control plane; multi-attach since Train; Barbican encryption; per-backend QoS/policy.
VMware vSAN	Clustered NVMe/SSD/HDD pool (policy-based)	Storage policies (RAID 1/5/6, FTT); dedupe/-compression; stretched cluster; vSphere integration.

Thuật ngữ: managed disk — *đĩa được quản lý tập trung;* hyperdisk — *ổ đĩa siêu hiệu năng*

Emerging Trends & Integration with Cloud Services

- **Disaggregated Storage:** NVMe-oF / RoCE / EBOF (Elastic Block Over Fabric) decouples compute.
- **Smart NIC offload:** DPUs handle I/O and encryption, reducing host latency.
- **Container Integration:** CSI drivers (Cloud Storage Interface) for Kubernetes.
- **Hybrid Backups:** Snapshots $\rightarrow$ Object Storage (S3, Azure Blob, GCS).
- **Next Wave:** zonal replica pools, I/O telemetry for AIOps, ML-based tier optimization.

Thuật ngữ: disaggregated storage = *lưu trữ tách rời tính toán*; DPU = *bộ xử lý dữ liệu mạng*

Object Storage Basics

- **Model:** bucket/container $\rightarrow$ object (key, data, metadata).
- **Scale:** scale-out via sharding/partitioning; billions of objects; large data (logs, media, backups).
- **Features:** pre-signed URL, lifecycle tiers, versioning, server-side encryption.
- **Access:** HTTP/HTTPS (REST); strong API contracts; eventual &/or strong read-after-write consistency (provider-specific).

Thuật ngữ: pre-signed URL = *URL đã ký*; server-side encryption = *mã hoá phía máy chủ*

Internal Architecture & Consistency

- **Frontends:** stateless API gateways/routers; request auth (sigv4/SAS/OAuth).
- **Metadata plane:** namespace index, bucket policies, version maps, object manifests.
- **Data plane:** erasure coding or replicated chunks across failure domains (AZs).
- **Placement:** partition by key prefix/hash; hot partitions auto-split/merge.
- **Consistency:** modern clouds offer **strong** read-after-write for new puts/overwrites; cross-region replication may be async.

Thuật ngữ: erasure coding = *mã hoá xoá*; failure domain = *miền lỗi*

Performance Patterns & Cost Controls

- **Throughput:** parallelism + multi-part upload; range GET for partial reads.
- **Key design:** random/hashed prefixes to avoid hot partitions; avoid sequential keys.
- **Latency classes:** frequent access vs infrequent vs archive; retrieval SLA differs.
- **Cost levers:** storage class, lifecycle policies, intelligent tiering/auto-class, compression.
- **Data access accel:** edge caches/CDN; S3 Select/Blob Query for in-place filtering.

Thuật ngữ: multi-part upload = *tải nhiều phần*; hot partition = *phân vùng nóng*

Provider Implementations (Well-known Examples)

- **AWS S3:** Standard/IA/One Zone-IA/Glacier (Instant/Flexible/Deep Archive), Intelligent-Tiering; Access Points; CRR/SRR.
- **Azure Blob:** Hot/Cool/Archive; immutable policies (time-based/legal hold); SAS; hierarchical namespace (Data Lake Gen2).
- **GCP Cloud Storage:** Standard/Nearline/Coldline/Archive; Autoclass; Turbo/dual-region options; uniform IAM.
- **OpenStack Swift:** proxy + account/container/object rings; EC or replication; TempURL; middleware pipeline.
- **S3-compatible (on-prem):** MinIO, Ceph RGW; CSI for Kubernetes; tiering to public clouds.

Thuật ngữ: SAS = *chữ ký truy cập chia sẻ*; CRR/SRR = *sao chép chéo vùng/cùng vùng*

Security, Governance & Emerging Trends

- **Identity/Policy:** IAM + bucket/container policies; block public access; VPC endpoints/PrivateLink.
- **Encryption:** SSE with provider KMS or customer-managed keys; client-side encryption for end-to-end.
- **Data protection:** Versioning, Object Lock (WORM), replication (same/cross-region), integrity checks (ETag/MD5/SHA-256).
- **Eventing:** native notifications to queues/bus (EventBridge/Event Grid/Pub/Sub) for data pipelines.
- **Trends:** storage-lake convergence (lakehouse), multi-region active/active, governance catalogs, zero-ETL analytics on objects.

Thuật ngữ: WORM = *ghi một lần, đọc nhiều lần*; PrivateLink = *kết nối riêng qua mạng nhà cung cấp*

File Storage Concepts

- **Protocols:** NFSv3/v4.1/v4.2, SMB 2/3; POSIX semantics, advisory/mandatory locking.
- **Namespace:** hierarchical paths, directories, ACLs, quotas.
- **Performance:** throughput vs concurrency; metadata ops (stat/readdir) are bottlenecks.
- **Use cases:** home dirs, web content, render farms, EDA/HPC, legacy apps.
- **Access modes:** multi-client shared (RWX), file/byte-range locking, close-to-open consistency (NFS).

Thuật ngữ: locking = *khoá tệp*; metadata ops = *tác vụ siêu dữ liệu*; RWX = *đọc-ghi-thực thi dùng chung*

Internal Architecture & Scale-Out NAS

- **Control plane:** export/share policies, authN/Z (LDAP/AD, Kerberos), snapshots.
- **Data plane:** NAS heads $\rightarrow$ backends (RAID/erasure) via SAS/NVMe; scale-out adds *multiple* heads.
- **Metadata:** single vs distributed MDS; directory sharding; journaling.
- **Parallelization:** pNFS layouts (files/objects/blocks); SMB Multichannel; NFS over RDMA.
- **Multi-protocol:** NFS+SMB to same dataset (name-mapping, ACL translation).

Thuật ngữ: MDS = *máy chủ siêu dữ liệu*; pNFS = *NFS song song*; RDMA = *truy cập bộ nhớ từ xa*

Performance, Tuning & Operations

- **Throughput vs IOPS:** large sequential IO vs small random; many small files hurt.
- **Client tuning:** NFS `rsize/wsize`, `nconnect`, attribute caching; SMB Multichannel, credits.
- **Latency:** network RTT, server CPU, metadata locks; prefer close clients/AZ locality.
- **Hotspots:** single directory hot keys; hash/fan-out directory design.
- **Ops:** snapshots/clones, quotas (user/group/project), tiering to object, backups.

Thuật ngữ: snapshot = *bản chụp nhanh*; fan-out = *toả nhánh*

Provider Implementations (Well-known Examples)

- **AWS EFS:** regional, NFSv4.1; classes: Standard/IA; throughput modes: Bursting/Provisioned; EFS-to-S3 lifecycle.
- **AWS FSx:** NetApp ONTAP (NFS/SMB/SnapMirror), Lustre (HPC), OpenZFS; multi-AZ options.
- **Azure Files:** SMB/NFS; tiers Hot/Cool/Premium; **ANF** (Azure NetApp Files) for high IOPS/low latency.
- **GCP Filestore:** Basic/Enterprise/High Scale; NFSv3/v4.1; regional HA in Enterprise.
- **OpenStack Manila:** drivers for CephFS, NetApp, GlusterFS, Dell/EMC, etc.; share networks, access rules.
- **HPC/On-prem:** Lustre, IBM Spectrum Scale (GPFS), BeeGFS, Isilon/PowerScale, NetApp ONTAP.

Thuật ngữ: HA = *tính sẵn sàng cao*

Security, Governance & Advanced Patterns

- **Identity/ACLs:** POSIX vs NFSv4/SMB ACLs; AD/Kerberos; least privilege shares/exports.
- **Encryption:** at-rest (KMS/Key Vault/Cloud KMS) & in-flight (TLS, SMB encryption, Kerberos signing).
- **Data protection:** snapshots, replication (sync/async, cross-AZ/region), ransomware-resilient immutable snaps.
- **Kubernetes:** CSI drivers for RWX; dynamic provisioning; workload-affinity to reduce latency.
- **Trends:** disaggregated NAS over NVMe-oF, DPU offload for crypto/IO, multi-protocol data lakes, policy-driven tiering to object.

Thuật ngữ: immutable snapshot = *bản chụp không thể sửa*; CSI = *giao diện lưu trữ cho container*

Case Setup & SLOs (Cross-Cloud)

- **Workloads & I/O:** DB (8–64 KB random RW), media render (1–8 MB seq), analytics (mixed; large scans).
- **Protocols:** iSCSI / NVMe-oF (block), NFSv4.1/4.2 & SMB 3.x (file).
- **SDN overlay:** VXLAN/Geneve overhead ($\approx$50–60 B) $\Rightarrow$ plan MTU/headroom; rely on PMTU discovery.
- **Routing fabric:** east–west via L3 gateways (qrouter/TGW/vWAN/Cloud Router) with policy; avoid L2 extension.
- **SLO targets:** p99 latency (block: $\leq$2–5 ms; file: $\leq$5–15 ms), sustained throughput (GB/s), durability (nines), RPO/RTO.
- **Policy/QoS:** DSCP marking at host; per-class queues; IOPS/MBps limits where supported.

Thuật ngữ: PMTU = *phát hiện MTU đường đi*; east–west = *luồng nội bộ giữa VNET/VPC/VN*

OpenStack (OVN) — iSCSI/NFS/NVMe-oF

- **Overlay/MTU:** ML2/OVN (Geneve). Set tenant MTU (e.g., 1450/9000-overhead) and underlay MTU consistently; verify with `tracepath` / `ping -M do`.
- **Cinder (block):** iSCSI or Ceph RBD. For iSCSI: dm-multipath, `no_path_retry`, udev rules; for NVMe-oF: enable NVMe multipath.
- **Manila (file):** NFS/SMB exports via share networks; align export subnet with router namespaces; security groups → minimal, allow only storage flows.
- **Routing/policy:** Inter-tenant via logical routers → gateway chassis (NAT optional). OVN ACLs at LR/LSP; ECMP for scale.
- **QoS/DSCP:** Neutron QoS (rate limit) + OVN QoS rules; DSCP mark on instances (tc/iptables mangle) → preserve in underlay if supported.
- **Validate:** `fio` (iodepth sweep), `iperf3` (parallel flows), `ovn-trace` (policy path), `ovn-appctl` latency.

Thuật ngữ: ECMP = *cân bằng tải đa đường*; LR/LSP = *router/port logic*

AWS (Transit Gateway) — EFS/FSx & Hybrid iSCSI

- **File services: EFS** (NFSv4.1) for shared POSIX; **FSx for ONTAP** (NFS/SMB/iSCSI features) or **FSx for Lustre** (HPC).
- **Block paths:** EBS is local to EC2 (no network). For SAN-like iSCSI, use FSx/ONTAP or on-prem SAN via DX/VPN.
- **SDN path:** VPC $\leftrightarrow$ TGW (route tables per attachment); insert **Network Firewall/GWLB** for policy.
- **MTU/PMTU:** Use provider defaults per ENI/VPC; verify with `tracepath` across TGW/ENI and DX/VPN to avoid fragmentation.
- **QoS/DSCP:** Mark at EC2; preserve across appliances; use policing on NVA if needed (not native in VPC fabric).
- **Tuning:** EFS throughput mode (Bursting/Provisioned), concurrency via multiple NFS clients; FSx/ONTAP: junctions, aggregates, FlexVol/QoS.
- **Validate:** `fio` on EFS mount (nconnect), `nfsstat`, CW metrics (PercentIOLimit), `mtr` across TGW.

Thuật ngữ: DX = *Direct Connect*; GWLB = *cân bằng tải cổng*

Azure (vWAN/Hub-VNet) — ANF & Azure Files

- **File services: Azure NetApp Files (ANF)** for high IOPS/low latency (NFS/SMB); **Azure Files** (SMB/NFS) for broad compatibility.
- **SDN path:** Spoke VNets ↔ **vWAN Hub** or Hub-VNet; centralized policy via **Azure Firewall**/NVA; UDR for steering.
- **Identity/ACL:** AD/Kerberos for SMB; POSIX modebits/ACLs for NFS; carefully map IDs across domains.
- **MTU/PMTU:** Keep consistent per NIC/vNIC and gateways; validate PMTU on ER/VPN paths; avoid L2 stretch, prefer routed.
- **QoS:** ANF capacity pools/volume QoS; Azure Files tiers (Hot/Cool/Premium); traffic shaping on NVA.
- **Validate:** `fio` over NFS/SMB, `nfsstat`/`smbstatus`, Connection Monitor, `psping` for tail latency.

Thuật ngữ: UDR = *định tuyến do người dùng*; ER = *ExpressRoute*

GCP (Cloud Router) — Filestore & Hybrid NVMe-oF

- **File services: Filestore** (NFSv3/v4.1) tiers: Basic/Enterprise/High Scale; regional HA in Enterprise.
- **Block/high-perf:** For extreme IOPS, consider local SSD or Hyperdisk (for DB) + app-level sharding; NVMe-oF usually via partner NVAs.
- **SDN path:** VPCs via **VPC peering** or **Cloud Router** across **Cloud Interconnect**/VPN; centralize policy with firewalls/NVAs.
- **MTU/PMTU:** Keep per-VPC/NIC consistent; verify PMTU across Interconnect; avoid fragmentation on storage flows.
- **QoS/DSCP:** Mark at VM; enforce in NVAs; use Filestore performance guidance (num threads, rsize/wsize).
- **Validate:** `fio` profiles (randread/randwrite, bs=4k/64k/1m), `iperf3` with multiple streams, `traceroute`/`tracepath`.

Thuật ngữ: Interconnect = *kết nối chuyên dụng tới GCP*; NVA = *thiết bị ảo mạng*

Data Redundancy

- **Replication (x3):** improves availability and reads; higher cost.
- **Erasure coding (k+m):** capacity efficiency $= k/(k+m)$; overhead $= (k+m)/k$; best for cold/large objects.
- **RAID:** 0/1/5/6/10; parity write penalties; rebuild risk windows.
- **Durability vs availability:** data loss vs uptime.

Thuật ngữ: erasure coding = *mã xoá*; availability = *khả dụng*

Cloud Mapping (OpenStack / AWS / Azure / GCP)

Need	**OpenStack**	**AWS**	**Azure / GCP**
Block	Cinder (LVM/Ceph)	EBS (gp3, io1/io2; Multi-Attach)	Managed Disks / PD (balanced/ssd)
Object	Swift (replication/EC)	S3 (strong consistency)	Blob / GCS (Std/Near/-Cold/Archive)
File	Manila (NFS/SMB)	EFS / FSx	Azure Files (SMB/NFS) / Filestore
Keys	Barbican	KMS	Key Vault / KMS
Backup	Snapshots $\rightarrow$ object	EBS snapshots $\rightarrow$ S3	Snapshots $\rightarrow$ Blob/GCS

Thuật ngữ: strong consistency = *nhất quán mạnh*; Filestore = *dịch vụ NFS GCP*

Hands-on: OpenStack

Goal: Cinder/Swift/Manila via Ansible; snapshot & verified restore.

1. Create *volume type* + *QoS*; attach to VM; format/mount; run `fio`.
2. Swift container; upload; set `X-Delete-After`; verify expiry.
3. Manila NFS share; IP access; mount on 2 VMs; test locks/concurrent writes.
4. Snapshot volume; archive to Swift; **restore** to new VM; verify SHA-256.

Thuật ngữ: QoS = *chất lượng dịch vụ*; expiry = *hết hạn*; SHA-256 = *băm kiểm toàn vẹn*

Ansible: Volume Type + QoS (OpenStack)

```
- hosts: controller
  tasks:
    - name: Create QoS and volume type, then associate
      openstack.cloud.qos:
        state: present
        name: ssd-qos
        specs:
          total_iops_sec: "3000"

    - name: Create volume type
      openstack.cloud.volume_type:
        state: present
        name: ssd-qos-type

    - name: Associate QoS to type
      openstack.cloud.volume_type:
        state: present
        name: ssd-qos-type
        qos_specs: ssd-qos

    - name: Create 20GB volume using the type
```

Manila NFS: Export & Mount (OpenStack)

```
- hosts: controller
  tasks:
    - name: Create NFS share
      openstack.cloud.share:
        state: present
        name: wk6-share
        size: 10
        share_proto: NFS
        is_public: false
      register: create_share

    - name: Get export location
      openstack.cloud.share_info:
        name: wk6-share
      register: share_info

    - set_fact:
        wk6_share_export: "{{ share_info.shares[0].export_locations[0].path }}"

- hosts: apps
  become: true
```

Swift Lifecycle & Backup (OpenStack)

```
- hosts: controller
  tasks:
    - name: Create container with metadata
      openstack.cloud.object_container:
        state: present
        name: wk6-backups
        metadata: { "retention": "180d" }

    - name: Upload with expiry (X-Delete-After)
      openstack.cloud.object:
        state: present
        container: wk6-backups
        name: snap-{{ ansible_date_time.epoch }}.raw
        filename: /tmp/snap.raw
        headers:
          X-Delete-After: "15552000" # 180 days

    - name: Create snapshot
      openstack.cloud.volume_snapshot:
        state: present
        name: wk6-db-snap-{{ ansible_date_time.date }}
```

Hands-on: AWS

Goal: EBS (gp3/io2, Multi-Attach), S3 (strong consistency), EFS.

1. Create EBS gp3; attach to EC2; benchmark `fio`; try Multi-Attach with a cluster FS.
2. S3 bucket: enable versioning; pre-signed URL; lifecycle to Glacier.
3. EFS: mount from 2 EC2; test locks; observe CloudWatch metrics.

Thuật ngữ: Multi-Attach = *gắn nhiều phiên*; Glacier = *lưu trữ lạnh*

Hands-on: Azure

Goal: Managed Disks, Blob, Azure Files (SMB/NFS, Premium).

1. Managed Disk: attach to VM; `fio` gp/premium; monitor IOPS.
2. Blob: container, SAS tokens, lifecycle (cool/archive); test restore.
3. Azure Files: SMB or NFS (Premium); try SMB Multichannel if supported.

Thuật ngữ: SAS = *chữ ký chia sẻ*; Multichannel = *đa kênh SMB*

Hands-on: GCP

Goal: Persistent Disk (pd-balanced/pd-ssd), Cloud Storage (Std/Near/Cold/Archive), Filestore.

1. PD: attach to GCE; run `fio`; snapshots.
2. Cloud Storage: classes; signed URL; lifecycle; check retrieval cost.
3. Filestore: NFS mount from 2 VMs; test POSIX locks.

Thuật ngữ: GCE = *máy ảo GCP*; PD = *đĩa bền*

Performance & Cost

- **Benchmark:** `fio` 4k randread (QD=1/16/32), 4k 70/30, 1M sequential; record p50/p95/p99.
- **Tiers:** hot/warm/cold; compression; columnar formats (Parquet/ORC) for analytics.
- **Egress/requests:** batch, CDN, concurrency.
- **FinOps:** track $/TB$-month, $/IOPS$; prune orphaned snapshots.

Thuật ngữ: p50/p95/p99 = *bách phân vị*; egress = *dữ liệu ra*

Security & Compliance

- **At rest:** LUKS, backend-native, KMS/Barbican; per-project keys.
- **In transit:** TLS/mTLS; NFS/SMB encryption where supported; IPsec for iSCSI if needed.
- **Access:** RBAC/ABAC; bucket policies; block public access; least privilege.
- **Immutability:** WORM/object lock; protect snapshots/backups from deletion.

Thuật ngữ: ABAC = *kiểm soát theo thuộc tính*; WORM = *ghi một lần đọc nhiều lần*

Threat Model (STRIDE) for Storage

- **Spoofing:** unauthorized bucket access ⇒ IAM/RBAC, short-lived pre-signed URLs.
- **Tampering:** object/volume modification ⇒ versioning, WORM, checksum, signatures.
- **Repudiation:** missing audit ⇒ enable logs (Cinder/Manila/Swift/EBS/S3/EFS).
- **Information Disclosure:** public buckets, exposed snapshots ⇒ policies & encryption.
- **DoS:** IO storms ⇒ QoS limits; API rate limits.
- **EoP:** container/hypervisor escape ⇒ patching, network policy, isolated IAM.

Thuật ngữ: audit = *ghi nhật ký kiểm toán*; rate limit = *giới hạn tốc độ yêu cầu*

Decision Flow

1. **Access pattern?** random vs sequential; small vs large objects.
2. **Consistency?** strong vs eventual; single vs multi-region.
3. **Failure domain?** disk/node/AZ/region.
4. **SLO & Cost?** p95 latency, $/TB$-month, $/IOPS$.
5. **Governance?** PII, retention, immutability, legal hold.

Thuật ngữ: failure domain = *miền lỗi*; governance = *quản trị dữ liệu*

Comparison: Services & Features

	OpenStack	**AWS**	**Azure**	**GCP**
Block	Cinder (QoS, snapshots)	EBS (gp3, io2, Multi-Attach)	Managed Disks (Std/Premium)	PD (standard/bal-anced/ssd)
Object	Swift (replication/EC)	S3 (strong R/W/LIST)	Blob (hot/cool/archive)	GCS (Std/Near/-Cold/Archive)
File	Manila (NFS/SMB)	EFS/FSx	Azure Files (SMB/NFS)	Filestore (NFS)
Consistency	Backend-dependent	Strong	Strong (Blob)	Strong reads (ops vary)
Keys	Barbican	KMS	Key Vault	KMS
Notes	OVN SDN common	Nitro + NVMe	SMB Multichannel	Regional/multi-reg buckets

Thuật ngữ: LIST = *liệt kê*; Nitro = *ảo hoá phần cứng AWS*; Multichannel = *đa kênh SMB*

Assessment & Deliverables

- **Report (2–3 pages):** architecture for 3 workloads; technical & cost rationale; clear SLOs.
- **Evidence:** screenshots: volumes/attach, buckets/containers, share mounts, `fio`, restores.
- **Files:** playbooks, scripts, lifecycle, recovery runbook.
- **Rubric (10 pts):** functionality (4), performance (2), security (2), docs (2).

Thuật ngữ: runbook = *sổ tay quy trình*; rubric = *thang điểm chi tiết*

Wrap-up

- Match block/object/file to workload and budget.
- Choose redundancy wisely (replication/EC/RAID), encrypt, apply lifecycle.
- SDN impacts storage data paths: MTU, QoS, multipath.
- Next: Container Fundamentals.

Thuật ngữ: data plane = *mặt phẳng dữ liệu*; control plane = *mặt phẳng điều khiển*