Week 4: Fundamentals of Virtual Networks

NT524 — Cloud Architecture and Security

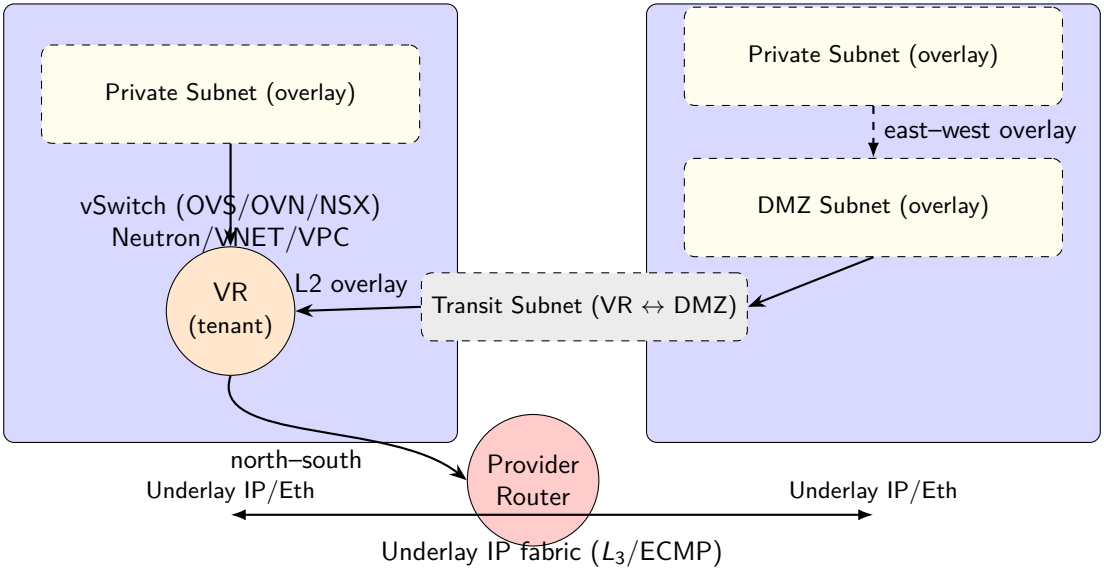
PhD. Nguyen Ngoc Tu

October 1, 2025

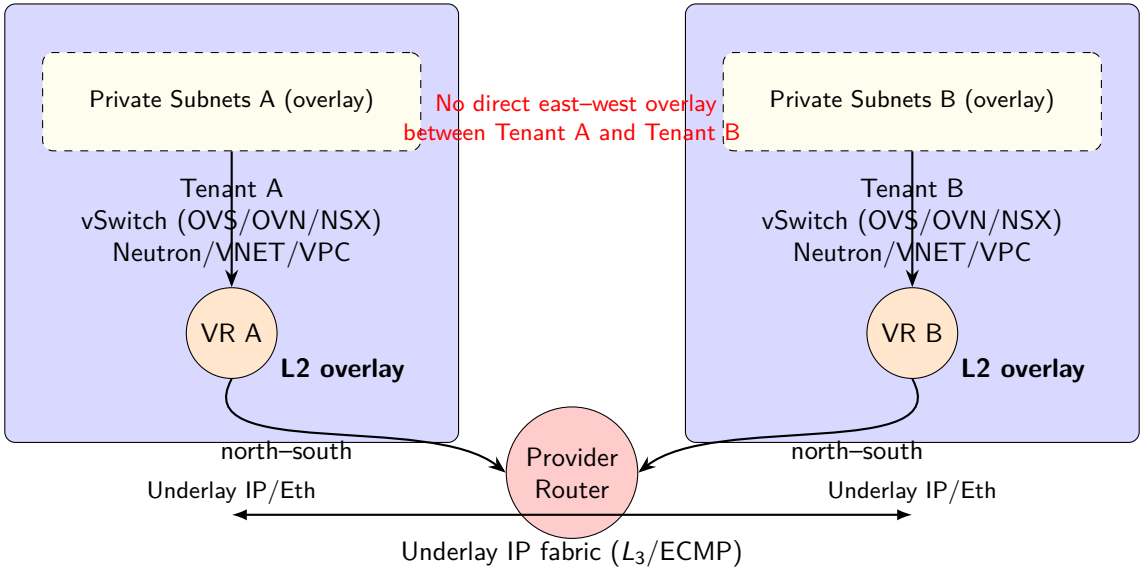
Thuật ngữ: Máy ảo (VM), trình ảo hoá (hypervisor); thành phần mạng (Network/Subnet/Router/SG/Floating IP); hạ tầng mạng (overlay network/underlay network), vòng đời mạng (network lifecycle).

Outline

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



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



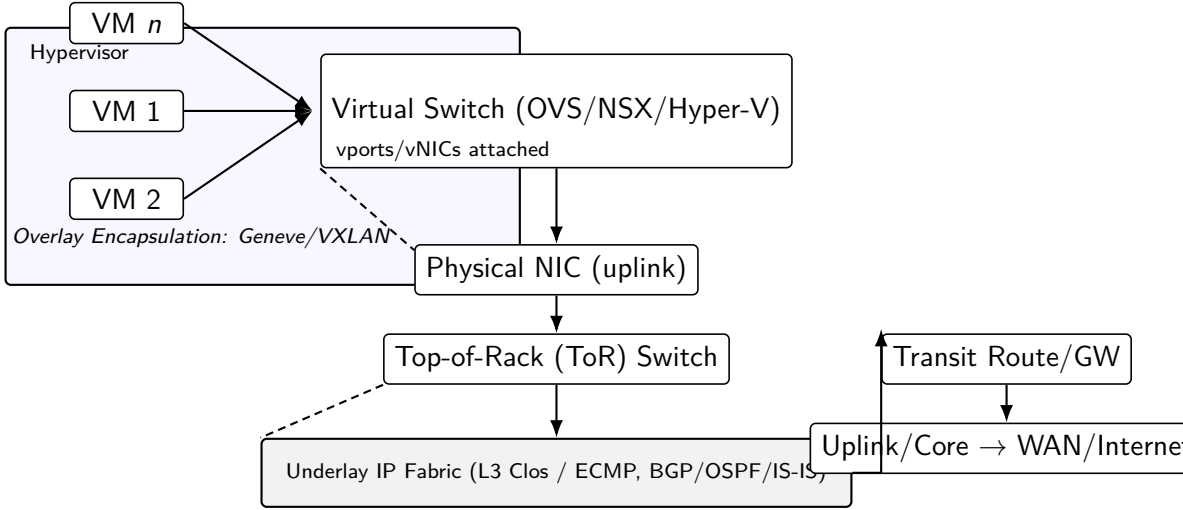
Overlay–Underlay Mapping (Concepts)

- **Underlay (mạng vật lý):**
  - Physical IP network (switches, routers, fabric).
  - Provides basic **IP reachability** between nodes.
  - Examples: Data center leaf–spine fabric, WAN backbone.
- **Overlay (mạng ảo):**
  - 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.
- **Mapping (ánh xạ):**
  - Overlay packets are encapsulated inside underlay IP packets.
  - **VNI (Virtual Network Identifier)** $\leftrightarrow$ Underlay UDP port + IP path.
  - Example: Tenant subnet 10.0.1.0/24 $\rightarrow$ VXLAN VNI 5001 $\rightarrow$ transported over underlay IP fabric.

Overlay–Underlay Mapping (Industry)

- **OpenStack (Neutron):**
  - Tenant networks use VXLAN/GENEVE with unique VNI.
  - Underlay IP fabric carries encapsulated packets.
  - Mapping managed by OVS/OVN or SDN controller.
- **AWS (VPC):**
  - VPC(Virtual private cloud) is logical overlay, backed by AWS underlay fabric.
  - ENI (Elastic Network Interface) acts as overlay port.
  - Customers do not see underlay, only overlay CIDRs.
- **Azure (VNET):**
  - VNets are overlays mapped to physical underlay.
  - Encapsulation/protocol hidden, but functions similar to VXLAN.
  - UDR (User-Defined Routes) influence overlay routing.
- **GCP(VPC):**
  - VPC (Virtual private cloud) is global overlay across regions.
  - Underlay = Google's backbone network.
  - Cloud Router integrates overlay with on-prem via BGP.

Underlay vs. Overlay (Hypervisor → Fabric → Transit)



Control planes (examples): OpenStack Neutron / VMware NSX / AWS VPC / Azure VNet / GCP VPC
Roles: vSwitch encapsulates tenant traffic (overlay) → pNIC → ToR →
Underlay fabric forwards outer IP (no tenant state) → Transit GW for N–S or E–W between VRFs/tenants.

VMs & Hypervisors (Cross-Cloud Overview)

Platform	Hypervisor	Public IP Concept	Notes
OpenStack (on-prem)	KVM/QEMU (via libvirt)	Floating IP	Flexible backends (Ceph/Cinder/OVS/OVN); Neutron networking.
VMware vSphere	ESXi	Public IP via Edge/LB	NSX provides SDN; VMDK disks; snapshots as delta.
Microsoft Azure	Hyper-V	Public IP	VNet/NSG; Managed Disks; PIP can attach to NIC/LB.
AWS EC2	Xen/Nitro Hypervisor (KVM-based)	Elastic IP (EIP)	VPC/SG/NACL; EBS volumes or instance store NVMe.
Google Compute Engine	KVM	External IP	VPC/Firewall; Persistent/Local SSD; per-NIC ext IP.

Thuật ngữ: IP công khai: FIP/EIP/PIP/External IP.

Virtual Disks & Snapshots (Forensics-aware)

Platform	Disk Types	Snapshot/Location Notes
OpenStack (KVM)	qcow2/raw; Cinder	`/var/lib/nova/instances/<id>/disk` or Ceph RBD; Glance images; Cinder snapshots.
VMware	VMDK	`/vmfs/volumes/<datastore>/<vm>/<vm>.vmdk`; deltas `*-delta.vmdk`.
Hyper-V	VHDX	`C:\ProgramData\Microsoft\Windows\Virtual Hard Disks`.
AWS	EBS, instance store	Snapshots stored in EBS; instance store ephemeral NVMe `/dev/nvme*`.
Azure	Managed Disks	Snapshots of managed disks; ephemeral OS disk option.
GCP	Persistent/Local SSD	Snapshots of PD; Local SSD ephemeral at `/dev/nvme*`.

Thuật ngữ: Đĩa ảo: qcow2/VMDK/VHDX; ảnh chụp (snapshot); đĩa tạm (ephemeral) vs thường trực (persistent).

Network Components (1/2): Workloads & L3 Foundations

1. **End Hosts & Ports (vNICs)**
   Workloads (VMs/containers/bare-metal) attach via *ports* (virtual NICs). A port carries MAC/IP, anti-spoof, QoS, and policy handles.
2. **Security Groups (SG)**
   Stateful, per-port firewalls enforced at the hypervisor datapath (OVS/OVN/NSX). Control east—west and north—south at the workload edge.
3. **Subnet (Private & Public)**
   IP pools (CIDR, gateway, DNS, host-routes). Private = internal reachability; Public = directly routed or via NAT.

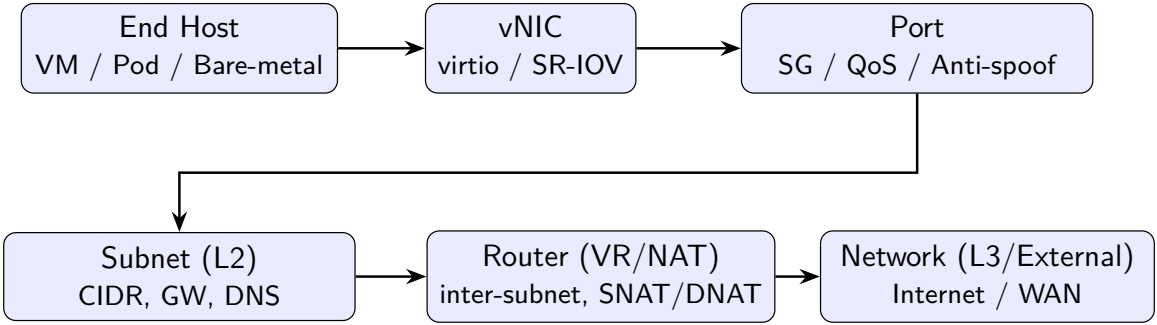
Thuật ngữ: (1)vNIC: Virtual Network Interface Card (network adapter); (2) CIDR: Classless Inter-Domain Routing (allocating IP addresses); (3) Subnet = dải IP/CIDR với gateway/DNS/route; Private (nội bộ), Public (ra ngoài).

Network Components (2/2): Fabric, Middleware & External

1. **Logical Networks / Encapsulation**
   L2 domains: VLAN (on-prem), VXLAN/GENEVE (overlay). Encapsulations provide tenant isolation across the underlay.
2. **Middleware / Network Nodes**
   - **Router (VR)** — tenant L3 gateway; inter-subnet routing; SNAT/DNAT; static routes.
   - **Edge/DMZ Hosts** — firewall/WAF/LB/IDS; often multi-homed (e.g., DMZ + Transit).
   - **Transit Networks** — dedicated subnets for VR $\leftrightarrow$ DMZ/FW; policy choke point.
3. **Provider / External Connectivity**
   Provider router, underlay L3/ECMP fabric; Floating/Public IPs or VIPs for Internet exposure.

Thuật ngữ: (1) Mạng logic: VLAN/VXLAN/Geneve; (2) Nút trung gian: Router ảo, DMZ/Firewall, Transit; (3) Kết nối ra ngoài: router nhà cung cấp, IP công khai/VIP; (4) Chuỗi từ ứng dụng tới ngoài Internet: vNIC $\rightarrow$ Network/Subnet $\rightarrow$ VR $\rightarrow$ Transit/Edge $\rightarrow$ Provider.

Network mapping Path



Thuật ngữ: **CIDR** (*Classless Inter-Domain Routing*): cách biểu diễn dải địa chỉ IP kèm theo độ dài mặt nạ mạng. Ví dụ: `10.0.1.0/24` (mặt nạ mạng là /24).

SNAT (*Source NAT*): thay đổi địa chỉ IP nguồn trong gói tin, thường dùng để các máy trong mạng nội bộ truy cập Internet thông qua một địa chỉ IP công khai duy nhất.

DNAT (*Destination NAT*): thay đổi địa chỉ IP đích trong gói tin, thường dùng để công bố dịch vụ nội bộ ra bên ngoài (ví dụ ánh xạ Floating/Public IP tới máy chủ nội bộ).

Network Types & Segmentation

Type (L2 Network)	Where / Scope	Design Notes (simple)
VLAN (802.1Q)	On-prem / provider edge; ToR–server; SR-IOV	Classic L2 segment; ∼4094 IDs; trunking; a *Network* can contain multiple subnets.
VXLAN (UDP/4789)	Cloud overlays (OpenStack-/OVN, etc.); L3 ECMP underlay	L2-over-L3; VNI per tenant/segment; scales broadcast domains; needs MTU headroom.
Geneve (UDP/6081)	Modern DC/SDN (NSX-T, OVN)	Like VXLAN but extensible (TLVs); common in OVN/NSX pipelines.

Thuật ngữ: Geneve (Generic Network Virtualization Encapsulation,IETF (RFC 8926, 2020)): Cơ chế đóng gói mạng ảo Geneve

Encapsulation Headers (well-known protocols)

Protocol	Header fields	Notes / Use case
GRE (Generic Routing Encapsulation, RFC 2784)	Flags, Protocol Type	Simple L3 tunneling; no security; overhead $\approx$ 24 B.
VXLAN (Virtual eXtensible LAN, RFC 7348)	8-byte VXLAN header: Flags, `VNI` (24-bit)	UDP encapsulation (default port 4789); maps tenant L2 to underlay L3.
NVGRE (Network Virtualization using GRE)	GRE Key field $\Rightarrow$ `VSID` (24-bit)	Microsoft-centric; less used today; limited ECMP support.
GENEVE (Generic Network Virtualization Encapsulation, RFC 8926)	Base header: Flags, `VNI`; TLV Options (flexible)	Future-proof design; allows metadata extension (security, telemetry).
IPsec ESP (Encapsulating Security Payload, RFC 4303)	SPI, Seq Number, IV, Auth Tag	Adds confidentiality + integrity; used in VPN overlay.

Key idea: Overlay encapsulation adds extra headers on top of underlay IP/UDP to virtualize L2/L3 networks.

Geneve (Generic Network Virtualization Encapsulation) — RFC 8926, 2020

Aspect	Details
Design goals	Unify VXLAN/NVGRE/STT; extensible, common overlay standard for virtual networks.
Encapsulation	Encapsulates Ethernet/IP payload over UDP; default port **6081**.
Header (high level)	Version (2b), Opt Len (6b), Flags (incl. OAM/Critical), Protocol Type (16b), **VNI 24-bit** ($\sim$16M), **Options (TLV)** for metadata/policy/telemetry.
Key advantages	Flexible TLV options; runs on any IP underlay (ECMP friendly); multi-vendor (OVN, NSX-T, K8s/CNI); can carry rich ops/telemetry.
Quick comparison	**VXLAN**: UDP/4789, 24-bit VNI, minimal options. **NVGRE**: GRE, 24-bit VNI. **STT**: TCP/7471. **Geneve**: UDP/6081, 24-bit VNI + flexible TLVs.

Thuật ngữ: Geneve = Giao thức đóng gói mạng ảo hóa tổng quát: UDP/6081; VNI 24-bit ($\approx$16 triệu). So với VXLAN (UDP/4789) và NVGRE (GRE), Geneve hỗ trợ tuỳ chọn TLV linh hoạt để mang metadata /policy /telemetry.

End Host (Network Endpoint)

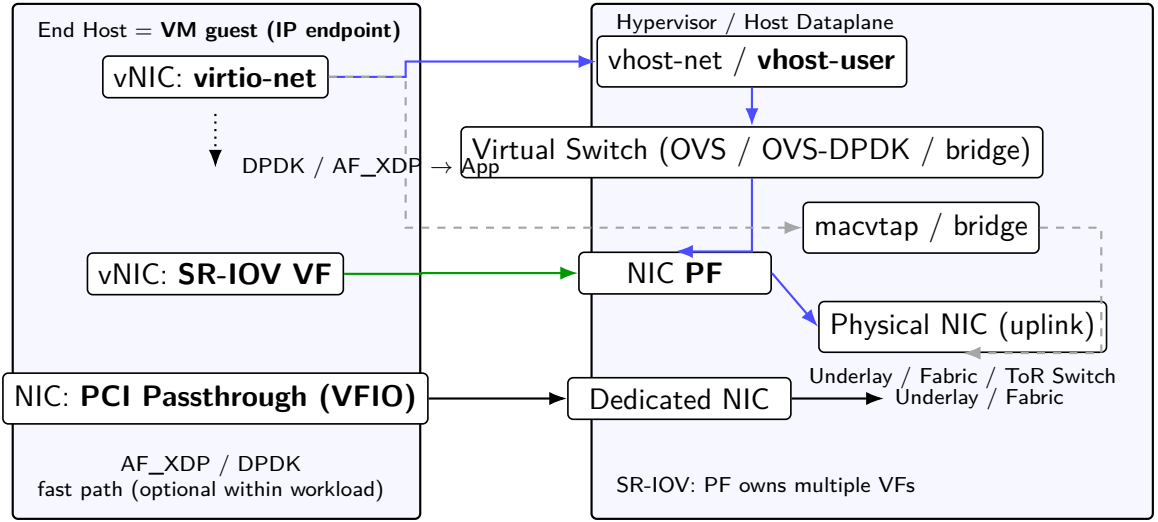
- **Definition:** IP-terminating endpoint: **physical host (bare metal)** or **VM guest**. *Not* containers/pods, hypervisors, or switches.
- **Role:** Source/sink of packets; owns L2 interface(s) and L3 IP address(es); attaches to an access port or vSwitch.
- **Context:** In virtualization, the *VM* is the end host; the hypervisor is infrastructure. (Pod/container networking is covered separately in port-based orchestration.)
- **Data-plane acceleration:** virtio-net, vhost-user (OVS-DPDK), DPDK/vDPA; **SR-IOV VFs** for near line-rate into a VM.
- **Multi-homing:** Multiple pNICs and/or per-VM **vNICs** bound to different VLANs/subnets as needed.

Thuật ngữ: End Host = nút mạng có IP: máy chủ vật lý hoặc máy ảo; *không* phải pod/container. Tăng tốc I/O: virtio, vhost-user, DPDK/vDPA, SR-IOV.

vNIC — Virtual Network Interface (virtio / SR-IOV / passthrough)

- **Role:** connect End Host networking stack to virtual/physical NIC. (**vNIC**, **giao diện mạng ảo**).
- **Techniques:**
  - **virtio (+ vhost/vhost-user)**: para-virtualized device; good balance of performance and manageability (works with OVS, vSwitches).
  - **SR-IOV (Virtual Functions)**: direct hardware VFs to VMs/containers — near-line rate throughput and low CPU overhead; limited by VF count and mobility constraints.
  - **PCI Passthrough (VFIO)**: full device ownership for a VM — maximal performance, minimal sharing.
  - **macvtap / bridge modes**: simple host bridging; easy but limited for advanced features.
  - **AF_XDP / XDP / DPDK paths**: for user-space accelerated packet I/O in workloads (via vhost/DPDK integration).
- **Trade-offs / Guidance:**
  - Use SR-IOV for throughput-sensitive, latency-sensitive tenants when VF count and orchestration constraints are acceptable.
  - Use virtio + vhost-user for flexibility (live migration, centralized vSwitch) and when advanced switching (OVS, vxlan/geneve offloads) is needed.
  - Consider DPDK paths when building purpose-built NFV data plane appliances.

vNIC — Virtual Network Interface (virtio / SR-IOV / passthrough)



Port — Policy Switching (SG / QoS / Anti-spoof)

- **Role:** policy enforcement point at attachment (security groups, QoS, anti-spoof). (**port**, **cổng ảo**).
- **Techniques:**
  - **Virtual Switches:** Open vSwitch (OVS), OVS-DPDK for high throughput switching and flow programming.
  - **eBPF / XDP-based datapaths (Cilium, etc.)**: implement L3/L4/L7 policy in kernel with high scale and observability, minimal packet copies.
  - **Linux netfilter / nftables / iptables / tc**: traditional host-level packet filtering and shaping.
  - **Hardware offloads:** TSO/GSO, checksum offload, flow director; use NIC offload features carefully (SR-IOV/virtio interactions).
  - **Anti-spoofing / source validation:** implement port-security (mac/ip binding), reverse-path filtering, and SG policy enforcement at vSwitch/NIC.
- **Guidance:**
  - For cloud-scale multi-tenant: prefer eBPF/CNI solutions for policy at pod-level (Cilium) and OVS/OVS-DPDK for legacy OpenStack-style setups.
  - Enforce anti-spoofing at the earliest possible point (vNIC/vSwitch) to reduce lateral attack surface.

Subnet (L2 domain) — CIDR, GW, DNS

- **Role:** logical L2 domain and addressing scope. (**subnet**, **mạng con**).
- **Techniques / Patterns:**
  - **Overlay networks:** GENEVE (preferred for extensibility), VXLAN — encapsulate L2/L3 across L3 underlay.
  - **Underlays:** EVPN + BGP as control plane for VXLAN/Geneve (EVPN-VXLAN / EVPN-GENEVE) to support large fabrics and multitenancy.
  - **DNS DHCP orchestration:** integrate IPAM with orchestration (Neutron, CNI IPAM) for predictable addressing.
  - **MTU planning:** adjust MTU for encapsulation overhead (overlay header); avoid fragmentation.
- **Guidance:**
  - Use **Geneve** when you need TLV options / metadata in headers (telemetry, security tags). Ensure underlay supports required latency and path MTU.
  - For on-prem DC fabrics, prefer EVPN control plane for scale and multi-tenancy.

Router (VR / NAT Gateway) — Inter-Subnet, SNAT / DNAT

- **Role:** L3 routing, NAT (SNAT/DNAT), edge services, VRF/tenant isolation. (**router**, **cổng NAT**).
- **Techniques:**
  - **Software routers:** Linux kernel routing (iproute2, nftables), FRRouting (FRR) for full BGP/OSPF/EVPN control-plane.
  - **High-performance dataplane:** DPDK, VPP (FD.io) for fast NAT forwarding in NFV appliances.
  - **Distributed NAT models:** kube-proxy variants / eBPF-based Service NAT (scale to large clusters).
  - **Stateful connection tracking:** conntrack for SNAT/DNAT; optimize conntrack table size and timeouts for scale.
- **Guidance:**
  - For control-plane-rich fabrics (BGP/EVPN), use FRR as a routing daemon integrated with orchestration.
  - For very high throughput NAT, use DPDK/VPP appliances or offload NAT to smart NICs where possible.
  - Monitor conntrack pressure and use connection-affinity / consistent hashing for load-balanced flows.

Network (L3 / External) — Internet / WAN / Load Balancing

- **Role:** external connectivity, ingress/egress policies, load balancing. (**mạng ngoài**, **Internet / WAN**).
- **Techniques:**
  - **Load balancers:** HAProxy/nginx for L7; MetalLB (BGP/Layer-2) for on-prem Kubernetes external IPs.
  - **Edge routing / interconnection:** BGP peering, multi-homing, EVPN interconnects.
  - **Service publishing / DNAT:** Floating IPs / Elastic IPs mapped to internal services with DNAT + LB.
  - **Security at edge:** WAF, DDoS protection, ACLs on edge routers or cloud provider gateways.
- **Guidance:**
  - For bare-metal K8s, MetalLB in BGP mode gives production-grade advertiseable IPs.
  - Ensure edge and underlay routing are coordinated (BGP / route reflectors / peering policies).

Path: End Host → vNIC → Port → Subnet (L2) → Router (VR/NAT) → Network (L3)

Path element	OpenStack	AWS	Azure	GCP	NSX-T
End Host (compute)	Server (Nova)	EC2	VM	VM	VM
vNIC	virtio/SR-IOV	ENI attach	NIC	nic0/nic1	vNIC
Port (attach point)	**Neutron Port**	**ENI**	**NIC**	NIC on instance	**Logical Port**
Subnet (L2 + IP pool)	Subnet (CIDR)	Subnet (VPC)	Subnet (VNet)	Subnet (VPC)	Segment + IP pool
Router / NAT	Neutron Router (+ SNAT/DNAT)	Route Table + IGW/NAT GW	Route Table + NAT GW/Public IP	Default IGW + Cloud NAT/Public IP	Tier-1 Router (+ NAT) → Tier-0
Network (L3/External)	Provider/External + Floating IP	VPC + Elastic IP	VNet + Public IP	VPC + External IP	NSX fabric / Edge uplink

Thuật ngữ: **CIDR** (khối IP), **SNAT** (đổi IP nguồn), **DNAT** (đổi IP đích). Cổng mặc định (gateway) nằm trên **Router/VR** của Subnet.

Hands-on: OpenStack (Neutron)

```
# L2 + L3
openstack network create netA
openstack subnet create --network netA --subnet-range 10.0.1.0/24 \
  --gateway 10.0.1.1 --dns-nameserver 8.8.8.8 netA-sub
openstack router create rA
openstack router set rA --external-gateway public
openstack router add subnet rA netA-sub

# Port + VM + reachability
openstack port create --network netA --fixed-ip ip-address=10.0.1.10 vm1-port
openstack server create --image ubuntu --flavor m1.small \
  --nic port-id=$(openstack port show -f value -c id vm1-port) vm1
openstack security group rule create --proto tcp --dst-port 22 default
openstack security group rule create --proto icmp default
openstack floating ip create public
openstack server add floating ip vm1 <FIP>
```

Thuật ngữ: Floating IP gắn qua **router** và **provider network**. Ingress/Egress đều đi qua Neutron router.

Hands-on: AWS (VPC) – Network Setup

```
VPC_ID=$(aws ec2 create-vpc --cidr-block 10.0.0.0/16 \
  --query Vpc.VpcId --output text)
SUBNET_ID=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block 10.0.1.0/24 \
  --query Subnet.SubnetId --output text)
IGW_ID=$(aws ec2 create-internet-gateway \
  --query InternetGateway.InternetGatewayId --output text)
aws ec2 attach-internet-gateway --vpc-id $VPC_ID --internet-gateway-id $IGW_ID
RTB_ID=$(aws ec2 create-route-table --vpc-id $VPC_ID \
  --query RouteTable.RouteTableId --output text)
aws ec2 create-route --route-table-id $RTB_ID \
  --destination-cidr-block 0.0.0.0/0 --gateway-id $IGW_ID
aws ec2 associate-route-table --route-table-id $RTB_ID --subnet-id $SUBNET_ID
SG_ID=$(aws ec2 create-security-group --group-name sgA \
  --description sgA --vpc-id $VPC_ID --query GroupId --output text)
```

Thuật ngữ: Thiết lập VPC, Subnet, IGW, Route Table và Security Group. Đây là phần hạ tầng mạng (network foundation).

Hands-on: AWS (VPC) – Instance Public IP

```
aws ec2 authorize-security-group-ingress --group-id $SG_ID \
  --protocol tcp --port 22 --cidr 0.0.0.0/0

INSTANCE_ID=$(aws ec2 run-instances --image-id ami-xxxx \
  --instance-type t3.micro \
  --subnet-id $SUBNET_ID --security-group-ids $SG_ID \
  --query Instances[0].InstanceId --output text)

ALLOC_ID=$(aws ec2 allocate-address --query AllocationId --output text)
aws ec2 associate-address --instance-id $INSTANCE_ID \
  --allocation-id $ALLOC_ID
```

Thuật ngữ: Elastic IP (**EIP**) gắn trực tiếp **NIC/instance**. Internet Gateway (**IGW**) cho ingress/egress; NAT Gateway dùng khi egress từ private subnet.

Hands-on: Azure (VNet)

```
az group create -n rg-netA -l eastus
az network vnet create -g rg-netA -n vnetA --address-prefix 10.0.0.0/16 \
  --subnet-name subA --subnet-prefix 10.0.1.0/24
az network public-ip create -g rg-netA -n pipA --sku Standard
az network nic create -g rg-netA -n nicA --vnet-name vnetA \
  --subnet subA --public-ip-address pipA
az network nsg create -g rg-netA -n nsgA
az network nsg rule create -g rg-netA --nsg-name nsgA -n allow-ssh \
  --priority 1000 --access Allow --protocol Tcp --direction Inbound \
  --destination-port-ranges 22
az network nic update -g rg-netA -n nicA --network-security-group nsgA
az vm create -g rg-netA -n vm1 --image UbuntuLTS --size Standard_B1s \
  --nics nicA --admin-username azureuser --generate-ssh-keys
```

Thuật ngữ: Public IP (**PIP**) gắn trực tiếp vào **NIC** hoặc Load Balancer. Outbound có thể đi qua **NAT Gateway** hoặc Load Balancer.

Hands-on: GCP (VPC)

```
gcloud compute networks create vpc-a --subnet-mode=custom
gcloud compute networks subnets create sub-a --network=vpc-a \
  --range=10.0.1.0/24 --region=us-central1
gcloud compute firewall-rules create allow-ssh --network=vpc-a \
  --allow tcp:22 --direction=INGRESS --source-ranges=0.0.0.0/0
gcloud compute instances create vm1 --zone=us-central1-a --subnet=sub-a \
  --image-family=ubuntu-2204-lts --image-project=ubuntu-os-cloud

# Private-only egress (Cloud NAT):
# gcloud compute routers create r-a --network=vpc-a --region=us-central1
# gcloud compute routers nats create nat-a --router=r-a \
#   --auto-allocate-nat-external-ips --region=us-central1 \
#   --nat-all-subnet-ip-ranges
```

Thuật ngữ: External IP gắn trực tiếp vào **NIC**. Cloud Router + **Cloud NAT** dùng cho egress từ subnet riêng tư.

NSX-T quick steps & cross-cloud gotchas

NSX-T (conceptual)

1. Create Segment `seg-A` (Geneve), IP pool/DHCP `10.0.1.0/24`.
2. Tier-1 Router `t1-A`: attach `seg-A` (GW `10.0.1.1`).
3. Connect `t1-A` to Tier-0 (Edge uplink) for external reachability.
4. NAT on `t1-A` (SNAT/DNAT) as needed.
5. Attach VM vNIC to `seg-A` (Logical Port) $\rightarrow$ gets IP.

Cross-cloud sanity checks

- Overlay MTU headroom (VXLAN/Geneve) — set underlay MTU ≥ 1600 when possible.
- Default gateway resides on the Router/VR interface of the Subnet.
- Security Groups (stateful at Port/NIC) vs. edge firewalls/NACLs (often stateless).
- Multi-NIC hosts: add routes/policies to avoid asymmetric return paths.

Thuật ngữ: Geneve (UDP/6081), VXLAN (UDP/4789). MTU cần headroom cho encapsulation. Gateway của Subnet nằm trên Router/VR.

DHCP & IPv6

- **Subnet (mạng con):** defines **CIDR**, default gateway (**cổng mặc định**), DNS scope.
  - Logical L2/L3 boundary for tenant networks.
  - Overlay (VXLAN/GENEVE) or underlay EVPN as transport.
  - **MTU planning:** adjust for encapsulation overhead to avoid fragmentation.
- **DHCP (Dynamic Host Configuration Protocol):**
  - DHCPv4: leases IP, mask, router, DNS to VMs/pods. (**DHCP cấp phát IP**).
  - Orchestrators (OpenStack Neutron, Kubernetes CNI plugins) usually run DHCP agent or IPAM controllers.
  - Stateless workloads: IPAM integration with orchestration for predictable assignment.
- **IPv6:**
  - **SLAAC (Stateless Address Auto-Config)**: Router Advertisements (RA) provide prefix + default route. (**SLAAC, RA**).
  - **DHCPv6**: stateful leases; better for enterprise-style tracking. (**DHCPv6**).
  - Must **allow ICMPv6/Neighbor Discovery (ND)** in security groups/firewall; otherwise IPv6 breaks. (**ICMPv6/ND**).
  - **RA Guard / DHCPv6 Shield**: prevent rogue RA/DHCPv6 servers in tenant networks.

DHCP & IPv6

- **Metadata services (dịch vụ metadata):**
  - Provide per-instance config (user-data, SSH keys) for cloud-init, ignition, or agent.
  - Delivered via link-local IP (169.254.169.254 for IPv4, `fe80::/64` scoped for IPv6).
  - Security: isolate metadata path (e.g., Neutron metadata proxy) to avoid tenant cross-access.
- **Guidance:**
  - Use **SLAAC** for lightweight, scalable IPv6 in containerized environments.
  - Use **DHCPv6** when auditability / IP control is needed (e.g., enterprise clouds).
  - Always test MTU across overlay networks; ensure DHCP relay / RA forwarding works in virtual routers.

Router, NAT & Public Addressing (Cross-cloud view) – Part 1

- **Role of Router (Bộ định tuyến ảo):**
  - Provides L3 connectivity between subnets (inter-subnet routing).
  - Handles **SNAT** (nguồn) for outbound, **DNAT** (đích) for inbound traffic.
  - Policy: ACLs, security groups, distributed firewall integration.
- **OpenStack (Neutron Router):**
  - Router attaches tenant subnet to provider/external network.
  - **Floating IP (FIP)**: Public IP mapped to private IP via DNAT.
  - SNAT: tenant workloads share router's external IP for outbound Internet.
  - NAT performed in **network node** or **DVR (Distributed Virtual Router)** for scale.
- **AWS:**
  - **Internet Gateway (IGW)**: attaches VPC to Internet.
  - **Elastic IP (EIP)**: static public IPv4, bound to ENI (Elastic Network Interface) or directly to instance.
  - NAT Gateway/Instance for private subnet egress.
  - Security: SG + NACL enforced at VPC level.

Router, NAT & Public Addressing (Cross-cloud view) – Part 2

- **Azure:**
  - **Public IP (PIP)** resource: can bind to NIC, VM, or Load Balancer.
  - **User-Defined Routes (UDR)**: customize routing beyond system routes.
  - Outbound SNAT: via Azure NAT Gateway or Load Balancer.
  - Inbound DNAT: via LB rules or PIP on NIC.
- **Google Cloud Platform (GCP):**
  - **External IP**: assigned to VM NIC (ephemeral or static).
  - **Cloud NAT**: scalable, managed SNAT for private subnets.
  - **Cloud Router**: dynamic routing (BGP) for hybrid / multi-cloud.
  - HA setup with redundant Cloud Routers for resilient connectivity.
- **Key Guidance:**
  - **Cloud-native NAT** scales better than DIY VM NAT.
  - Use **public IP binding models** that match workload needs (per-VM vs load-balanced).
  - Hybrid/multi-cloud: integrate **dynamic routing (BGP, Cloud Router/Transit Gateway)** to avoid static route sprawl.
  - Security: minimize DNAT exposure, rely on LB + firewall policies.

Security Services & Policy Layers – Part 1

- **Port-level firewall (tường lửa cấp cổng):**
  - **OpenStack:** Security Groups (SG) – **stateful**, per port/vNIC.
  - **Azure:** Network Security Groups (NSG) – same concept.
  - **GCP:** VPC firewall rules – stateful, applied at NIC/tag/project.
  - **AWS:** Security Groups (SG) – stateful, attached to ENI.
  - Use case: **microsegmentation** (**phân đoạn vi mô**) for east–west traffic.
- **Edge / subnet-level filter (lọc gói ở rìa mạng):**
  - Stateless filtering – no connection tracking.
  - **OpenStack:** rare, sometimes via NFV firewall.
  - **Azure:** Network Virtual Appliance (NVA) + User-Defined Routes (UDR).
  - **GCP:** hierarchical firewall + org policy.
  - **AWS:** NACL – stateless, subnet-level, explicit allow/deny.
  - Use case: coarse-grained filtering, compliance, DMZ (**vùng đệm an ninh**).

Security Services & Policy Layers – Part 2

- **Bandwidth control (điều tiết / QoS):**
  - **OpenStack:** QoS policies at port/network.
  - **Azure:** shaping via Load Balancer tiers, QoS in ExpressRoute.
  - **GCP:** egress bandwidth caps per VM or project.
  - **AWS:** shaping via LB/WAF tiers, instance policing.
  - Use case: tenant fairness, SLA enforcement, cost control.
- **Modern practices:**
  - Shift to **Zero Trust** – enforce SG/NSG/firewall on all paths.
  - **Distributed firewalls (DFW):** hypervisor/CNI-level enforcement.
  - **Service mesh (lưới dịch vụ):** L7 policy, mTLS, per-service rules.
  - Integration with **SIEM/SOAR** for monitoring & automation.

Thuật ngữ: SG/NSG/GCP firewall (**có trạng thái**), NACL (**không trạng thái**), QoS (**điều tiết**)

.

Network Lifecycle Manager

- **Network service (API)**: **Neutron (OpenStack)**, **VPC (AWS/GCP)**, **VNet (Azure)**.
- Express desired state: networks, subnets, route/IGW, SG/NSG/NACL, public IPs.
- Automate with **Ansible** collections: `openstack.cloud`, `amazon.aws`, `azure.azcollection`, `google.cloud`.

Thuật ngữ: "Dịch vụ mạng": Neutron/VPC/VNet; mô tả trạng thái mong muốn; tự động hoá bằng Ansible.

Ansible — OpenStack Example (excerpt)

```
- hosts: localhost
  collections: [openstack.cloud]
  tasks:
    - os_network: { name: "net-app" }
    - os_subnet: { name: "sub-app", network_name: "net-app",
                   cidr: "10.50.0.0/24", dns_nameservers: ["1.1.1.1","8.8.8.8"] }
    - os_router: { name: "rt-core", network: "public" }
    - os_router_interface: { router: "rt-core", subnet: "sub-app" }
    - os_security_group: { name: "sg-web" }
```

Thuật ngữ: Ansible OpenStack: tạo network, subnet, router, giao diện, SG.

Ansible — AWS VPC Example (excerpt)

```
- hosts: localhost
  collections: [amazon.aws]
  tasks:
    - ec2_vpc_net: { name: "vpc-app", cidr_block: "10.60.0.0/16" }
    - ec2_vpc_subnet: { vpc_id: "{{ vpc_id }}", cidr: "10.60.1.0/24" }
    - ec2_vpc_igw: { vpc_id: "{{ vpc_id }}", state: "present" }
    - ec2_vpc_route_table: { vpc_id: "{{ vpc_id }}", routes: [{dest: "0.0.0.0/0", gateway_id
        : "{{ igw_id }}"}] }
    - ec2_security_group: { name: "sg-web", vpc_id: "{{ vpc_id }}", description: "web allow"
        }
```

Thuật ngữ: Ansible AWS: VPC, Subnet, IGW, Bảng định tuyến, SG.

Ansible — Azure VNet Example (excerpt)

```
- hosts: localhost
  collections: [azure.azcollection]
  tasks:
    - azure_rm_virtualnetwork: { resource_group: "rg-app", name: "vnet-app",
        address_prefixes: ["10.70.0.0/16"] }
    - azure_rm_subnet: { resource_group: "rg-app", name: "sub-app", address_prefix: "
        10.70.1.0/24", virtual_network: "vnet-app" }
    - azure_rm_networksecuritygroup: { resource_group: "rg-app", name: "nsg-web" }
    - azure_rm_publicipaddress: { resource_group: "rg-app", name: "pip-web",
        allocation_method: "Static" }
```

Thuật ngữ: Ansible Azure: VNet, Subnet, NSG, Public IP (PIP).

Ansible — GCP VPC Example (excerpt)

```
- hosts: localhost
  collections: [google.cloud]
  tasks:
    - gcp_compute_network: { name: "vpc-app", auto_create_subnetworks: "false" }
    - gcp_compute_subnetwork: { name: "sub-app", ip_cidr_range: "10.80.1.0/24", region: "
        asia-southeast1", network: "vpc-app" }
    - gcp_compute_firewall: { name: "fw-web", network: "vpc-app", allowed: [{ ip_protocol: "
        tcp", ports: ["80","443"] }] }
    - gcp_compute_address: { name: "eip-web", region: "asia-southeast1" }
```

Thuật ngữ: Ansible GCP: VPC, Subnet, Firewall, External IP.

Lab Aligned to Mapping Path

1. Create **Network** + **Subnet** (CIDR, DNS).
2. Create **Router** + set external gateway / IGW / default route.
3. Create **Port/NIC** and attach VM vNICs.
4. Configure **SG/NSG/Firewall** (baseline + app rules).
5. Allocate **Floating/Public/External IP** and bind to NIC/port.
6. Verify end-to-end reachability (ICMP/SSH/HTTP).

Thuật ngữ: Lab theo ánh xạ: Mạng/Subnet → Router/GW → Cổng/NIC → SG/NSG → IP công khai → Kiểm thử.

Validation Matrix (Evidence)

Check	Command/Evidence
Network/Subnet	`openstack network/subnet list` or cloud portal/API
Router/Route/IGW	`openstack router show`, AWS route table, Azure UDR, GCP routes
Port/NIC binding	`openstack port list -server <vm>`, ENI/NIC details
Security policy	SG/NSG/Firewall rule listings
Public IP	FIP/EIP/PIP/External IP assignment
MTU path	`ping -M do -s 1400`, `tracepath` evidence

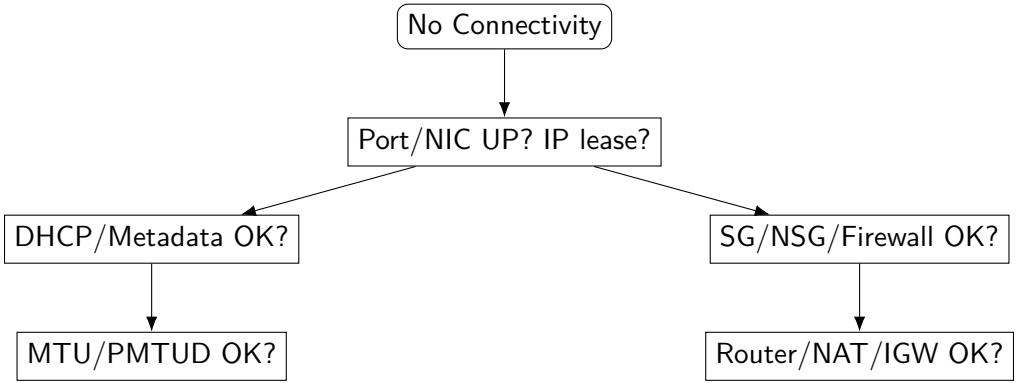
Thuật ngữ: Ma trận kiểm chứng: cấu hình mạng, cổng/NIC, chính sách bảo mật, IP công khai, MTU.

Troubleshooting Checklist (Ordered by Path)

1. Port/NIC bound? IP lease present? Metadata reachable?
2. Router/IGW/default route correct? NAT path valid?
3. SG/NSG/Firewall rules direction/stateful expectations correct?
4. MTU/PMTUD/ICMP? Any "Frag Needed" blocked?
5. Provider network constraints (quotas/limits)?

Thuật ngữ: Chẩn đoán theo thứ tự: Cổng/NIC → Router/NAT → SG/NSG → MTU/ICMP → Giới hạn nhà cung cấp.

Troubleshooting Decision Tree (Sketch)



Thuật ngữ: Cây quyết định: Cổng/NIC → DHCP/SG → MTU → NAT/IGW.

Cross-Cloud Control-Plane Mapping

Concept	**OpenStack**	**Azure**	**VMware**	**AWS**	**GCP**
Compute	Nova	Azure VM	vCenter	EC2	GCE
Network	Neutron	VNet	NSX	VPC	VPC
Security	SG	NSG	DFW	SG / NACL	Firewall
Images	Glance	Templates	Templates	AMI	Images
Block	Cinder	Managed Disks	vSAN	EBS	Persistent Disk

Thuật ngữ: Ánh xạ mặt phẳng điều khiển: Compute, Network, Security, Image, Block giữa các nhà cung cấp.

Forensics & Snapshot Handling Tips

- Prefer platform-native snapshots; record IDs/timestamps; preserve chain-of-custody.
- Mount deltas with base (e.g., VMDK delta) or attach snapshots to analysis VMs (EBS/PD).
- Avoid powering on evidence VMs; work on copies; document hash/signature when applicable.

Thuật ngữ: Pháp y: dùng snapshot gốc, ghi nhận ID/giờ; gắn vào máy phân tích; không bật máy gốc; băm/biên bản.