Operational set of **best practices** for designing OpenStack networking for multi-tenant deployments that span multiple data centers (or regions).

**1 — High-level design goals**

When you design multi-tenant networking across DCs/regions, aim for:

* **Isolation:** Tenants must be isolated (L2/L3) unless explicitly shared.
* **Scalability:** Avoid L2 stretching where possible; design an underlay/overlay that scales.
* **Resilience:** No single-point failures for control and data planes.
* **Predictability & observability:** IPAM, quotas, telemetry, and logging.
* **Performance where needed:** SR-IOV, DPDK, trunking for tenants that need it.
* **Operational simplicity:** Clear ownership, automation, and templates.

**2 — Architectural patterns (common topologies)**

**A. Multi-region (cloud region per DC)**

* **Each region = full OpenStack control plane + Neutron deployment.**
* Regions are *independent* — use Keystone regions.
* Inter-region tenant connectivity is **L3** (VPN/SD-WAN/MPLS) or application layer replication.
* **When to use:** clear administrative boundaries, failover, regulatory isolation.

**Pros:** strong isolation, simple local networking.  
**Cons:** no native L2 tenant networks across regions.

**B. Multi-site with shared control plane (centralized control)**

* Single Keystone/glance/catalog across sites; compute and network nodes per site.
* Requires careful network design; generally more complex.

**Use only** if you need central control and can accept complexity.

**C. Multi-DC L2 stretched underlay (not recommended generally)**

* Physically extend L2 across sites (e.g., EVPN-VXLAN underlay that spans sites).
* **Risks:** broadcast domains scale poorly, failure domains larger, operations complex.

**Best practice:** avoid unless you have carrier-grade underlay (EVPN fabrics) and strict operational discipline.

**3 — Underlay / Overlay choices**

**Underlay:**

* Use a robust L3 fabric (BGP EVPN-capable switches) with consistent MTU settings.
* Plan **IP addressing** for management, storage, vtep/vxlan, and external provider networks.

**Overlay:**

* ML2 + driver choices: **OVN (recommended)** or ML2+OVS (OVS), or Linuxbridge for simpler setups.
* **Encapsulation:** prefer **Geneve** (future proof) or VXLAN. Geneve supports metadata/metadata options.
* **Why OVN:** built-in distributed logical routing/switching, simpler scaling and cleaner control plane.

**4 — Tenant network primitives and recommended usage**

* **Network (L2 domain):** create one per tenant need. Use tenant-scoped networks by default.
* **Subnet:** contains IP pool, gateway, DHCP. Integrate with IPAM (NetBox/Infoblox).
* **Router:** L3 connect tenant networks to external/public network via L3 agents. Use **DVR** where possible to distribute SNAT and avoid centralized bottlenecks.
* **Ports:** attach to VMs, support trunks for nested virtualization.
* **Floating IPs:** map public addresses to tenant private IPs — provide through dedicated NAT gateways.
* **Security Groups:** default micro-segmentation; use with host firewall and per-tenant policies.
* **QoS:** set bandwidth/DSCP policies via Neutron QoS.

**5 — Key Neutron services, how they interact**

* **neutron-server (API):** receives user/Horizon/Nova calls → talk to DB and plugin.
* **ML2 plugin + mechanism drivers:** map logical constructs to hosts (OVS/OVN/LinuxBridge).
* **L2 agent (ovs/bridge):** configures local switches, ports, integration with compute.
* **DHCP agent:** runs dnsmasq in network namespaces or centrally in OVN.
* **L3 agent:** creates qrouter namespaces or OVN routers; does NAT for external access.
* **Metadata agent:** proxies instance metadata requests.
* **FWaaS/NAT/Loadbalancer (Octavia):** optional services for firewall/edge load balancing.
* **Interaction with Nova:** Nova requests port creation before spawning VM; Nova uses Neutron to set up networking.
* **Interaction with Keystone:** authentication and service endpoints.
* **Interaction with external routing (BGP):** use FRR/ExaBGP or hardware BGP to advertise routes (esp. with EVPN or DVR SNAT).

**6 — Routing & east-west/west-east connectivity patterns**

* **Use DVR** to keep east-west traffic local on compute nodes and distribute SNAT for north-south.
* **For large multi-tenant clouds:** combine DVR + centralized SNAT for specific public IP needs or use distributed SNAT with per-node NAT when scale allows.
* **BGP Integration:** for scaling public networks & route advertisement, run BGP between OpenStack edge (FRR on network nodes) and physical routers — supports dynamic advertisement of tenant external prefixes and EVPN for underlay.
* **Avoid L2 stretch across DCs** — prefer L3, VPNs, or SD-WAN.

**7 — Addressing, VLAN planning, and IPAM**

* **VLAN pool plan:** reserve VLAN ranges per rack/tenant/vlan-type. Avoid overlapping with provider VLANs.
* **VXLAN/Geneve IDs**: plan large enough ID space for tenants.
* **MTU:** overlay encapsulation increases packet size — set underlay MTU to at least 1600 (or use jumbo frames) to avoid fragmentation.
* **IPAM integration:** use NetBox/Infoblox or the OpenStack IPAM plugin to avoid conflicts.
* **Naming & tagging:** standardize network and subnet names (tenant-net, tenant-subnet, az tags).

**8 — Security best practices**

* **Default-deny security groups** for tenants; only open required ports.
* **Microsegmentation:** use security groups + OVN ACLs (or NSX/Calico with Neutron plugins) for east-west restrictions.
* **Separate management, data, storage networks:** isolate traffic types physically or via VLANs.
* **RBAC & Neutron:** limit who can create provider/shared networks; use policy.json to restrict operations.
* **Audit & logging:** enable Neutron API, agent logs, and flow logs (OVS/OVN or sFlow/IPFIX on physical switches).
* **Limit shared networks:** make provider/shared networks admin-only except where explicitly required.

**9 — High availability & scaling**

* **Control plane HA:** run multiple neutron-server replicas behind load balancer; make DB and message bus HA.
* **Distributed agents:** run L3/DHCP agents on multiple nodes for failover.
* **Scale data plane:** scale ML2/OVS/OVN agents per compute node; use hardware offloads (SR-IOV) for high throughput VMs.
* **Octavia/Loadbalancer:** run Amphora VMs in separate project with autoscaling.
* **Monitoring:** Prometheus + Grafana for agent health, interface stats, tunnel utilization.

**10 — Performance: SR-IOV, DPDK, offloads**

* Use **SR-IOV** or **DPDK** for tenant workloads requiring high throughput/low latency.
* Provide **trunked ports** for nested VMs/containers.
* Plan NIC passthrough policy to ensure security and tenant isolation.

**11 — Operational tools & automation**

* Manage networks via **Infrastructure as Code** (Terraform/Ansible using OpenStack modules).
* Centralize IPAM and use templates for network/subnet/router provisioning.
* Test tenant lifecycle with CI: create network, subnet, router, create VM, attach floating IP, cleanup.

**12 — Multi-region considerations**

* **Regions** are the recommended approach: each region is an independent deployment in Keystone.
* **Do not assume L2 across regions** — instead:
  + Use **L3 VPNs or SD-WAN** for tenant connectivity across sites.
  + Use **application replication** or cross-region services rather than cross-region L2.
* Use a central orchestration layer (Terraform, Juju, Ansible) to keep region configs consistent.

**13 — Disaster recovery & backup**

* Backup Neutron DB and configuration.
* Keep a documented network recovery runbook (how to rebuild routers, reattach floating IPs).
* Test failover scenarios: agent failures, region isolation, DB failover.

**14 — Common pitfalls & how to avoid them**

* **MTU mismatches → fragmentation:** set underlay MTU > overlay overhead.
* **L2 stretch across DCs** causing broadcast storms — prefer L3.
* **Loose provider networks** — accidentally sharing a provider network across tenants. Make provider networks admin-only.
* **Policy.json ignored** (snap-based bundles) — ensure management plane allows policy overrides or use non-snap deployments for heavy customization.
* **Insufficient IPAM** causing address collisions.

**15 — Example recommended stack for production multi-tenant, multi-DC**

* Underlay: L3 BGP EVPN fabric, MTU 1600+.
* Overlay: OVN with Geneve.
* Routing: DVR + FRR for BGP integration on edge nodes.
* Edge/Public: Dedicated edge nodes (NAT/Load balancer) with BGP to provider routers.
* Service HA: multiple neutron-server, HA DB (Galera/PG cluster), RabbitMQ cluster.
* IPAM: NetBox (or Infoblox) + automation.
* Monitoring: Prometheus + Grafana + ELK for logs.

**16 — Practical checklist before deployment**

1. Decide per-region vs per-DC boundaries.
2. Choose ML2 driver (OVN recommended) and encapsulation (Geneve).
3. Design underlay IPs, VLAN ranges, and MTU.
4. Plan external/provider networks and BGP/EVPN strategy.
5. Select IPAM and integrate with workflow.
6. Define policy.json and RBAC for network ops.
7. Plan HA for agents and DB.
8. Implement CI for network lifecycle testing.
9. Test tenant isolation, floating IPs, DVR behavior, and failover.
10. Document runbooks and rollback steps.

**Short example: launching a tenant network (commands)**

# create tenant network

openstack network create tenant-net

# create subnet

openstack subnet create --network tenant-net \

--subnet-range 10.10.10.0/24 tenant-subnet --gateway 10.10.10.1

# create router and attach to external network 'public'

openstack router create tenant-router

openstack router set tenant-router --external-gateway public

openstack router add subnet tenant-router tenant-subnet