Google Cloud - Networking

Virtual Private Cloud (VPC):

What

VPC is Google Cloud's global private network. It provides an isolated, software-defined network where you can deploy resources securely.

Why (Use Cases)

• **Host Compute Engine, GKE, and App Engine resources.**
– Every workload in GCP runs inside a VPC, making it the foundation of networking.

• **Segment workloads into private, controlled networks.**
– Helps isolate dev, staging, and prod environments or separate departments for security.

• **Provide hybrid connectivity to on-premises systems.**
– VPC can connect to your data center via VPN or Interconnect.

Key Features

• **Global VPC spanning multiple regions.**
– Unlike AWS, GCP VPCs are global, so you don't need to manage multiple regional networks.

• **Subnets defined at regional level.**
– You allocate IP ranges region by region for better control.

• **Private IP ranges (RFC 1918).**
– Ensures workloads use private IPs, not exposed to the internet by default.

• **Default VPC auto-created with subnets in each region.**
– Projects start with a ready-made VPC to simplify initial setup.

• **Peering and Shared VPC supported.**
– Enables secure project-to-project connectivity.

Pricing Model

• **No charge for VPC creation.**
– The network itself is free.

• **Pay for egress traffic (between zones, regions, or internet).**
– Only data leaving a zone or region is billed.

Limitations

• **Cannot extend across organizations.**
– Each VPC is tied to one org/project.

• **Peering has no transitive routing.**
– If A peers with B and B peers with C, A cannot reach C unless directly peered.

Integration

• **Compute Engine, GKE, App Engine.**
– All workloads attach to a VPC.

• **Cloud Interconnect and Cloud VPN for hybrid connections.**
– Extend to on-premises networks.

Best Practices

• **Use custom mode VPC for production.**
– Avoid IP conflicts in default mode.

• **Segment workloads with separate subnets.**
– Logical isolation improves security.

• **Use labels and firewall rules for security.**
– Organize and enforce policies consistently.

Subnets:

What

Subnets divide a VPC into regional IP ranges for resource placement.

Why (Use Cases)

• **Control IP addressing and resource distribution.**
– Allocate IPs per environment, team, or workload.

• **Apply firewall rules at subnet level.**
– Enforce security policies on all resources in that subnet.

Key Features

• **Regional (not zonal).**
– Subnets work across zones in the same region.

• **Can span across multiple zones.**
– Gives resilience if one zone fails.

• **IP ranges defined using CIDR blocks.**
– Allows precise network design (e.g., /24, /16).

• **Support for secondary IP ranges (for GKE).**
– Essential for Kubernetes Pod and Service IPs.

Pricing Model

• **No cost for subnets themselves.**
– Free to create; traffic costs apply.

• **Charged for egress traffic across regions.**
– Moving data between regions incurs charges.

Limitations

• **Cannot change subnet region after creation.**
– Region is fixed once created.

Integration

• **GKE Pods (secondary IP ranges).**
– Needed for container networking.

• **Compute Engine instances.**
– VMs pull IPs from subnets.

Best Practices

• **Plan CIDR ranges carefully to avoid overlap.**
– Prevents issues when peering.

• **Use secondary ranges for containerized workloads.**
– Keeps Kubernetes IPs separate from VM IPs.

Firewall Rules:

What

Firewall rules control ingress and egress traffic in a VPC.

Why (Use Cases)

• **Secure workloads from unauthorized access.**
– Protects against unwanted internet traffic.

• **Allow only necessary ports/services.**
– For example, only open 80/443 for web servers.

Key Features

• **Stateful rules.**
– Responses are automatically allowed.

• **Default rules (allow internal, deny external).**
– Baseline protection for every new VPC.

• **Priority-based enforcement (0 = highest).**
– Rules are evaluated in order of priority.

• **Can filter by tags or service accounts.**
– Apply rules dynamically to groups of VMs.

Pricing Model

• **No direct cost; billed for traffic allowed/blocked.**
– Rules themselves are free.

Limitations

• **Not transitive across peered VPCs.**
– Each VPC enforces its own rules.

Integration

• **Works with Compute Engine, GKE.**
– Protects VMs and containers.

• **IAM can control who manages firewall rules.**
– Ensures governance.

Best Practices

• **Use least-privilege rules.**
– Open only what's needed.

• **Deny all ingress by default; open only what's needed.**
– Improves security posture.

Cloud Load Balancing:

What

Cloud Load Balancing is a fully managed software-defined load balancer that distributes traffic across instances globally or regionally.

Why (Use Cases)

• **Distribute HTTP(S), TCP/UDP traffic.**
– Ensures fair distribution.

• **Provide high availability and scalability.**
– Reroutes to healthy backends automatically.

• **Global app delivery with single anycast IP.**
– Simplifies global DNS management.

Key Features

• **Global and regional load balancing.**
– Match scope to your workload.

• **Anycast IP for global access.**
– Directs users to nearest healthy backend.

• **Supports SSL offloading.**
– Frees backend resources.

• **Integrated with autoscaling.**
– Handles sudden spikes in traffic.

Pricing Model

• **Billed for forwarding rules, data processed, and SSL certificates.**
– Based on configuration and usage.

Limitations

• **Some types (like Internal LB) are regional only.**
– Not all are global.

Integration

• **Works with MIGs, GKE Services.**
– Backends scale automatically.

• **Can front App Engine, Cloud Run.**
– Extends serverless to enterprise traffic.

Best Practices

• **Use health checks for backend validation.**
– Sends traffic only to healthy backends.

• **Use HTTPS load balancer for global traffic.**
– Adds security and reach.

Cloud CDN:

What

Cloud CDN caches web and media content at Google's edge locations for faster delivery.

Why (Use Cases)

• **Reduce latency for global users.**
– Brings content closer to them.

• **Serve static content like images, videos, APIs.**
– Reduces origin load.

Key Features

• **Integrated with HTTPS Load Balancing.**
– Easy to enable.

• **Over 150 edge points of presence.**
– Wide coverage.

• **Cache invalidation support.**
– Purge outdated objects instantly.

• **Signed URLs and signed cookies for access control.**
– Secure premium content.

Pricing Model

• **Pay for cache egress and cache fill.**
– Hits save money, misses cost backend egress.

Limitations

• **Only works with HTTPS Load Balancer.**
– Requires LB pairing.

Integration

• **Compute Engine, Cloud Storage buckets.**
– Deliver VM or object data faster.

• **App Engine, Cloud Run.**
– Cache serverless responses.

Best Practices

• **Use signed URLs for secure content delivery.**
– Prevents unauthorized access.

• **Set appropriate cache-control headers.**
– Controls freshness and reduces cost.

Cloud Interconnect:

What

Provides dedicated or partner-managed private connectivity between on-premises and Google Cloud.

Why (Use Cases)

• **High-bandwidth, low-latency workloads.**
– Suitable for heavy data transfer.

• **Regulatory compliance avoiding internet routing.**
– Meets strict industry standards.

Key Features

• **Dedicated Interconnect: 10–100 Gbps links.**
– Direct high-speed connection.

• **Partner Interconnect: 50 Mbps–10 Gbps via providers.**
– Lower entry barrier through partners.

• **SLA-backed availability.**
– Guaranteed uptime.

Pricing Model

• **Port charges + egress usage fees.**
– Based on capacity and data sent.

Limitations

• **Physical setup required for Dedicated Interconnect.**
– Needs colocation.

Integration

• **Hybrid cloud deployments.**
– Extends on-prem into GCP.

• **Works with Shared VPC and VPN.**
– Can combine for redundancy.

Best Practices

• **Use Partner Interconnect for quick setup.**
– Faster to deploy.

• **Reserve Dedicated Interconnect for mission-critical workloads.**
– Ensures consistent performance.

Cloud VPN:

What

Cloud VPN creates an IPSec-encrypted tunnel between your on-premises network and Google Cloud.

Why (Use Cases)

• **Secure connectivity over public internet.**
– Protects data in transit.

• **Quick hybrid cloud setup.**
– Fast to configure.

Key Features

• **Classic VPN (single tunnel, 99.9% SLA).**
– Entry-level.

• **HA VPN (dual tunnel, 99.99% SLA).**
– Redundant and reliable.

• **Dynamic routing with BGP support.**
– Adapts to network changes.

Pricing Model

• **Billed for tunnel and egress traffic.**
– Tunnel uptime + data volume.

Limitations

• **Latency depends on internet quality.**
– Not as reliable as Interconnect.

Integration

• **Works with VPC, Interconnect for hybrid setups.**
– VPN + Interconnect = redundancy.

Best Practices

• **Use HA VPN for production.**
– Ensures high uptime.

• **Prefer Interconnect for consistent performance.**
– Use VPN as backup.

VPC Peering:

What

VPC Peering allows private connectivity between two VPC networks.

Why (Use Cases)

• **Connect workloads in separate projects or organizations.**
– Enables communication without public IPs.

Key Features

• **Traffic stays on Google's private network.**
– Secure and fast.

• **No bandwidth bottleneck.**
– Same performance as internal traffic.

• **Simple setup without gateways.**
– No routers needed.

Pricing Model

• **No charge for peering itself.**
– Free to create.

• **Charged for egress if crossing regions.**
– Data transfer billed regionally.

Limitations

• **No transitive peering.**
– Must peer directly.

• **Cannot apply firewall rules across peers.**
– Each VPC enforces its own.

Integration

• **Connect Dev/Prod environments.**
– Allows controlled cross-project access.

• **Multi-project architectures.**
– Common in enterprises.

Best Practices

• **Avoid overlapping CIDR ranges.**
– Prevents conflicts.

• **Use Shared VPC for larger orgs.**
– Easier to manage.

Shared VPC:

What

Shared VPC lets multiple projects share a centralized VPC network.

Why (Use Cases)

• **Centralize networking and security for large organizations.**
– Simplifies management.

• **Isolate environments by project but use a single VPC.**
– Combines separation and centralization.

Key Features

• **One host project shares subnets with service projects.**
– Centralized subnets.

• **IAM used to grant project-level permissions.**
– Fine-grained control.

• **Centralized firewall and routing policies.**
– Enforced consistently.

Pricing Model

• **No extra cost; normal network charges apply.**
– Free to configure.

Limitations

• **Only available within the same organization.**
– Org-scoped.

Integration

• **Enterprises with multiple teams/projects.**
– Provides shared backbone.

Best Practices

• **Use for compliance-heavy environments.**
– Helps enforce uniform policies.

• **Centralize firewall/security rules for consistency.**
– Reduces drift.

Private Google Access:

What

Private Google Access allows VMs without external IPs to reach Google APIs and services using their internal IPs.

Why (Use Cases)

• **Securely access Google services without exposing workloads to the internet.**
– Keeps traffic private.

• **Required for compliance/security-sensitive environments.**
– Meets regulations.

Key Features

• **Works for VMs on subnets without external IPs.**
– Private-only instances can reach APIs.

• **Supports access to Google APIs, Cloud Storage, BigQuery, etc.**
– Covers most Google services.

• **Configured at the subnet level.**
– Turned on per subnet.

Pricing Model

• **Standard network egress charges apply.**
– Traffic billed normally.

Limitations

• **Only works for Google APIs/services, not arbitrary internet destinations.**
– Limited scope.

Integration

• **Compute Engine, GKE, App Engine private workloads.**
– Common with private apps.

Best Practices

• **Enable for all private subnets that need Google API access.**
– Ensures apps work without external IPs.

• **Use with Cloud NAT for broader internet access when required.**
– Complements NAT.

Private Service Connect (PSC):

What

Private Service Connect enables private, internal connections to Google services, partner services, or other VPCs.

Why (Use Cases)

• **Access Google services without traversing the internet.**
– Adds privacy.

• **Provide services privately to consumers in other VPCs.**
– SaaS-style setups.

• **Connect to SaaS providers securely.**
– Private consumption.

Key Features

• **Uses internal IP addresses.**
– Keeps traffic private.

• **Supports Google APIs, third-party services, and cross-VPC communication.**
– Wide coverage.

• **Consumer and producer models.**
– Flexible roles.

Pricing Model

• **Charged per GB of egress traffic.**
– Pay for usage.

Limitations

• **Not supported in every region.**
– Regional limitations apply.

Integration

• **Works with Cloud Storage, BigQuery, Pub/Sub, and SaaS solutions.**
– Integrates widely.

Best Practices

• **Prefer PSC over public endpoints for compliance/security.**
– Safer option.

• **Use service attachment for multi-tenant architectures.**
– Scales better.

Identity-Aware Proxy (IAP):

What

IAP provides zero-trust access control for applications running on App Engine, Cloud Run, and GKE, or behind HTTPS Load Balancers.

Why (Use Cases)

• **Secure web apps without using VPNs.**
– Access without network tunnels.

• **Grant access based on user identity and context.**
– Identity-based protection.

Key Features

• **Enforces access using IAM policies.**
– Centralized control.

• **Supports multi-factor authentication (MFA).**
– Adds security.

• **Provides audit logs of access.**
– Helps compliance.

Pricing Model

• **No extra cost; pay for load balancing/egress traffic.**
– Free feature.

Limitations

• **Requires HTTPS Load Balancer for GCE/GKE backends.**
– LB needed.

Integration

• **App Engine, Cloud Run, GKE, Compute Engine.**
– Works across compute.

• **Works with Google Identity or external IdPs.**
– Flexible identity sources.

Best Practices

• **Use instead of VPN for app-level security.**
– Simpler and safer.

• **Apply least-privilege IAM roles (IAP-secured Web App User).**
– Enforces minimal access.

Cloud NAT:

What

Cloud NAT (Network Address Translation) allows private resources without external IPs to reach the internet securely.

Why (Use Cases)

• **Allow private VMs to fetch updates or call APIs.**
– Ensures patching works without external IPs.

• **Reduce public IP usage for compliance/cost.**
– Fewer external IPs needed.

Key Features

• **Scales automatically.**
– Adapts to workload size.

• **Works with regional subnets.**
– Regional deployment.

• **Supports TCP/UDP protocols.**
– General purpose.

Pricing Model

• **Charged per VM per hour + egress traffic.**
– Cost depends on traffic and usage.

Limitations

• **One-way only → external services can’t initiate connections back.**
– Outbound only.

Integration

• **Compute Engine, GKE nodes.**
– Common with private workloads.

• **Often paired with Private Google Access.**
– Complements API access.

Best Practices

• **Use Cloud NAT instead of assigning external IPs.**
– Reduces exposure.

• **Monitor NAT usage with logs.**
– Keeps visibility.