08-11-2021

All subnetworks in auto mode vpc are in the 10.128.0.0/9 range, all subnetworks are /20 (expandable to /16, to increase further, convert to custom mode)

If a new region is added, a subnet is automatically created for it

Auto mode can be converted to custom mode but not viceversa

First two addresses reserved - .0 is for network, .1 is for subnets gateway

Last two also reserved - last is broadcast address

entire documentations for all networking services, overviews, how-to-guides, troubleshooting guides and best practices and manyy Qwiklabs.

## Defining and Implementing Networks

[Reserving an internal IP address](https://cloud.google.com/compute/docs/ip-addresses/reserve-static-internal-ip-address) takes that address out of the dynamic allocation pool and prevents it from being used for automatic allocations.

External ip is mapped to internal ip - network stores a lookup table

Cloud DNS has 100% uptime SLA

Alias IP range for a VM: subset of secondary IP range of the subnet its in

Routes for traffic within vpc and that going outside

Firewall must also allow

Firewall applied at network level

But connections allowed or denied at instance level

Hierarchical fwr - gotonext

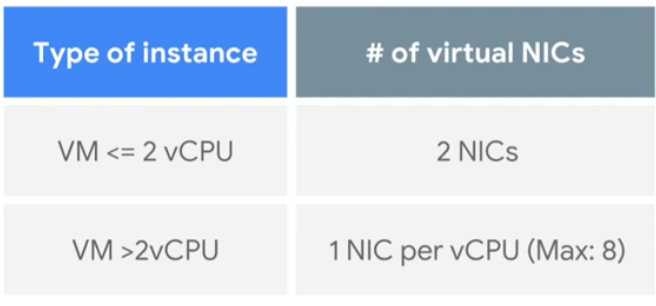
Lower level rules canNOT override higher rules

Can narrow VMs by using target network or target service account

When are multiple NICs used:

* Intrusion detection and prevention
* WAF
* Network appliance that does load balancing
* WAN optimization between networks

Cannot have two interfaces from same network



Internal dns resolves to nic0

Routes:

Default route to internet - applied to all instances in network - all subnets

Even default route to a subnet - same as above

You can ping **privatenet-us-vm** by its name because VPC networks have an internal DNS service that allows you to address instances by their DNS names instead of their internal IP addresses. When an internal DNS query is made with the instance hostname, it resolves to the primary interface (nic0) of the instance. Therefore, this only works for **privatenet-us-vm** in this case.

In a multiple interface instance, every interface gets a route for the subnet that it is in. In addition, the instance gets a single default route that is associated with the primary interface eth0. Unless manually configured otherwise, any traffic leaving an instance for any destination other than a directly connected subnet will leave the instance via the default route on eth0. <https://cloud.google.com/vpc/docs/create-use-multiple-interfaces#linux>

the **Network Admin** role has permissions to list but not modify/delete firewall rules.

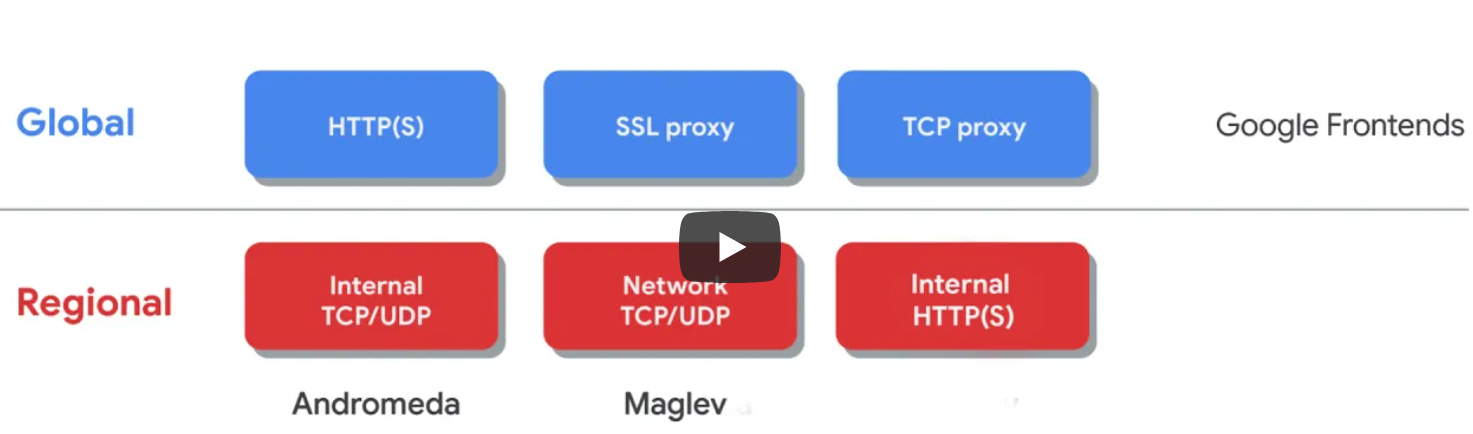
VM from service project can ping VM in another service project if proper firewall rules are there

The [**compute.networkUser**](https://cloud.google.com/compute/docs/access/iam#compute.networkUser) and [**compute.networkAdmin**](https://cloud.google.com/compute/docs/access/iam#compute.networkAdmin) roles have different permission sets. The **compute.networkUser** role includes permissions that are not available in the **compute.networkAdmin** role.

* Managed instance groups used with Shared VPC require [making the Google APIs service account a Service Project Admin](https://cloud.google.com/vpc/docs/provisioning-shared-vpc#sa-as-spa) because tasks like automatic instance creation via autoscaling are performed by that service account.

Compute Engine internal DNS names created in a network are not accessible to peered networks. The IP address of the VM should be used to reach the VM instances in peered network.

Deleting one side of the VPC Peering connection removes all peering routes.



virtual gateways do **not** respond to ICMP traffic

or decrement IP TTL headers.

Except for the default network, you must explicitly create higher priority [ingress firewall rules](https://cloud.google.com/vpc/docs/firewalls#priority_order_for_firewall_rules) to allow instances to communicate with one another.

The MTU is the size, in bytes, of the largest packet supported by a network layer protocol, including both headers and data. In Google Cloud, you set the MTU for each VPC network, and VM instances that use that network must also be configured to use that MTU for their interfaces.

Regional MIGs are better than zonal MIGs

1. HTTP(S) load balancer
   1. Global
   2. Url based
   3. Layer 7
   4. 80, 8080 or 443
   5. Url maps, but usually to group which is closest to request, next closest if not enough compute capacity on closest backend
   6. Timeout and session affinity for backends
   7. Anycast IP > Global forwarding rule > Target proxy > URL map > healthcheck > Backend
   8. HTTPS: At least one signed SSL certificate, client ssl session terminated at load balancer
   9. QUIC transport layer protocol
   10. Cloud armor
   11. NEGS - zonal, internet (outside of GCP) and serverless

CDN:

Cache hits and misses are logged in Cloud Logs

Cache modes: 3

The static content rules described here apply only to successful responses (for example, HTTP **200 OK** responses). By contrast, error responses are cached based on the [negative caching](https://cloud.google.com/cdn/docs/using-negative-caching) settings, regardless of content type.

Cache can be invalidated from Load Balancer page

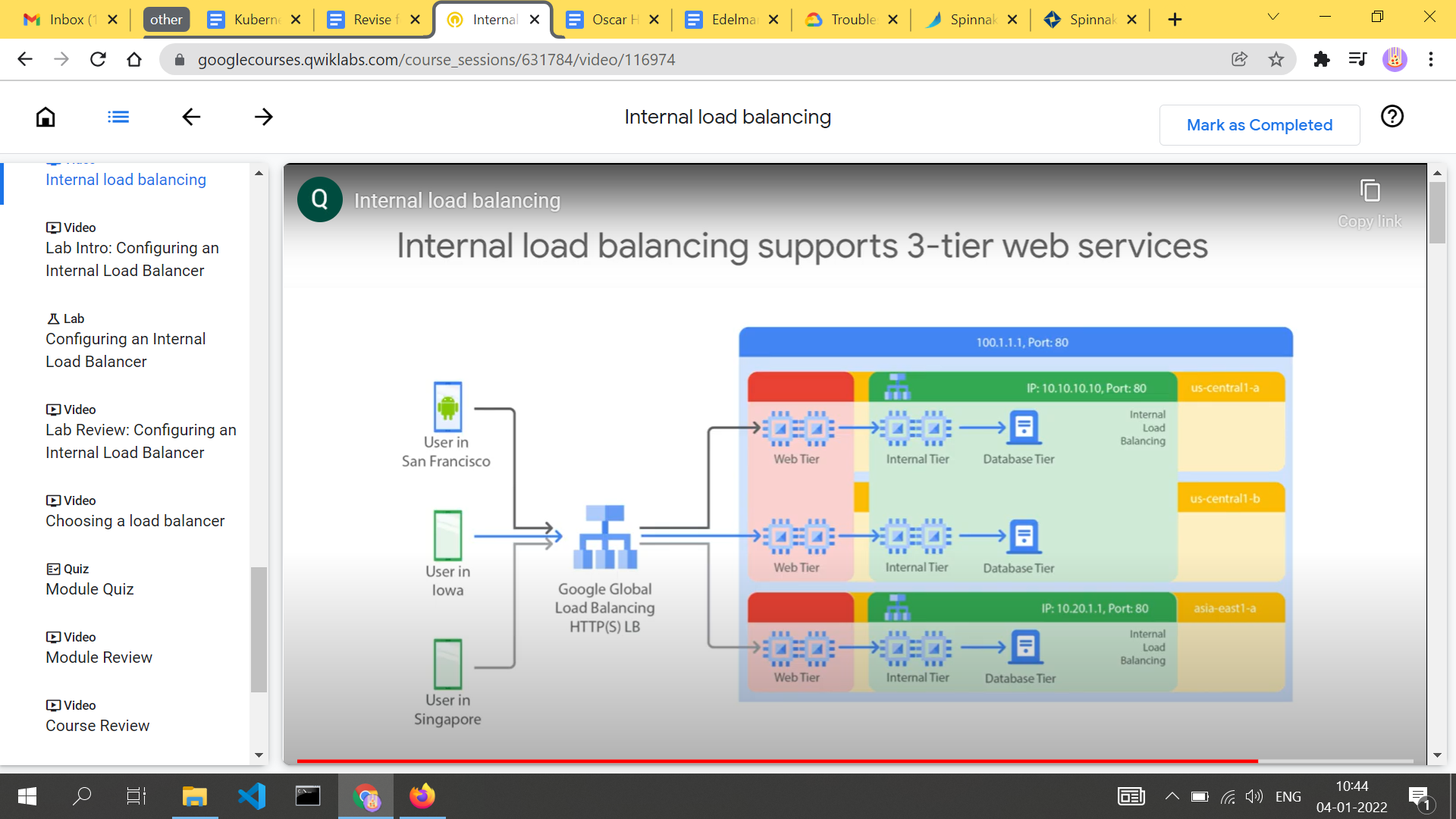
Each cache entry in a Cloud CDN cache is identified by a *cache key*. When a request comes into the cache, the cache converts the URI of the request into a cache key, and then compares it with keys of cached entries. If it finds a match, the cache returns the object associated with that key.

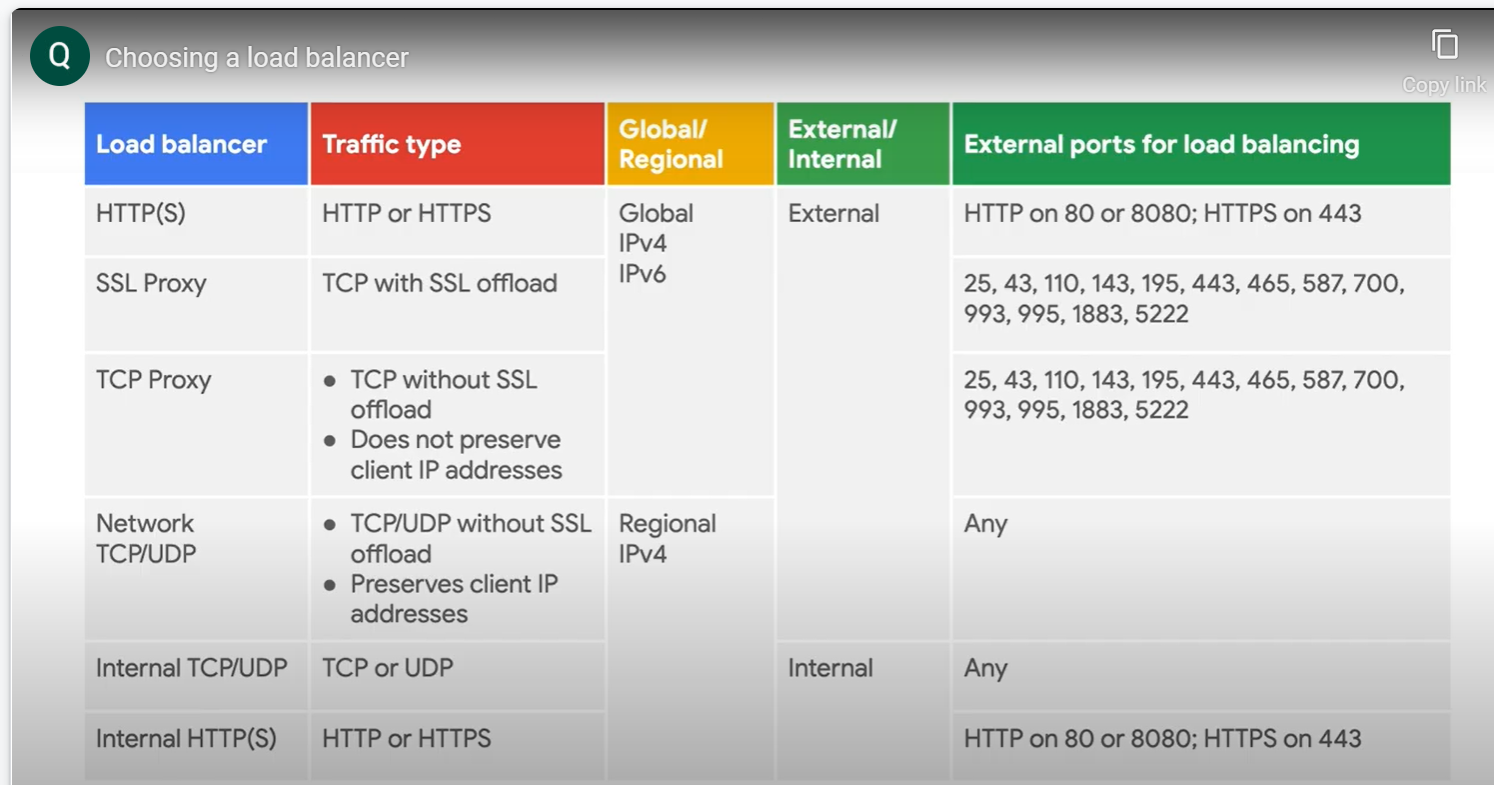
1. SSL proxy load balancer
   1. Global
   2. Ssl session termination at load balancer
   3. Security patching, intelligent routing
2. TCP proxy load balancer
   1. Global
   2. Unencrypted, non HTTP
   3. TCP session termination at load balancer
   4. Load balancer to backend can be tcp or ssl
   5. Security patching, intelligent routing

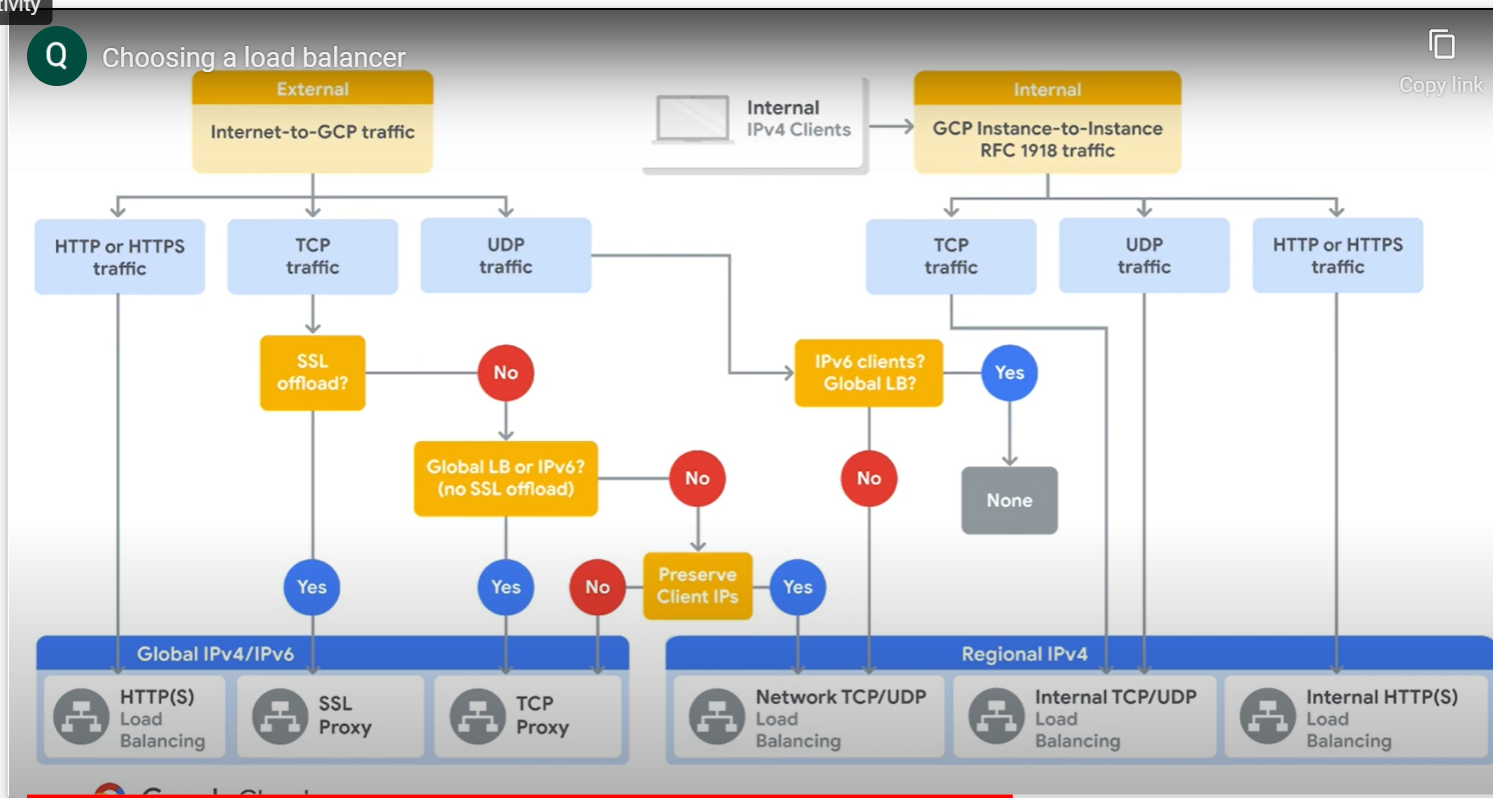
**Regional:**

1. Network TCP/UDP load balancing
   1. Non proxied
   2. **Preserves client IP address**
   3. pass-through load balancer, you control access to the load balancer's backends using Google Cloud firewall rules.
   4. Forwarding rule based traffic routing
      1. Hash of port, protocol type, source IP and destination IP
   5. Tcp, ssl, udp on ports that are not supported on global ssl/tcp proxy LBs
2. Internal TCP/UDP
   1. Simpler config, reduced latency
   2. Andromeda
   3. Session terminates at instance
3. Internal HTTP(S)
   1. Envoy proxy

3 tier web application with internal LB:







## Hybrid Connectivity

Peering, Cloud VPN is Layer3

Interconnect is Layer2

HA VPN does not support static routes

HA VPN can also be configured with one active interface and one external IP address, but it wont have the 99.99% uptime SLA

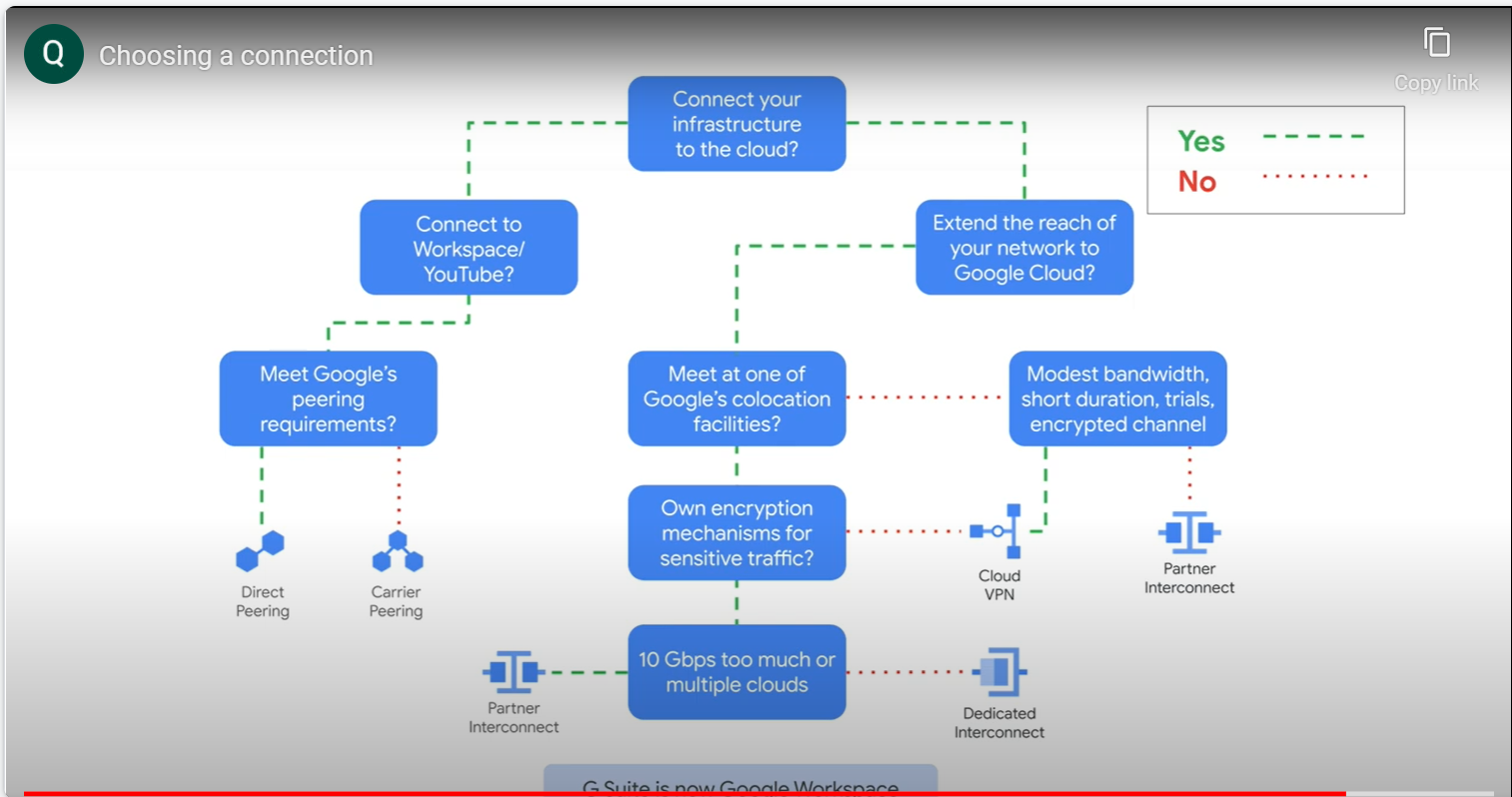
Dynamic routing with cloud router

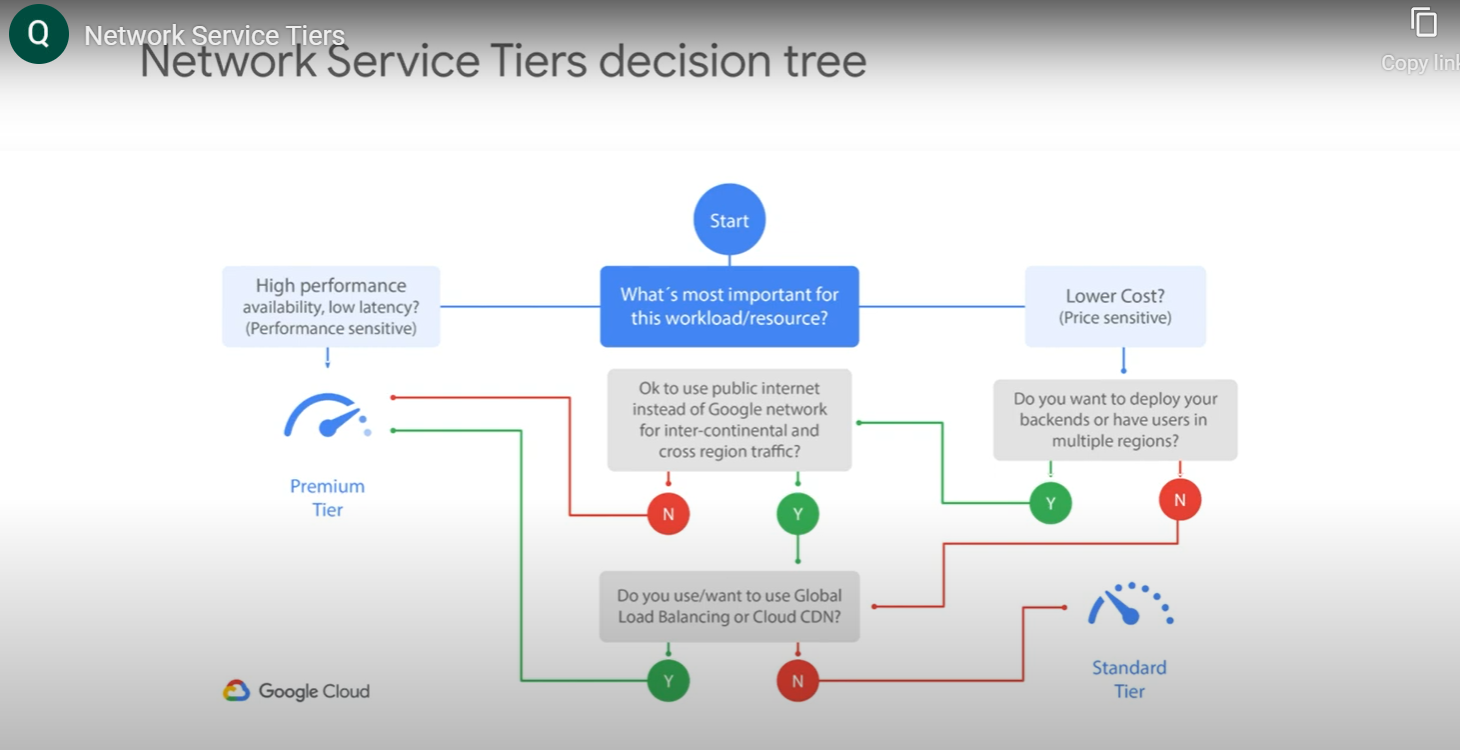
Needs additional IP to be assigned to Cloud Router and peer VPN gateway (two ends of VPN tunnel) from 169.254.0.0/16 (link local addresses)

When you run HA VPN tunnels between two Google Cloud VPCs, you need to make sure that the tunnel on **interface0** is connected to **interface0** on the remote VPN gateway. Similarly, the tunnel on **interface1** must be connected to **interface1** on the remote VPN gateway.

Peering doesnt have SLA

Access is over public IP addresses



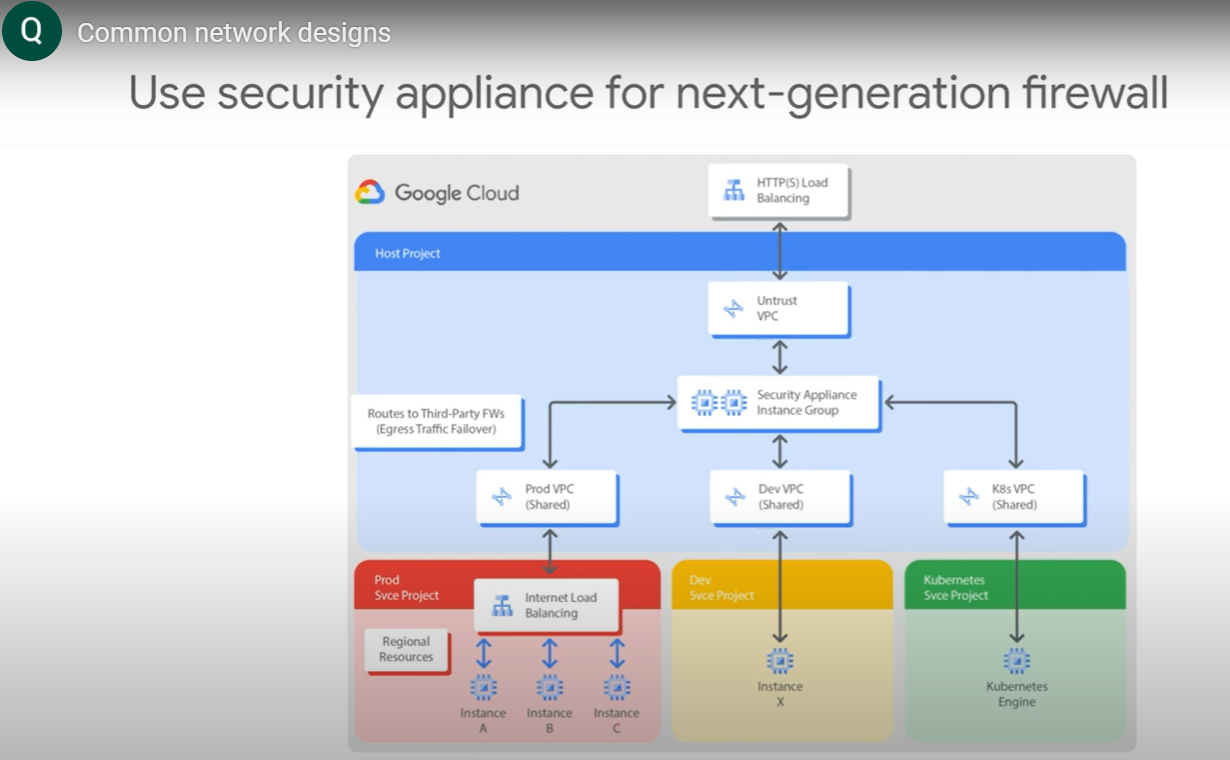


Billing for resources in projects using shared VPCs are attributed to service project

Egress traffic to Google products, like YouTube, Maps or Drive is not charged for.

There is a charge for egress between zones in the same region.

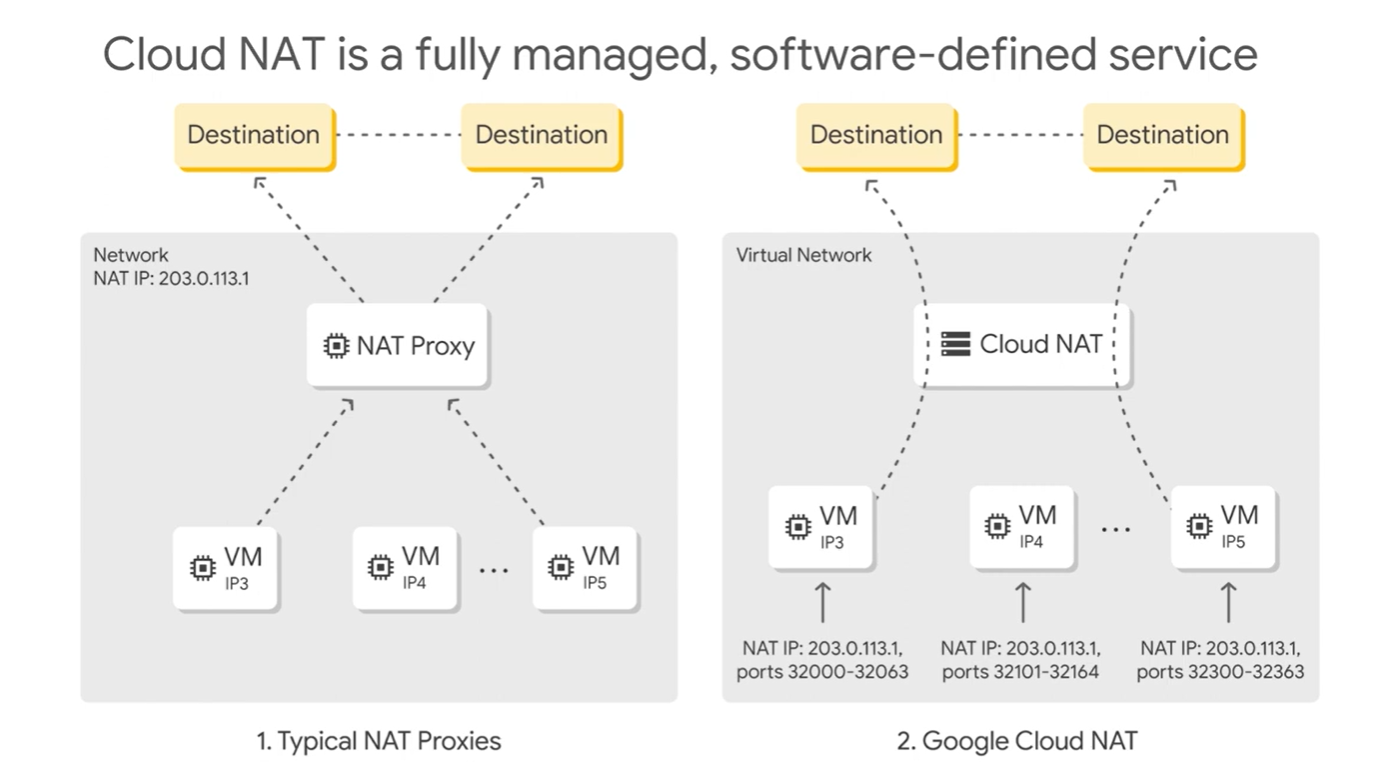
Network designs:

1. Availability: regional MIG, placing instances across two zones in same region
2. Globalization: across failure domains, can route requests to region closest to user
3. Access internal LB through global LB
4. Access internal LB through cloud VPN or xconnect
5. Security appliance for NGFW
   1. 
   2. NS traffic - client server traffic
   3. EW traffic - between internal networks

PGA allows VMs without external IP addresses to reach public addresses of google apis and services

PGA for onpremises hosts is similar, but its for onprem instances without external IPs

Private services access is implemented through VPC peering connection between our VPC network and service providers VPC network (what we did for vertex ai private endpoint)



These NAT IPs are programmed by andromeda and configured inside the VMs

Can have multiple NAT IP addresses per gateway

Manual or auto mode - for allocation of NAT IP addresses

If PGA is enabled and there is no NAT, then sudo apt-get update should only work for Google Cloud packages because **vm-internal** only has access to Google APIs and services!

When Cloud NAT logging is enabled, one log entry can be generated for each of the following scenarios:

* When a network connection using NAT is created.
* When a packet is dropped because no port was available for NAT.

Can have BYOL for marketplace solutions

Direct Peering can only access external IP addresses. ?

<https://cloud.google.com/vpc/docs/private-access-options#pga-supported>

1. Private Service Connect (3) - endpoint
2. Private Google Access (2) - for internal IP only
3. Private Services Access - network peering
4. Serverless VPC Access - for serverless

Uptime check - checks availability of service (app engine, compute engine instance, url of a host, aws load balancer or host) from specific regions - can have alerting policies and check latency

Network Intelligence Center - connectivity tests

Topology

Performance dashboards

Firewall insights - cloud monitoring metrics and recommender insights

Because Network Load Balancing is a pass-through load balancer, you control access to the load balancer's backends using Google Cloud firewall rules. You must create ingress allow [firewall rules](https://cloud.google.com/vpc/docs/firewalls) or an ingress allow [hierarchical firewall policy](https://cloud.google.com/vpc/docs/firewall-policies) to permit health checks and the traffic that you're load balancing.

Near real time log updates - every 5 seconds

VPC Flow Logs - from instances

Packet Mirroring - also from instances - duplicates traffic (hence bandwidth used also increases)

NAT Logging

Resources can't be relocated from one VPC network to another without being recreated.

Group applications into fewer subnets with larger address ranges

* Static and dynamic routes are not propagated.
* Source tags and source service accounts of the sending VM are not propagated across VPC Network peering.
* There are no additional charges; you are billed as if the VMs were in the same VPC network.

Don't use Cloud VPN as a transit network between on-premises networks, as explained in the [Cloud VPN documentation](https://cloud.google.com/network-connectivity/docs/vpn/concepts/overview#vpn-as-data-transfer-network).

Up to 3 Gbps per Cloud VPN tunnel

Dedicated Interconnect - L2

Partner Interconnect - L3

* Each vCPU has a 2 Gbps egress cap for peak performance, up to a theoretical maximum of 16 Gbps.
* Limit of 8 interfaces per instance.

Is VPN (peering) transitive? - **i guess no since cloud router can't advertise routes it has learnt to another BGP session**

However, even though it is possible to use tags for target filtering in this manner, we recommend that you use service accounts where possible. Target tags are not access-controlled and can be changed by someone with the instanceAdmin role while VMs are in service. Service accounts are access-controlled, meaning that a specific user must be explicitly authorized to use a service account. There can only be one service account per instance, whereas there can be multiple tags. Also, service accounts assigned to a VM can only be changed when the VM is stopped.

Despite the next-hop gateway's name, the traffic path from instances to the Google APIs remains within Google's network.

## Build and Secure Networks

## Network Performance and Optimization

With higher latency, extra caching layers or WAN accelerators might have to be used.

The latency between the EU and Asia locations is very high - Compute Engine does not have a direct link it can use between Europe and Asia at this time.

From a networking point of view, it is recommended that if you run a service using only **ONE** global location, let that location be in Central US. Depending on how your user-base is split, US East or West might also be recommended.

Traceroute shows all Layer 3 (routing layer) hops between the hosts.

You will see that within a region, the bandwidth is limited by the [2 Gbit/s per core](https://cloud.google.com/compute/docs/networks-and-firewalls#egress_throughput_caps) egress cap.

## Ensure Access & Identity

## [Exam Topics](https://www.examtopics.com/exams/google/professional-cloud-network-engineer/view/)

* Project wide ssh keys, how does ssh work in compute engine VMs - **done**
* TMNA: Implicit FW rule [ingress or egress] are NOT logged…
* For cases where you have multiple on-premises routers connected to a single Cloud Router, the Cloud Router learns and propagates routes from the router with the lowest ASN.
* After you order a Dedicated Interconnect connection, Google sends you and the NOC (technical contact) an email with your [Letter of Authorization and Connecting Facility Assignment (LOA-CFA)](https://cloud.google.com/network-connectivity/docs/interconnect/concepts/terminology#loa) (one PDF file per connection). You must send these LOA-CFAs to your vendor so that they can install your connections. If you don't, your connections won't get connected.

If you can't find the LOA-CFAs in your email, retrieve them from the Google Cloud Console. You can also respond to your order confirmation email for additional assistance.

After the status of an Interconnect connection changes to PROVISIONED, the LOA-CFA is no longer valid, necessary, or available in the Cloud Console.

* Depending on the type of load balancer you are using, you can add instance groups to a target pool or backend service. To learn more about load balancing, see [Choosing a load balancer](https://cloud.google.com/load-balancing/docs/choosing-load-balancer).
* You can configure 2 different MED values for each BGP neighbor in your single on-prem router , to influence ISP(GCP)'s 2 separate routers to select which path they send traffic towards you. The lower MED value is preferred.
* Each Cloud VPN tunnel associated with a **Classic VPN gateway** must connect to a unique peer VPN gateway, as identified by the peer gateway's IP address. If you need to create a second tunnel to the same peer gateway, you must create that tunnel from a different Cloud VPN gateway.
* When a user SSHs into any VM in a Google Cloud Platform Project, their public key is added to the project’s metadata, meaning this key could be used to access any VMs running in that project. Google makes it easy to block project-wide SSH keys from a given VM,

## [VCE Guide questions](https://vceguide.com/which-two-methods-can-you-use-to-accomplish-this-3/)

* You must use legacy HTTP health checks for [target pool based-network load balancers](https://cloud.google.com/load-balancing/docs/network/networklb-target-pools). You can select or create a legacy health check when you complete the network load balancer's configuration in the Cloud Console.
* GKE - network policy - firewalls between pods and services
* GKE private endpoint

<https://cloud.google.com/blog/products/gcp/uusing-googles-cloud-networking-products-a-guide-to-all-the-guides>

TFTP is a UDP based protocol

Cloud NAT allows outbound connections and the inbound responses to those connections. Each Cloud NAT gateway performs source NAT on egress, and destination NAT for established response packets.

Cloud NAT does *not* implement unsolicited inbound connections from the internet. DNAT is only performed for packets that arrive as responses to outbound packets.

* Google [Cloud](https://cloud.google.com/vpc/docs/routes) does not allow you to create a custom static route that has an equal or narrower destination than any subnet route or peering subnet route.
* Conversely, Google Cloud does not allow you to create a new subnet or peering subnet route whose destination exactly matches or is broader than (would contain) an existing custom static route.
* When Cloud Routers learn prefixes that exactly match or fit within the destination of a subnet or peering subnet route, Google Cloud ignores these custom dynamic routes. However, a dynamic route's destination can *contain* (can have a smaller subnet mask than) a subnet's IP range.
* Tagged routes are not exported or imported through peering.
* IKEv1 does not support multiple local traffic and remote traffic selectors.
* static routes do not work with Dedicated Interconnect.
* striping traffic over multiple flows will increase the amount of bandwidth consumed by the application (each TCP flow is 3Gbps).
* If the rule is invalid or not relevant, disable it by using the evaluatePreconfiguredExpr expression, and specify the rule's ID in the exclude ID list argument.
* Cache hits are served even if the downstream Google Cloud Armor security policy would prevent that request from reaching the CDN origin server.

<https://docs.google.com/forms/d/e/1FAIpQLServ0tNGkr-dYAfmez_Gdk74dmVypZjzUKrkVFtFcArzhmPow/viewscore?viewscore=AE0zAgAeud8f96_ABYS9xG4x5nRm76jWrrugI_cFojPxxd3YBH2IMbTCMHmeXxRZ6_9x1_4>

## [LearnGCPwithMahesh](https://www.youtube.com/c/LearnGCPwithMahesh/search)

Datastore is mini version of Bigtable

Cloud SQL can be created even without any VPC network present (with only public IP). Need to set authorized network CIDRs and stuff for security

## [Jayendra Patil](https://jayendrapatil.com/google-cloud-professional-cloud-network-engineer-certification-learning-path/)

## Cloud Next videos

1. VPC Best practices
   1. Something about routers - check
2. Load Balancing
   1. General approach to do global load balancing in other clouds is DNS based load balancing with multiple regional VIPs, but this has many problems
3. Cloud OnAir: Networking 105: How to use GCP DNS
   1. Cannot do DNS forwarding to another VPC, has to be to or from onprem
      1. **Forwarding zones for outbound**
      2. **Policy for inbound**
   2. DNS peering **zone** - essentially forwarding between two VPCs
      1. Is unidirectional
   3. Alternative Name Server - dont want to do any resolving on GCP - forward everything - **DNS policy**
      1. Forwarding zone approach is preferred over using a DNS policy that enables an alternative name server. It preserves access to Compute Engine internal DNS names, and public IP addresses are still resolved without an extra hop through an on-premises name server.
   4. Need to have only one forwarding zone out of GCP, and can peer other zones to this zone
      1. Because if multiple forwarding zones, onprem might send it to some random zone, becomes obvious when you have many forwarding zones
      2. All routing setup needs to be done in only one forwarding zone vala project
   5. **Troubleshooting:** primary interface of VM is on the same network as the private zone that you have created
   6. Cross project binding for shared vpc service projects
   7. Google Cloud first looks for myapp.dev.gcp.example.lan in a zone that serves dev.gcp.example.lan, if accessible, before it looks for it in a zone that serves gcp.example.lan, if accessible.
   8. The IP address used for inbound forwarding must be in the same region as the VPN tunnel or VLAN attachment.
   9. custom static routes with network tags are not valid for forwarding DNS queries. Ensure that the route used to reach the on-premises name server does not specify a network tag.
4. [Cloud OnAir: CE Chat: Google Cloud Networking 102 - Cloud Routing and VPC Peering](https://www.youtube.com/watch?v=jQc9P7xA_wU)
   1. <https://cloud.google.com/vpc/docs/routes#types_of_routes>
   2. Preprogrammed routes cant be modified, default routes (for eg to IGW) can be
   3. Routing table lives on every VM
   4. Route priority = BGP MED
   5. In case of failure, data plane not affected, packets will still be sent, enable graceful restart, so on prem router doesnt start tearing down routes
   6. Need to add fwr after doing VPC peering
   7. VPC peering means sharing preprogrammed routes
   8. **Best practices**: Bidirectional Forwarding Detection configured with default settings detects failure in 5 seconds, compared to 60 seconds for BGP-based failure detection.
   9. The on-premises router's hold timer is set to at least 60 seconds.
   10. Graceful restart
   11. Redundant routers and BGP sessions
   12. **Troubleshooting**: *Router appliance requires Network Connectivity Center to operate.* That is, you can't configure standalone router appliance instances that peer with a Cloud Router or with other peer routers. You must configure router appliance instances as part of a Network Connectivity Center spoke.
   13. On-premises routes without a MED value are take priority (value 0)
   14. Dedicated Interconnect: Layer 2, Partner: Layer 3 **?**
   15. A single Cloud Router can't re-advertise routes learned from one BGP peer to other BGP peers, including to Cloud Routers in other VPC networks. - can be sorted using VPC peering and importing custom routes/ shared VPC/ multiple VPNs
   16. On a multi-NIC VM, each NIC gets different routes
   17. Traffic is routed asymmetrically when ingress and egress traffic use different paths. Asymmetric routing happens when the preferred path advertised by your on-premises router and the Cloud Router don't align.
5. [Scalable and Manageable: A Deep-Dive Into GKE Networking Best Practices (Cloud Next '19)](https://www.youtube.com/watch?v=fI-5LkBDap8)

the Cloud Router that you use must not already manage a BGP session for a VLAN attachment associated with a Partner Interconnect connection because of the attachment's [specific ASN requirements.](https://cloud.google.com/network-connectivity/docs/interconnect/how-to/partner/creating-vlan-attachments#creating_vlan_attachments)

HA VPN Gateway > Peer VPN Gateway > Router > VPN tunnels > BGP session > Verify > additional tunnels

Classic VPN: With **route-based VPN**, **you specify only the remote traffic selector**. If you need to specify a local traffic selector, create a Cloud VPN tunnel that uses policy-based routing instead.

**Console:** Dynamic (BGP) provides the easiest to configure and most resilient IPsec VPN configuration. If your device does not support BGP you should use Route Based VPN and you should only use Policy-based routing for configurations that cannot support other routing methods.

If you convert an auto mode VPC network to a custom mode VPC network, you might have to delete and re-create VPN tunnels that use policy-based routing if the tunnel relied on default values for local IP ranges.

**Having two routes to 0.0.0.0/0 pointing to a VPN tunnel and one to IG - couldnt SSH - showed below error when trying by IAP**

| student\_04\_583981a29544@cloudshell:~ (qwiklabs-gcp-02-763ca33700f4)$ gcloud compute ssh cloud-loadtest --zone "us-east1-b" --tunnel-through-iap  ERROR: (gcloud.compute.start-iap-tunnel) Error while connecting [4033: 'not authorized'].  kex\_exchange\_identification: Connection closed by remote host  Connection closed by UNKNOWN port 65535  Recommendation: To check for possible causes of SSH connectivity issues and get  recommendations, rerun the ssh command with the --troubleshoot option.  gcloud compute ssh cloud-loadtest --project=qwiklabs-gcp-02-763ca33700f4 --zone=us-east1-b --troubleshoot  Or, to investigate an IAP tunneling issue:  gcloud compute ssh cloud-loadtest --project=qwiklabs-gcp-02-763ca33700f4 --zone=us-east1-b --troubleshoot --tunnel-through-iap  ERROR: (gcloud.compute.ssh) [/usr/bin/ssh] exited with return code [255].  student\_04\_583981a29544@cloudshell:~ (qwiklabs-gcp-02-763ca33700f4)$ gcloud compute ssh c |
| --- |

**Forwarding rules:**

External forwarding rules accept traffic from client systems that have internet access

Cloud NAT never performs NAT for traffic sent to the select external IP addresses for Google APIs and services

[Load balancing health checks](https://cloud.google.com/compute/docs/load-balancing/health-checks) help direct traffic away from non-responsive instances and toward healthy instances; these health checks do not cause Compute Engine to recreate instances. On the other hand, [managed instance group health checks](https://cloud.google.com/compute/docs/instance-groups/autohealing-instances-in-migs) proactively signal to delete and recreate instances that become UNHEALTHY.

| **pod-ipv4-range** | **cluster-ipv4-cidr** |
| --- | --- |
| in nodepool command | in cluster command |
| name of subnet sec range | cidr range (which exactly?) |
| ip allocation policy, alias ip (vpc native) | routes based clusters |

VM exam 10 questions

<https://www.whizlabs.com/blog/google-cloud-certified-professional-network-engineer-questions/> 25 questions - review again once

Specify the remote traffic selector of a policy-based Cloud VPN tunnel when you create it. If you use the Cloud Console to create the Cloud VPN tunnel, custom static routes whose destinations correspond to the CIDRs of the remote traffic selector are automatically created. IKEv1 [limits remote traffic selectors to a single CIDR](https://cloud.google.com/network-connectivity/docs/vpn/concepts/choosing-networks-routing#ts-ip-ranges).

Creating a policy-based VPN tunnel with both traffic selectors set to 0.0.0.0/0 is functionally equivalent to creating a route-based VPN.

A best practice is to use 30 or fewer CIDRs per traffic selector so that you don't create an IKE proposal packet that exceeds the maximum MTU.

| **Type of cluster** | **Default size for pod** | **Pod range** | **Service range** |
| --- | --- | --- | --- |
| **VPC native gke mnged** | /14 | /21 to /9 | /27 to /16 |
| **VPC native user mnged** |  | /24 to /9 | /28 to /16 |
| **Autopilot gke managed (?)** |  |  |  |
| **Routes based gke mnged** |  |  | - |
| **Routes based user managed** | /14 | /19 to /9 | - |

VPN - 1.5 to 3 Gbps

Partner interconnect - 50 Mbps to 50 Gbps

Dedicated xconnect - 10 to 200 Gbps

Direct peering -

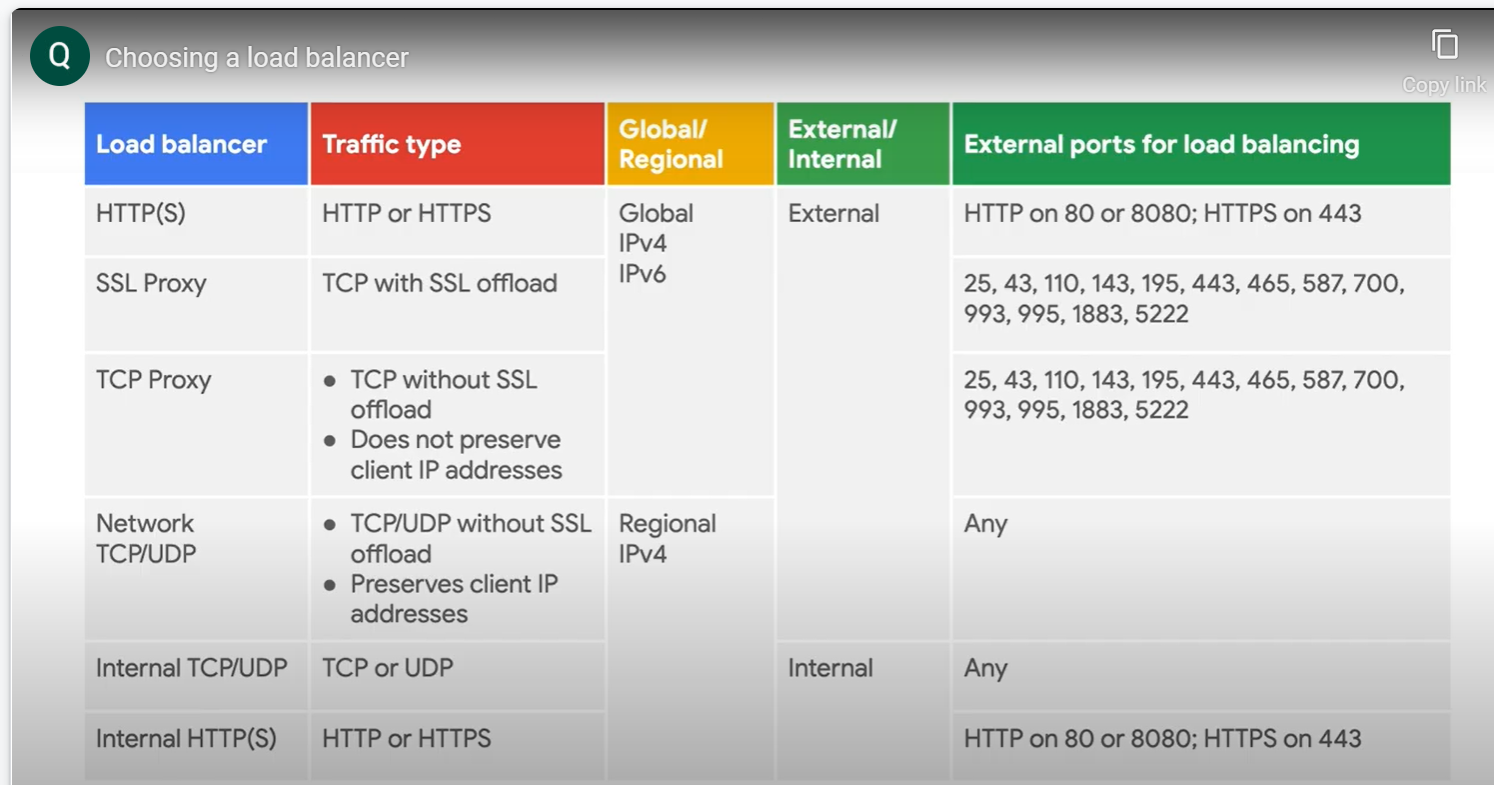
Carrier peering -

Private peering if more than 1Gbps

**be clear with these basic network concepts. You should also have a Clear understanding of OSI layer and how TCP and UDP works.**

1. **TFTP** - UDP/IP protocols using well-known port number 69
2. **IKE** - UDP port 500, if NAT, UDP 4500
3. **IKEv2** - same as IKE
4. **VOIP** - TCP or UDP port 5060
5. **SMTP** - TCP/UDP 25
6. **POP** - TCP ports 110 and 995
7. **IMAP** - TCP ports 143 and 993
8. **FTP** - TCP protocol using port 21 for control, 20 for data transfer
9. **SFTP** - TCP port 22 since over SSH
10. **SSL** - TCP port 443
11. **ICMP** - IP protocol 1 (TCP is IP protocol 6)

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=voip>

Check which LB works on which ports: 

Balancing modes for different LBs:

1. All HTTP LBs - have **rate** option
2. All proxy LBs - have **utilization** option
3. All TCP/UDP LBs - have **connection** option

Session Affinity:

* Client IP for all
* HTTP cookie, header field, generated cookie for HTTP
* Client IP ( + Client Port + Destination IP + Dest Port + Protocol) for TCP/UDP (**but not proxy**)

Use Rate or Connection balancing mode to reduce chance of breaking session affinity

Exceptions:

1. Target pool based NLB - need legacy health checks
2. Cdn does not work with IAP
3. Cloud armor does not work with cache hits
4. Packet mirroring user for creating policies
5. Packet mirroring admin on resources which are sources
6. Dns peering mein specify svc account which has dns peer role on producer project
7. To enable logging for , do it while updating, cannot do while creating
8. DNS policies for
   1. Enable logging
   2. Server policies
9. Cloud Router for Partner interconnect needs public ASN, for all others, private ASN
10. Ikev1 in static routing does not support multiple local/remote traffic selector ranges
11. Cdn works only with https
12. Cloud armor works with all proxy based LBs
13. Internet NEG only works with global https classic LB