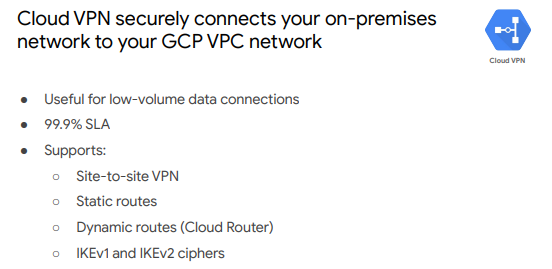
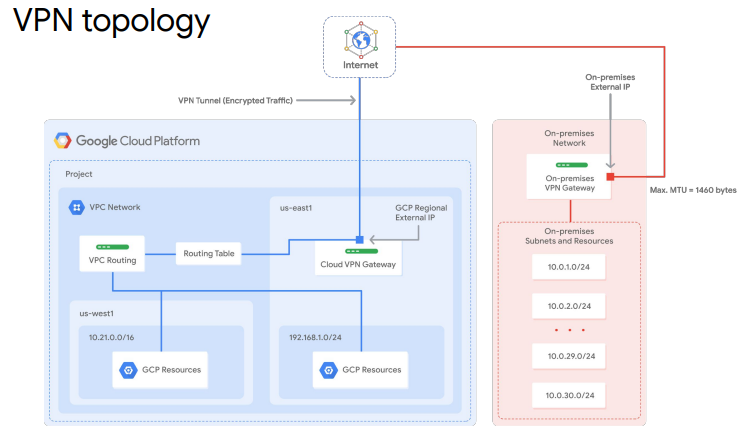
**Essential Google Cloud Infrastructure – Scaling and automation**

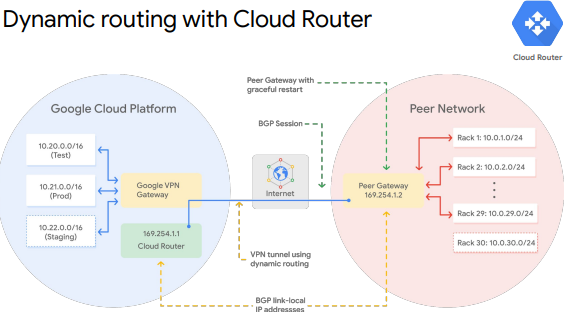
**Interconnecting networks**

Cloud VPN

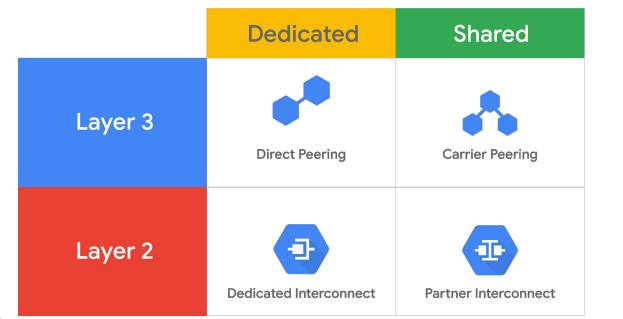


^ securely connects on-prem network to GCP VPC network through IPsec VPN. Traffic is encrypted through one gateway and decrypted by the other VPN gateway. Protecting data as it travels over the public internet. Useful for low volume data connections



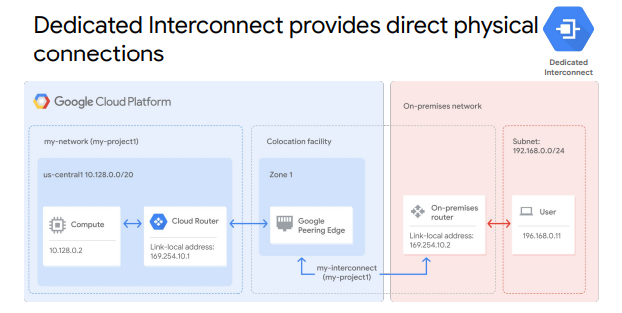


Cloud interconnect and peering



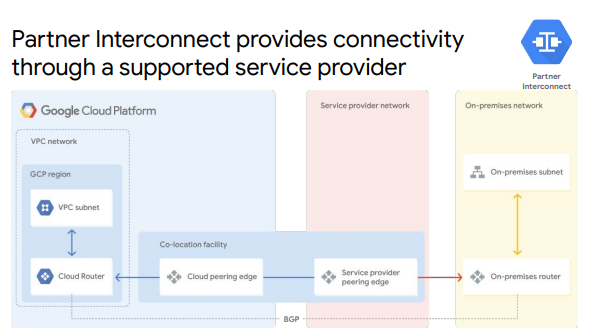
^ dedicated connections provide direct connection to Google’s network. Shared connections provide the connection through a partner. Layer 2 connections VLAN piped directly into GC environments. Layer 3 connections access to Google Workspace services, YouTube and GC APIs using public IP addresses

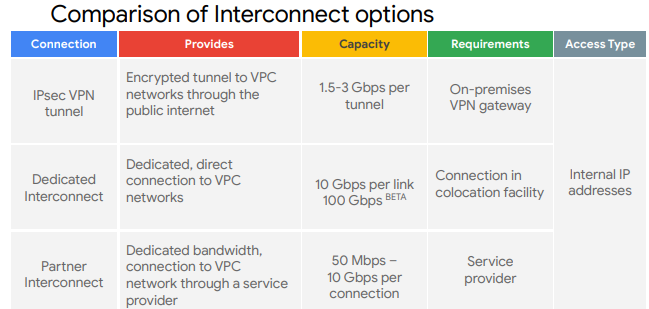
Cloud Interconnect



^ direct physical connection between on-prem and Google network. Enables transferring large amount of data between the networks. Requires your network to physically meet Google’s in supported colocation facility.

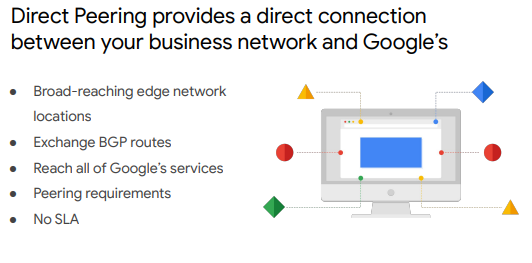
If you are too far away from a colocation facility, use partner connect

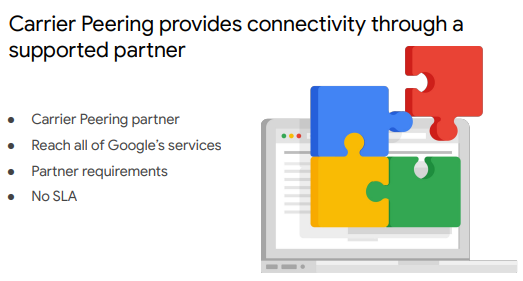




^IPsec = Cloud VPN

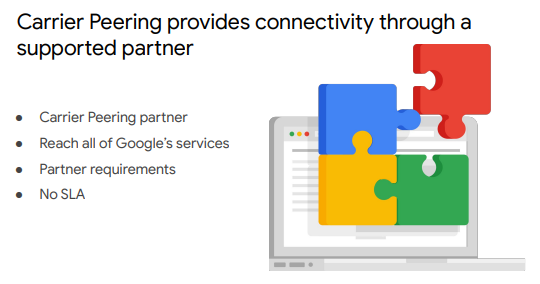
Peering

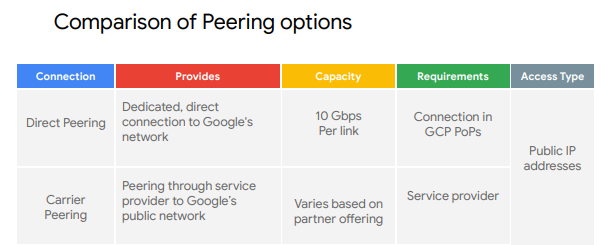




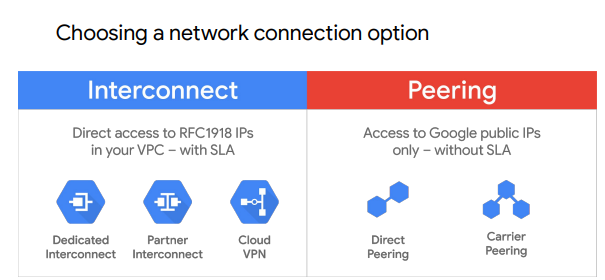
^ GCPs Edge Points of Presence (PoPs) are where Google network connects to other networks via peering.

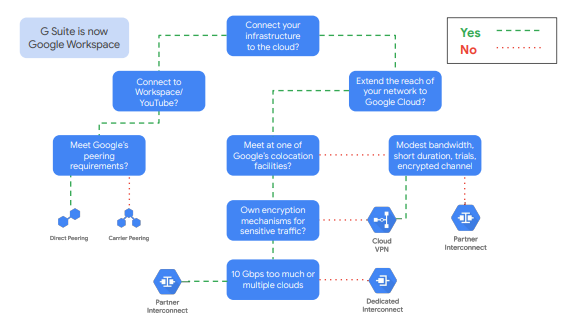
Similar to interconnect, if you are too far from a PoP you can peer via a partner





Choosing a connection

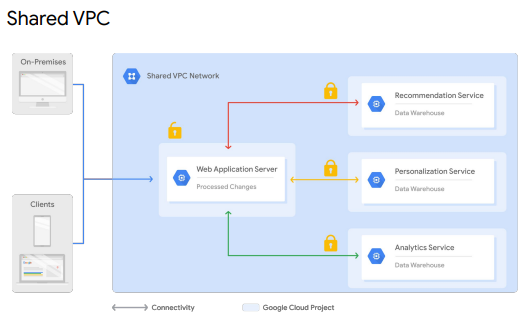




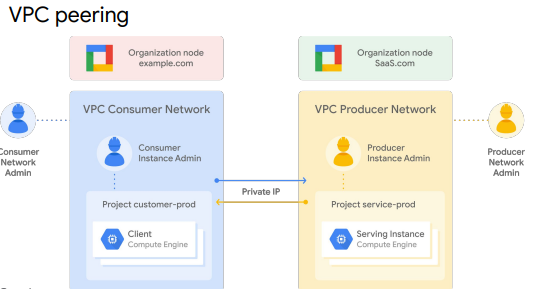
Shared VPC and VPC Peering

* Many organizations commonly deploy multiple, isolated projects with multiple VPC networks and subnets. Two configurations for sharing VPC networks across GCP projects

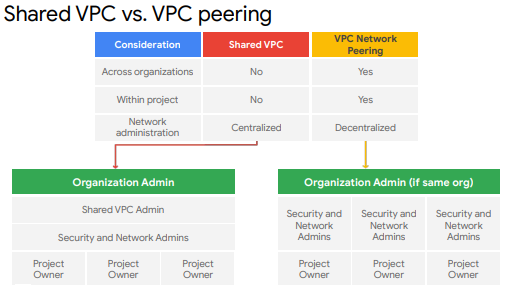
1. Shared VPC – share network across several projects in your GCP organisation
2. VPC Network Peering – configure private communication across projects in the same or different organisations



^ allows an organisation to connect resources from multiple projects to a common VPC network. Resources can then communicate securely using internal IP addresses from this network. When using shared VPC one project is designated as the host project and other projects are attached to it



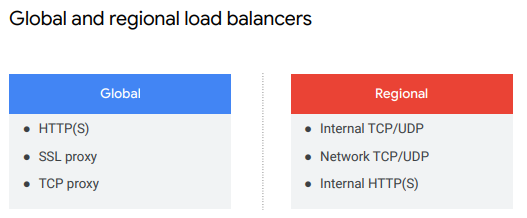
^ allows connectivity across VPC networks regardless of if they belong to the same project or organisation (depending on the firewall rules that allow or deny traffic between networks). In order for it to be established, both resources most peer the network to the corresponding network. Once it is done on both sides the VPC peering becomes active and routes are exchanged. The resources can then communicate privately using internal IP addresses. VPC peering is decentralised (or distributed) because each VPC network remains under control of separate admin groups and maintains their own firewall and routing tables



**Load Balancing and Autoscaling**

Cloud Load Balancing

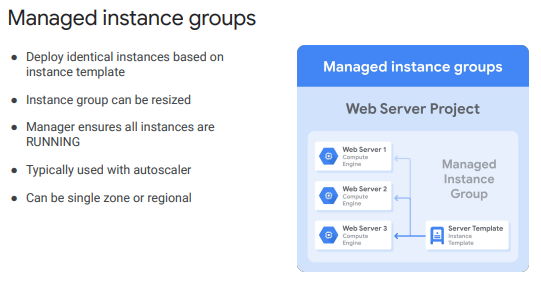
* Distribute load-balanced compute resources in single or multi region to meet your high availability requirements
* Put resources behind single anycast IP address
* Scale resources up or down with intelligent autoscaling



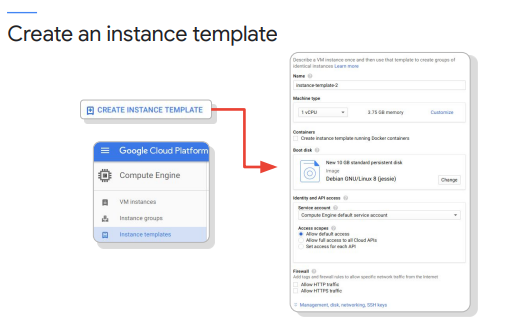
^ global leverages Google front ends. Used when users and instances are distributed globally where users all need access to the same resources and content, and you want to provide access using single IP address. Regional distributes traffic to instances in a single GCP region

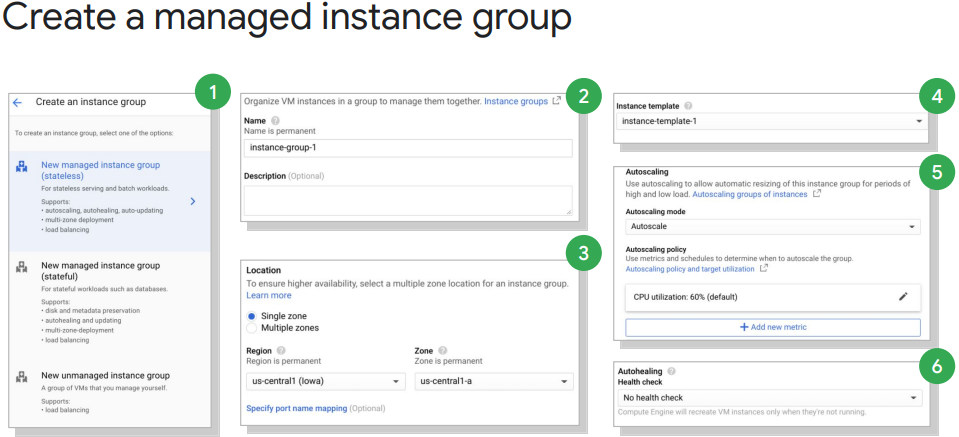
Managed instance groups

* Collection of identical VM instances you control using an instance template
* Easily update all by updating templating and performing rolling update
* Use with load balancing services so that if an instance in the groups stops another is auto created. The recreated instance has the same name and template as the previous one, ensuring all instances are running optimally
* Recommended to run them regionally rather than zonal as this allow the instance to be run over multiple zones preventing outages if there is an issue with the zone

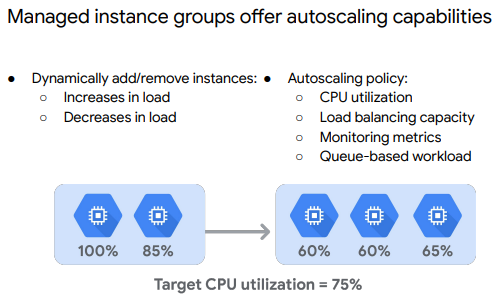


Need to have a template before you can create a managed instance group

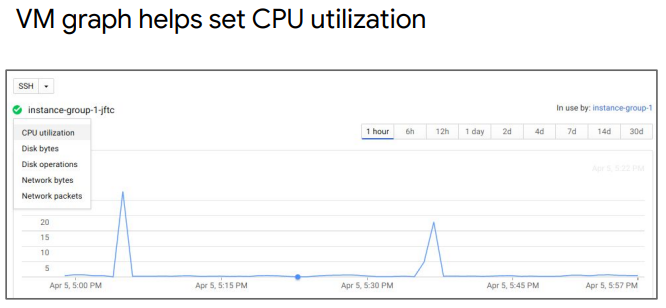


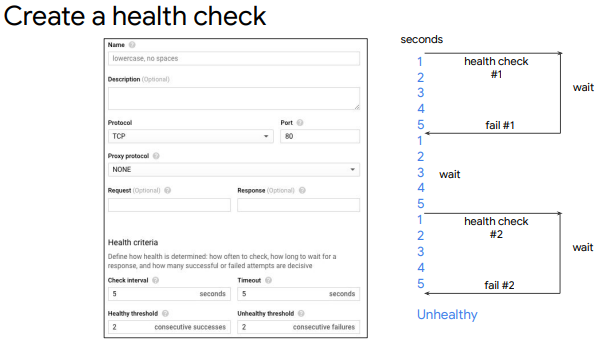


Autoscaling and health check



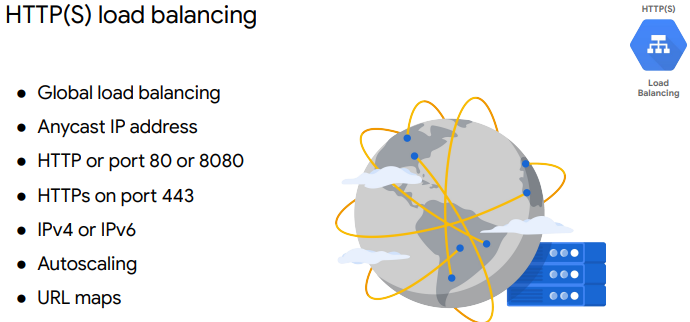
^in above example, autoscaler adds a third instance to split the load to keep CPU utilization below 75% and vice versa

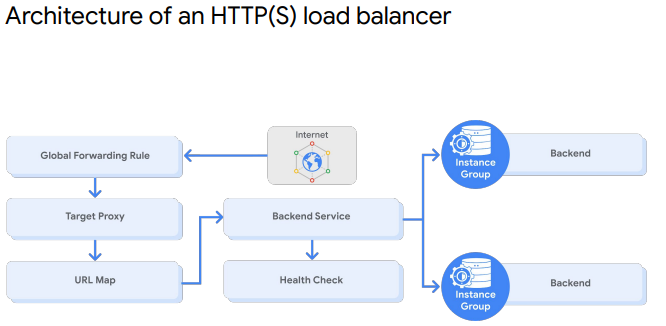
Can monitor the utilisation by clicking on an instance group (or individual VM) and a graph appears. Default is CPU but can see others such as disk and network usage

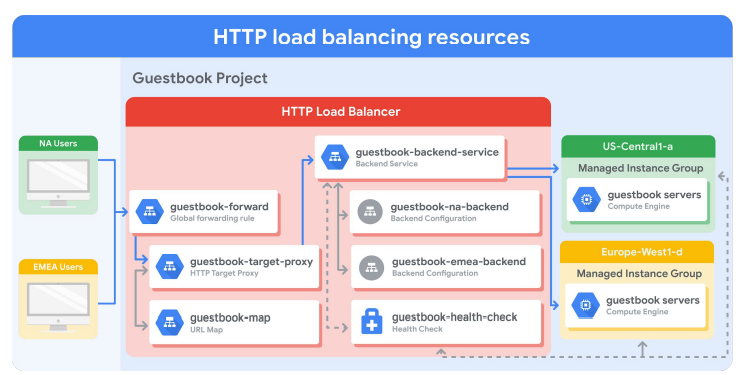
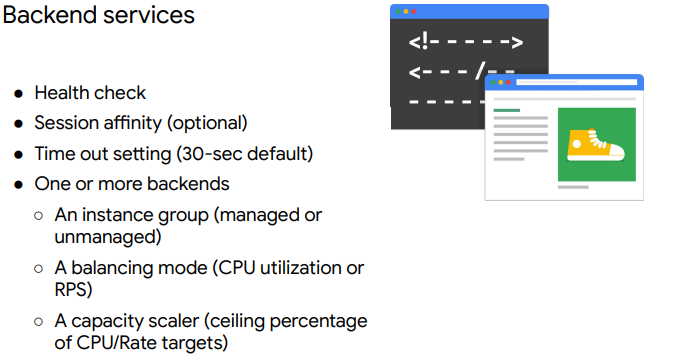


^define a protocol, port and health criteria. This then checks the health of an instance. Health criteria determine how often to check an instance is healthy

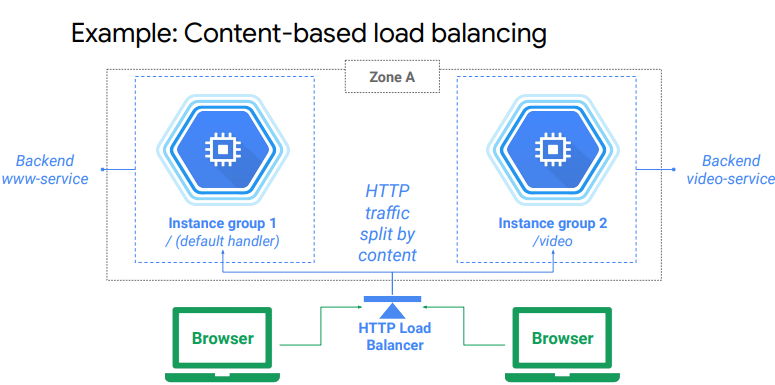
HTTP(S) Load Balancer

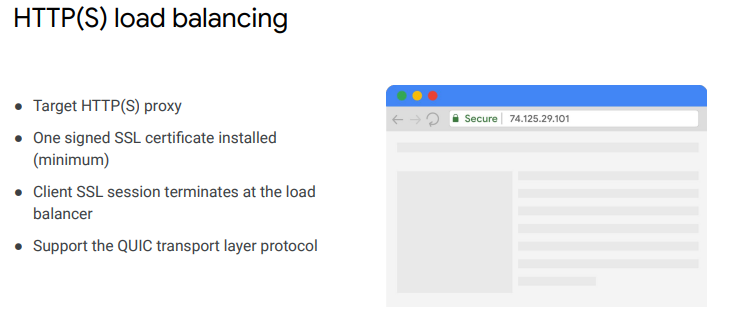






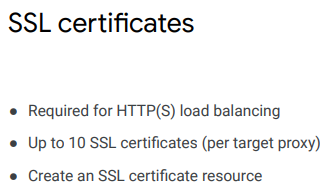
^ if an instance is full, the http load balancer will send the traffic to the next closest instance with capacity (cross region load balancing)



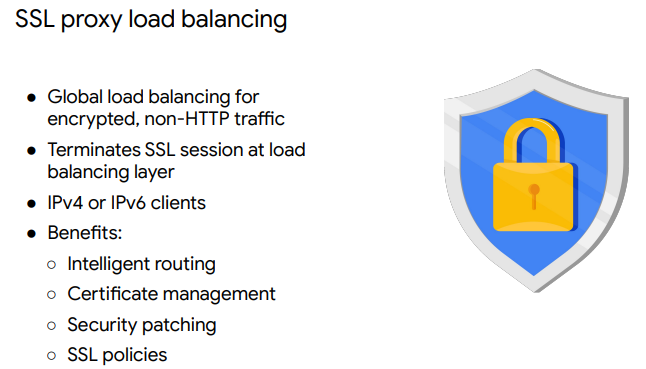


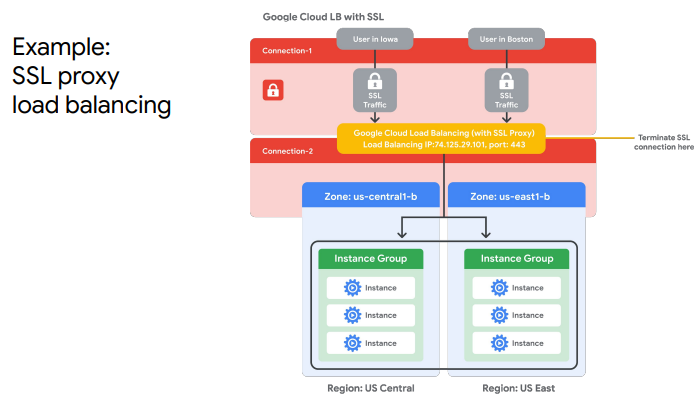
^Differences between HTTP(S) and HTTP load balancing

* An HTTP(S) load balancer uses a target HTTPS proxy instead of a target HTTP proxy
* An HTTP(S) load balancer requires at least one signed SSL certificate installed on the target HTTPS proxy for the load balancer
* The client SSL session terminates at the load balancer
* HTTP(S) load balancers support the QUIC transport layer protocol

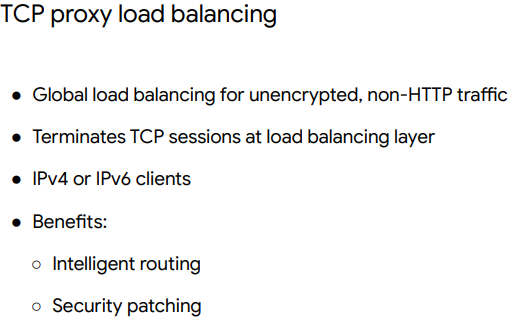


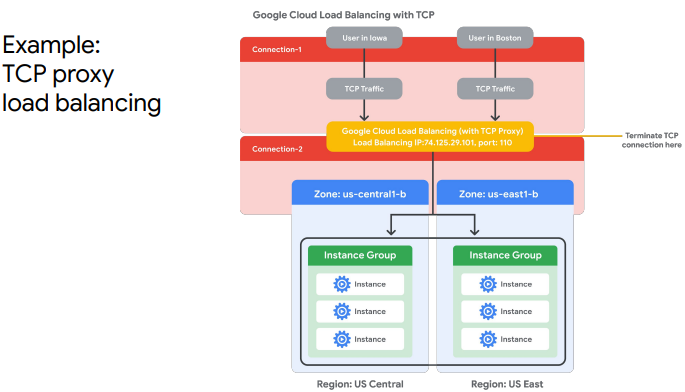
SSL Proxy Load Balancing



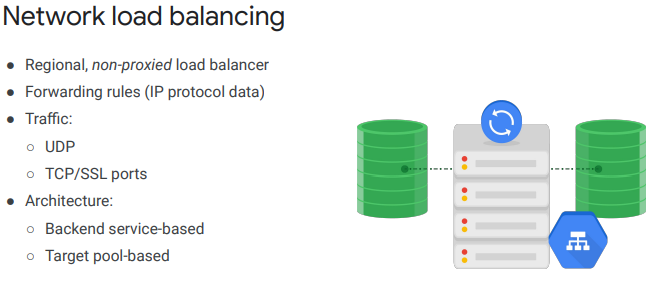


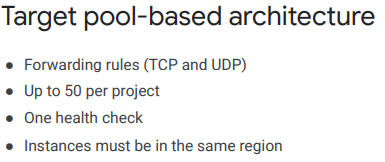
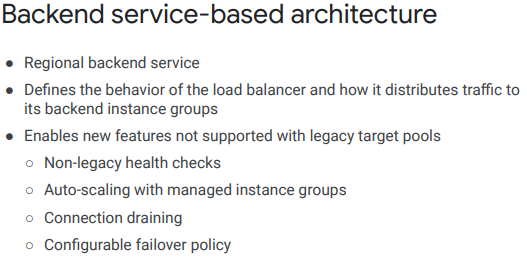
TCP proxy Load balancing





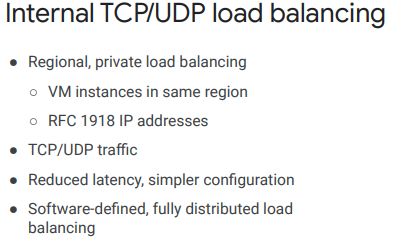
Network Load Balancing

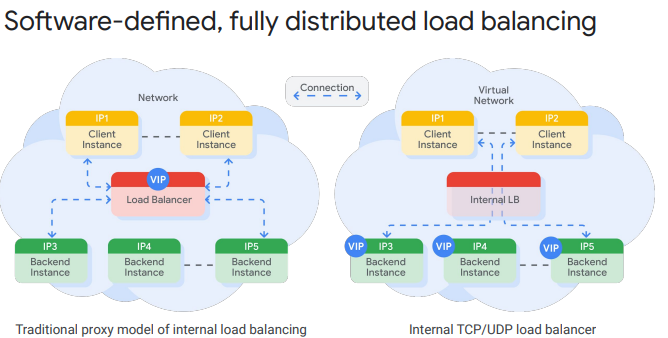


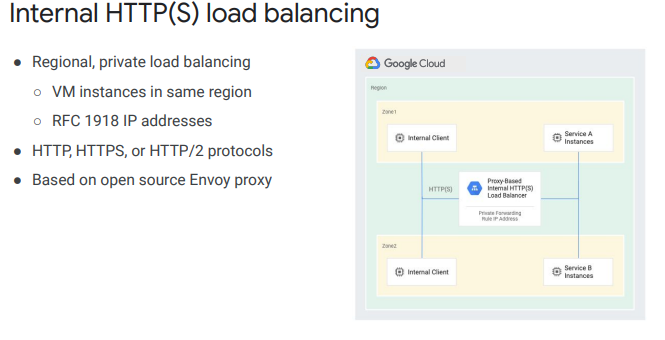


Internal Load Balancing

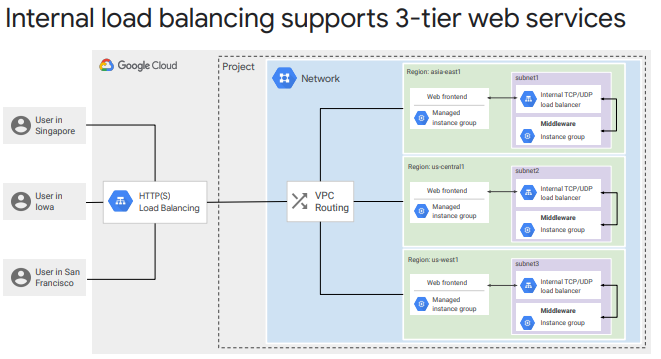
* run and scale your services behind a private load balancing IP address
* only accessible through the internal IP addresses of virtual machine instances that are in the same region
* lower latency as request stay within Google’s network (no external IP addresses)



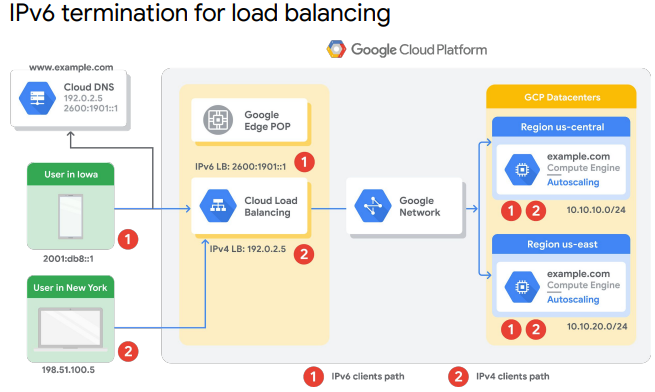




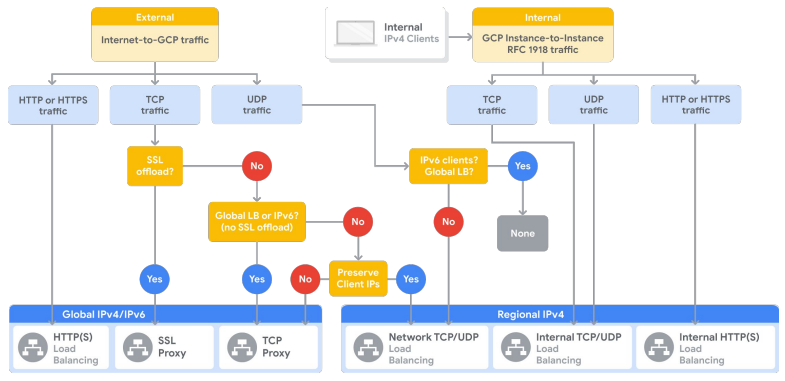
^proxy based, regional, layer 7 load balancer. Enables you to run and scale services behind internal load balancing IP address



Choosing a load balancer



^IPv6 supported by HTTP(s), SSL proxy and TCP proxy load balancing. IPv6 termination for these load balancers enables IPv6 requests to be proxied to your backend IPv4. This is done by terminating the IPv6 connection and placing the same request into an IPv4 connection. On the reverse connection, the IPv4 connection is terminated into an IPv6 connection which is passed to the IPv6 client



**Infrastructure Automation**