GOOGLE CLOUD PLATFORM

LOAD BALANCING

FEATURES

- Ability to distribute load-balanced compute resources in single or multiple high availability regions
- Ability to put your resources behind a single anycast IP and to scale your resources up or down with intelligent Autoscaling
- Ability to serve content as close as possible to your users, on a system that can respond to over 1 million queries per second
- Cloud Load Balancing is fully integrated with Cloud CDN for optimal content delivery.
- Software defined, managed service - It is not instance or device based, so you do not need to manage a physical load balancing infrastructure.

TYPES OF CLOUD LOAD BALANCING

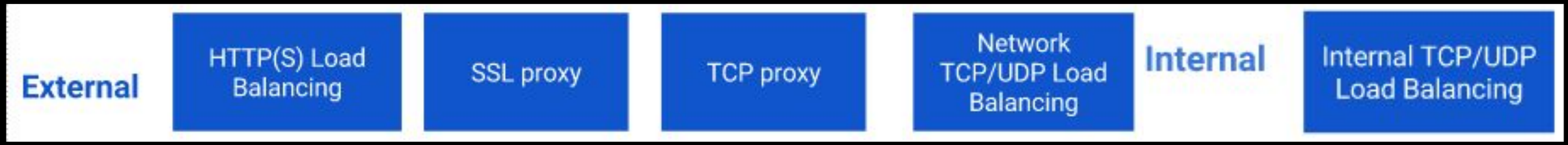
- Types of load balancers
  - Global versus regional load balancing
  - External versus internal load balancing
  - Traffic type

- Global load balancing
  - Users and instances are globally distributed
  - Need access to the same applications and content
  - Want to provide access using a single anycast IP address
  - Need Support for IPv6

- Regional load balancing
  - Regional load balancing is used when users and instances are concentrated in one region and you only require IPv4 termination

TYPES OF CLOUD LOAD BALANCING

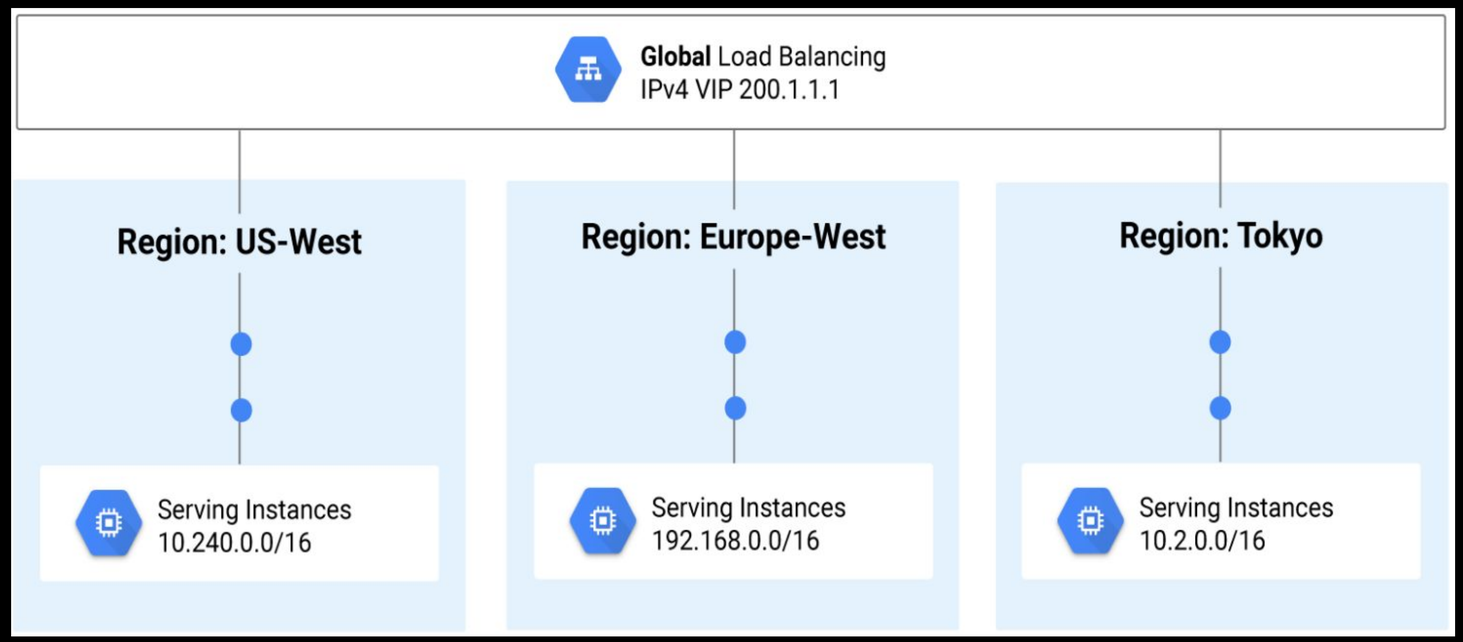
- External versus internal load balancing
  - External load balancers distribute traffic coming from the internet to your GCP network.
  - Internal load balancers distribute traffic within your GCP network.
- Traffic type
  - HTTP/HTTPS traffic require global, external load balancing.
  - TCP traffic can be handled by global, external load balancing; external, regional load balancing; or internal, regional load balancing.
  - UDP traffic can be handled by external regional load balancing or internal regional load balancing.



GLOBAL LOAD BALANCING

- HTTPS, HTTP, or TCP/SSL
- Single anycast IP address
- Instances globally distributed
- Health checks
- IP address and cookie-based affinity
- IPv6 and IPv4 client termination

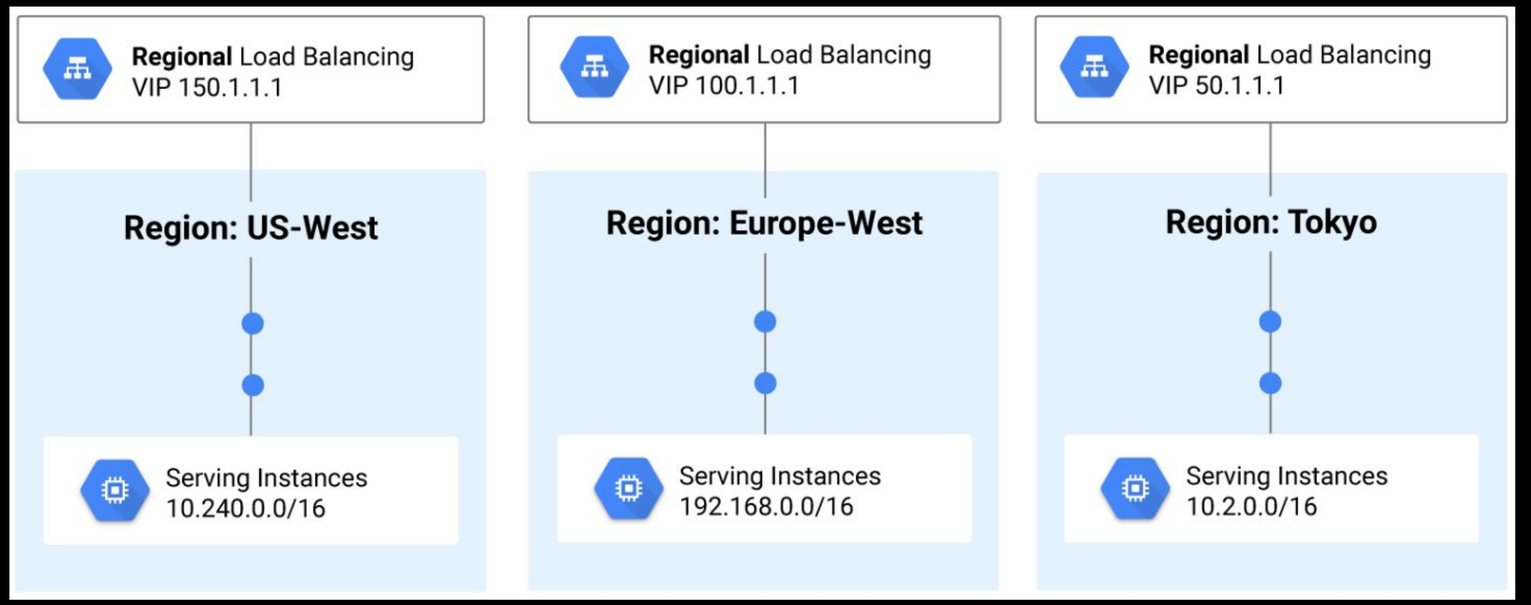
- Connection draining
- Autoscaling
- Monitoring and logging
- Load balancing for cloud storage
- Cross-region overflow and failover
- Requires Premium Tier of Network Service Tiers



REGIONAL LOAD BALANCING

- Internal TCP/UDP Load Balancing
- UDP or TCP/SSL traffic
- Instances in one region
- Single IP address per region
- Health checks

- Session affinity
- IPv4 only
- Autoscaling
- Standard Tier of Network Service Tiers



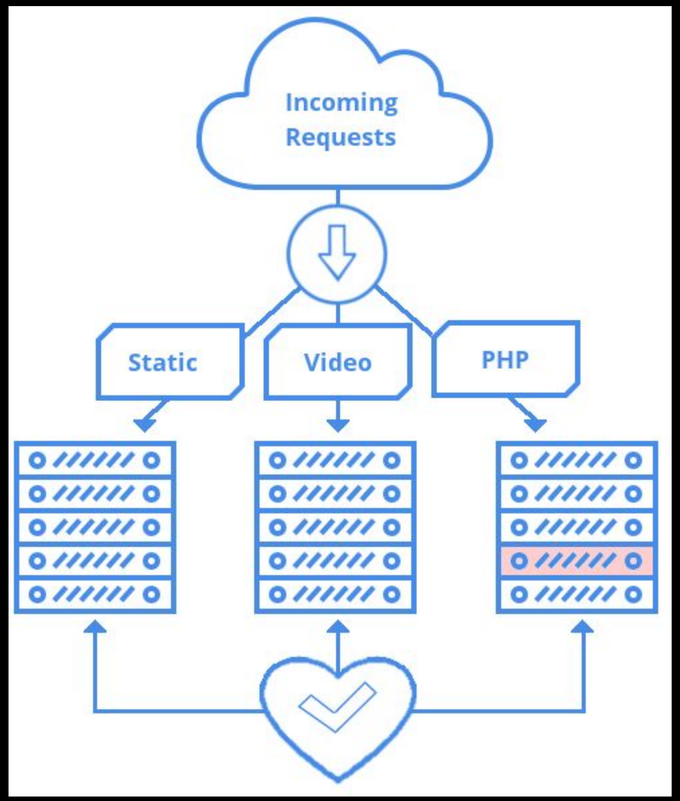
CLOUD LOAD BALANCERS TYPES

- **HTTP(S) Load Balancing**
  - Balances HTTP and HTTPS traffic across multiple backend instances and across multiple regions using a single global IP address, which simplifies DNS setup
- **SSL Proxy Load Balancing**
  - Allows you to enable encryption between your clients and the load balancing layer for non-HTTP(S) traffic
- **TCP Proxy Load Balancing**
  - Global load balancing service for non-HTTP traffic that automatically routes to the instances that are closest to the user.
- **Network TCP/UDP Load Balancing**
  - It is a regional, non-proxied, helps load balance traffic on your systems based on incoming IP protocol data, including address, port, protocol type.

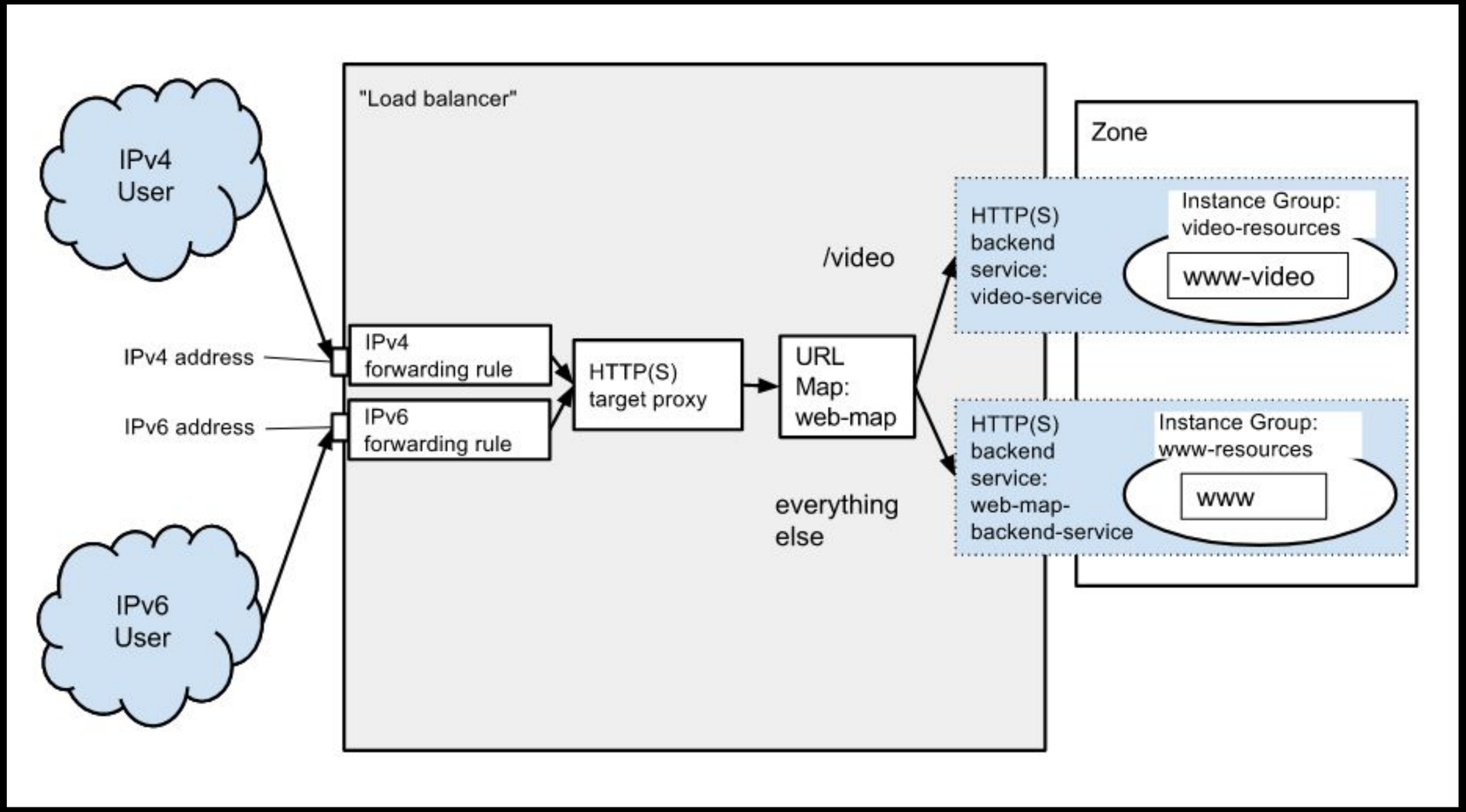
HTTP(S) LOAD BALANCER

- **Global Forwarding Rule** route traffic by IP address, port, and protocol to a load balancing target proxy
- **Target Proxies** route incoming requests to a URL map
- **URL Map** allows traffic to be directed to different matched backend instances
- **Backend Services** are groups of instances configured to deliver files
- **Health Checks** determine whether VM instances respond properly to traffic

- HTTPS load balancer uses a target HTTPS proxy and requires a SSL certificate



DEMO: SETUP HTTP(S) LOAD BALANCER



DEMO: SETUP HTTP(S) LOAD BALANCER

In this demo, we will create a load balancer that routes traffic to different VM instances based on the URL path - those that start with /video vs. the rest

1. Configuring instances
2. Create firewall rules
3. Create a global static external IP address that is the external IP address to reach the load balancer
4. Create Instance groups to hold the VM instances
5. Configuring the load balancing service
    1. Create load balancer
    2. Configure the backend for each instance
    3. Configure the health check for each instance
    4. Configure host and path rules
    5. Configure the frontend
    6. Configure forwarding rules
6. Sending traffic to your instances
7. Once everything is working, modify your firewall rules so HTTP(S) traffic to your instances can only come from your load balancing service.

DEMO: SETUP HTTP(S) LOAD BALANCER

Script for the **Startup script** field while creating the VM's

```
sudo apt-get update
sudo apt-get install apache2 -y
sudo a2ensite default-ssl
sudo a2enmod ssl
sudo service apache2 restart
echo '<!doctype html><html><body><h1>www</h1></body></html>' | sudo tee /var/www/html/index.html
```

```
sudo apt-get update
sudo apt-get install apache2 -y
sudo a2ensite default-ssl
sudo a2enmod ssl
sudo service apache2 restart
echo '<!doctype html><html><body><h1>www-video</h1></body></html>' | sudo tee /var/www/html/index.html
sudo mkdir /var/www/html/video
echo '<!doctype html><html><body><h1>www-video</h1></body></html>' | sudo tee /var/www/html/video/index.html
```

SSL PROXY LOAD BALANCING

- With SSL Proxy Load Balancing, SSL connections are terminated at the load balancing layer then proxied to the closest available instance group.
- SSL Proxy Load Balancing can handle HTTPS traffic, but should be used for other protocols that use SSL, such as Websockets and IMAP over SSL.
- SSL proxy can be deployed globally with instances in multiple regions, and the load balancer automatically directs traffic to the closest region that has capacity.
- Allows for end-to-end encryption for your SSL proxy deployment when you configure your backend service to accept traffic over SSL

SCALING BASED ON HTTP(S) LOAD

- An HTTP(S) load balancer spreads load across backend services, which distributes traffic among instance groups.
- Within the backend service, you can define the load balancing serving capacity of the instance groups associated with the backend.
- When you attach an autoscaler to an HTTP(S) load balancer, the autoscaler will scale the managed instance group to maintain a fraction of the load balancing serving capacity.
- Autoscaling only works with **maximum CPU utilization** and **maximum requests per second/instance** because the value of these settings can be controlled by adding or removing instances.