Network Design

Architecting with Google Cloud Platform: Design and Process

Google Cloud

Last modified 2018-08-08

Agenda

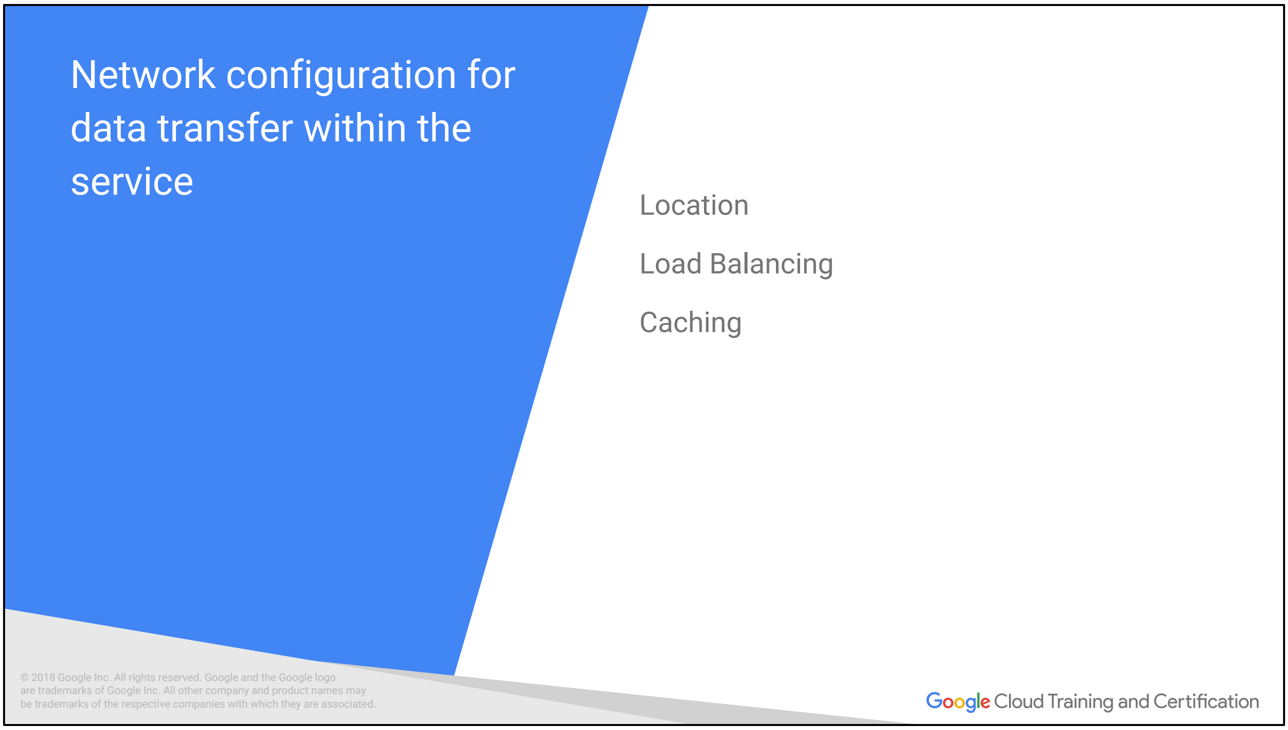
Network configuration for data transfer within the service

Network integration with other environments

Photo service: periodic slowdown

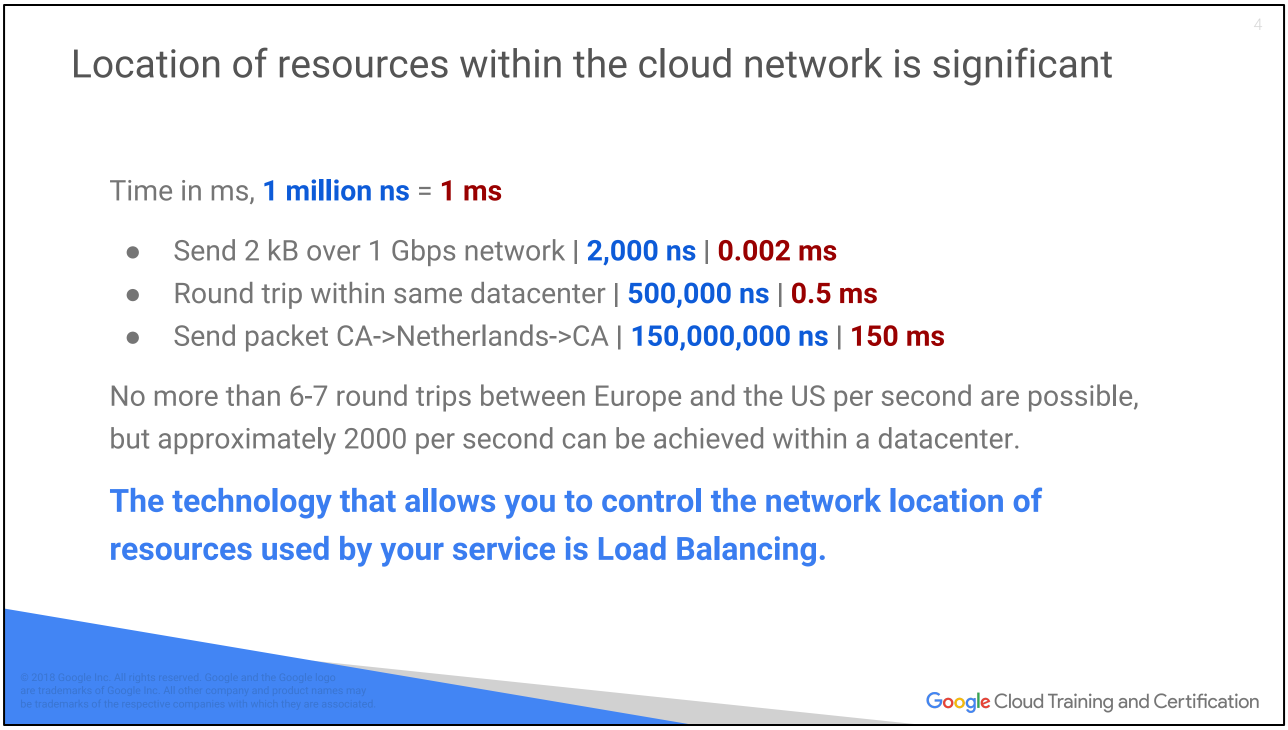
Design challenge #3: Growth

GCP lab Deployment Manager: Adding load balancing



Network configuration for data transfer within the service

- Location
- Load Balancing
- Caching



Location of resources within the cloud network is significant

Time in ms, **1 million ns** = **1 ms**

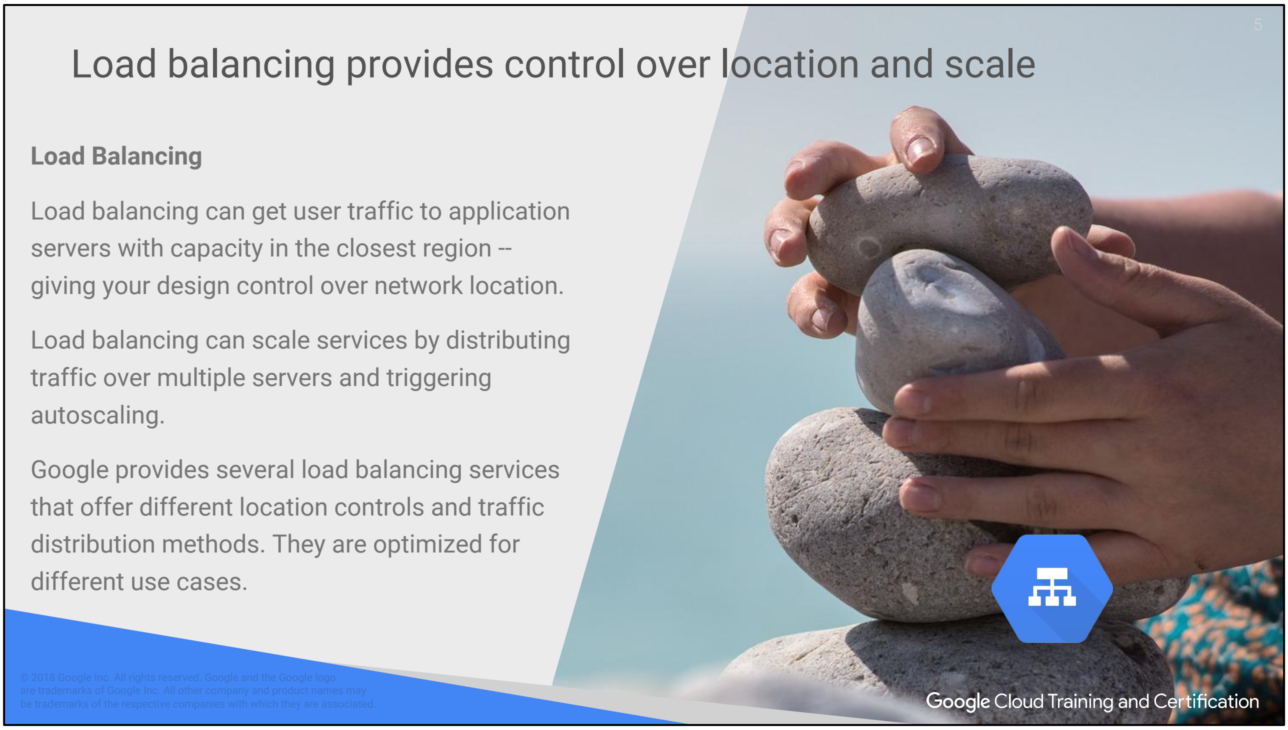
- Send 2 kB over 1 Gbps network | **2,000 ns** | **0.002 ms**
- Round trip within same datacenter | **500,000 ns** | **0.5 ms**
- Send packet CA->Netherlands->CA | **150,000,000 ns** | **150 ms**

No more than 6-7 round trips between Europe and the US per second are possible, but approximately 2000 per second can be achieved within a datacenter.

The technology that allows you to control the network location of resources used by your service is Load Balancing.

Location is significant. Only in this case, you pay more for something that is farther away.
Note: Describes VM-to-VM communications inside the Google Network.

You can use performance testing tools such as **iperf** to test timing.



Load balancing provides control over location and scale

Load Balancing

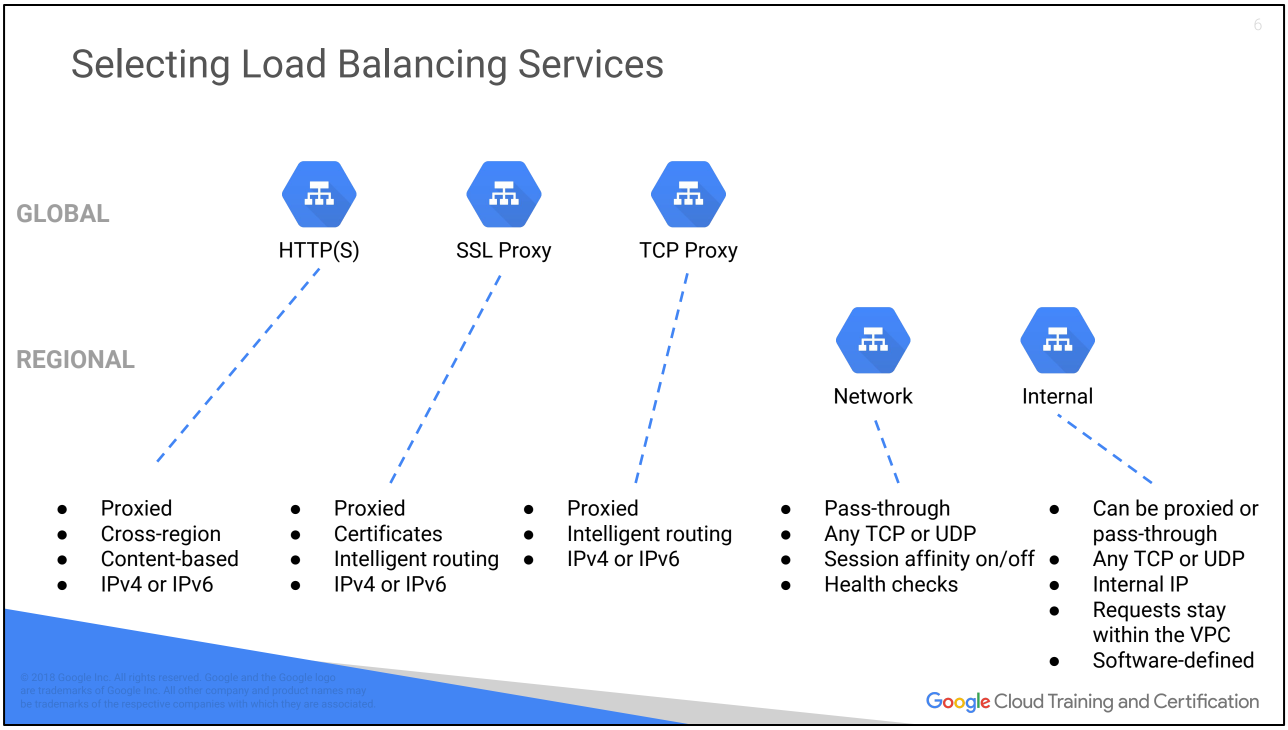
Load balancing can get user traffic to application servers with capacity in the closest region -- giving your design control over network location.

Load balancing can scale services by distributing traffic over multiple servers and triggering autoscaling.

Google provides several load balancing services that offer different location controls and traffic distribution methods. They are optimized for different use cases.

Network speed is just one factor in throughput. Network location is key. Parallelism is another factor. And load balancing combines both

https://pixabay.com/en/meditation-stone-towers-stone-tower-2262835/



Selecting Load Balancing Services

GLOBAL

REGIONAL

HTTP(S)

- Proxied
- Cross-region
- Content-based
- IPv4 or IPv6

SSL Proxy

- Proxied
- Certificates
- Intelligent routing
- IPv4 or IPv6

TCP Proxy

- Proxied
- Intelligent routing
- IPv4 or IPv6

Network

- Pass-through
- Any TCP or UDP
- Session affinity on/off
- Health checks

Internal

- Can be proxied or pass-through
- Any TCP or UDP
- Internal IP
- Requests stay within the VPC
- Software-defined

Google Cloud Training and Certification

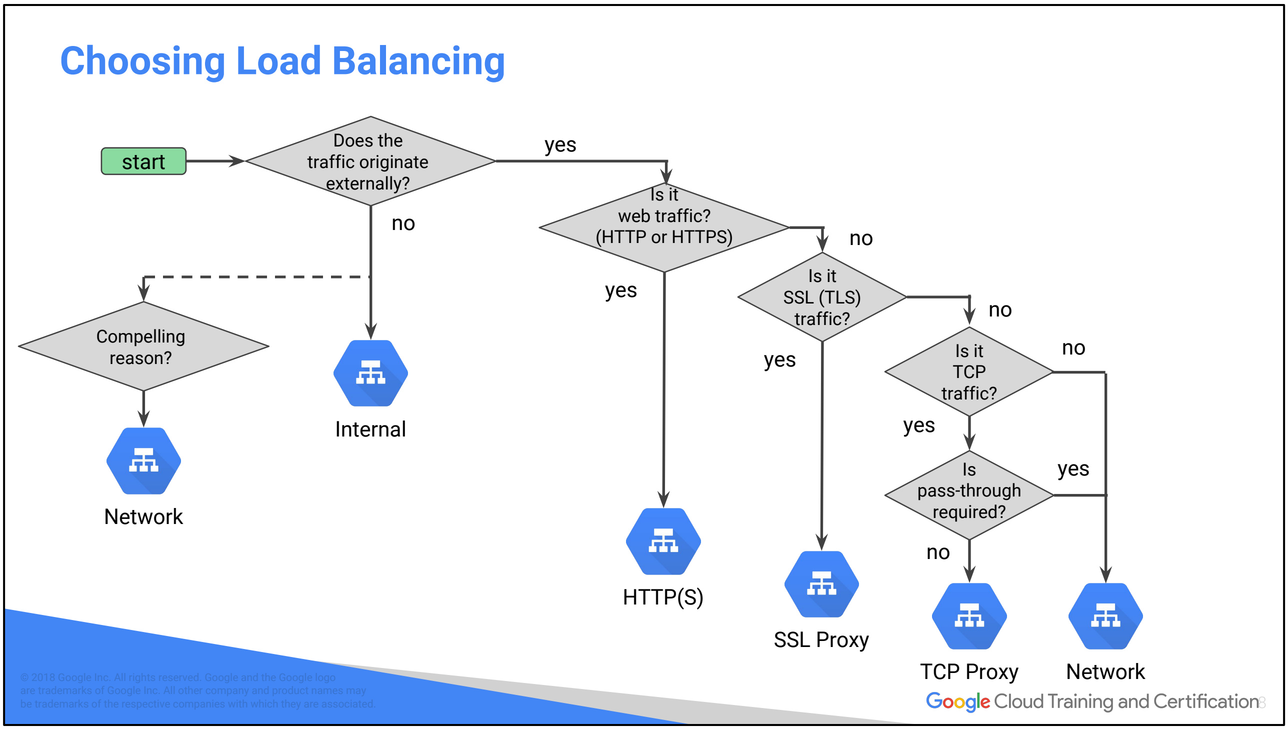
Load balancing is either proxied or pass-through. A proxied load balancer terminates the incoming connection and initiates a separate connection to the target (usually SSL or TCP). A non-proxied or pass-through load balancer redirects and distributes the traffic without terminating and initiating a separate connection. Global load balancers can direct traffic to the closest region with capacity, whereas regional load balancers direct traffic to or within a single region.
https://cloud.google.com/compute/docs/load-balancing/

HTTP(S) processes requests on port 80 or port 8080
SSL proxy supports the following ports: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995
TCP Proxy supports the following ports: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 99

Intelligent routing means that capacity is considered in the routing decision.
Internal load balancing is software-defined, meaning that there is no hardware interface to serve as a choke point or risk load balancer availability.

In general, for traffic originating externally, stick to the protocol-named service that is designed for and optimized for that protocol, unless you have a compelling reason.
For multi-tier internal traffic, use the internal load balancing service.
Then use the more general network load balancing service for anything else.

Maglev research paper: https://research.google.com/pubs/pub44824.html



Choosing Load Balancing

start

Does the traffic originate externally?

yes

Is it web traffic? (HTTP or HTTPS)

no

Is it SSL (TLS) traffic?

no

Is it TCP traffic?

no

yes

Is pass-through required?

yes

no

Network

no

Compelling reason?

Network

Internal

yes

HTTP(S)

yes

SSL Proxy

TCP Proxy

Google Cloud Training and Certification

HTTP(S) load balancing: You can configure URL rules that route some URLs to one set of instances and route other URLs to other instances. Requests are always routed to the instance group that has capacity and is closest to the user.
https://cloud.google.com/compute/docs/load-balancing/http/

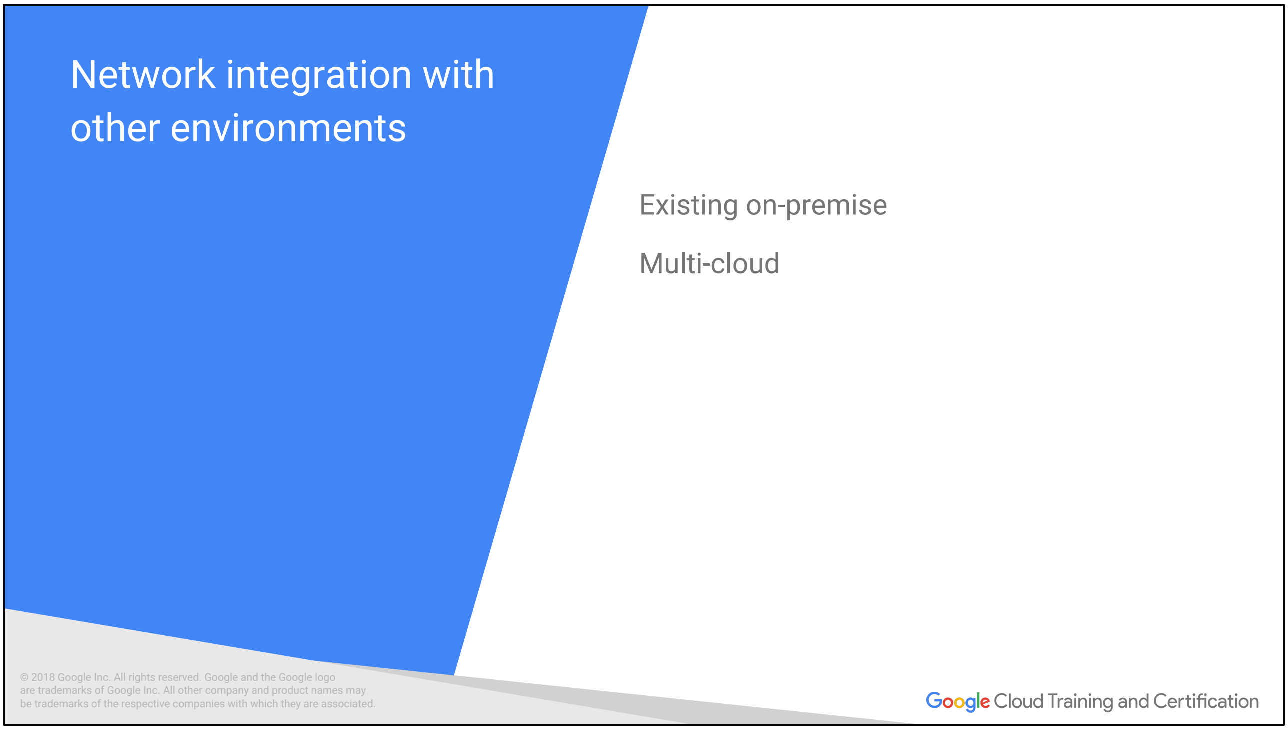
SSL proxy: A proxied global load balancing service that automatically directs SSL traffic to the closest region that has capacity.
https://cloud.google.com/compute/docs/load-balancing/tcp-ssl/

TCP proxy: Terminates IPv4 and IPv6 and initiates an IPv4 connection to the backend servers.
https://cloud.google.com/compute/docs/load-balancing/tcp-ssl/tcp-proxy

Network load balancing allows you to balance load of your systems based on incoming IP protocol data, such as address, port, and protocol type.
https://cloud.google.com/compute/docs/load-balancing/network/

Internal load balancing enables you to run and scale your services behind a private load balancing IP address which is accessible only to instances internal to your Virtual Private Cloud (VPC). https://cloud.google.com/compute/docs/load-balancing/internal/

Network load balancing was used for internal load balancing before the internal load balancing service was available. Configuration is significantly more complicated with network load balancing because you have to restrict access to the VPC using firewall rules and routes. You also must be plan for capacity of the load balancer itself, because choke points are possible and the load balancer could reach capacity and impact availability. If there is some reason Internal load balancing won't work in your situation, network load balancing is still an alternative. However, there are no common use cases.

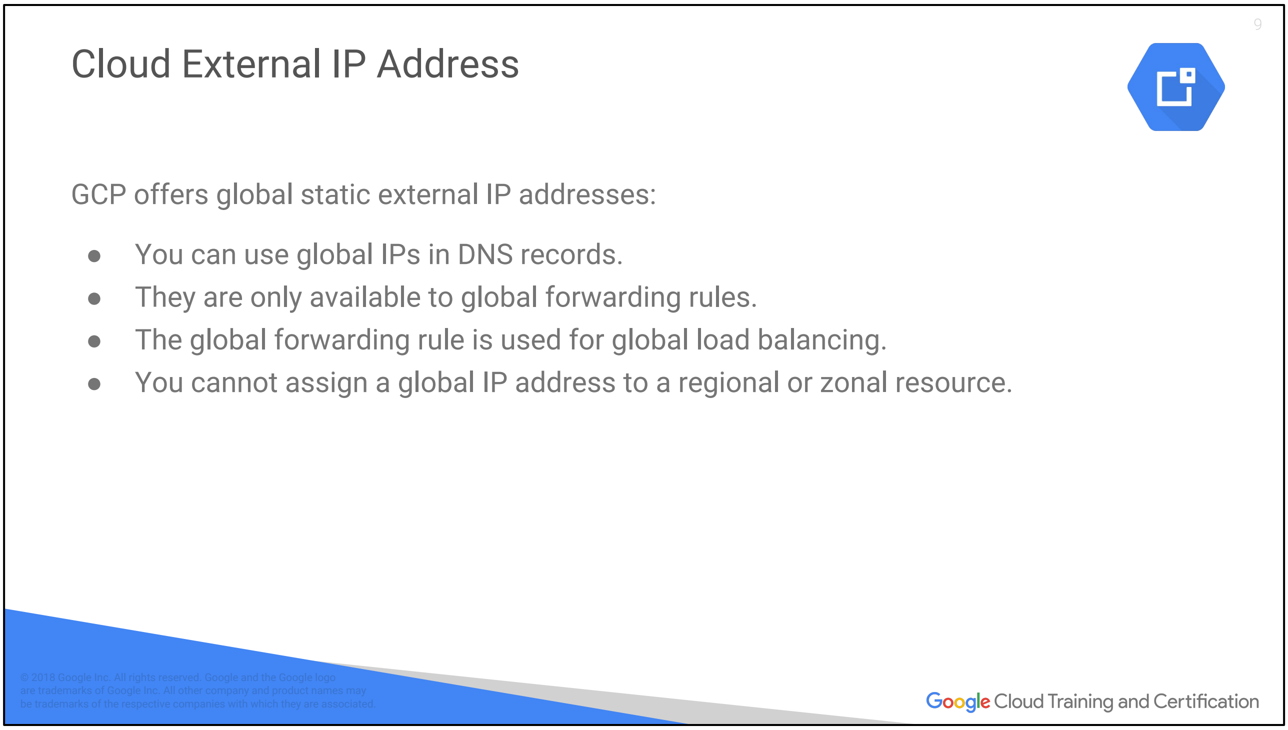


Network integration with other environments

Existing on-premise

Multi-cloud

Google Cloud Training and Certification



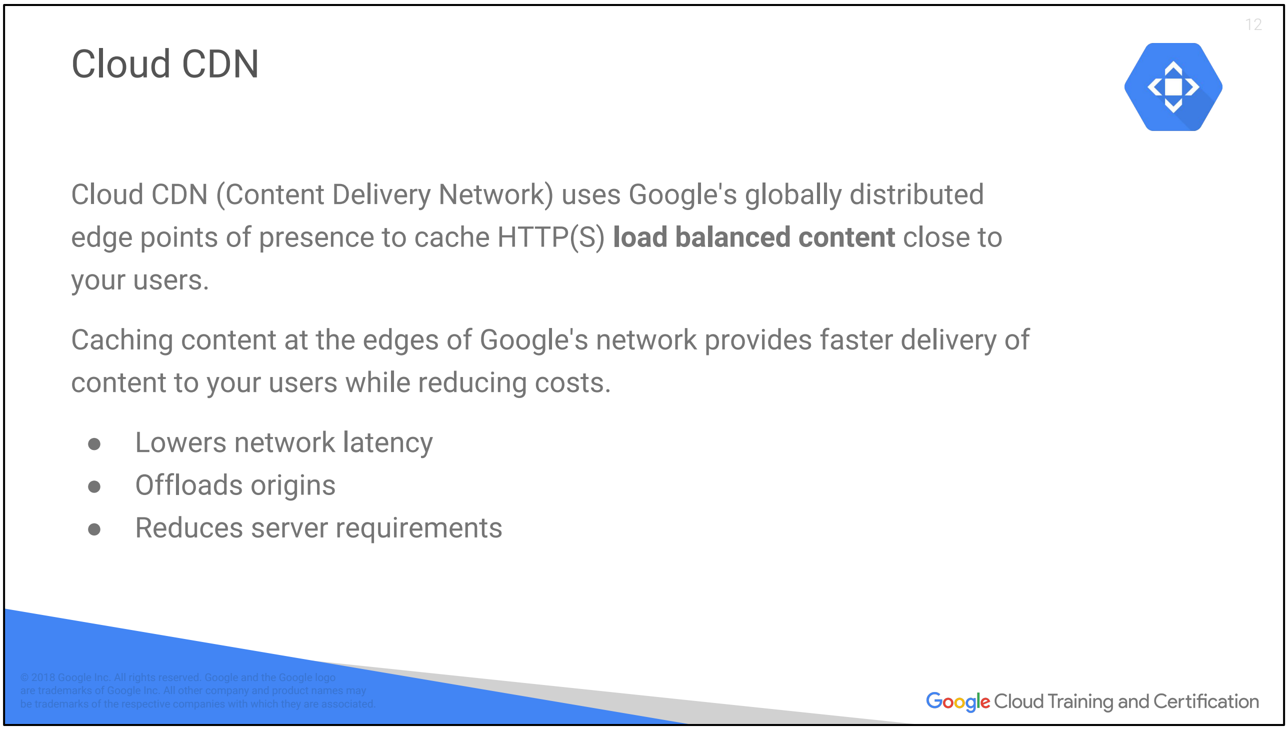
Cloud External IP Address

GCP offers global static external IP addresses:

- You can use global IPs in DNS records.
- They are only available to global forwarding rules.
- The global forwarding rule is used for global load balancing.
- You cannot assign a global IP address to a regional or zonal resource.

Google Cloud Training and Certification

To get a global static IP address for a GCP resource, configure global load balancing.



Cloud CDN

Cloud CDN (Content Delivery Network) uses Google's globally distributed edge points of presence to cache HTTP(S) **load balanced content** close to your users.

Caching content at the edges of Google's network provides faster delivery of content to your users while reducing costs.

- Lowers network latency
- Offloads origins
- Reduces server requirements

Google Cloud Training and Certification

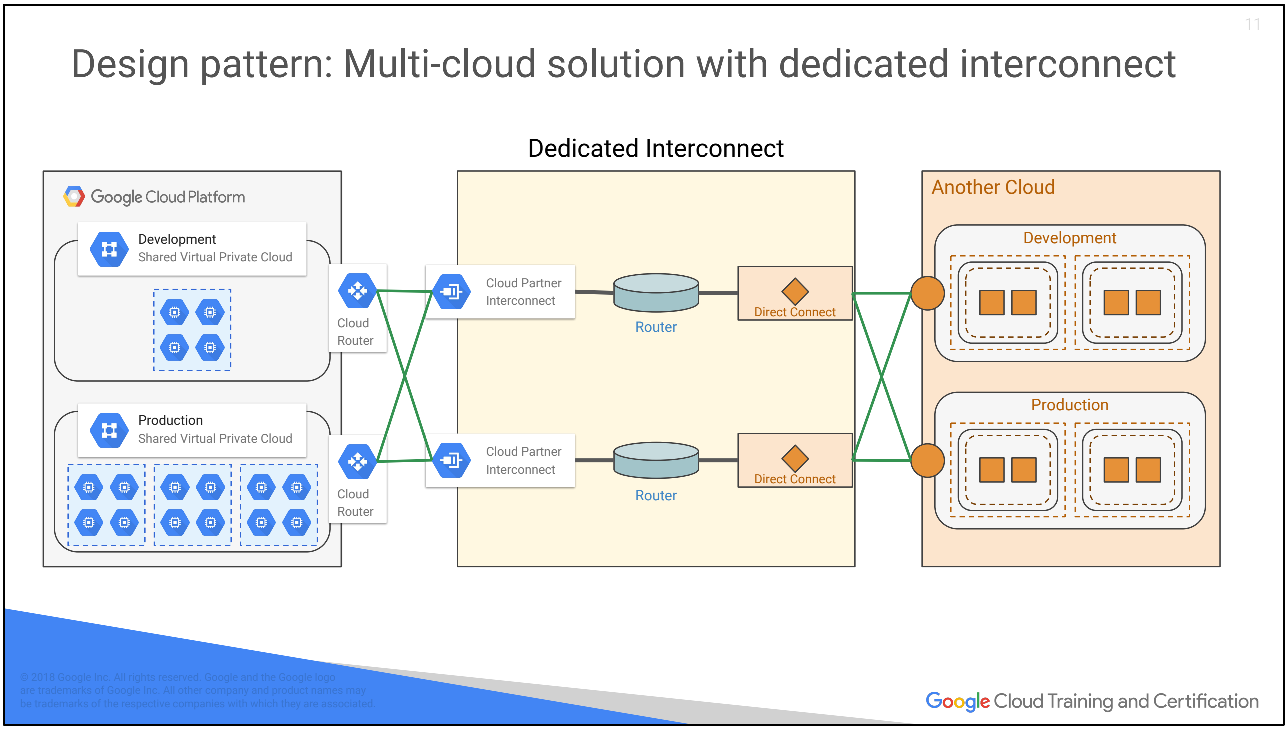
In certain circumstances caching can be a design issue. For example, if a value your application relies on is cached and you want to roll out a new version of the application that changes the value, the cached value could create issues that are difficult to troubleshoot.

In general, Cloud CDN handles caching transparently.

If you decide to use a 3rd party or open source cache as part of your solution, please investigate cache management.

Advice on 3rd party or open source cache products:

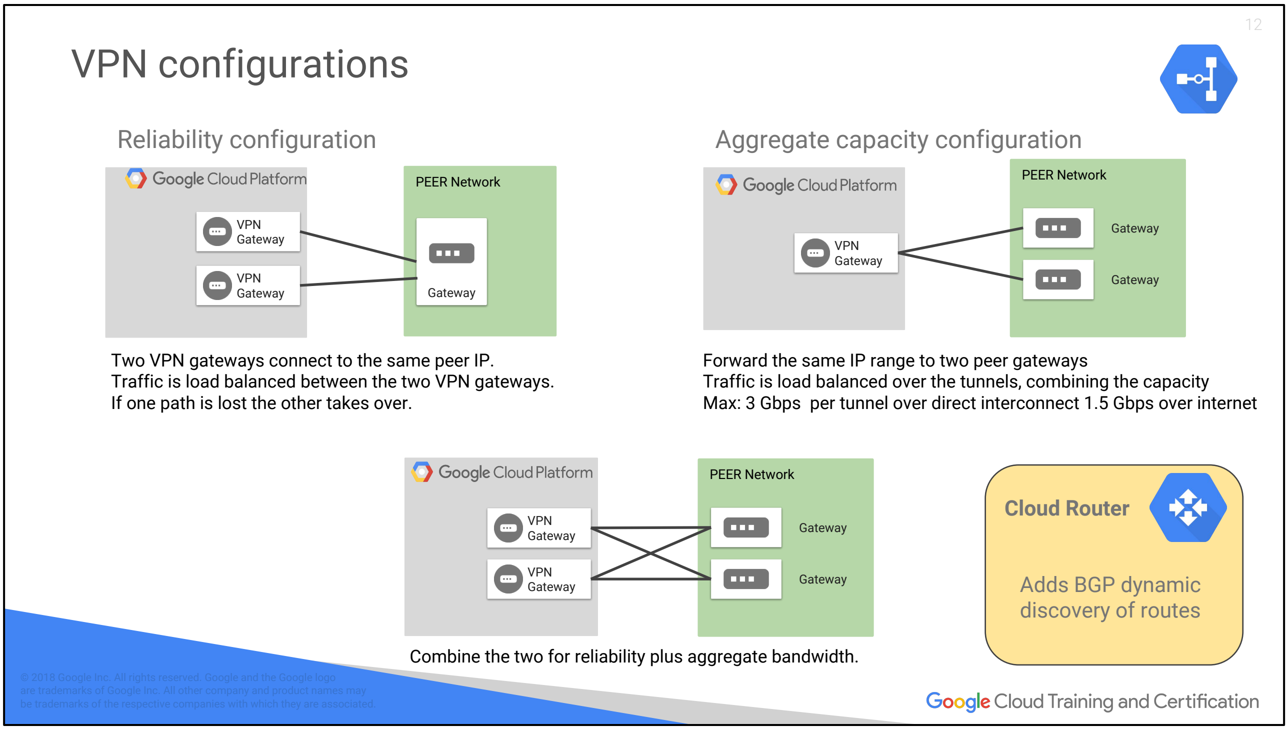
- Know the type of cache
- Performance (latency) cache: cache exists to serve data with lower average latency than from the backend.
- Capacity (throughput) cache: cache exists in order to serve higher throughput than the backend can deal with.
- Startup from cold is difficult, gradual ramp of traffic is required
- Be careful in use of caches if strict consistency matters
- Cache invalidation is complicated Cache invalidation is a process (purge, refresh, or ban) whereby entries in a cache are replaced or removed. It can be done explicitly, as part of a cache coherence protocol. https://en.wikipedia.org/wiki/Cache_invalidation



Design pattern: Multi-cloud solution with dedicated interconnect

Google Cloud Training and Certification

https://cloud.google.com/interconnect/docs



VPN configurations

Reliability configuration

Two VPN gateways connect to the same peer IP.
Traffic is load balanced between the two VPN gateways.
If one path is lost the other takes over.

Aggregate capacity configuration

Forward the same IP range to two peer gateways
Traffic is load balanced over the tunnels, combining the capacity
Max: 3 Gbps per tunnel over direct interconnect 1.5 Gbps over internet

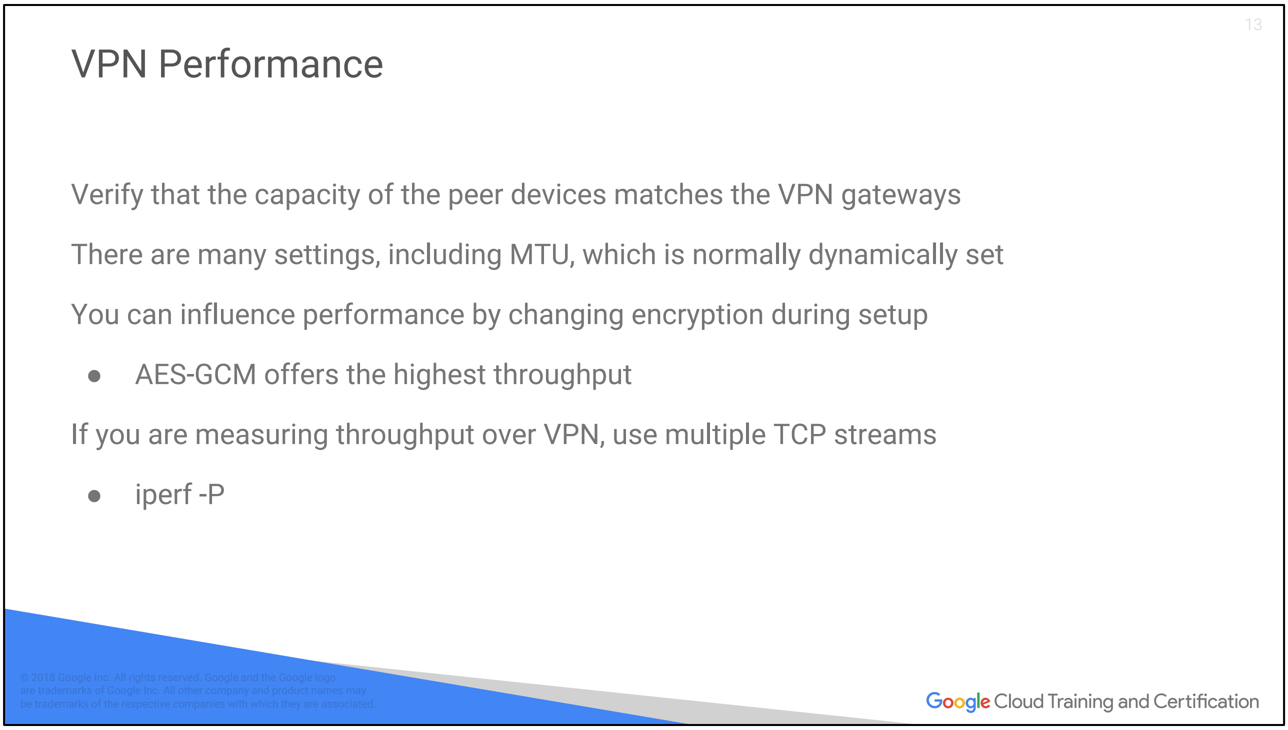
Combine the two for reliability plus aggregate bandwidth.

Cloud Router

Adds BGP dynamic discovery of routes

Google Cloud Training and Certification

https://cloud.google.com/compute/docs/vpn/advanced



VPN Performance

Verify that the capacity of the peer devices matches the VPN gateways

There are many settings, including MTU, which is normally dynamically set

You can influence performance by changing encryption during setup

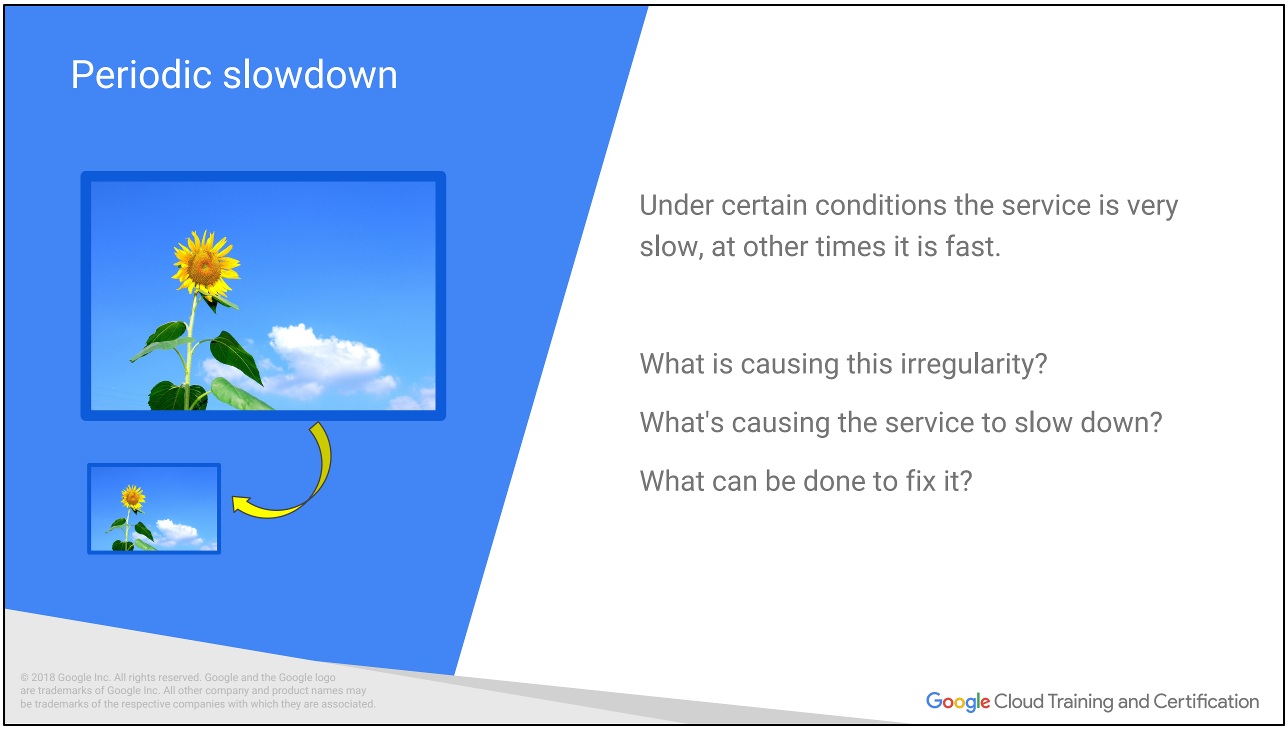
- AES-GCM offers the highest throughput

If you are measuring throughput over VPN, use multiple TCP streams

- iperf -P

Google Cloud Training and Certification

https://cloud.google.com/compute/docs/vpn/advanced#recommended_measures_to_increase_vpn_throughput



Periodic slowdown

Under certain conditions the service is very slow, at other times it is fast.

What is causing this irregularity?

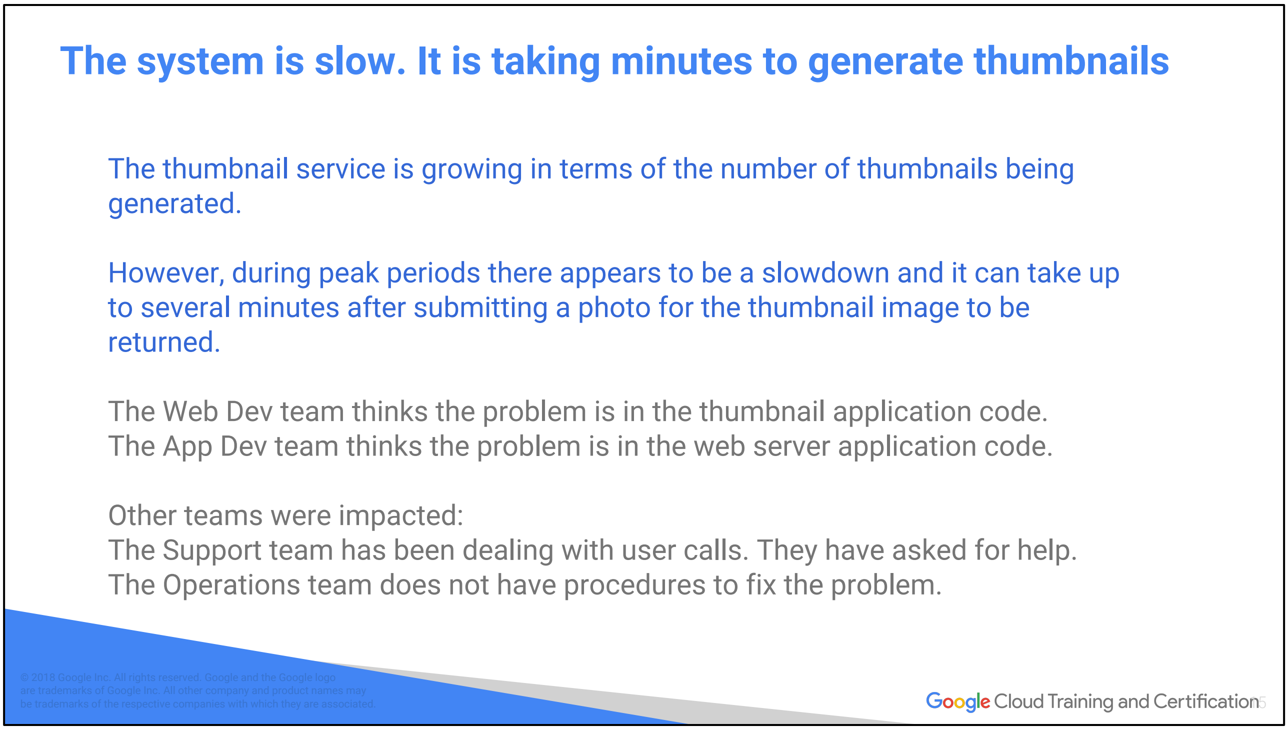
What's causing the service to slow down?

What can be done to fix it?

Google Cloud Training and Certification

Okay, so let's go back to our photo service. In this case, we have a periodic slowdown, which means that under certain conditions the service is very slow, but at other times it's fast. So, what could be causing this irregularity? What's causing the service itself to slow down? And, what can we do to fix it?

https://pixabay.com/en/summer-sunflower-flowers-sky-cloud-368224/



The system is slow. It is taking minutes to generate thumbnails

The thumbnail service is growing in terms of the number of thumbnails being generated.

However, during peak periods there appears to be a slowdown and it can take up to several minutes after submitting a photo for the thumbnail image to be returned.

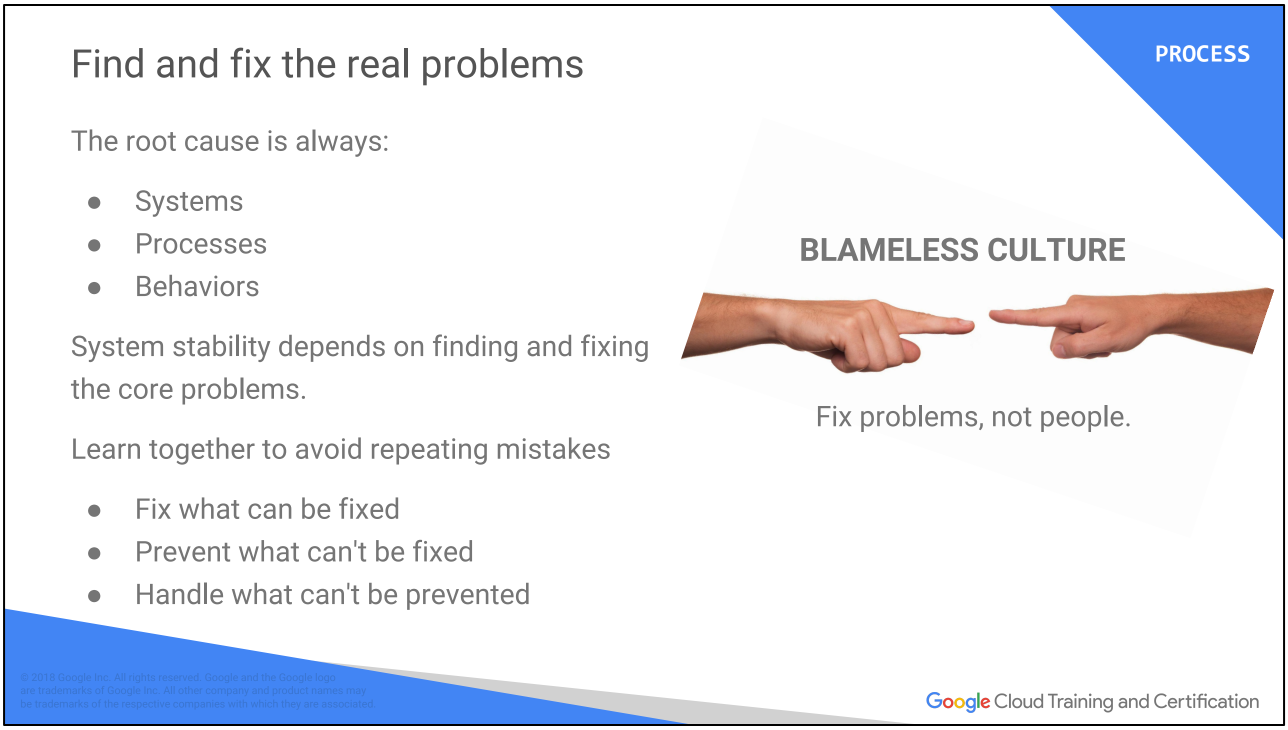
The Web Dev team thinks the problem is in the thumbnail application code.
The App Dev team thinks the problem is in the web server application code.

Other teams were impacted:
The Support team has been dealing with user calls. They have asked for help.
The Operations team does not have procedures to fix the problem.

So, the thumbnail service is growing and this can be seen through the number of thumbnails being generated, which is great. We're starting to get more popularity, but monitoring our log processing shows that there appears to be a slowdown during peak periods, and it can take up to several minutes after submitting a photo for the thumbnail image to be returned.

Now, this doesn’t really happen, but let’s take a fictitious scenario where a company has groups of teams that don't really get along, or will blame each other. So in this case, the Web Dev team thinks that the problem is the thumbnail application code. Well, guess what? The people who wrote the code - the App Dev team - think the problems in the web server application code, because it might not be handling sessions and so on.

But there are other teams that were impacted too. The Support team are dealing with user calls, so they're calling and asking for help. The Operations team doesn't have a procedure to fix the problem because they're the ones who manage the deployment and the production servers. They're not sure if it's the Web teams fault or the App team's fault. So, who's going to fix this?



PROCESS

Find and fix the real problems

The root cause is always:

- Systems
- Processes
- Behaviors

System stability depends on finding and fixing the core problems.

Learn together to avoid repeating mistakes

- Fix what can be fixed
- Prevent what can't be fixed
- Handle what can't be prevented

BLAMELESS CULTURE

Fix problems, not people.

Google Cloud Training and Certification

Google learned that the reliability of the service depends on how people work together to fix problems. Every new system or upgraded system goes through a period of stabilization. During that period you will need to respond to problems, find the root causes, and address the problems. If you stop looking after you have assigned blame to a person, but don't continue digging until you get to the systems, processes, or behaviors that must be changed -- you will leave the system broken, *and it will not stabilize.*

Consider this example:

The service was out.
1 .Why was the service out? Because the filesystem was full.
2. Why was the filesystem full? Because the person responsible for archiving old files failed to do so.
3. Why didn't they archive the old files, was the archive tool broken? No the tool wasn't broken.
4. Why didn't they archive the old files? The procedure documentation didn't tell them to archive the files.
So the root cause was a problem with the process documentation.

If the analysis had stopped at 2, the person might have been punished without solving the core system problem, which was absent procedures.

The service doesn't stabilize if you don't find and fix the real problems.

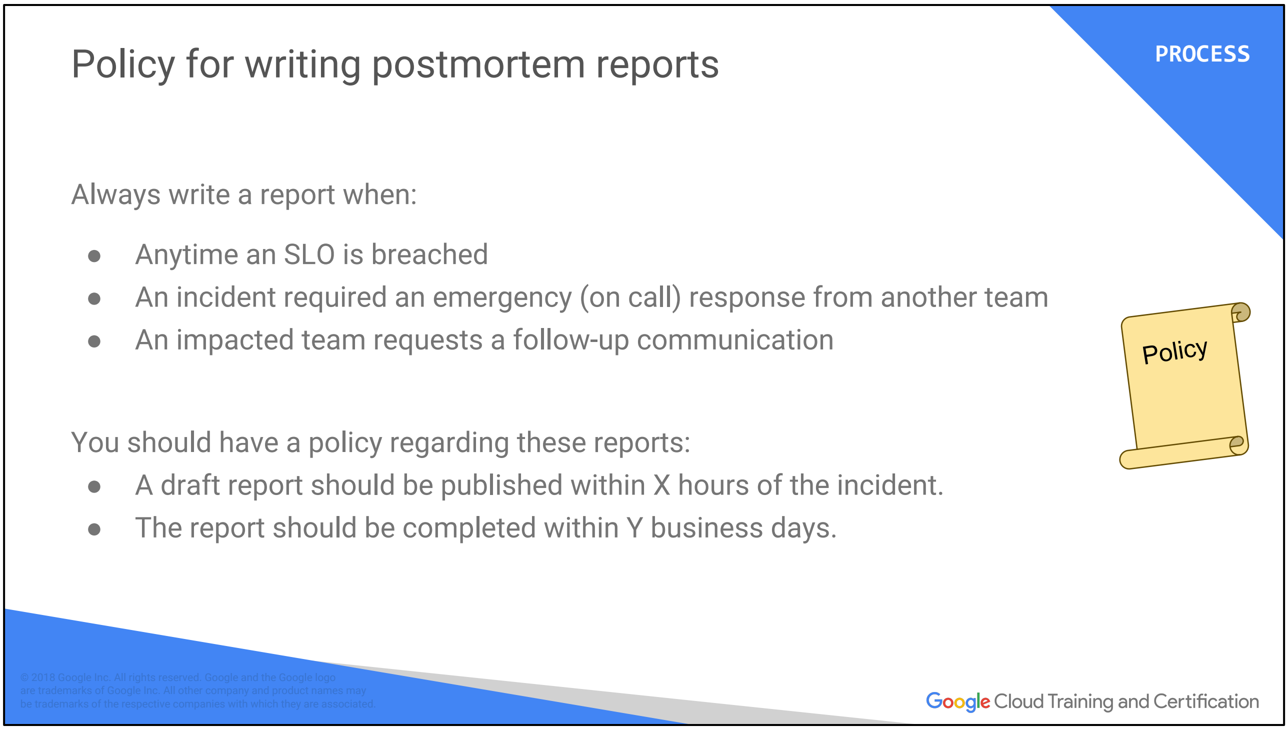
Learning together
Outages happen. And what may be clear to one person may not be clear to another. Some outcomes would not have been anticipated by anyone.

Blameless
Blame makes people afraid to bring real issues to light and is detrimental to a learning culture.
People are NEVER the root cause. There is something in the system, in the processes, or in the behaviors that IS the root cause and needs to be identified and fixed or mitigated.

Identify the actions that led to the incident.
Stick to the facts.
Keep communications simple. Use passive tense.
Consider modifying the auditing process.

Reference: SRE Book: Chapter 15 - Postmortem Culture: Learning from Failure

https://pixabay.com/en/pointing-accusation-accuse-blame-1991215/



PROCESS

Policy for writing postmortem reports

Always write a report when:

- Anytime an SLO is breached
- An incident required an emergency (on call) response from another team
- An impacted team requests a follow-up communication

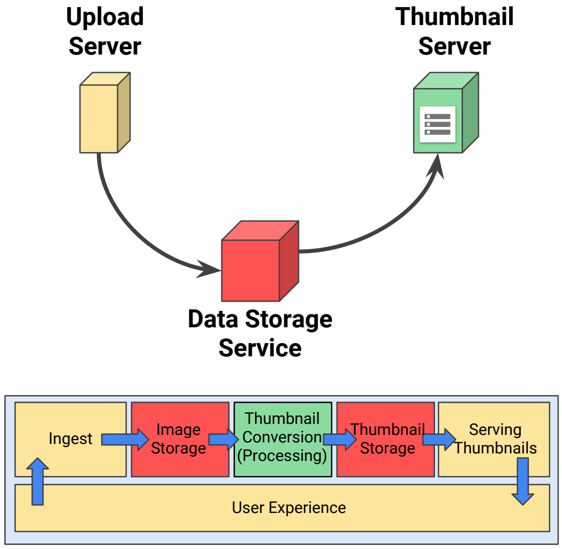
You should have a policy regarding these reports:

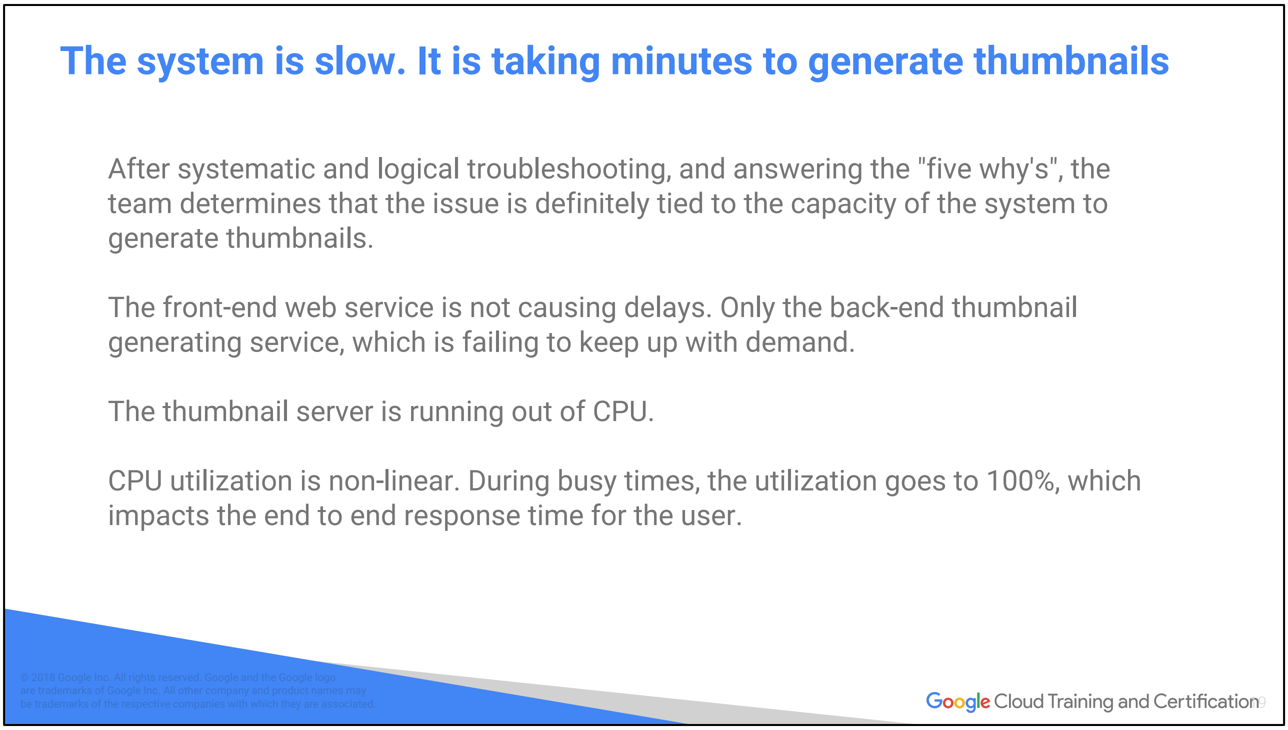
- A draft report should be published within X hours of the incident.
- The report should be completed within Y business days.

Google Cloud Training and Certification

It is important that teams learn from mistakes. Each postmortem written and read reduces the chances of repeating mistakes. Postmortem reports become a method for training people.

Refresher


Upload Server
Thumbnail Server
Data Storage Service
Ingest
Image Storage
Thumbnail Conversion (Processing)
Thumbnail Storage
Serving Thumbnails
User Experience



The system is slow. It is taking minutes to generate thumbnails

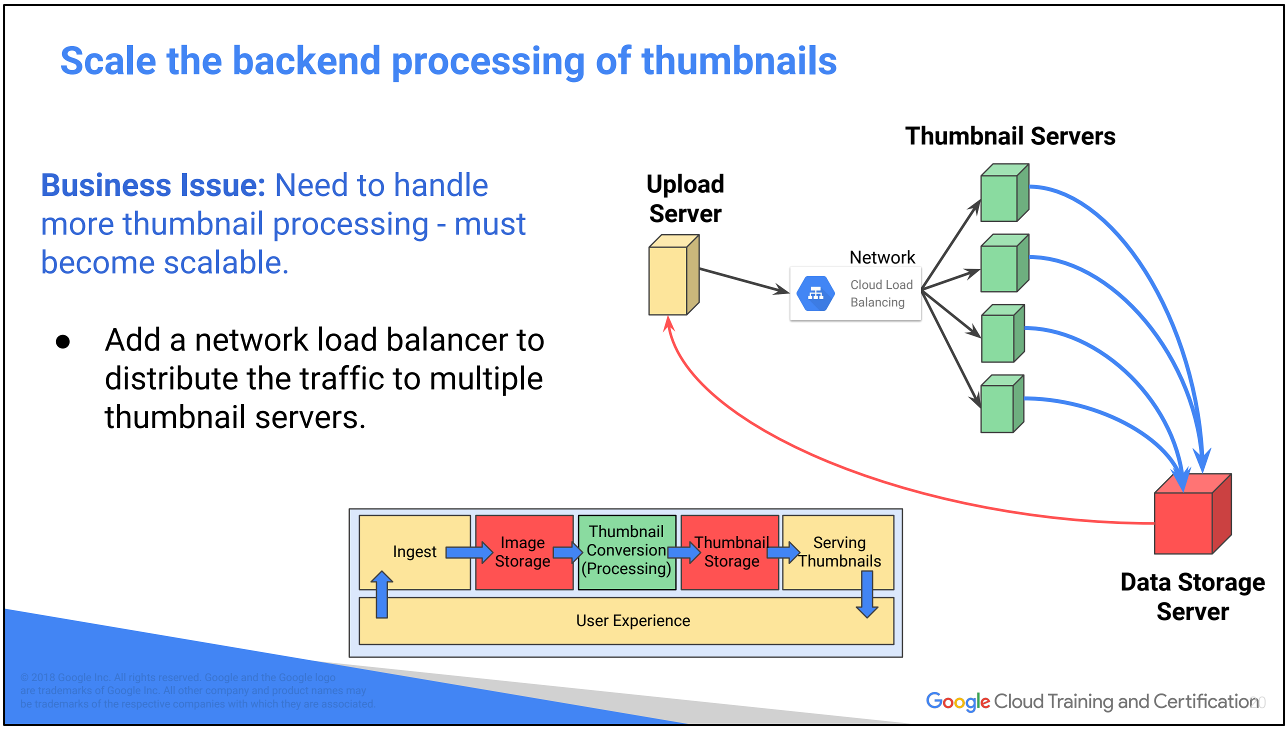
After systematic and logical troubleshooting, and answering the "five why's", the team determines that the issue is definitely tied to the capacity of the system to generate thumbnails.

The front-end web service is not causing delays. Only the back-end thumbnail generating service, which is failing to keep up with demand.

The thumbnail server is running out of CPU.

CPU utilization is non-linear. During busy times, the utilization goes to 100%, which impacts the end to end response time for the user.

Keep in mind that CPU utilization is not to be used as a service level indicator. It is not a direct measurement of customer pain.



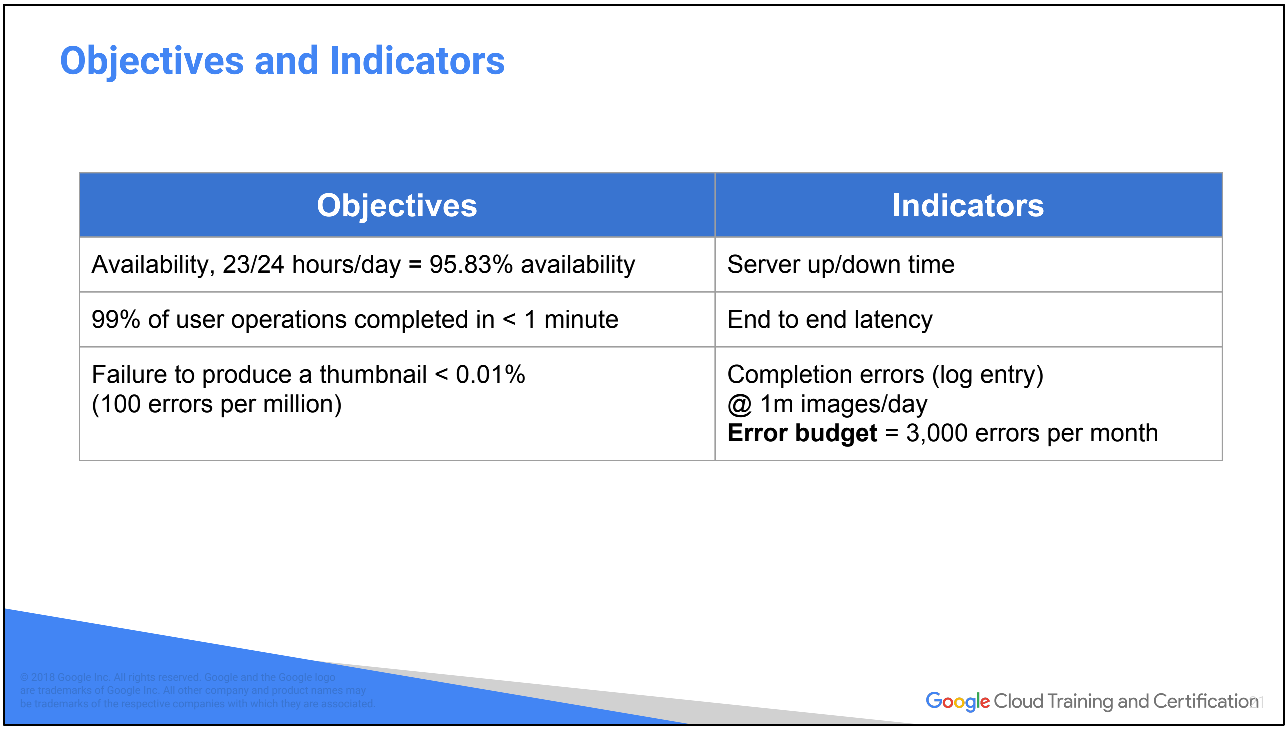
Scale the backend processing of thumbnails

Business Issue: Need to handle more thumbnail processing - must become scalable.

- Add a network load balancer to distribute the traffic to multiple thumbnail servers.

Google Cloud Training and Certification

So, here was our decision. We decided that if we need to handle more thumbnail processing, it's got to become more scalable. However, we didn't choose to simply throw more CPU and network at it because it was more of a single point of failure. Instead, we decided to add a load balancer and scale out the thumbnails servers. The great thing is that it's like a microservice in itself now. Because storage has been isolated to Google Cloud Storage, the same code can be distributed and it doesn't keep track of a queue or anything else. The upload server basically pulls whatever is on the data storage server, and load balances it as they come in. Technically, this is probably an internal load balancer, but we'll get into that a little bit later. In this case here, to help us with our greater than 80 percent CPU utilization, we want to distribute traffic requests from our business logic to the application servers in a cluster.



Objectives and Indicators

Objectives	Indicators
Availability, 23/24 hours/day = 95.83% availability	Server up/down time
99% of user operations completed in < 1 minute	End to end latency
Failure to produce a thumbnail < 0.01% (100 errors per million)	Completion errors (log entry) @ 1m images/day **Error budget** = 3,000 errors per month

Even though we've added a cluster of servers, we haven't changed anything that the user can measure. The performance is still a measure of the end to end latency, and the accuracy of the service is still based on the error logs.

We didn't need to adjust the SLOs because they are not based on the CPU load of the backend. Rather, the SLOs are based on the user experience.

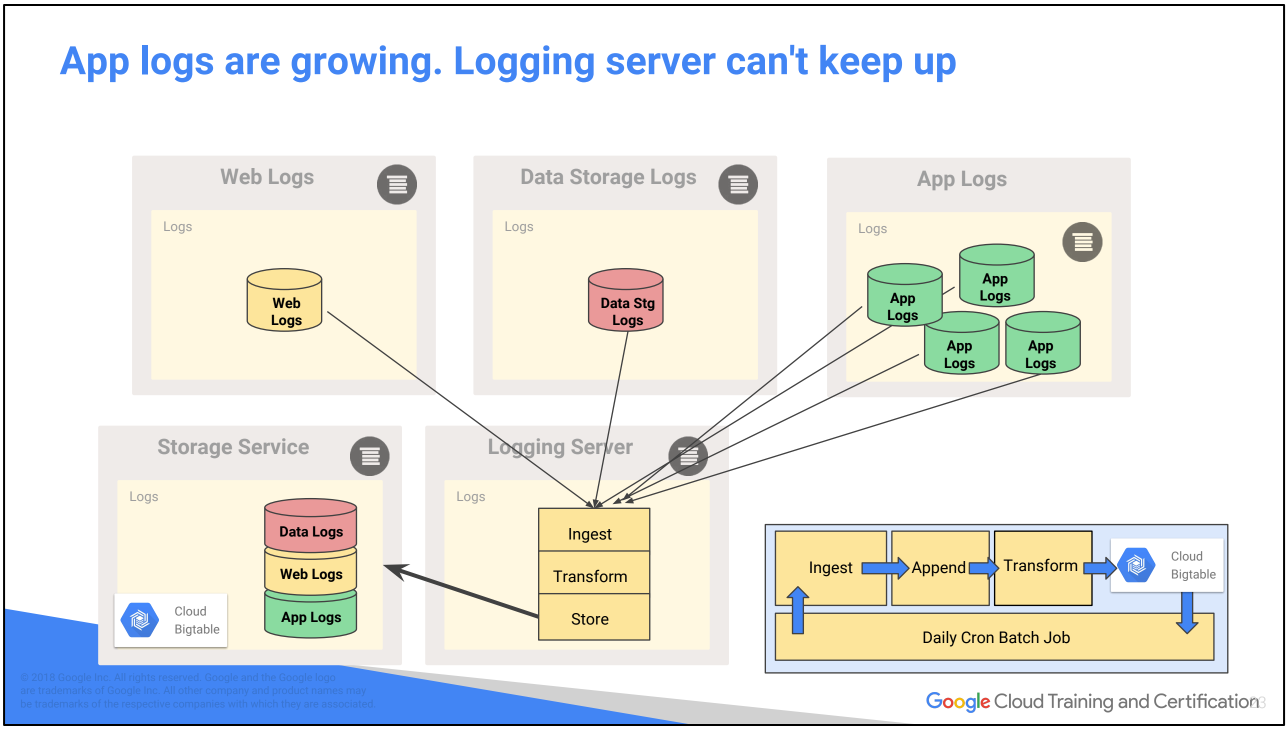
The autoscaling will help alleviate the CPU bottleneck because if the pool gets saturated it will autoscale.
That problem has been resolved. A new problem it reveals, however, is how long it will take for the autoscaling to catch up to the user demand. If the user demand is gradual, autoscaling will have no problem keeping up with demand. But if the demand is extremely bursty, other techniques and settings might be necessary: for example, might need to keep capacity at N+1 servers to give time for the pool to start up another server; changing the autoscaling trigger value; or using more sophisticated or custom metrics.

YOUR TURN

Design challenge #3

Growth

https://pixabay.com/en/the-strategy-win-champion-1080527/



App logs are growing. Logging server can't keep up

Google Cloud Training and Certification

If you expect to quickly outgrow local CPU, what is a way to scale the processing capability of the logic?

Take a few minutes to design your solution

Problem: Autoscaling of the application servers have produced logs that are outgrowing the processing capacity of the aggregation logging server.

Design a solution.

There are multiple designs possible depending on your assumptions. Your solution might be better than the one shown. The point of this exercise is to "think about the design" to develop your architecting skills.

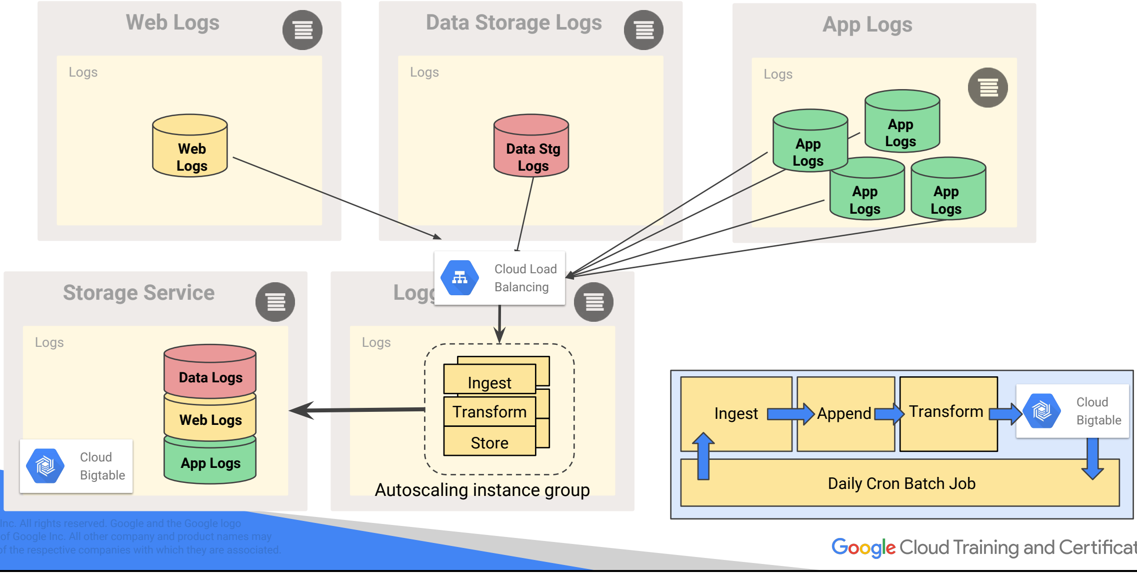
You can sketch your design in a tool like http://docs.google.com/drawings

One solution

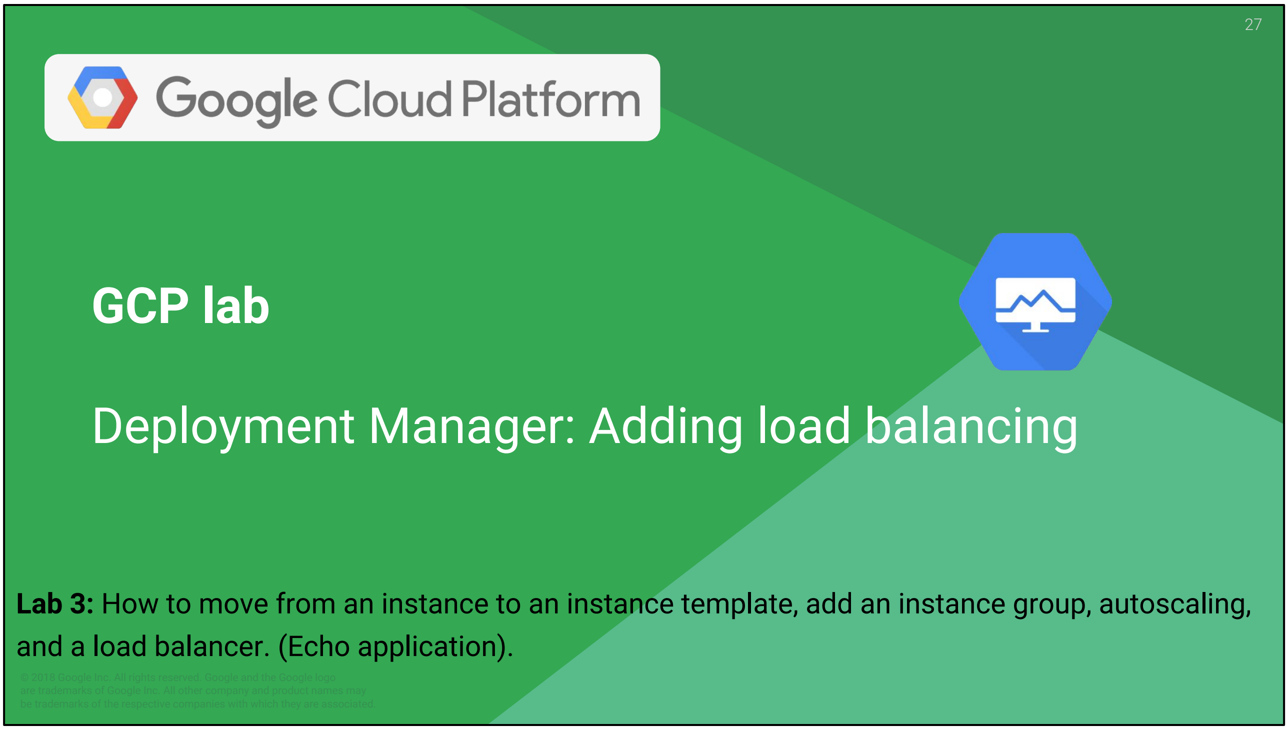
Remember, there are multiple valid solutions to this challenge.

Compare your design with the example solution.
Did your design account for all the elements addressed in the example solution?

App logs are growing. Logging server can't keep up



This proposed solution uses an internal load balancer.
An alternative could be Cloud Dataflow.



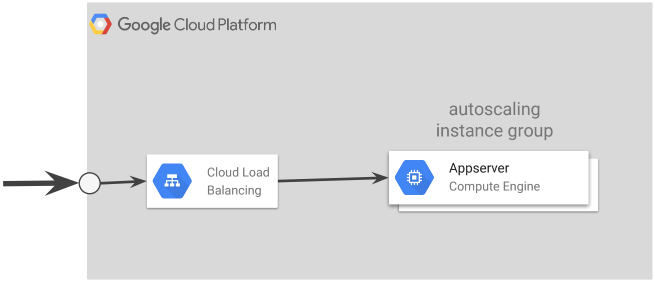
Google Cloud Platform

GCP lab

Deployment Manager: Adding load balancing

Lab 3: How to move from an instance to an instance template, add an instance group, autoscaling, and a load balancer. (Echo application).

Lab Deployment



Google Cloud Training and Certification