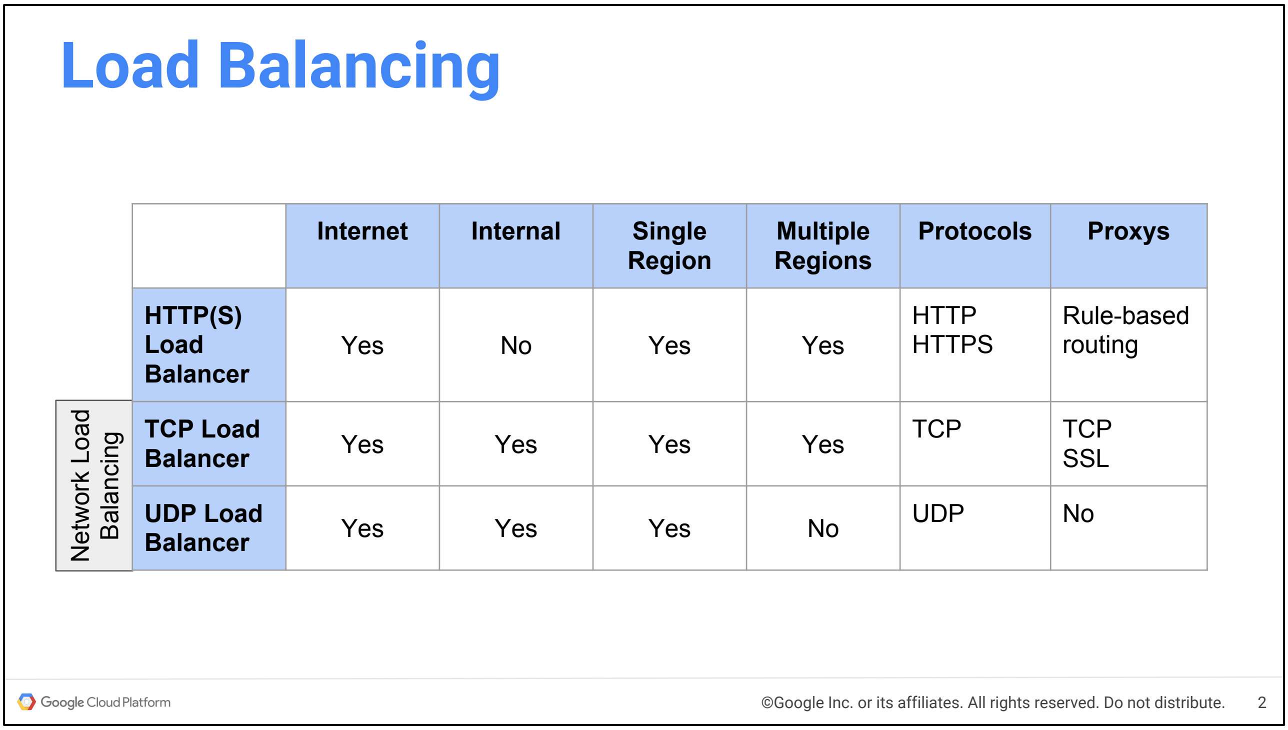
Google Cloud Platform

Infrastructure Automation

Load Balancing

v 1.0



Load Balancing

		Internet	Internal	Single Region	Multiple Regions	Protocols	Proxys
	HTTP(S) Load Balancer	Yes	No	Yes	Yes	HTTP HTTPS	Rule-based routing
Network Load Balancing	**TCP Load Balancer**	Yes	Yes	Yes	Yes	TCP	TCP SSL
	UDP Load Balancer	Yes	Yes	Yes	No	UDP	No

Google Cloud Platform

A proxy allows a single IP address to be used for all users around the world. Requests are automatically routed to the closest instance to the user. Check the documentation for current status of proxy services, they may be in Alpha or Beta.

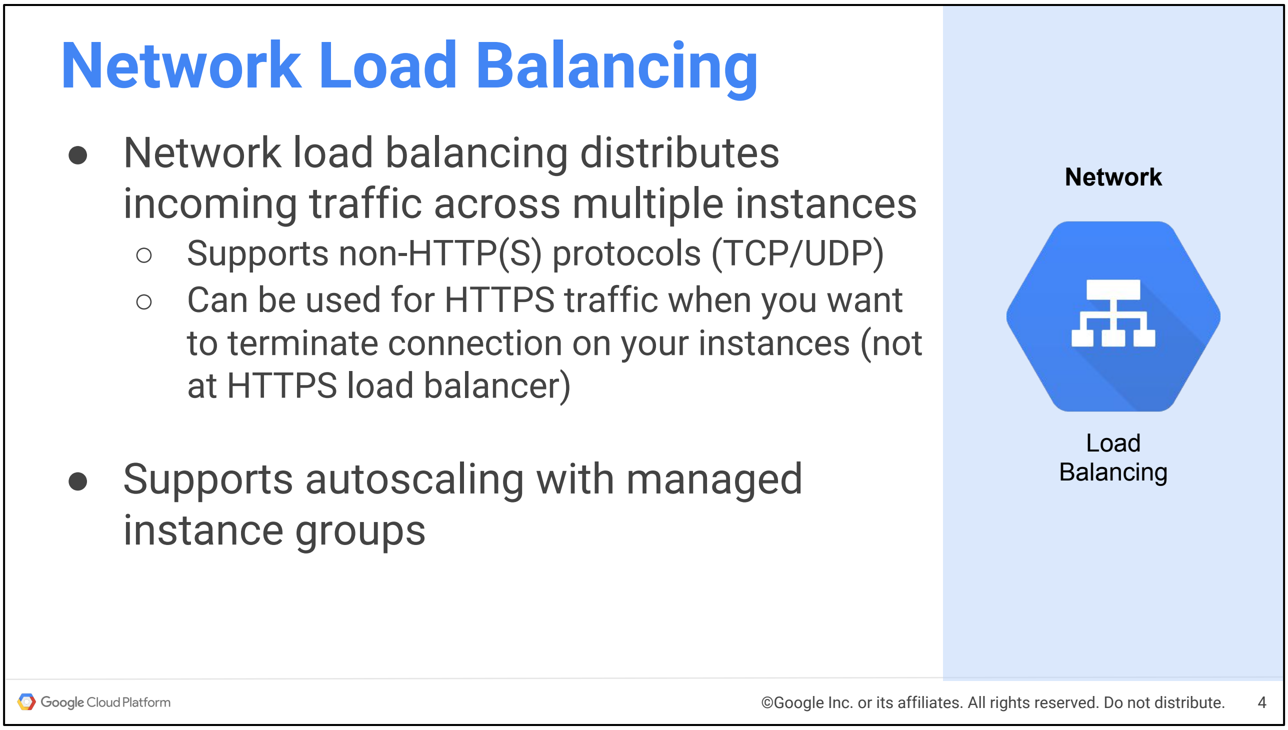
Although you *could* setup TCP Load Balancer with TCP proxy for port 80 (HTTP) and 443 (HTTPS), it is recommended that you use the HTTP(S) Load Balancer instead because it is designed specifically to deal with web traffic. Instead of simply directing the traffic to the nearest instance, you can establish rules to direct traffic to particular instances, for example, sending video requests to video servers and web page requests to web servers.

Common Load Balancing vocabulary

- Network load balancing
    - Distributes TCP/UDP traffic over a pool of VMs in a region
- HTTP(S) load balancing
    - Distributes HTTP(S) traffic to groups of VMs based on proximity to the user, the requested URL, or both
    - Cross-region load balancing, Content-aware load balancing
- SSL Proxy load balancing
    - Distributes SSL traffic based on proximity to the user
- Internal load balancing
    - Distributes traffic from VMs to other VMs in the same region

Agenda

1. Network Load Balancing
2. HTTP(S) Load Balancing
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. Review



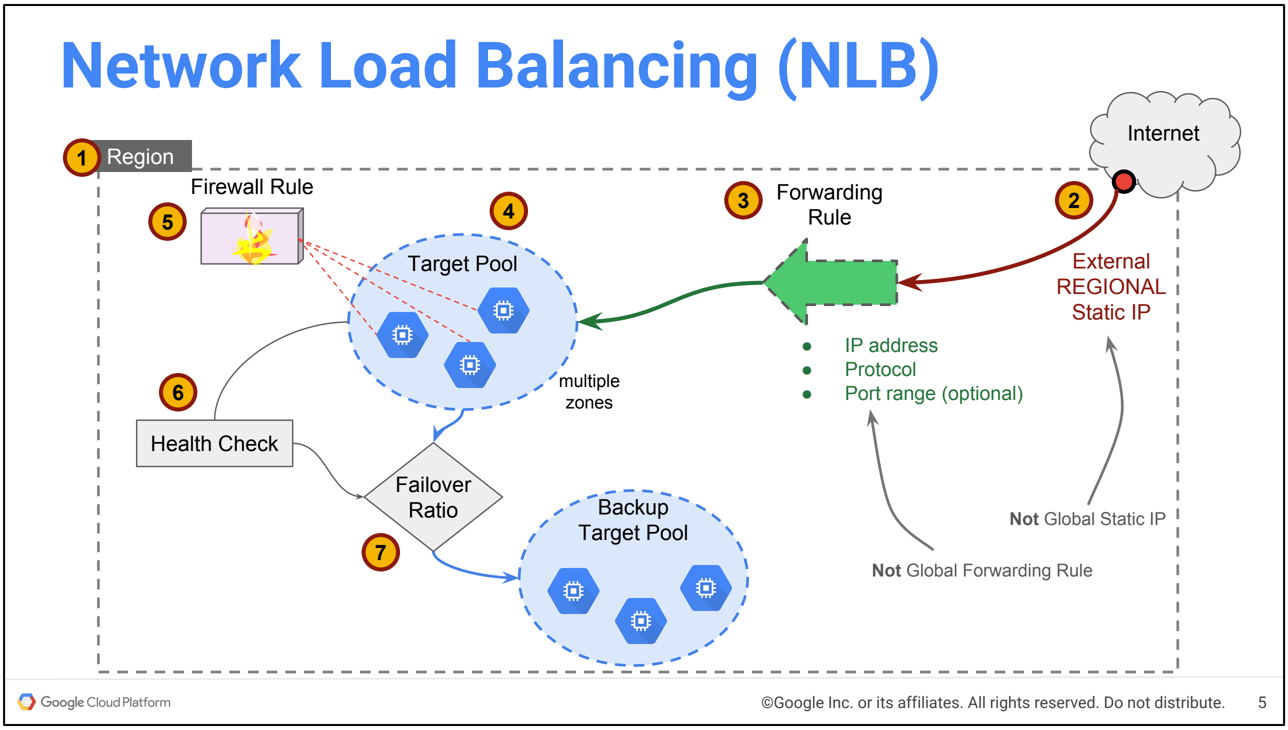
Network Load Balancing

- Network load balancing distributes incoming traffic across multiple instances
    - Supports non-HTTP(S) protocols (TCP/UDP)
    - Can be used for HTTPS traffic when you want to terminate connection on your instances (not at HTTPS load balancer)
- Supports autoscaling with managed instance groups

Network
Load
Balancing

Google Cloud Platform

For more information on network load balancing, see https://cloud.google.com/compute/docs/load-balancing/network/



Network Load Balancing (NLB)

1 Region

Internet

5 Firewall Rule

4 Target Pool

multiple zones

3 Forwarding Rule

2 External REGIONAL Static IP

- IP address
- Protocol
- Port range (optional)

6 Health Check

Failover Ratio

7

Backup Target Pool

Not Global Static IP

Not Global Forwarding Rule

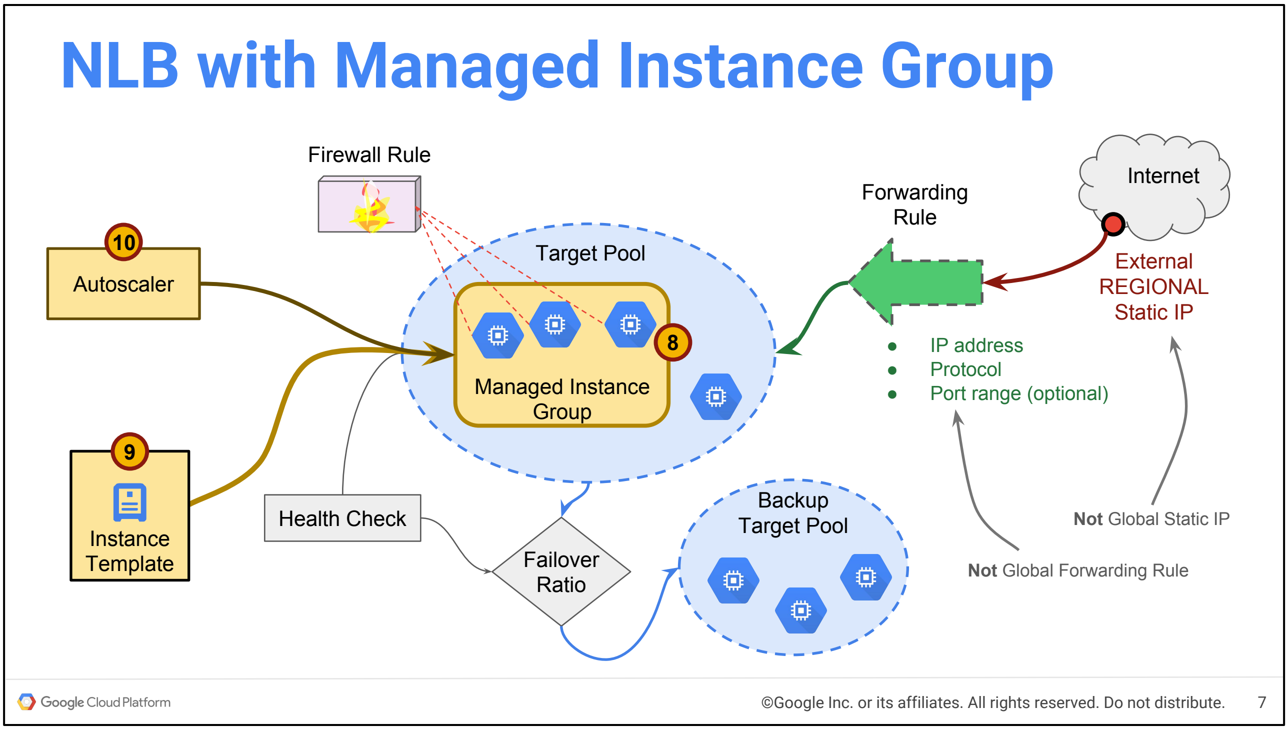
Google Cloud Platform

(1) Network Load Balancing occurs within a region. It can cover zones with a region (for high availability) but not multiple regions.
(2) Network Load Balancing (NLB) begins with an External REGIONAL Static IP. You can't use NLB with a Global Static IP.
(3) A Forwarding Rule is associated with the IP. Traffic arriving on the IP that matches a particular Protocol is directed by the Forwarding Rule. Optionally, you can specify a port range, otherwise all traffic of the specified protocol is forwarded. Forwarding Rules are regional. You can't use a Global Forwarding Rule with NLB.
(4) The Forwarding Rule sends the traffic to a Target Pool. The Target Pool contains instances and it uses a Load Distribution algorithm to distribute traffic to each of the instances in the pool. When the Forwarding Rule is created you need to reference the Target Pool, so that means the Target Pool must be created before the Forwarding Rule. The Target Pool can contain instances from multiple zones in the same region. If a zone is lost, the instances from other zones can continue, providing high availability.
(5) Of course, the instances can't receive traffic without a Firewall Rule to allow the particular protocol to be delivered to them.
(6) A Target Pool has a single Health Check that is used with all instances in the pool. A Health Check is an algorithm that determines whether an instance is "healthy" or not. For example, if the pool contained web servers, the check might be trying to read a particular web page. An error would indicate that the particular server wasn't healthy. An instance that is going out of service might intentionally do something to force the health check to fail, such as removing the web page used by the health check. This causes the target pool to stop sending new traffic to the instance so that it

can "drain" its sessions (complete work in progress) prior to going out of service. So the health check is used for traffic flow control within the pool.
(7) The number of healthy instances over the total number in the pool creates a ratio between 0 and 1. A Failover Ratio can be set such that if there are not enough healthy instances, traffic will be sent to instances in an (optional) backup pool. This service is only one layer deep. That is, you can create a backup pool for a Target Pool, but you can't create a backup for the backup. Note that the Backup Target Pool must be in the same Region as the Target Pool.

Note that instances are added to a Target Pool after they are operational. The Target Pool knows nothing about the origin of the instances.
It also cannot start new instances or shut down instances.

https://cloud.google.com/compute/docs/load-balancing/network/target-pools



NLB with Managed Instance Group

Firewall Rule

10 Autoscaler

Target Pool

Managed Instance Group 8

Forwarding Rule

Internet

External REGIONAL Static IP

- IP address
- Protocol
- Port range (optional)

9 Instance Template

Health Check

Failover Ratio

Backup Target Pool

Not Global Static IP

Not Global Forwarding Rule

Google Cloud Platform

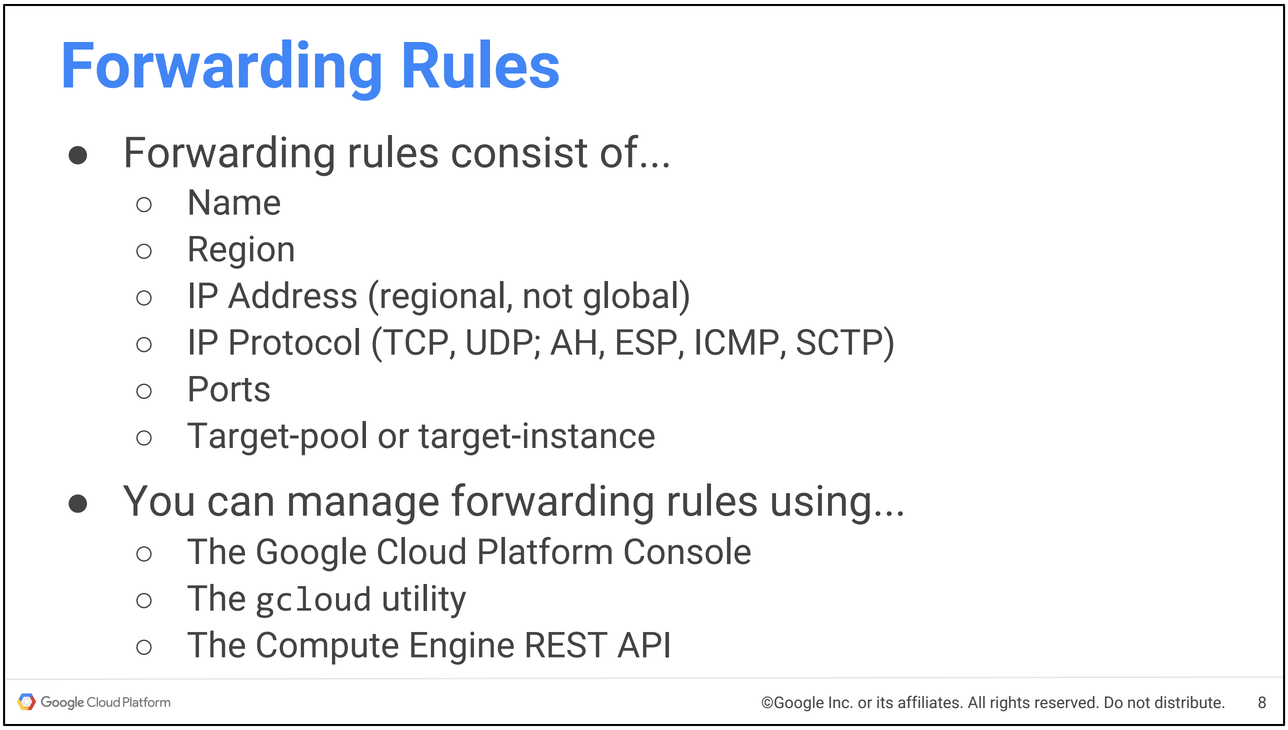
(7) When a Managed Instance Group is used with Network Load Balancing, the group adds or removes instances from the Target Pool automatically. You don't tell the Target Pool to use the Managed Instance Group, you tell the Managed Instance Group to use the Target Pool with this command: `gcloud compute instance-groups managed set-target-pools`

(8) The Managed Instance Group creates identical instances from an Instance Template. And it affords other commands for dealing with the entire group of instances so you don't have to perform the same action on each instance individually. For this reason, Managed Instance Groups are useful for software maintenance, because an update can be rolled out to all the instances in a group.

(9) Another benefit of a Managed Instance Group is that it can be used with an Autoscaler. The Autoscaler tells the Managed Instance Group when to create or shutdown instances. There are many modes, settings, and options for configuring the Autoscaler scaling policy. For example, the policy can be set to scale on CPU load or on Stackdriver metrics.

The concept of a Backup Pool is in conflict with the concept of Autoscaling. Therefore, Autoscaling cannot be configured for a Target Pool if it has a Backup Pool defined.

Autoscaling is discussed in detail in a subsequent module.



Forwarding Rules

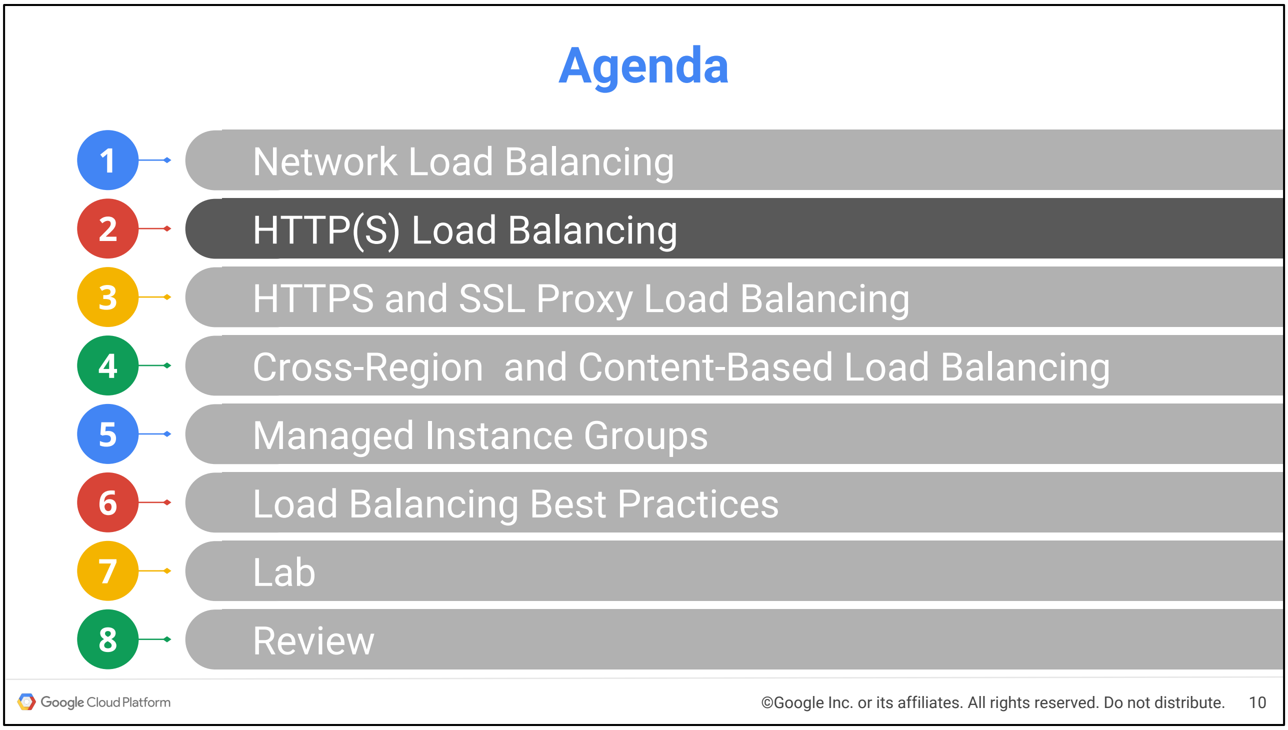
- Forwarding rules consist of...
  - Name
  - Region
  - IP Address (regional, not global)
  - IP Protocol (TCP, UDP; AH, ESP, ICMP, SCTP)
  - Ports
  - Target-pool or target-instance
- You can manage forwarding rules using...
  - The Google Cloud Platform Console
  - The `gcloud` utility
  - The Compute Engine REST API

Forwarding rules use for load balancing, thus pointing at a target pool, can only forward TCP & UDP packets. Forwarding rules for AH/ESP/ICMP/SCTP must forward to a target-instance as part of a port-forwarding setup.

For more information on forwarding rules, see https://cloud.google.com/compute/docs/load-balancing/network/forwarding-rules

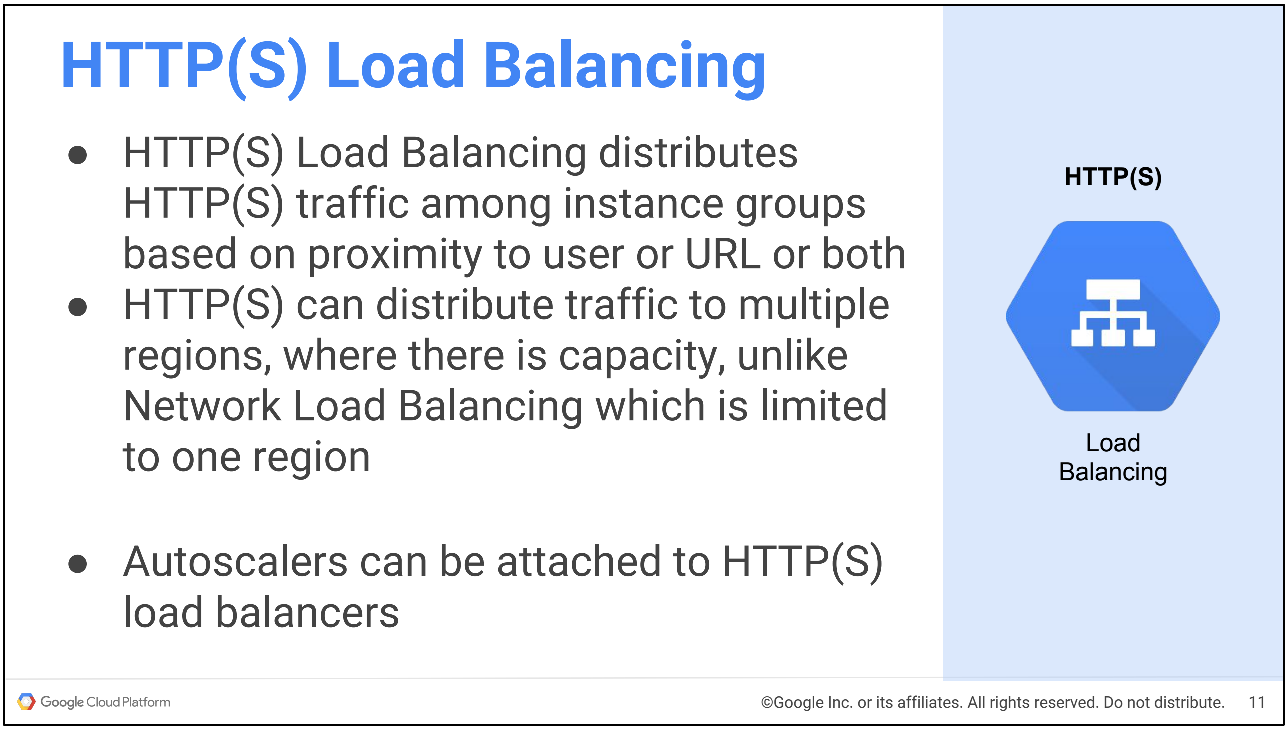
Target Pools

- Max 50 Target Pools per project
  - Instances can be in different zones but must be in the same region; add to pool at creation or use Instance Group
- SessionAffinity influences load distribution
  - NONE = a hash of 5 values is used to select the instance
    - Source IP, Source Port, Protocol, Destination IP, Destination Port
    - New connection - typically random assignment
  - CLIENT_IP_PROT - all connections from one client end up on the same instance
  - CLIENT_IP all connections from a client end up on the same instance regardless of protocol



Agenda

1. Network Load Balancing
2. **HTTP(S) Load Balancing**
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. Review



HTTP(S) Load Balancing

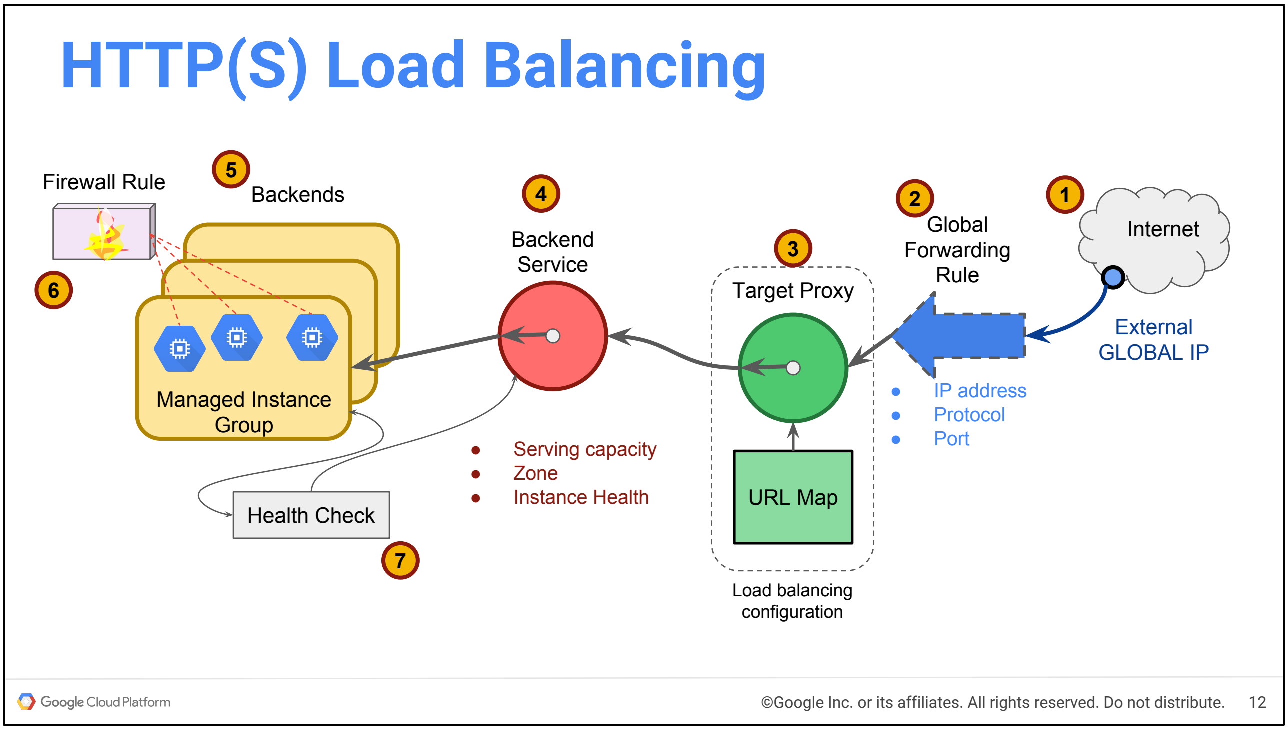
- HTTP(S) Load Balancing distributes HTTP(S) traffic among instance groups based on proximity to user or URL or both
- HTTP(S) can distribute traffic to multiple regions, where there is capacity, unlike Network Load Balancing which is limited to one region
- Autoscalers can be attached to HTTP(S) load balancers

HTTP(S)

Load Balancing

Google Cloud Platform

For more information on network load balancing, see https://cloud.google.com/compute/docs/load-balancing/network/



HTTP(S) Load Balancing

6 Firewall Rule

5 Backends

Managed Instance Group

7 Health Check

4 Backend Service

- Serving capacity
- Zone
- Instance Health

3 Target Proxy

URL Map

Load balancing configuration

2 Global Forwarding Rule

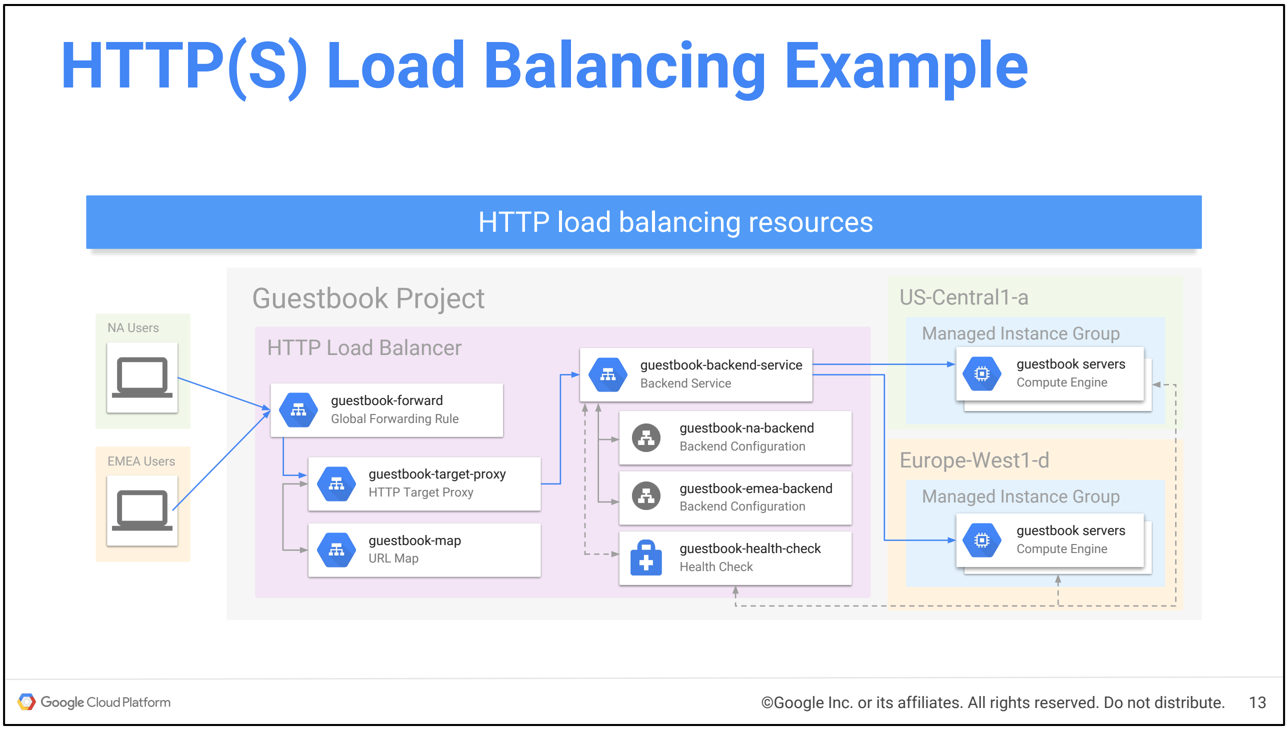
- IP address
- Protocol
- Port

1 Internet

External GLOBAL IP

Google Cloud Platform

(1) Recall that external IPs are related to internal resources through Network Address Translation (NAT). External REGIONAL IPs can be used by VMs or by Network Load Balancers. External GLOBAL IPs can only be used by GLOBAL forwarding rules. External GLOBAL IPs cannot be assigned to any regional or zonal resource. External IPs can only be assigned to one Global Forwarding Rule.
(2) One Global Forwarding Rule can only forward traffic for a single protocol and port, as follows: **HTTP** { **80** | 8080 }, **HTTPS** { **443** }
SSL { 25 | 43 | 110 | 143 | 105 | **443** | 465 | 587 | 700 | 993 | 995 }, **TCP** { **25** | 43 | 110 | 143 | 105 | 443 | 465 | 587 | 700 | 993 | 995 }
The Global Forwarding Rule requires a Target Proxy to already exist before it can be configured. Support for HTTP and HTTPS requires two rules.
(3) A Target Proxy is specific to a single protocol (HTTP, HTTPS, SSL, TCP)



HTTP(S) Load Balancing Example

HTTP load balancing resources

NA Users

EMEA Users

Guestbook Project

HTTP Load Balancer

guestbook-forward
Global Forwarding Rule

guestbook-target-proxy
HTTP Target Proxy

guestbook-map
URL Map

guestbook-backend-service
Backend Service

guestbook-na-backend
Backend Configuration

guestbook-emea-backend
Backend Configuration

guestbook-health-check
Health Check

US-Central1-a

Managed Instance Group

guestbook servers
Compute Engine

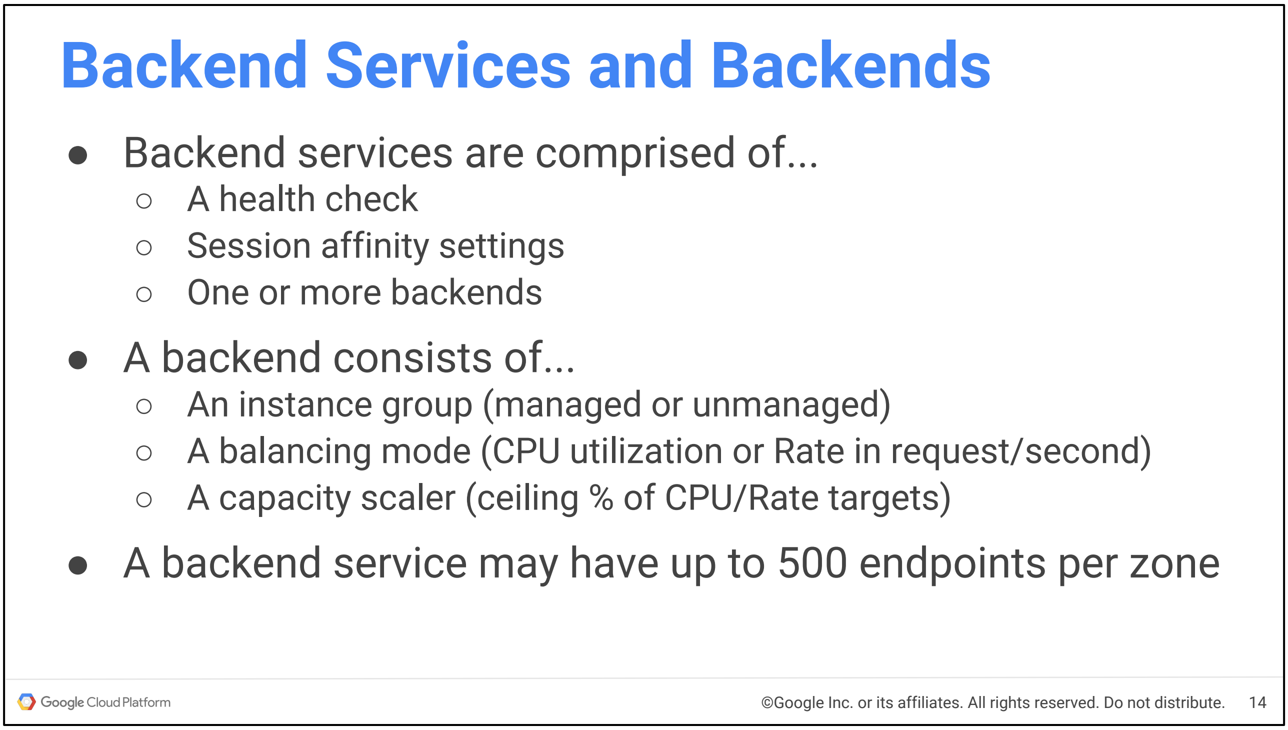
Europe-West1-d

Managed Instance Group

guestbook servers
Compute Engine

Google Cloud Platform

Distributes HTTP(S) traffic among instance groups based on proximity to user or URL or both
For an overview of setting up HTTP(S) Load Balancing, see https://cloud.google.com/compute/docs/load-balancing/http/#illegal_request_handling



Backend Services and Backends

- Backend services are comprised of...
  - A health check
  - Session affinity settings
  - One or more backends
- A backend consists of...
  - An instance group (managed or unmanaged)
  - A balancing mode (CPU utilization or Rate in request/second)
  - A capacity scaler (ceiling % of CPU/Rate targets)
- A backend service may have up to 500 endpoints per zone

Google Cloud Platform

The backend service will not send requests to instances reported as unhealthy by the Health Check.

Session affinity can be set so that requests from the same client end up going to the same instance.

Client IP affinity directs requests from the same origin IP to the same server. NAT and other network routing technologies can cause requests from multiple different users to look like they come from the same address causing many users to get routed to the same instance. On the other hand, one user who moves from network to network may be seen as two different users, and not end up directed to the same instance.

Generated cookie affinity causes the load balancer to issue a cookie named GCLB on the first request from a user, then directs subsequent requests with the same cookie to the same instance.

Session affinity can break…

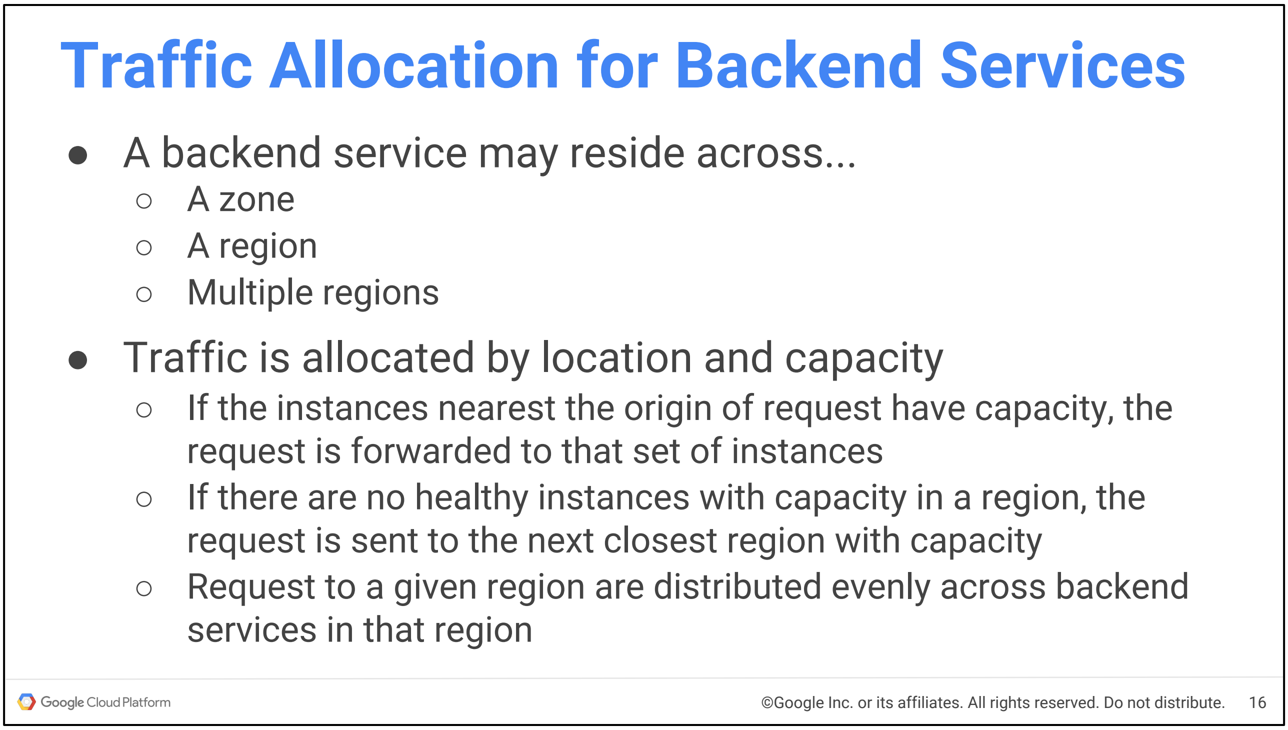
- If an instance group runs out of capacity and traffic is routed to another zone
- If autoscaling change capacity and load is reallocated
- If the target instance fails health checks

The balancing mode and capacity scaler designate how maximum utilization for a backend. The balancing mode uses one of the following metrics to determine if the backend instance group is at capacity and the load balancer needs to send requests to another backend:

- CPU utilization (average of CPU use across all instances in backend)
- Rate: Maximum Requests/second/instance
- Rate: Maximum Requests/second/group
- CPU utilization and Rate (either RPS/instance or RPS/group) - either condition triggers at-capacity state

The Capacity scaler is an additional setting that directs the load balancer to only direct requests to a given backend instance group as long as utilization is below a % of the balancing mode maximum. For example, if balancing mode is utilization and Max CPU utilization is 80%, setting the capacity scaler to 50% would mean the load balancer would see the backend as being at capacity when CPU utilization is at 40% average across the instance group.

For more information on Backend Services and Backends, see https://cloud.google.com/compute/docs/load-balancing/http/backend-service

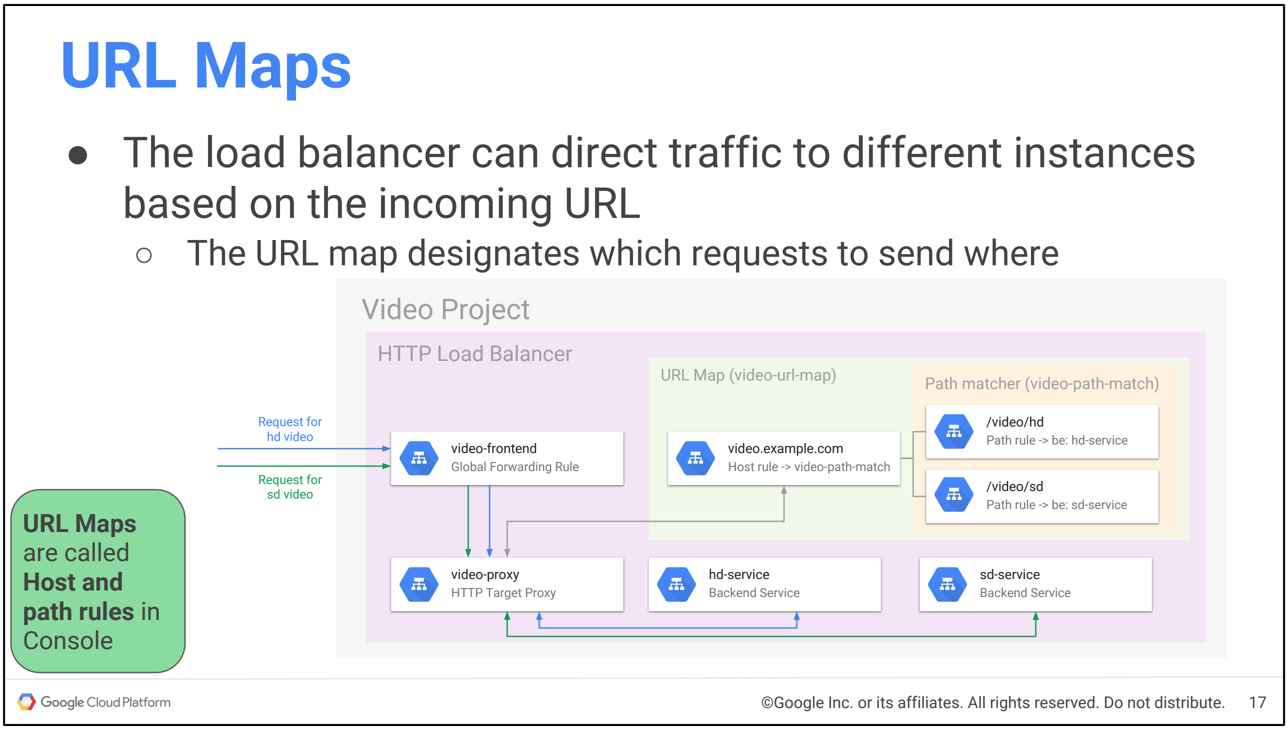


Traffic Allocation for Backend Services

- A backend service may reside across...
  - A zone
  - A region
  - Multiple regions
- Traffic is allocated by location and capacity
  - If the instances nearest the origin of request have capacity, the request is forwarded to that set of instances
  - If there are no healthy instances with capacity in a region, the request is sent to the next closest region with capacity
  - Request to a given region are distributed evenly across backend services in that region

Google Cloud Platform

A backend service can span multiple regions. At the same time, there can be multiple backend services within the same region.



URL Maps

- The load balancer can direct traffic to different instances based on the incoming URL
  - The URL map designates which requests to send where

URL Maps are called **Host and path rules** in Console

Google Cloud Platform

Notes:

URL Maps are comprised of:

- Host rules
- Path matchers
  - Path rules

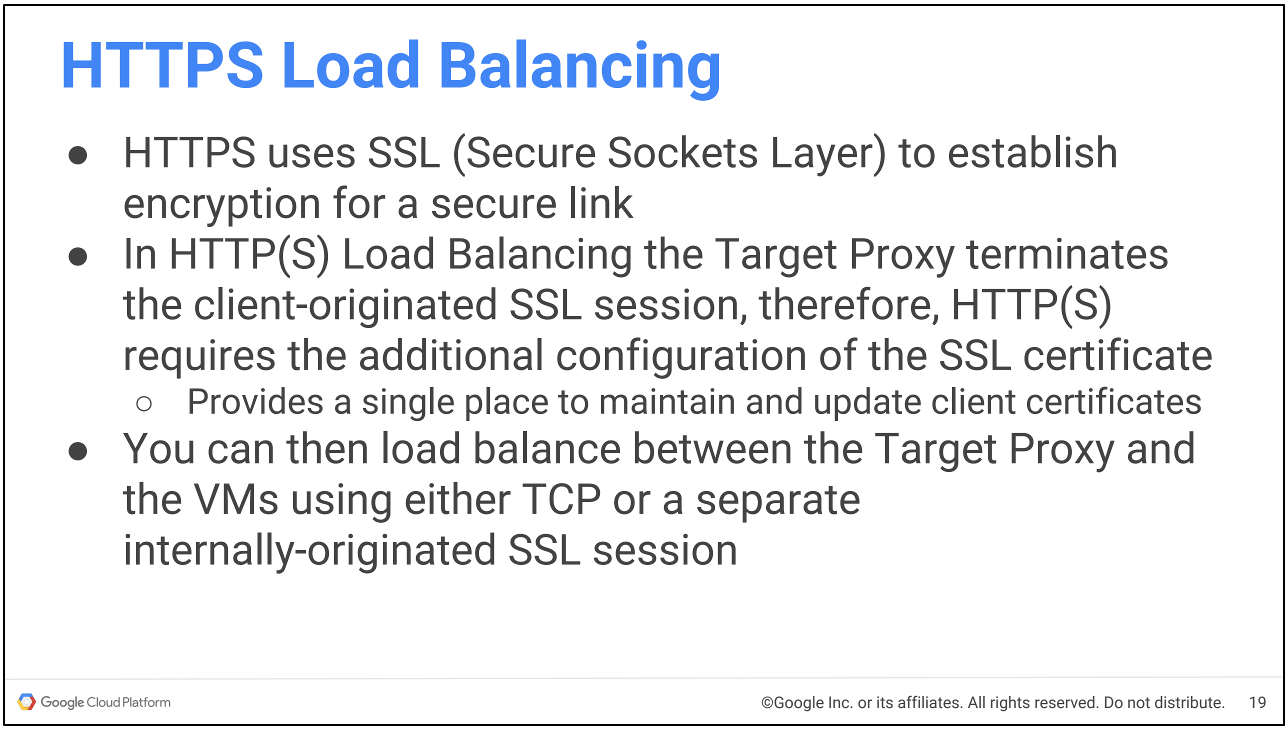
Host rules map host names to patch matchers. Patch matchers are comprised of one or more path rules with URL paths mapped to specific backend services.

The default configuration for a URL map is a single patch matcher (/*), which is created automatically and routes all traffic to a the default service.

For more information on URL Maps, see https://cloud.google.com/compute/docs/load-balancing/http/url-map

Agenda

1. Network Load Balancing
2. HTTP(S) Load Balancing
3. **HTTPS and SSL Proxy Load Balancing**
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. Review



HTTPS Load Balancing

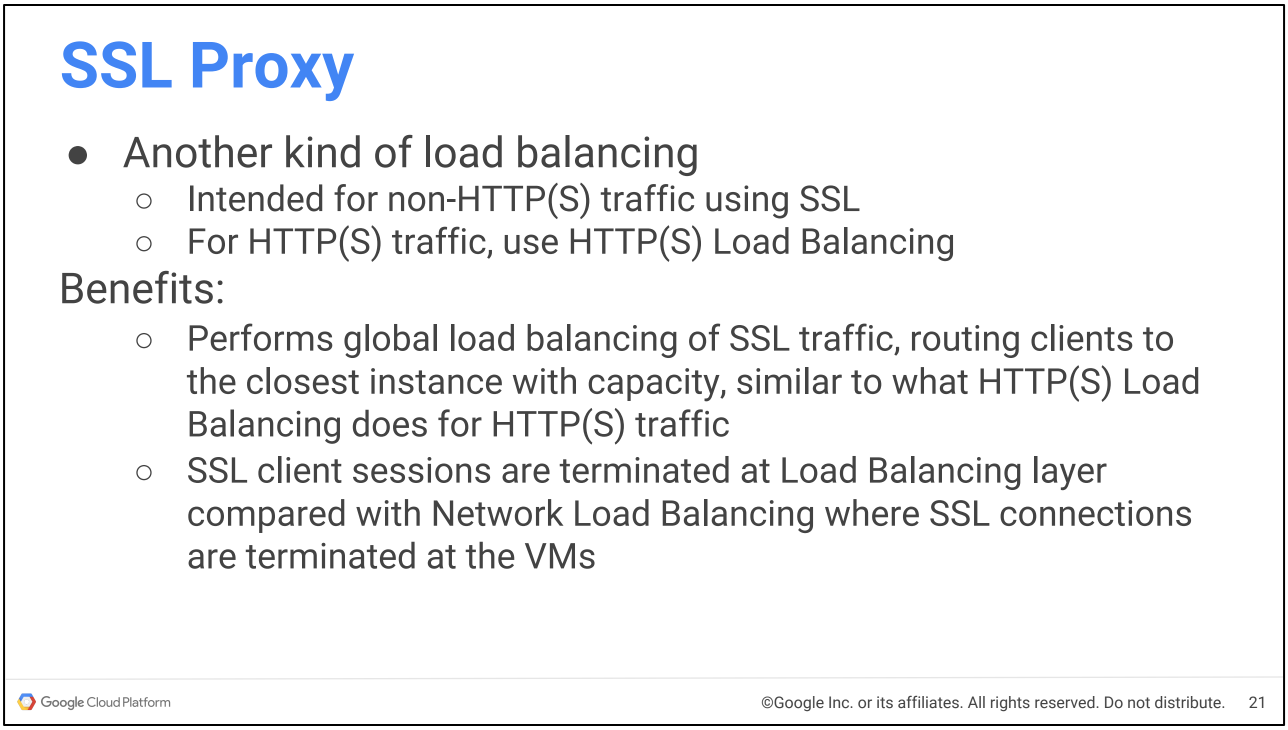
- HTTPS uses SSL (Secure Sockets Layer) to establish encryption for a secure link
- In HTTP(S) Load Balancing the Target Proxy terminates the client-originated SSL session, therefore, HTTP(S) requires the additional configuration of the SSL certificate
  - Provides a single place to maintain and update client certificates
- You can then load balance between the Target Proxy and the VMs using either TCP or a separate internally-originated SSL session

Google Cloud Platform

When using Network Load Balancing with HTTPS, the SSL session terminates on the VMs.

HTTPS additional information

- An HTTPS target proxy accepts only TLS 1.0 and up when terminating client SSL requests
    - Speaks only TLS 1.0 and up when backend protocol is HTTPS
- The load balancer blocks illegal requests
    - See documentation for details
- Load Balancing Logging writes request information into Stackdriver logs
    - Logs HTTPRequest log fields and statusDetails that explains why the load balancer returned a given status
    - Check documentation for release status



SSL Proxy

- Another kind of load balancing
  - Intended for non-HTTP(S) traffic using SSL
  - For HTTP(S) traffic, use HTTP(S) Load Balancing

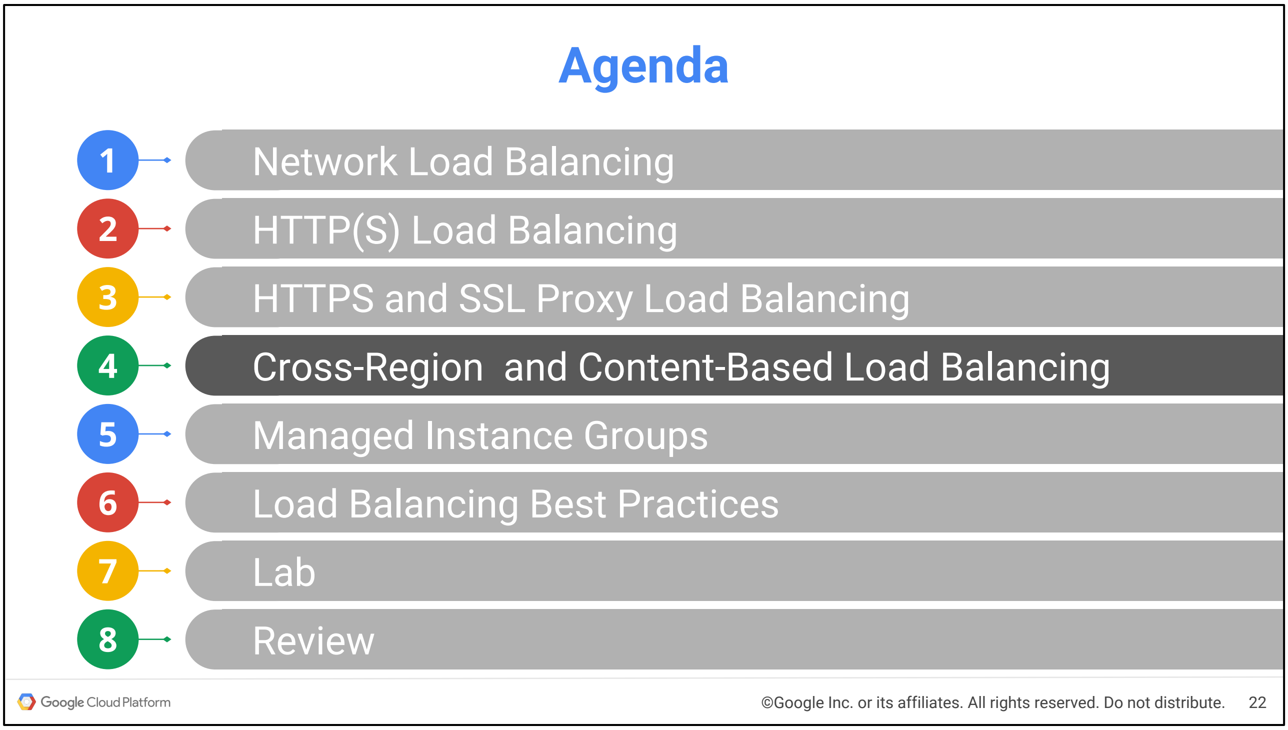
Benefits:

  - Performs global load balancing of SSL traffic, routing clients to the closest instance with capacity, similar to what HTTP(S) Load Balancing does for HTTP(S) traffic
  - SSL client sessions are terminated at Load Balancing layer compared with Network Load Balancing where SSL connections are terminated at the VMs

Google Cloud Platform

For more information on the Google Cloud SSL proxy, see https://cloud.google.com/compute/docs/load-balancing/tcp-ssl/

- Intelligent routing
- Reduced CPI load on instances
- Certificate management
- Security patching

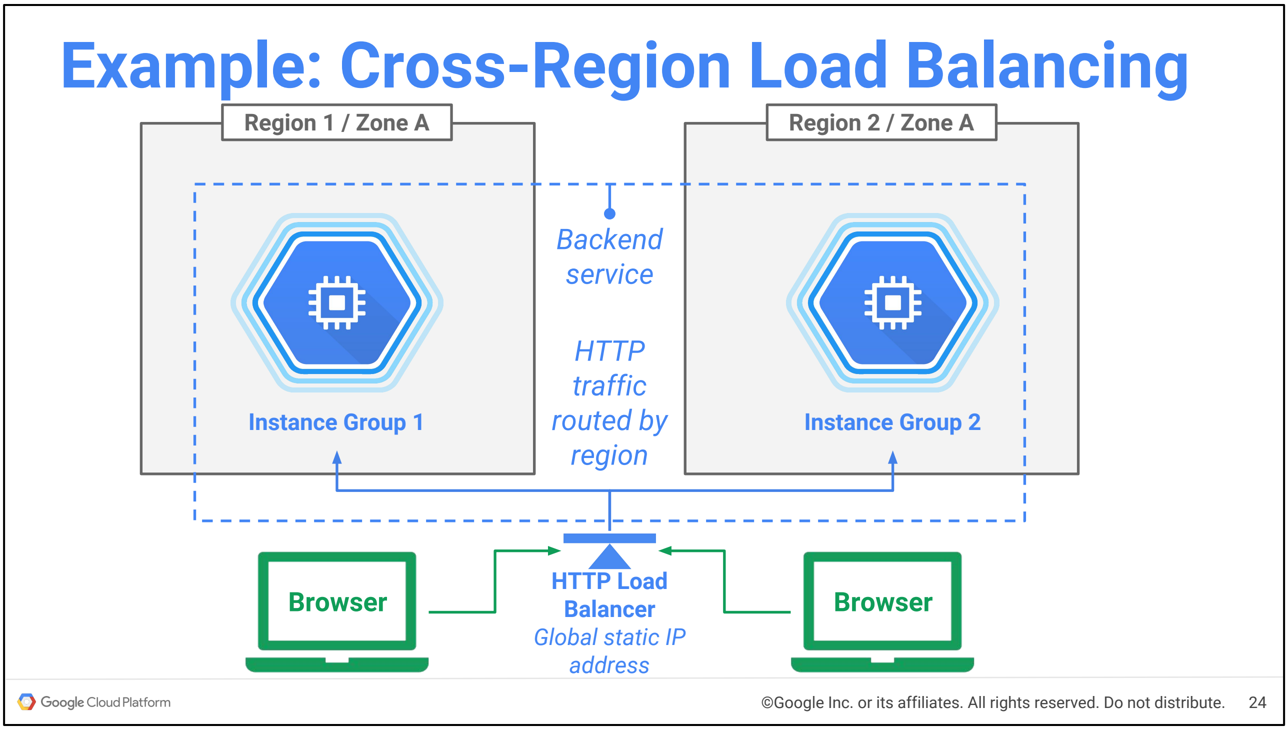


Agenda

1. Network Load Balancing
2. HTTP(S) Load Balancing
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. Review

Cross-Region Load Balancing

- HTTP/HTTPS only
- Cross-region using a single global IP address
- Requests routed to the closest region
- Automatically reroutes to next closest once capacity is reached
- Eliminates need for DNS-based load balancing



Example: Cross-Region Load Balancing

Region 1 / Zone A

Region 2 / Zone A

Backend service

Instance Group 1

HTTP traffic routed by region

Instance Group 2

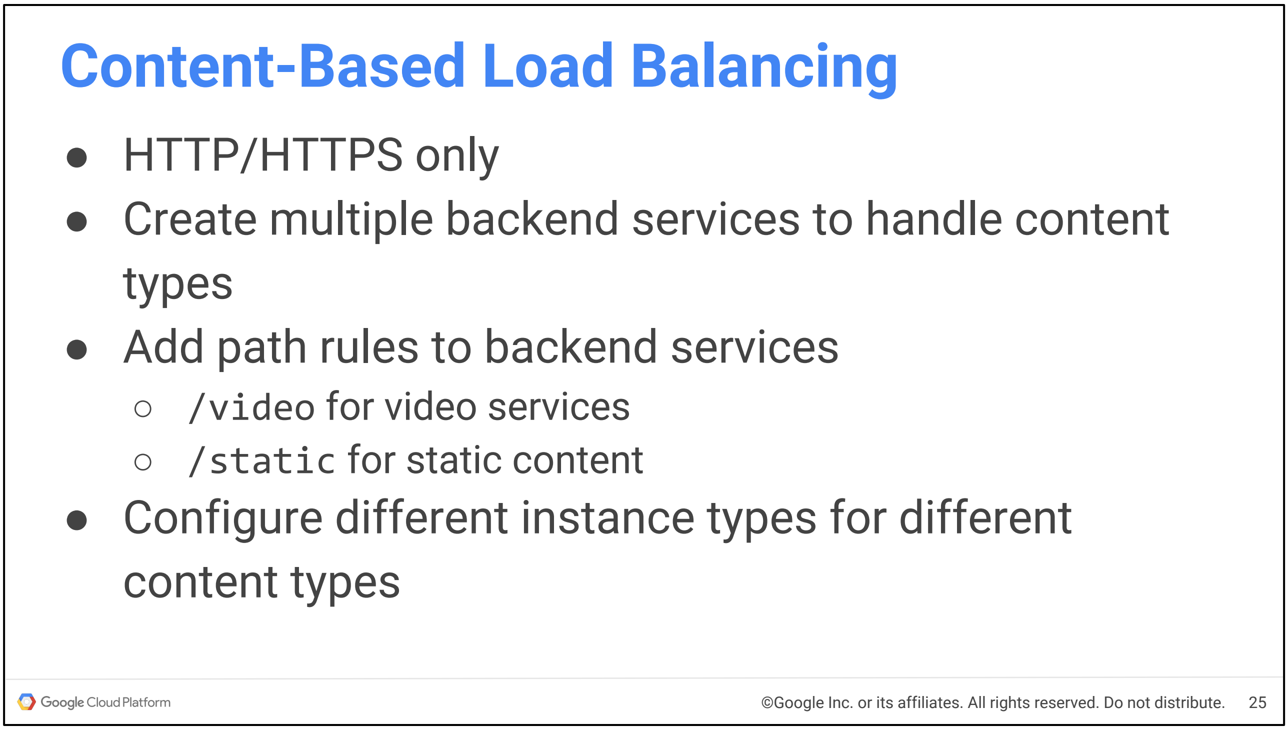
Browser

HTTP Load Balancer

Global static IP address

Browser

Google Cloud Platform



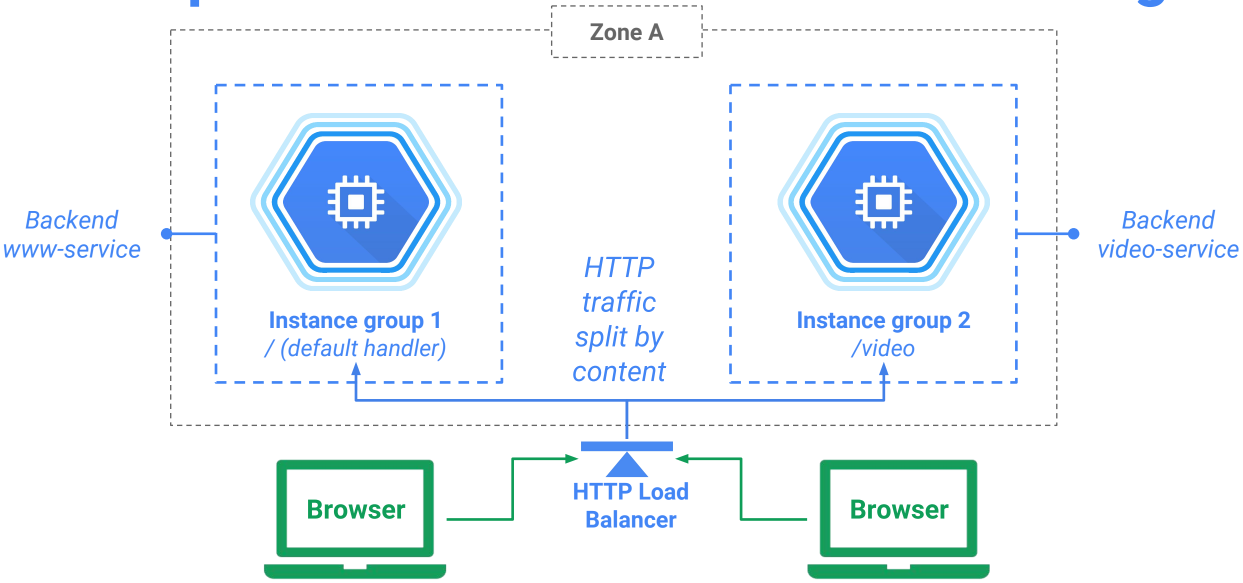
Content-Based Load Balancing

- HTTP/HTTPS only
- Create multiple backend services to handle content types
- Add path rules to backend services
  - `/video` for video services
  - `/static` for static content
- Configure different instance types for different content types

Google Cloud Platform

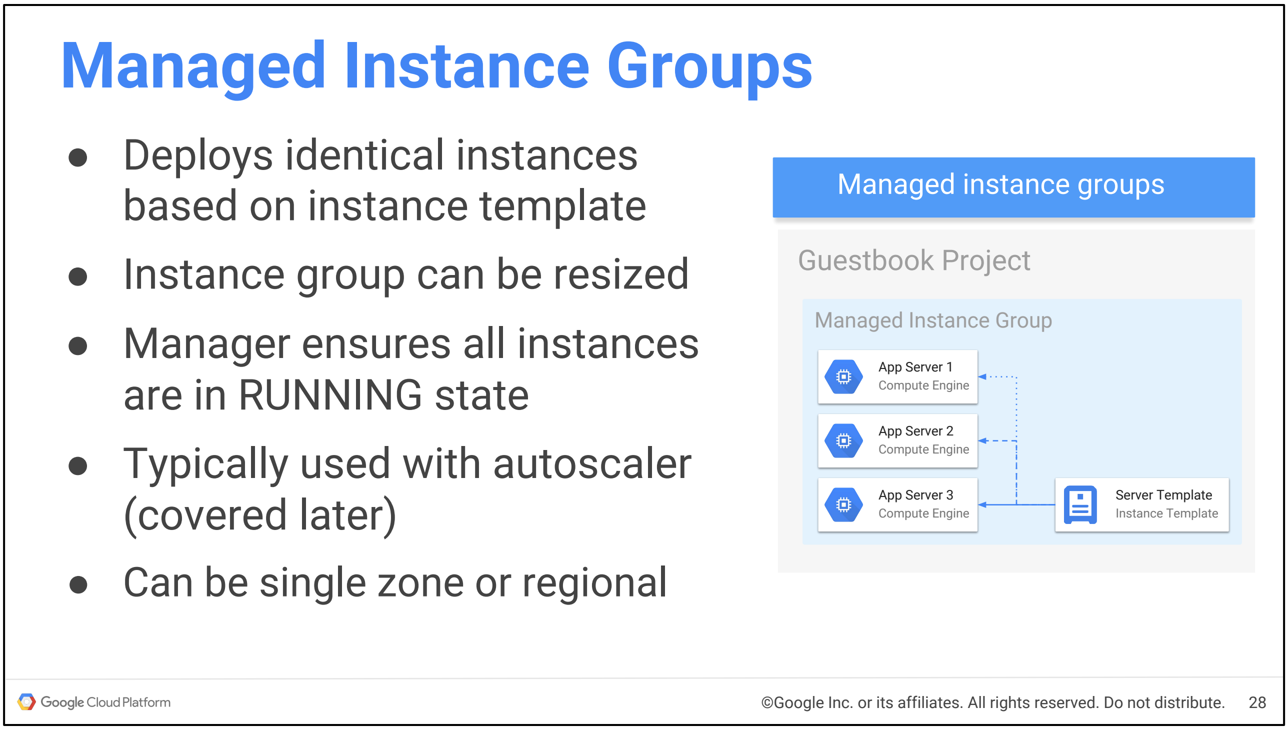
https://cloud.google.com/compute/docs/load-balancing-and-autoscaling

Example: Content-Based Load Balancing



Agenda

1. Network Load Balancing
2. HTTP(S) Load Balancing
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. **Managed Instance Groups**
6. Load Balancing Best Practices
7. Lab
8. Review



Managed Instance Groups

- Deploys identical instances based on instance template
- Instance group can be resized
- Manager ensures all instances are in RUNNING state
- Typically used with autoscaler (covered later)
- Can be single zone or regional

Managed instance groups

Guestbook Project

Managed Instance Group

App Server 1
Compute Engine

App Server 2
Compute Engine

App Server 3
Compute Engine

Server Template
Instance Template

Google Cloud Platform

Use managed instance groups for most situations. Managed instance groups provide the following benefits:

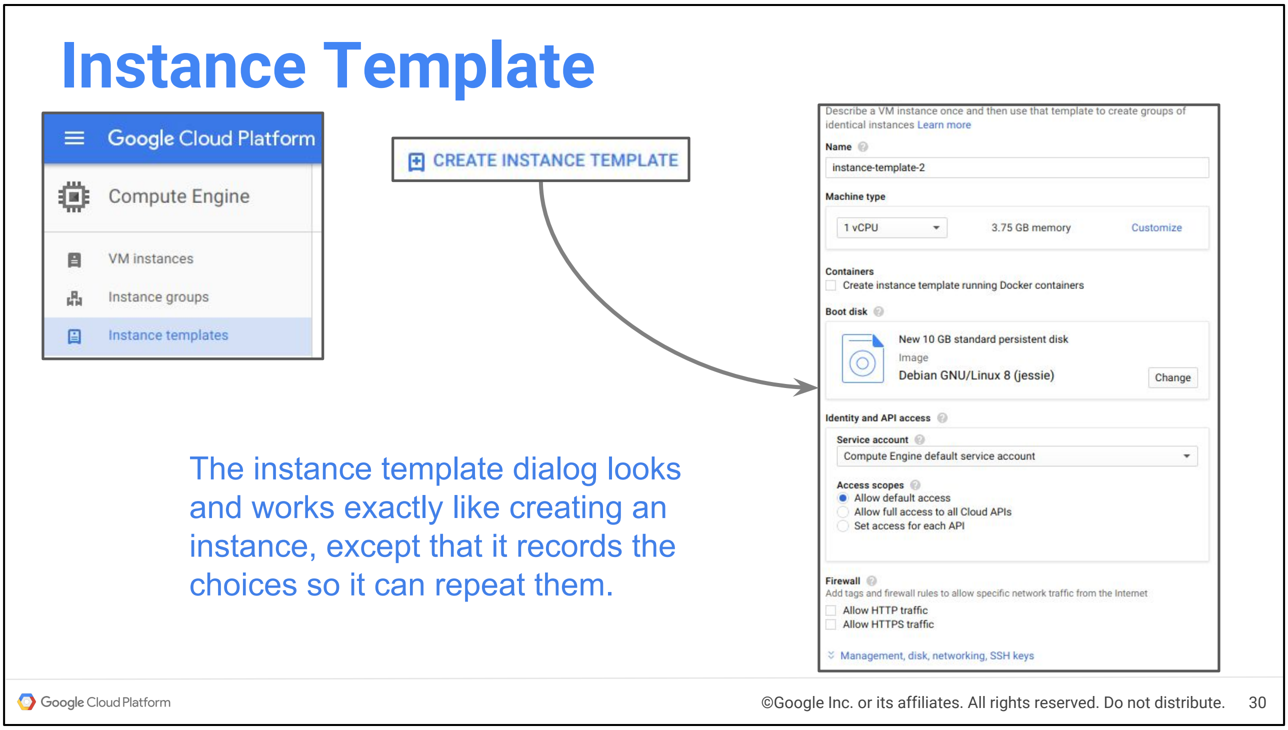
- Managed instance groups use an instance template to define the properties for every instance in the group. You can easily update all of the instances in the group by specifying a new template in a rolling update.
- When your applications require additional compute resources, managed instance groups can automatically scale the number of instances in the group.
- Managed instance groups can work with load balancing services to distribute network traffic to all of the instances in the group.
- If an instance in the group stops, crashes, or is deleted by an action other than the instance groups commands, the managed instance group automatically recreates the instance so it can resume its processing tasks. The recreated instance uses the same name and the same instance template as the previous instance, even if the group references a different instance template.
- Managed instance groups can automatically identify and recreate unhealthy instances in a group to ensure that all of the instances are running optimally.

For more information on managed instance groups, see:

https://cloud.google.com/compute/docs/instance-groups/#create_managed_group.

In addition to managed instance groups that belong in single zones, you can create regional managed instance groups, which distributes instances across multiple zones in the same region. Regional managed instance groups improve your application availability by spreading your instances across three zones. For example, a regional managed instance group in us-east1 will have instances in all three zones within us-east1: us-east1-d, us-east1-b, and us-east1-c. Regional managed instance groups also supports autoscaling, network load balancing, and HTTP(S) load balancing.

For more information on distributing instances using regional managed instance groups, see: https://cloud.google.com/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups.



Instance Template

Google Cloud Platform

Compute Engine

VM instances

Instance groups

Instance templates

CREATE INSTANCE TEMPLATE

Describe a VM instance once and then use that template to create groups of identical instances Learn more

Name

instance-template-2

Machine type

1 vCPU ▾ 3.75 GB memory Customize

Containers

□ Create instance template running Docker containers

Boot disk

New 10 GB standard persistent disk

Image

Debian GNU/Linux 8 (jessie)

Change

Identity and API access

Service account

Compute Engine default service account ▾

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

□ Allow HTTP traffic

□ Allow HTTPS traffic

Management, disk, networking, SSH keys

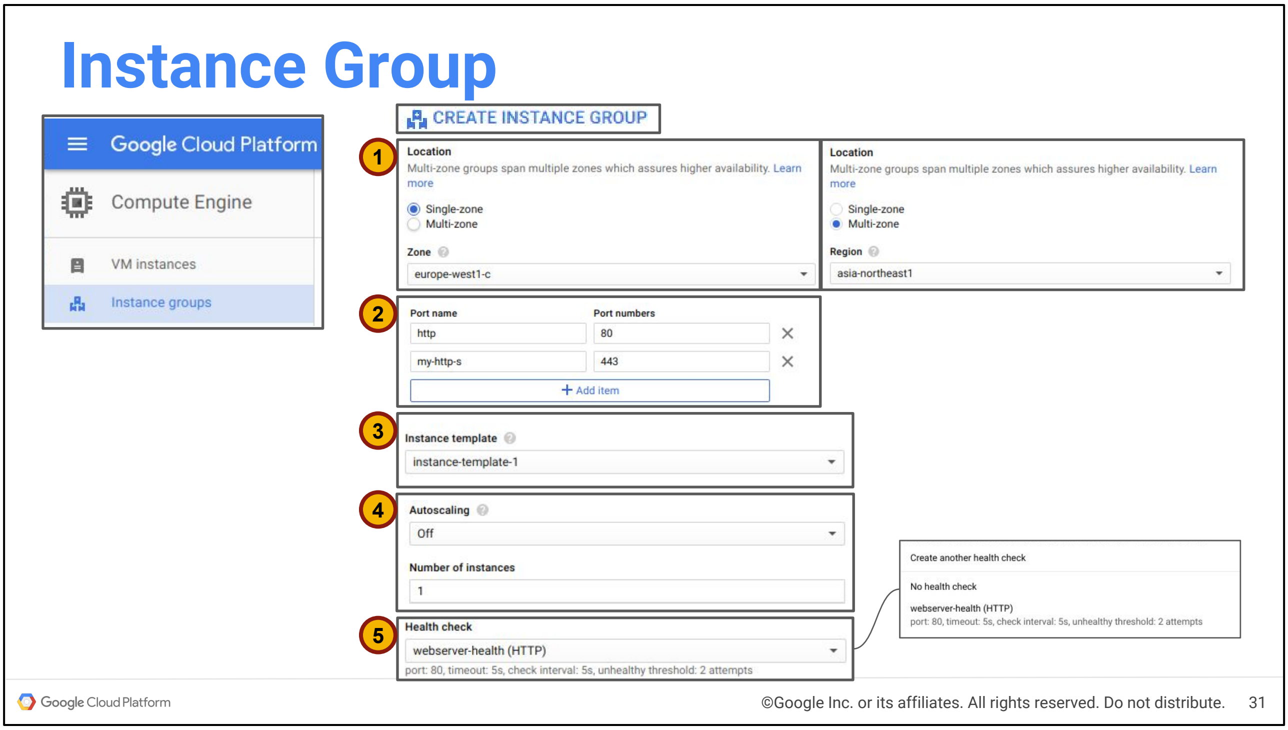
The instance template dialog looks and works exactly like creating an instance, except that it records the choices so it can repeat them.

Google Cloud Platform

Selecting **[X]** Allow HTTP traffic or **[X]** Allow HTTPS traffic will create firewall the rules.

Managed instance groups

- Create an instance template
- Create a managed instance group of n instances
- The instance group manager automatically populates the instance group based on the instance template



Instance Group

Google Cloud Platform

Compute Engine

VM instances

Instance groups

CREATE INSTANCE GROUP

1

Location

Multi-zone groups span multiple zones which assures higher availability. Learn more

● Single-zone
○ Multi-zone

Zone

europe-west1-c

Location

Multi-zone groups span multiple zones which assures higher availability. Learn more

○ Single-zone
● Multi-zone

Region

asia-northeast1

2

Port name	Port numbers
http	80
my-http-s	443

+ Add item

3

Instance template

instance-template-1

4

Autoscaling

Off

Number of instances

1

5

Health check

webserver-health (HTTP)

port: 80, timeout: 5s, check interval: 5s, unhealthy threshold: 2 attempts

Create another health check

No health check

webserver-health (HTTP)
port: 80, timeout: 5s, check interval: 5s, unhealthy threshold: 2 attempts

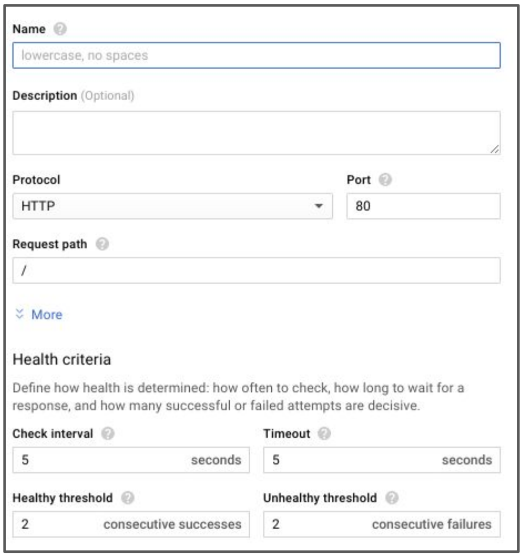
Google Cloud Platform

Instance Group configuration.
(1) Location of VMs. Single zone (pick the zone), Multi-zone (pick the region). VMs are evenly distributed in three zones in the region.
(2) Symbolic port name. The name is used by the Forwarding Rules to pass traffic to the instance group.
(3) This is where you select the pre-existing Instance Template that will be used to create VMs in the group.
(4) Autoscaling [Off] and the number of instances in the group. For availability when using a Multi-zone group, 3 minimum (one in each zone) is recommended. (*Autoscaling is covered in the next module*).
(5) Health check. When the health check is set, if a VM meets the "unhealthy" conditions, it is deleted and a new replacement is created. This process is called **Autohealing**.

Use cases for Managed Instance Groups

- Scale up / scale down
  - Target with autoscaler
- Deploy software / rolling update
- High availability

Create another health check



Name
lowercase, no spaces

Description (Optional)

Protocol: HTTP

Port: 80

Request path: /

More

Health criteria

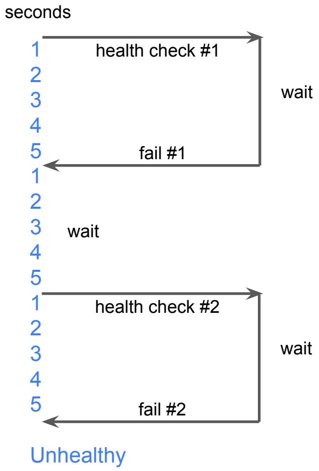
Define how health is determined: how often to check, how long to wait for a response, and how many successful or failed attempts are decisive.

Check interval: 5 seconds

Timeout: 5 seconds

Healthy threshold: 2 consecutive successes

Unhealthy threshold: 2 consecutive failures



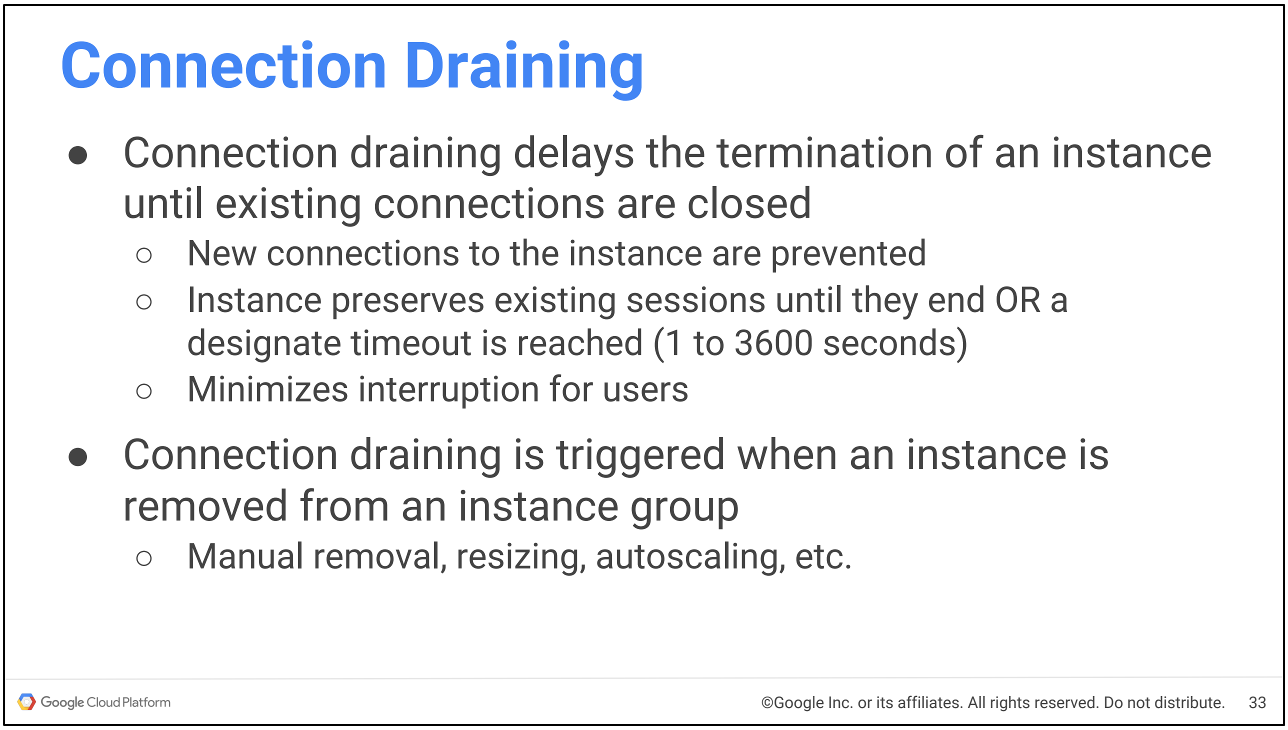
seconds

1 health check #1
2
3 wait
4
5 fail #1
1
2
3 wait
4
5
1 health check #2
2
3 wait
4
5 fail #2

Unhealthy

Protocol: HTTP or HTTPS

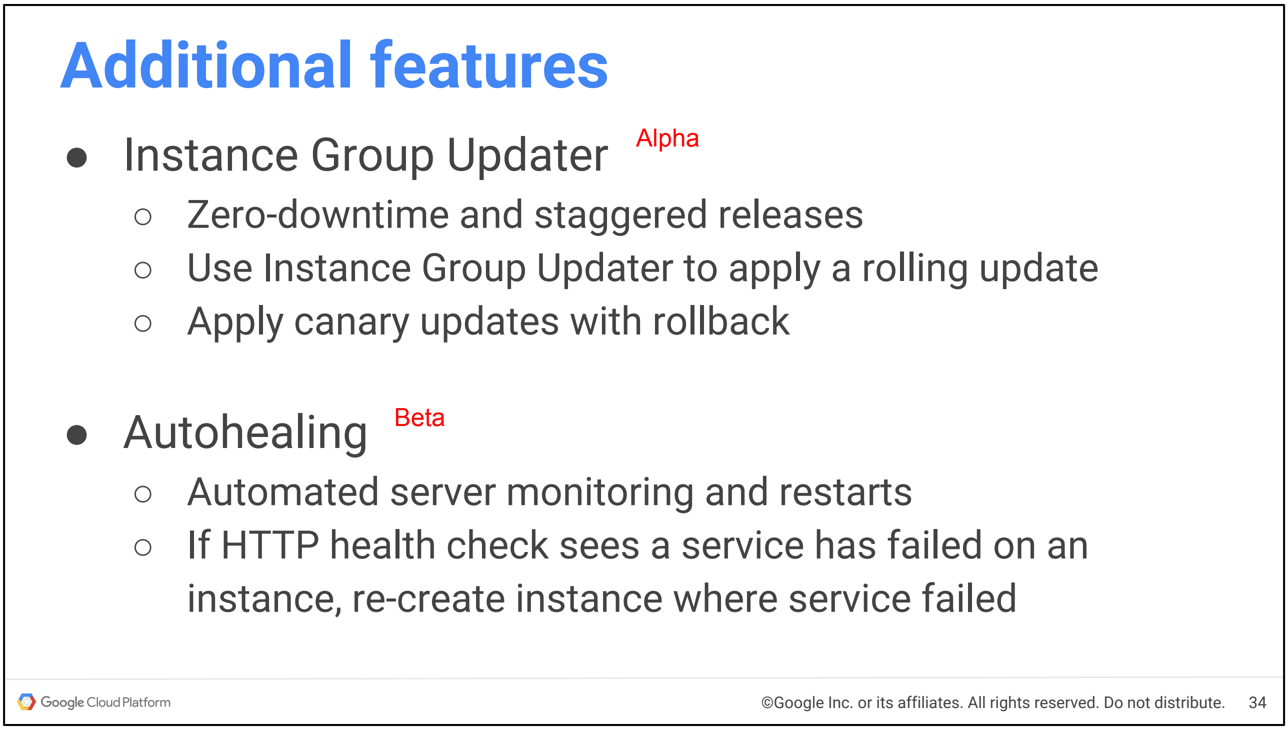
Check interval - how long to wait between attempts to check the health of a particular VM
Timeout - how long to wait for a response before declaring that the check attempt has failed
Health threshold - how many consecutive successes indicates that the VM is "healthy" and should receive traffic
Unhealthy threshold - how many consecutive failures indicates that the VM is "unhealthy" and should be replaced



Connection Draining

- Connection draining delays the termination of an instance until existing connections are closed
  - New connections to the instance are prevented
  - Instance preserves existing sessions until they end OR a designate timeout is reached (1 to 3600 seconds)
  - Minimizes interruption for users
- Connection draining is triggered when an instance is removed from an instance group
  - Manual removal, resizing, autoscaling, etc.

For more information on connection draining, see https://cloud.google.com/compute/docs/load-balancing/enabling-connection-draining



Additional features

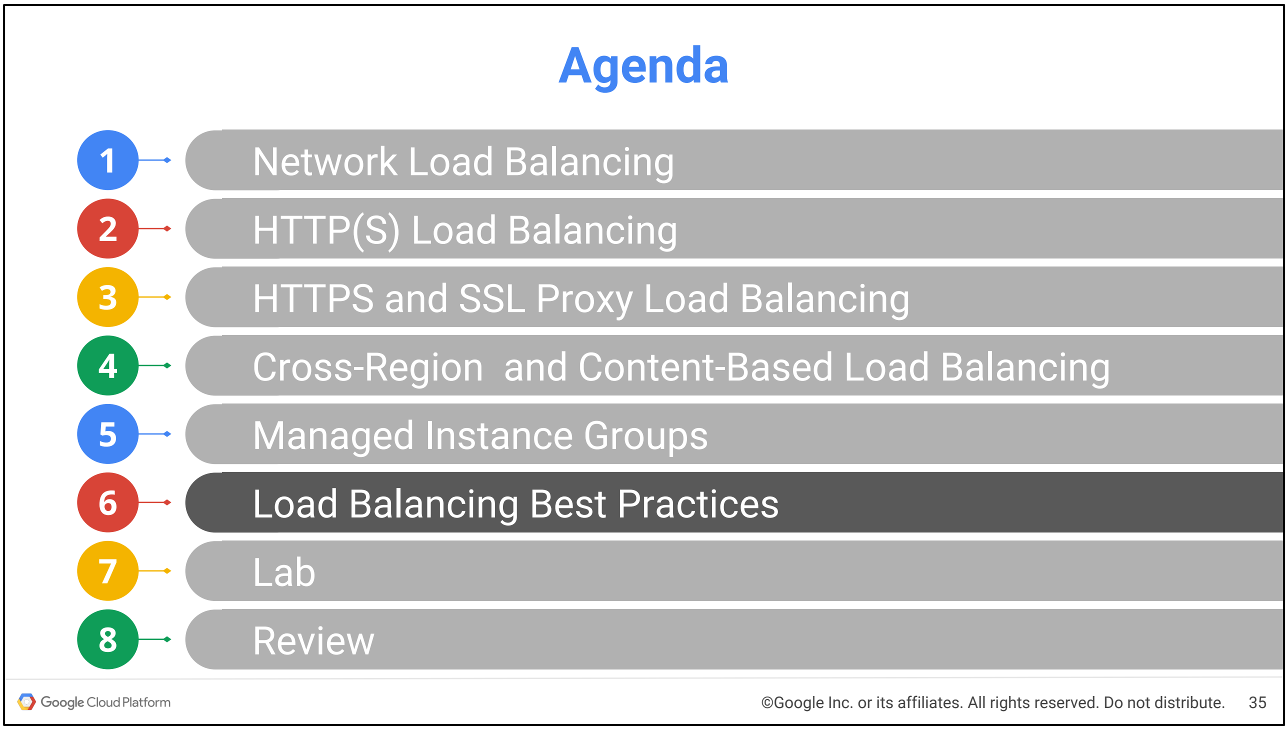
- Instance Group Updater Alpha
  - Zero-downtime and staggered releases
  - Use Instance Group Updater to apply a rolling update
  - Apply canary updates with rollback
- Autohealing Beta
  - Automated server monitoring and restarts
  - If HTTP health check sees a service has failed on an instance, re-create instance where service failed

Google Cloud Platform

Need to update software on running service, you'd kill a server and replace it with another. All new instances are in the new template.
The instance Group Update updates instances by applying a new instance template. Updates can be proactive or opportunistic, and can roll out gradually to all instances or only to a subset of instances (and rolled forward after test period). For more information on rolling updates, see: https://cloud.google.com/compute/docs/instance-groups/updating-managed-instance-groups

For more information on autohealing, see: https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-managed-instances#monitoring_groups

For more information on rolling updates, see: https://cloud.google.com/compute/docs/instance-groups/#starting_an_update. For more information on autohealing, see: https://cloud.google.com/compute/docs/instance-groups/#monitoring_groups.



Agenda

1 Network Load Balancing

2 HTTP(S) Load Balancing

3 HTTPS and SSL Proxy Load Balancing

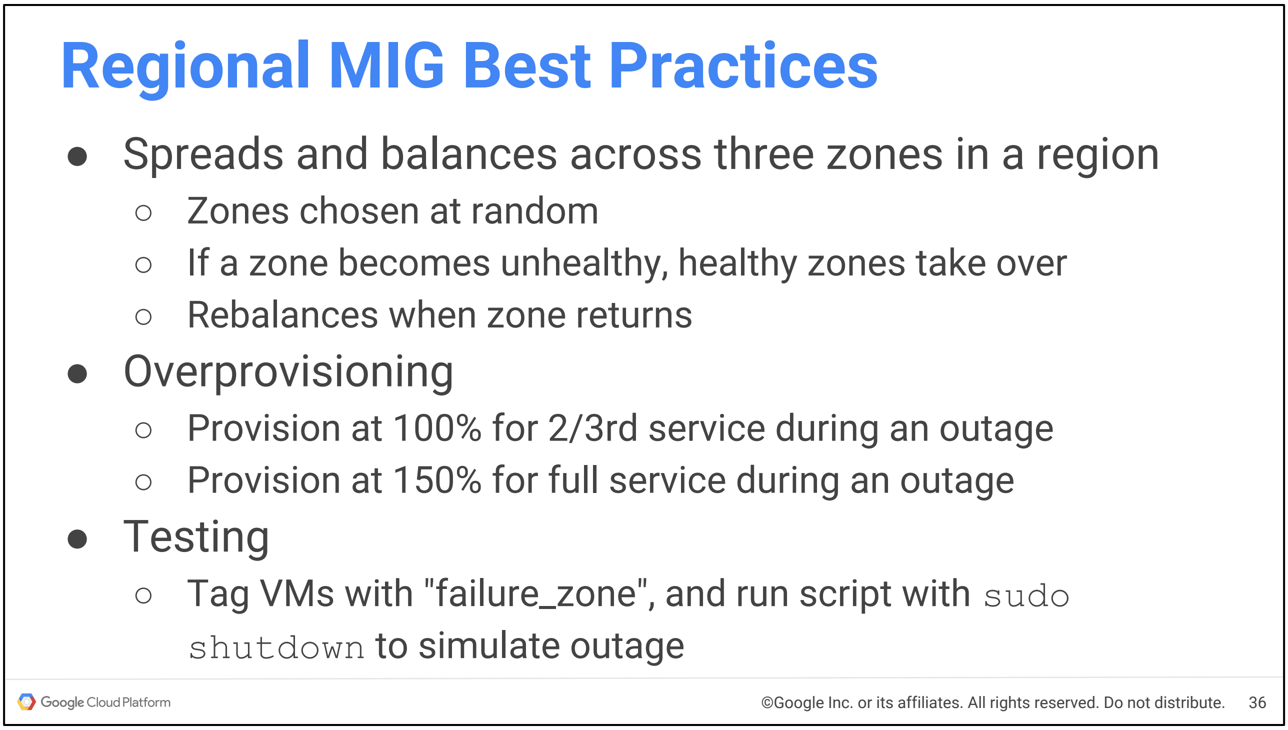
4 Cross-Region and Content-Based Load Balancing

5 Managed Instance Groups

6 Load Balancing Best Practices

7 Lab

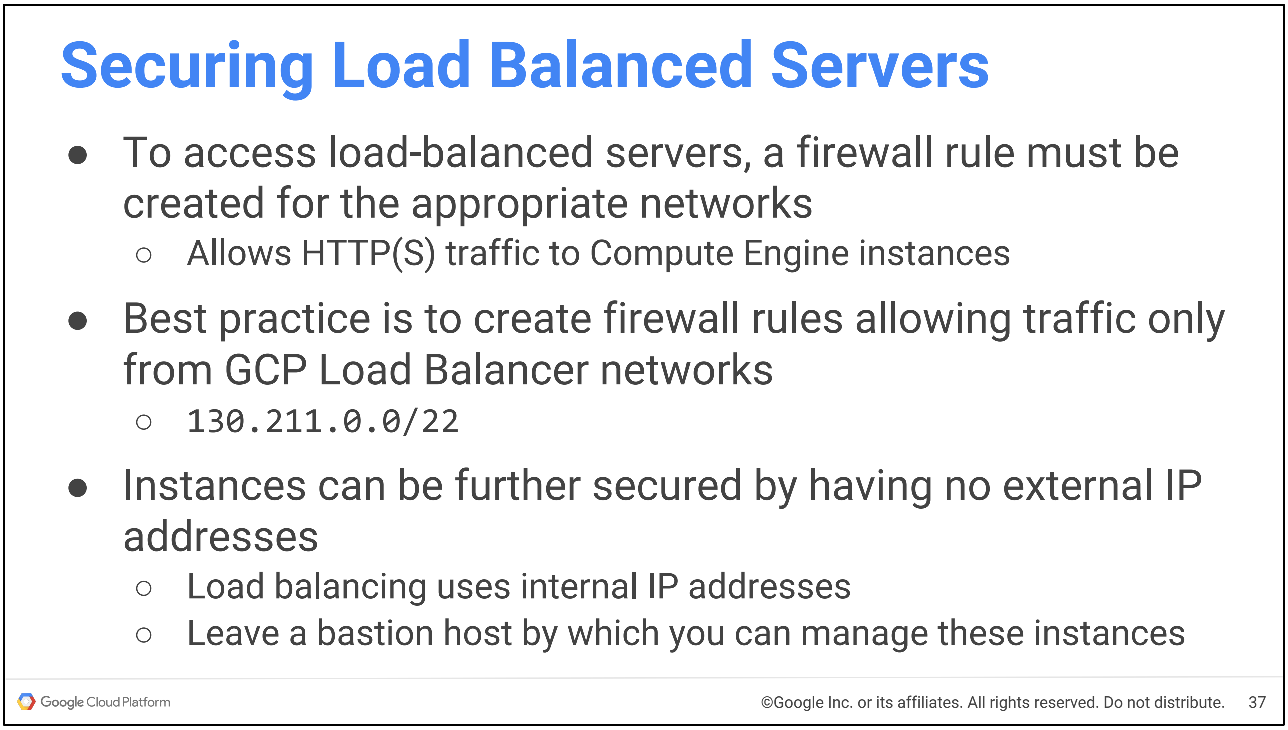
8 Review



Regional MIG Best Practices

- Spreads and balances across three zones in a region
  - Zones chosen at random
  - If a zone becomes unhealthy, healthy zones take over
  - Rebalances when zone returns
- Overprovisioning
  - Provision at 100% for 2/3rd service during an outage
  - Provision at 150% for full service during an outage
- Testing
  - Tag VMs with "failure_zone", and run script with `sudo shutdown` to simulate outage

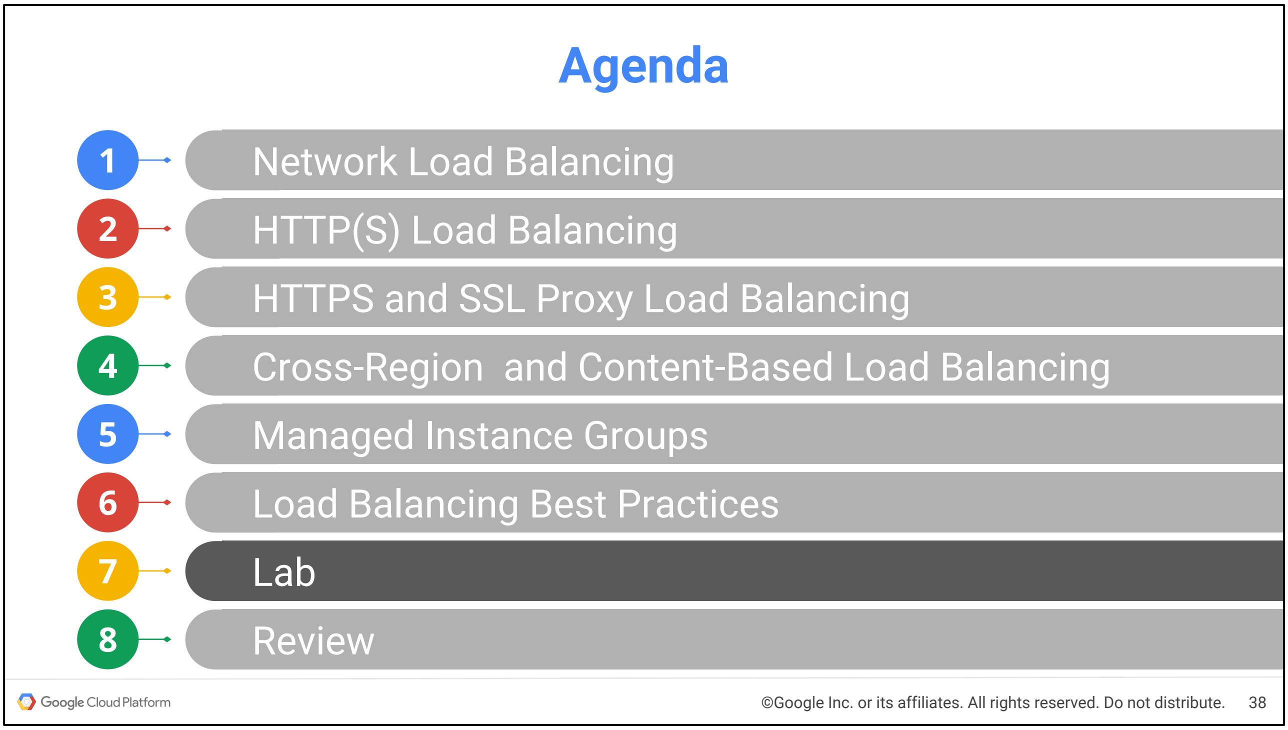
https://cloud.google.com/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups



Securing Load Balanced Servers

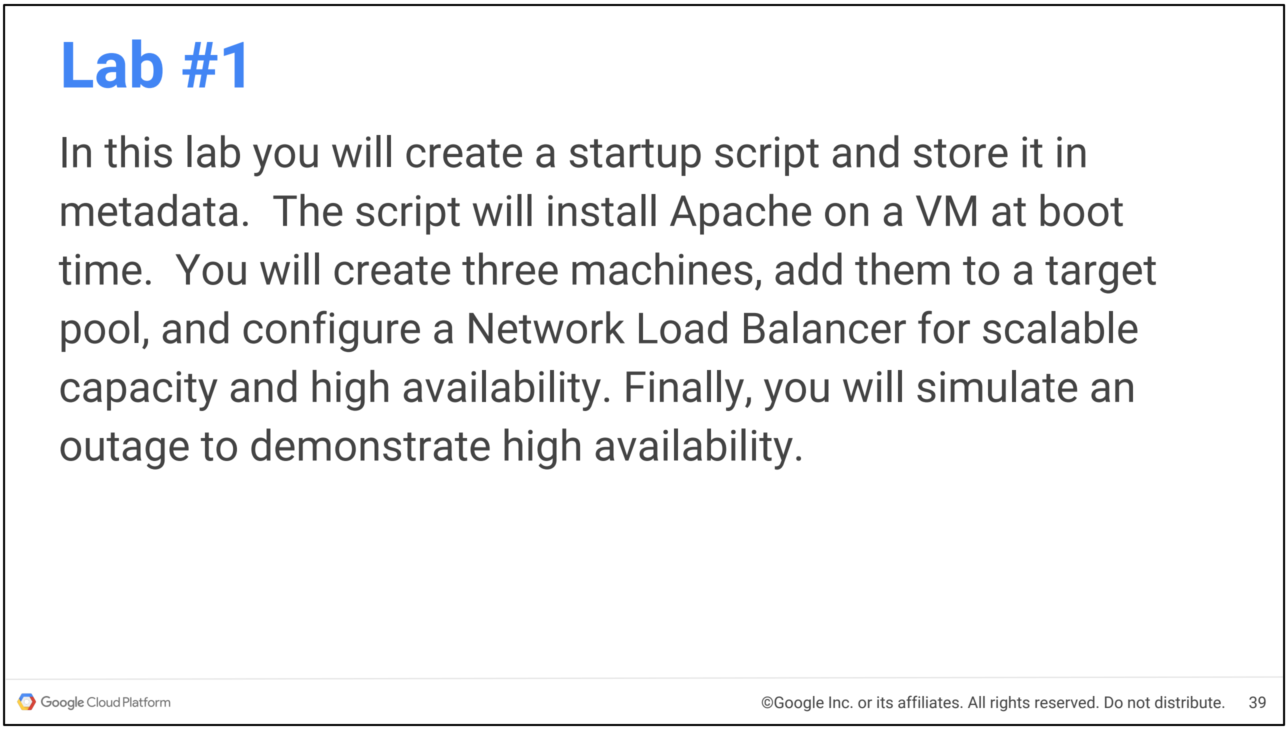
- To access load-balanced servers, a firewall rule must be created for the appropriate networks
  - Allows HTTP(S) traffic to Compute Engine instances
- Best practice is to create firewall rules allowing traffic only from GCP Load Balancer networks
  - `130.211.0.0/22`
- Instances can be further secured by having no external IP addresses
  - Load balancing uses internal IP addresses
  - Leave a bastion host by which you can manage these instances

Google Cloud Platform



Agenda

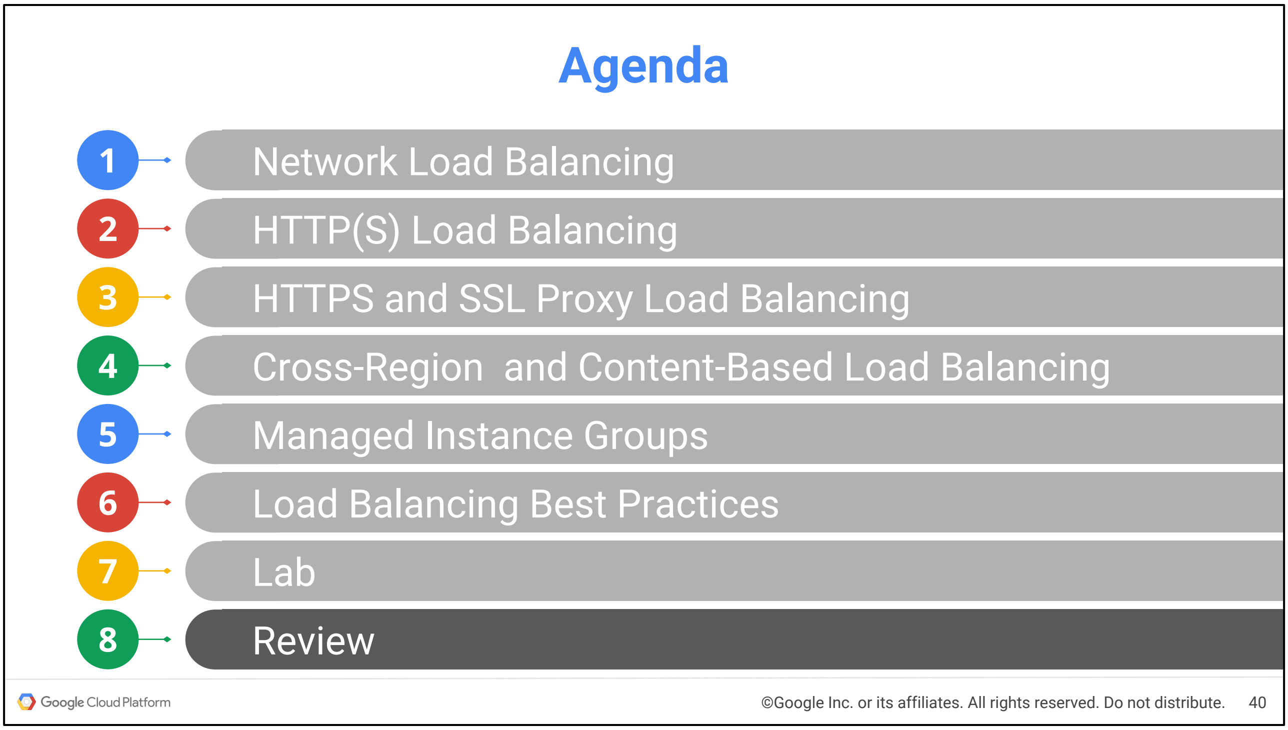
1. Network Load Balancing
2. HTTP(S) Load Balancing
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. Review



Lab #1

In this lab you will create a startup script and store it in metadata. The script will install Apache on a VM at boot time. You will create three machines, add them to a target pool, and configure a Network Load Balancer for scalable capacity and high availability. Finally, you will simulate an outage to demonstrate high availability.

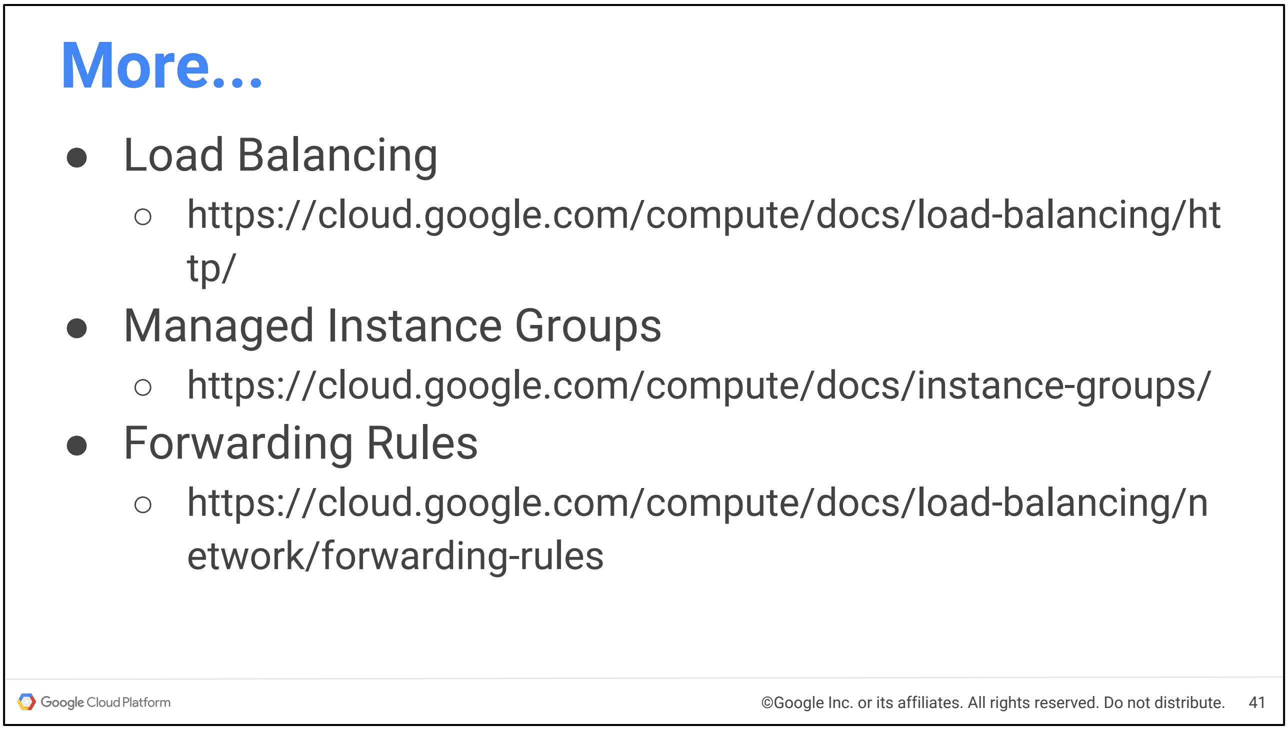
09-1 Virtual Machine Automation and LB



Agenda

1. Network Load Balancing
2. HTTP(S) Load Balancing
3. HTTPS and SSL Proxy Load Balancing
4. Cross-Region and Content-Based Load Balancing
5. Managed Instance Groups
6. Load Balancing Best Practices
7. Lab
8. **Review**

Google Cloud Platform



More...

- Load Balancing
  - https://cloud.google.com/compute/docs/load-balancing/http/
- Managed Instance Groups
  - https://cloud.google.com/compute/docs/instance-groups/
- Forwarding Rules
  - https://cloud.google.com/compute/docs/load-balancing/network/forwarding-rules

More to learn on this subject. Here are some suggestions and links.

More...

- Target Pools
  - https://cloud.google.com/compute/docs/reference/latest/targetPools
- Global Forwarding Rules
  - https://cloud.google.com/compute/docs/load-balancing/http/global-forwarding-rules
- Target Proxies
  - https://cloud.google.com/compute/docs/load-balancing/http/target-proxies

cloud.google.com