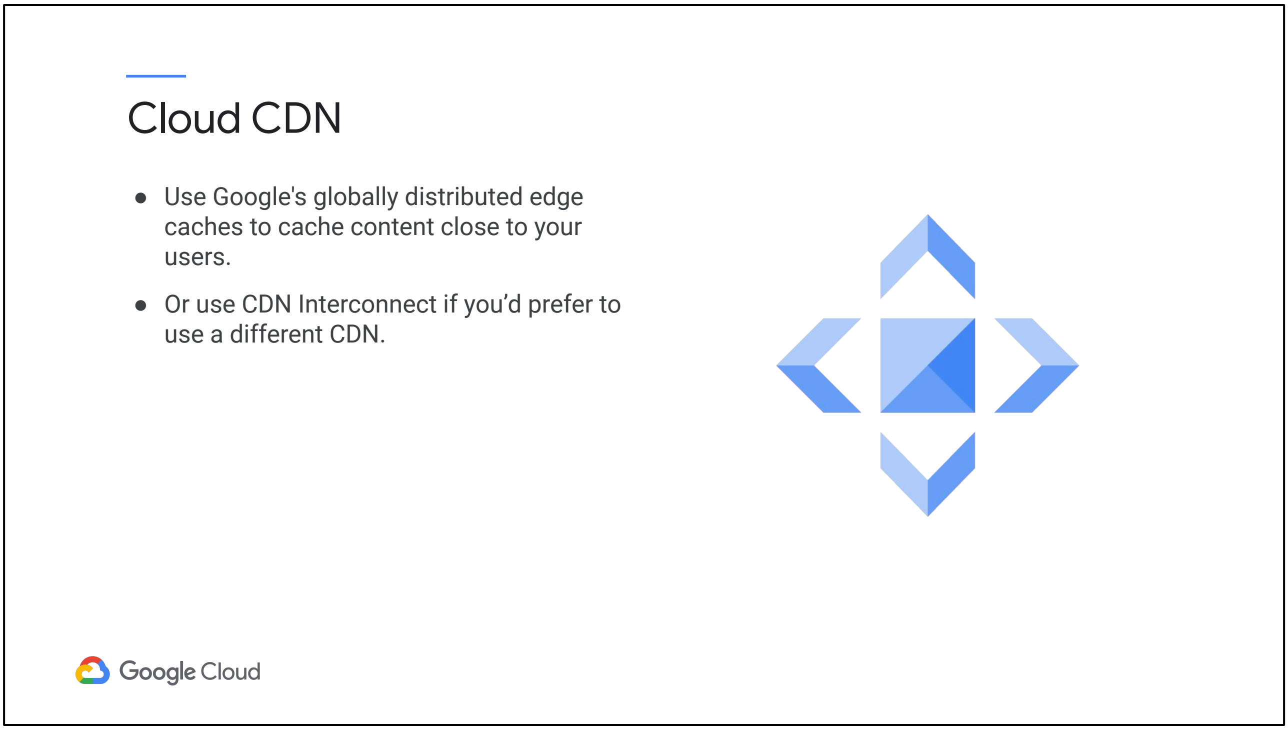
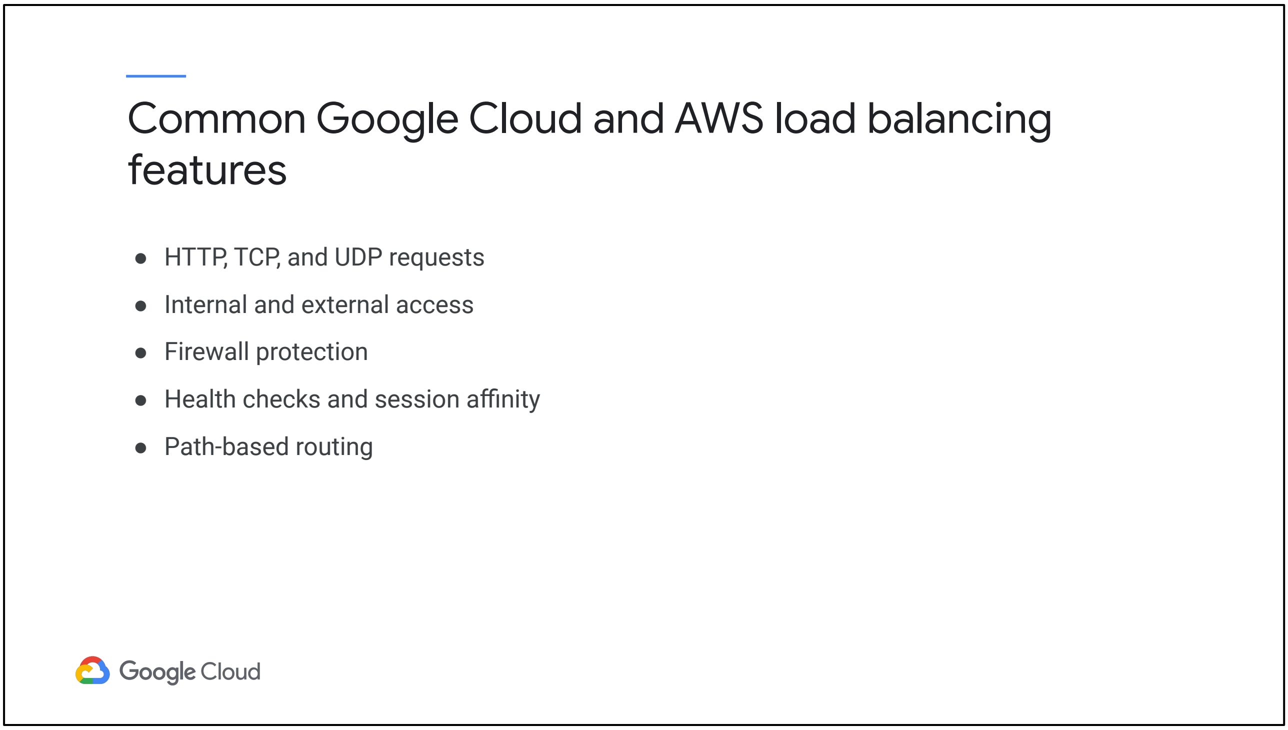


Cloud CDN

- Use Google's globally distributed edge caches to cache content close to your users.
- Or use CDN Interconnect if you'd prefer to use a different CDN.

Google has a global system of edge caches. You can use this system to accelerate content delivery in your application using Cloud CDN. Your customers will experience lower network latency, the origins of your content will experience reduced load, and you can save money too. Once you've set up HTTP(S) Load Balancing, simply enable Cloud CDN with a single checkbox.

There are lots of other CDNs out there, of course. If you are already using one, chances are, it is a part of Google Cloud’s CDN Interconnect partner program, and you can continue to use it.



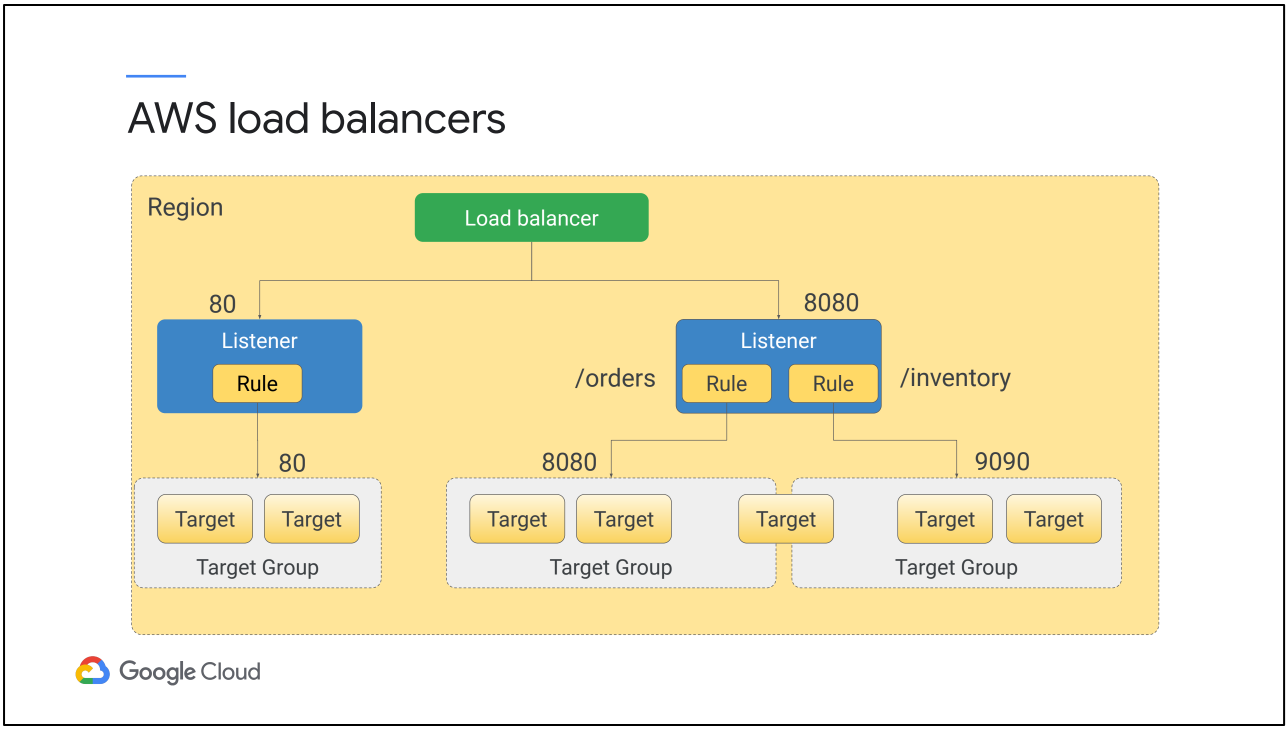
Common Google Cloud and AWS load balancing features

- HTTP, TCP, and UDP requests
- Internal and external access
- Firewall protection
- Health checks and session affinity
- Path-based routing

Google Cloud

There are several common features between Google and Amazon when comparing load balancing.

- Requests can be in HTTP, TCP, or UDP.
- Requests can come from the internet or from internal resources.
- The load balancers can be positioned behind a firewall to limit client access.
- Both support health checks and session affinity.
- In addition, Google's HTTP load balancer and Amazon's Application Load Balancer support path-based routing to micro-services, so we'll focus on them.



AWS load balancers

Region

Load balancer

80

Listener

Rule

8080

Listener

/orders

Rule

Rule

/inventory

80

8080

9090

Target

Target

Target Group

Target

Target

Target

Target Group

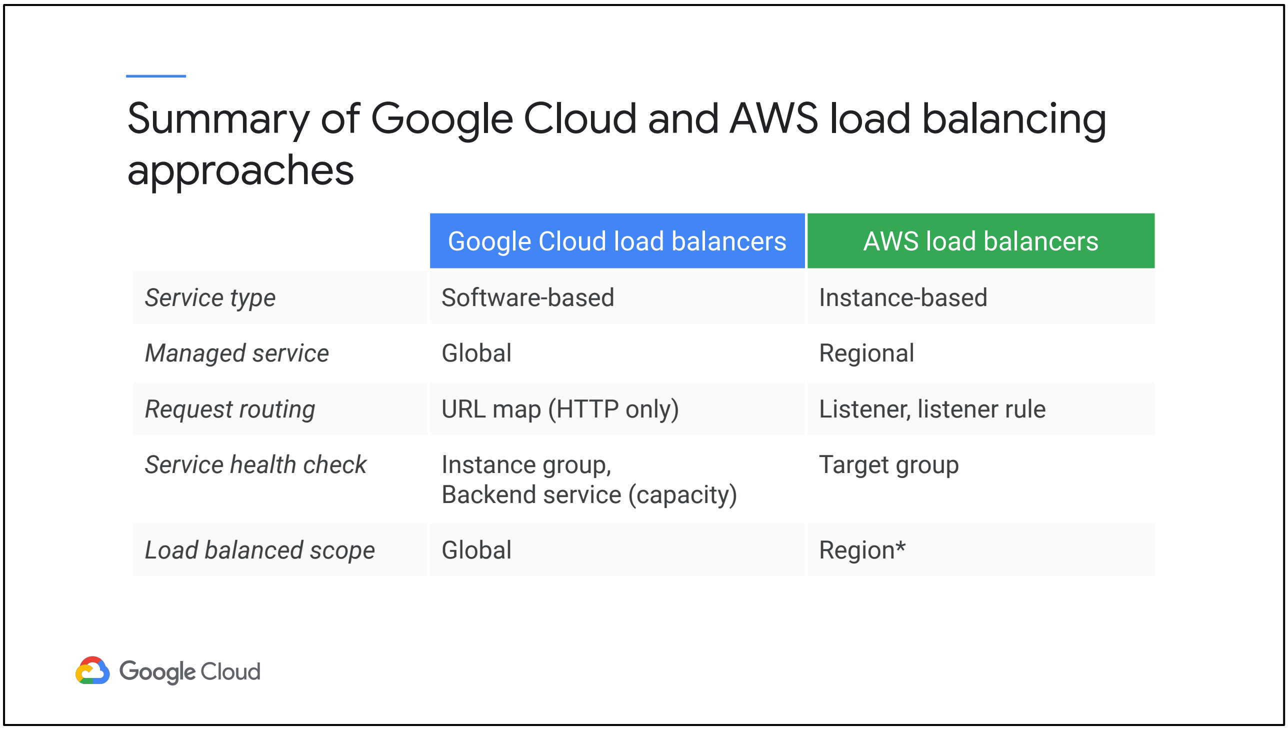
Target

Target

Target Group

AWS load balancers are a managed service that is instance-based, and the load balancer can take minutes to scale. They are built within a region and can only distribute traffic in the region. So, if you are expecting a significant amount of traffic to reach the load balancer at a predetermined time, it may be necessary to open a ticket with AWS support to pre-warm a load balancer to handle the traffic.

Elastic Load Balancers can handle HTTP(S), TCP, and application traffic.



Summary of Google Cloud and AWS load balancing approaches

	Google Cloud load balancers	AWS load balancers
Service type	Software-based	Instance-based
Managed service	Global	Regional
Request routing	URL map (HTTP only)	Listener, listener rule
Service health check	Instance group, Backend service (capacity)	Target group
Load balanced scope	Global	Region*

Google Cloud

Let's summarize the key differences between the approaches taken by Google Cloud and AWS to load balancing.

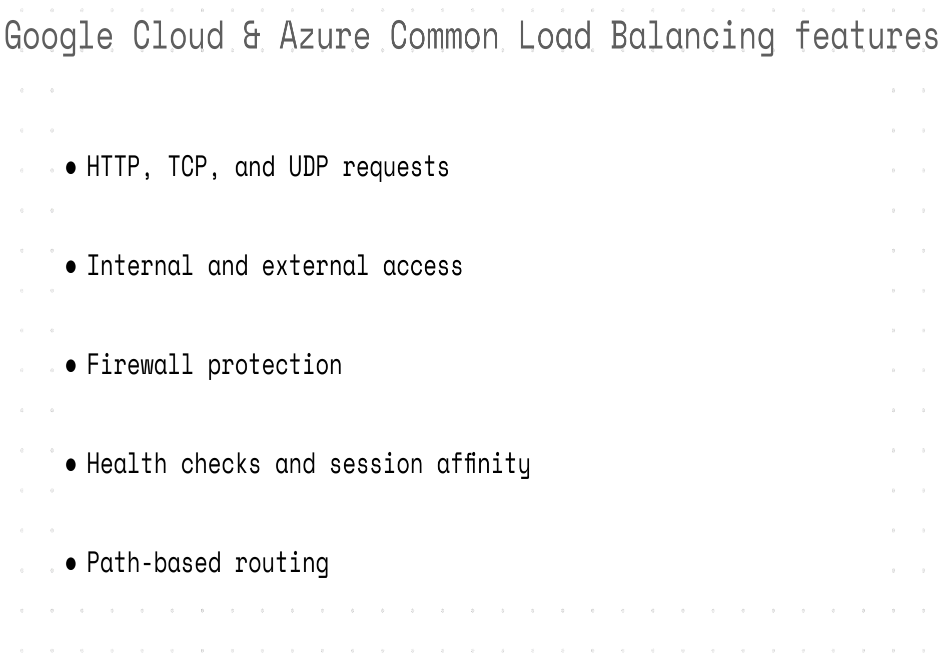
- Google Cloud load balancers are software-based. AWS load balancers run on EC2 instances. The difference in service type is the reason that you don't have to pre-warm load balancers in Google Cloud. The service can spin up quickly because a virtual machine is not used. An EC2 instance can take minutes to start. So if you are expecting a heavy amount of traffic in a short period of time on AWS, you may have to open a support ticket to manually start more load balancer virtual machines.

- Google Cloud load balancers are a global managed service. The load balancer can survive a region outage because the service is not built on a region. The service is built on the network edges. AWS load balancers are built in the regions. So, a region that goes down can take the load balancer with it.

- A Google Cloud load balancer can route traffic based on a URL for HTTP only. The URL can direct traffic to a particular backend service. AWS can route traffic based on a listener and a listener rule for HTTP and TCP.

- Both Google Cloud and AWS leverage health checks to ensure that traffic is only sent to healthy instances and can autoheal, which is an option to destroy virtual machines with new instances. But Google Cloud health checks can be used on both the load balancer, via the backend service, and the instance group. AWS only applies the health check on the target group.

- Google Cloud offers load balancers that can distribute traffic both globally and within a region. AWS load balancers can only distribute traffic within a region. To achieve load balancing at a global scale in AWS you would need to combine regional load balancers with AWS Global Accelerator

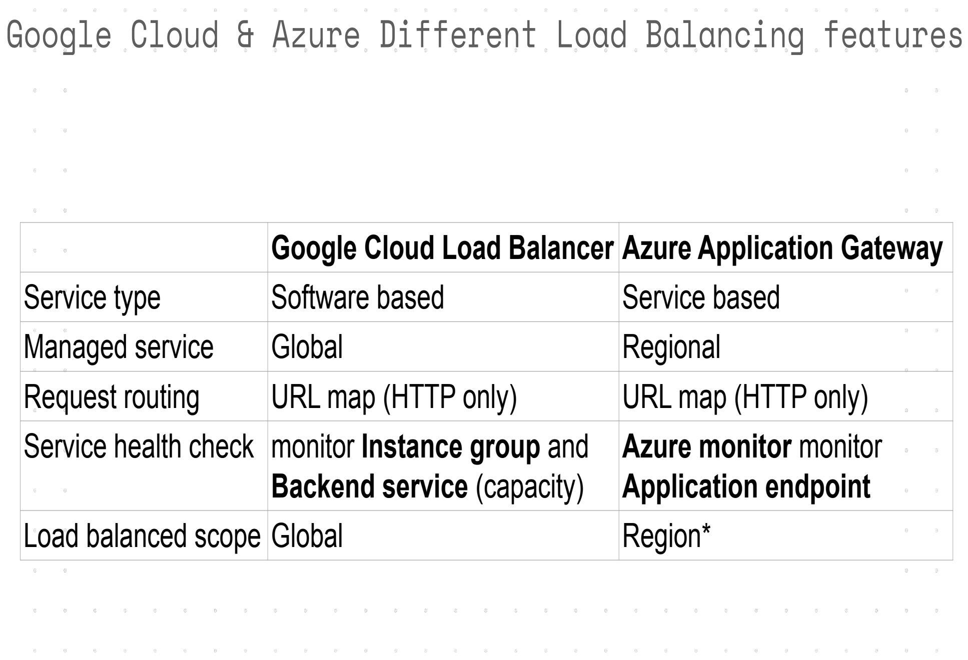


Google Cloud & Azure Common Load Balancing features

- HTTP, TCP, and UDP requests
- Internal and external access
- Firewall protection
- Health checks and session affinity
- Path-based routing

There are a number of common features between Google and Azure when comparing load balancing.

- Requests can be in HTTP, TCP, or UDP.
- Requests can come from the internet or from internal resources.
- The load balancers can be positioned behind a firewall to limit client access.
- Both support health checks and session affinity.
- In addition, Google's **HTTP(S) Load Balancing** and **Azure Application Gateway** support path-based routing to **micro-services**.



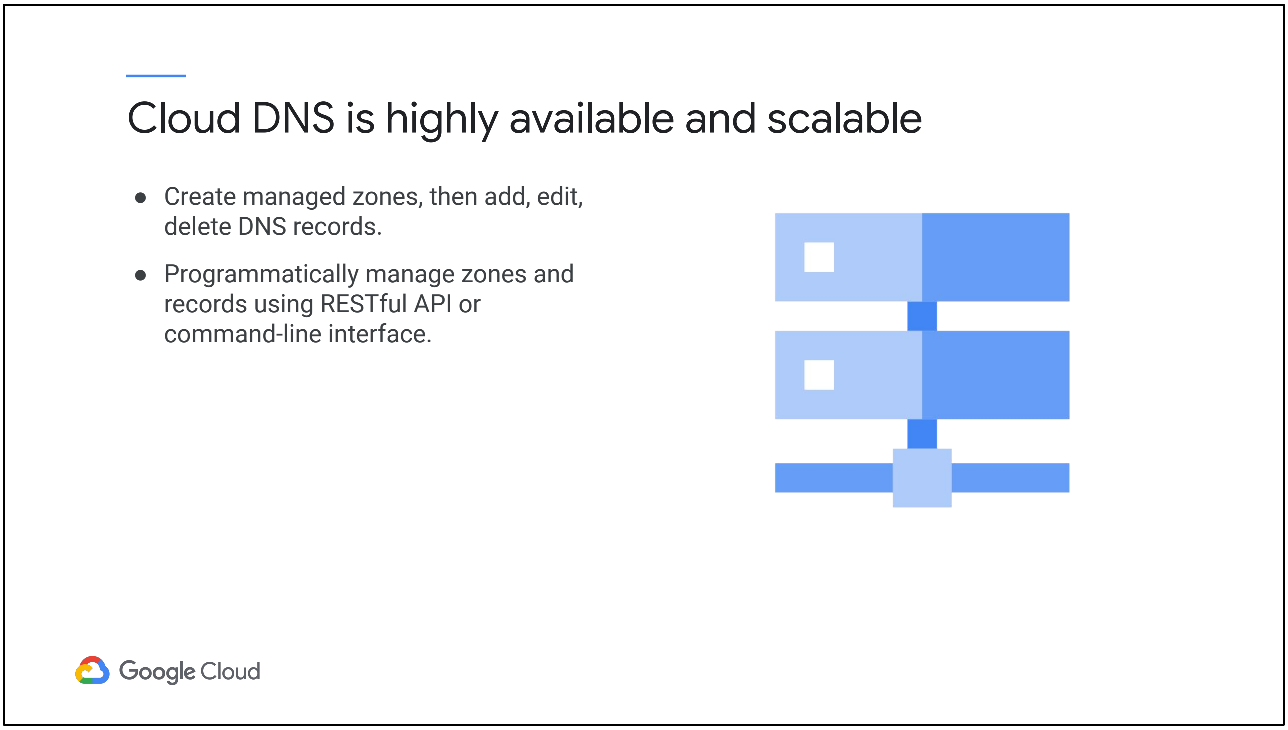
Google Cloud & Azure Different Load Balancing features

	Google Cloud Load Balancer	**Azure Application Gateway**
Service type	Software based	Service based
Managed service	Global	Regional
Request routing	URL map (HTTP only)	URL map (HTTP only)
Service health check	monitor **Instance group** and **Backend service** (capacity)	**Azure monitor** monitor **Application endpoint**
Load balanced scope	Global	Region*

Let's summarize the key differences between the approaches taken by Google Cloud and Azure to load balancing.

- The Google Cloud Load Balancer is a software based. Azure uses a service based approach in their **Azure Application Gateway**.
- The GCP Load Balancer is a global managed service.
  - GCP stands for Google Cloud Platform, which now is not an official abbreviation like in the past.
  - The load balancer can survive a region outage because the service is not built on a region.
  - The service is built on the network edges.

- **Azure Applications Gateway** is built in the regions.
    - So, if a region goes down it can take the load balancer with it.
- A GCP load balancer can route traffic based on a URL for HTTP only.
    - The URL can direct traffic to a particular backend service.
    - Azure uses a similar routing method to achieve this.
- Both GCP and Azure leverage health checks to ensure traffic is only sent to healthy instances and can autoheal, which is an option to destroy VMs with new instances.
    - But GCP health checks can be used on both
        - **backend services**
        - **instance groups**
    - Azure uses **Azure monitor** to monitor the **application endpoint**.
- GCP offers load balancers that can distribute traffic both globally and within a region.
    - Azure load balancers can only distribute traffic within a region.
    - You would need to use **Azure Traffic Manager** to take advantage of a cross-regional service.



Cloud DNS is highly available and scalable

- Create managed zones, then add, edit, delete DNS records.
- Programmatically manage zones and records using RESTful API or command-line interface.

Google Cloud

One of the most famous Google services that people don’t pay for is 8.8.8.8, which provides a public Domain Name Service to the world. DNS is what translates Internet hostnames to addresses, and as you would imagine, Google has a highly developed DNS infrastructure. It makes 8.8.8.8 available so that everybody can take advantage of it.

But what about the Internet hostnames and addresses of applications you build in Google Cloud?
Google Cloud offers Cloud DNS to help the world find them. It’s a managed DNS service running on the same infrastructure as Google. It has low latency and high availability, and it’s a cost-effective way to make your applications and services available to your users. The DNS information you publish is served from redundant locations around the world.

Cloud DNS is also programmable. You can publish and manage millions of DNS zones and records using the Cloud Console, the command-line interface, or the API.