GCP Certification Series: 2.4 Planning and configuring network resources

Prashanta Paudel

Oct 22, 2018 · 18 min read

Google is a global company and has the network all over the world. So, if you use google you can easily go global with few clicks. Even though not all companies are in the cloud all eventually they will feel the need to do so if they expand beyond their boundaries.

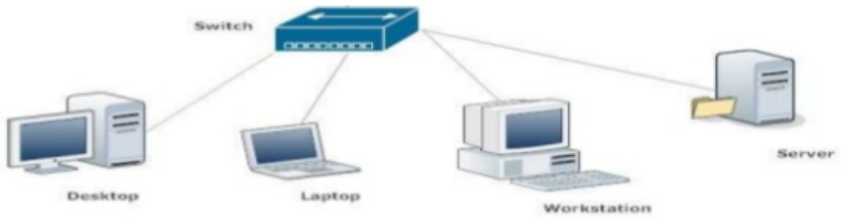
In this article, we will look at the networking aspect of the Google Cloud platform. We will see how projects are stored in the network, what resources should be placed where and what will be the good solution in this blog.

I will try to cover the concept part first then go in detail.

Computer Network

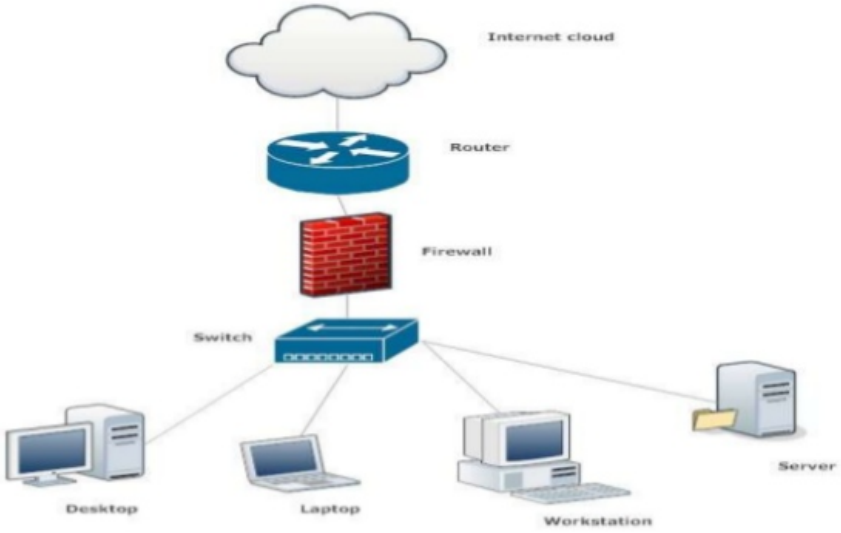
A computer network is a set of computers connected together for the purpose of sharing resources. It is often referred to as the data network. Computer network today may comprise of fiber, wireless and cable network. It is a physical layer of the OSI model. The most common resource shared today is the connection to the Internet.

In early days, a computer network used to be an interconnection between various devices within a small office or university. When computer become more and more used the network connecting them also developed. Earlier the most important device to connect to a network used to be the printer or server but now there are so many devices that hook up to a network that I cannot say this is the most important one.



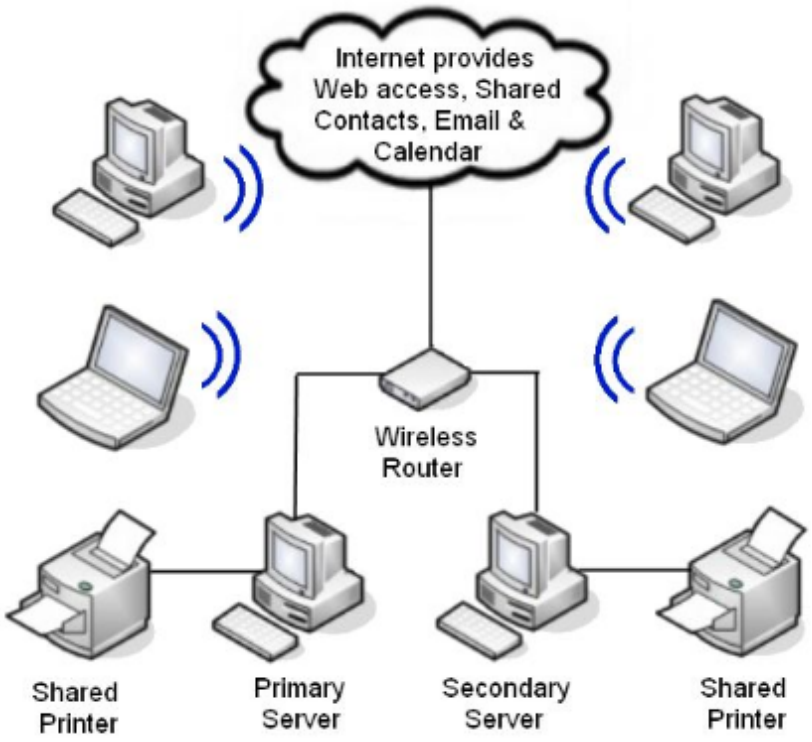
Simple network without internet—early days

After the development of the internet more and more servers and services started serving online but having a network owned far from the office location for computing purpose was not even imagined at that time. During the past 50 years, internet speed has changed from 56 kbps to 100's Gbps or more as the speed is increasing frequently.



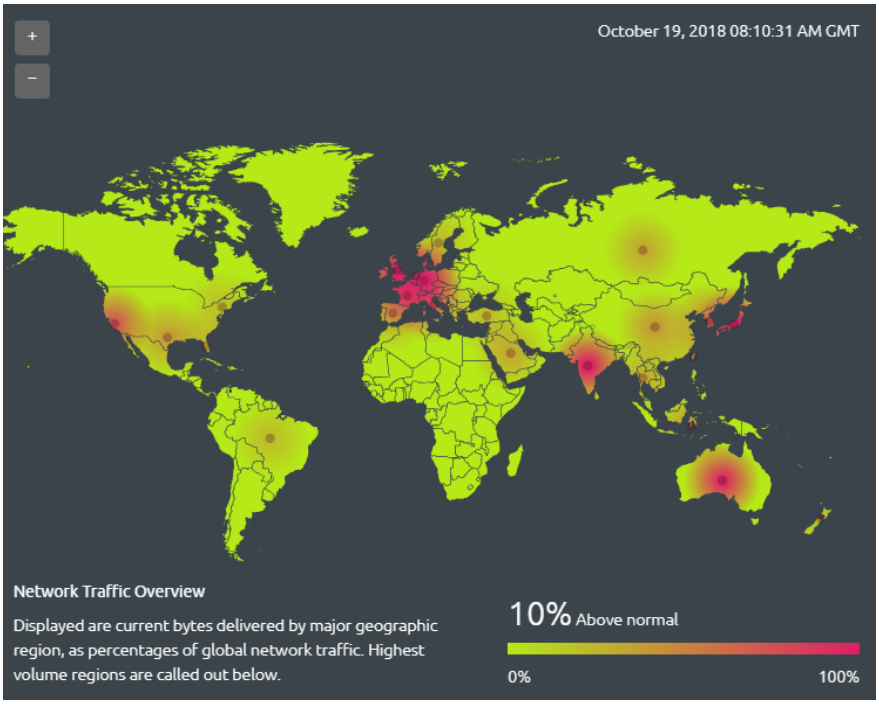
basic Network

Wireless internet and Wi-Fi became part of the lives for so many years now. 4G is available in most of the countries now and 5G is being launched now in Finland. So, network technology and speed have been developed at a fast pace.



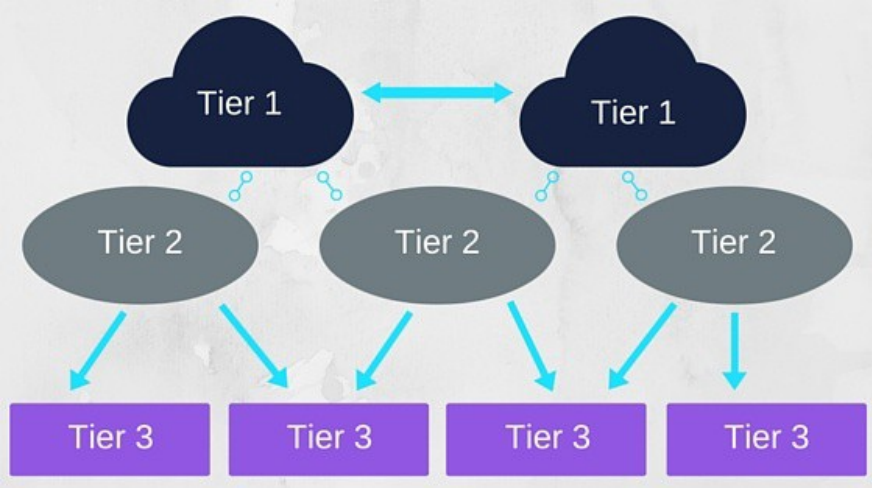
Basic office network today

If you look at the real-time network traffic today!

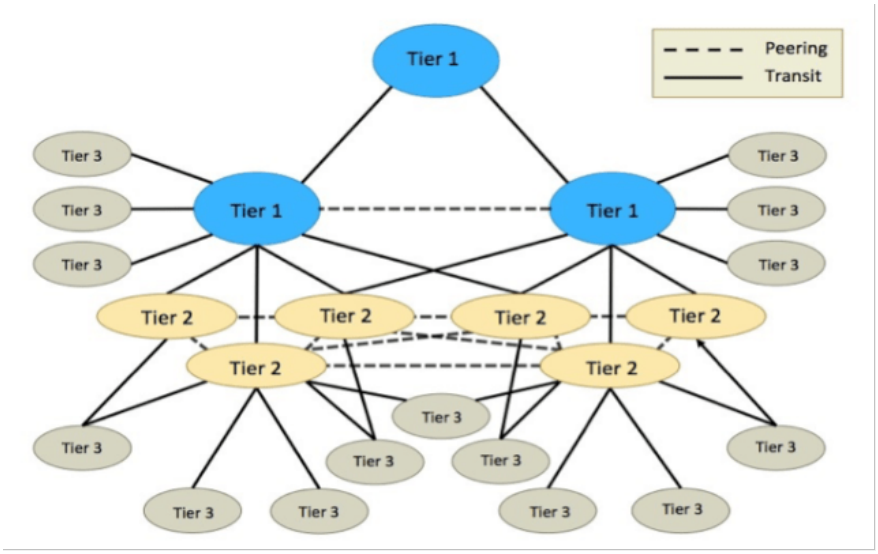


References: https://www.akamai.com/uk/en/solutions/intelligent-platform/visualizing-akamai/real-time-web-monitor.jsp

So, looking from top-view whole world's internet is divided into 3 tiers according to network providers



References: https://datapath.io/resources/blog/what-is-an-internet-service-provider/



References: https://orhanergun.net/2017/01/tier-1-tier-2-tier-3-service-providers/

Tier 1 can connect to each and every device in the world connected to the internet ***in a very short route.*** A Tier 1 network is an Internet Protocol network that can reach every other network on the Internet solely via settlement-free interconnection, also known as settlement-free peering. In other words, Tier 1 networks can exchange traffic with other Tier 1 network without having to pay any fees. Tier 1 Internet providers are the networks that provide the backbone of the Internet. We call them backbone Internet providers. These providers build

infrastructure such as the Atlantic Internet sea cables. They provide traffic to all other Internet providers, not end users.———— {Connection: peering Agreement }

List of Tier 1 network providers

Name	Headquarters	AS number	February 2018 degree[10][11]	Fiber Route Miles	Fiber Route km	Peering Policy
AT&T[12]	United States	7018	2,228	410,000	660,000[13]	AT&T Peering policy
CenturyLink (formerly Level 3, Qwest, Savvis, Global Crossing, TW Telecom and Exodus) [14] [15]	United States	209 3356 3549 4323	1,888 4,976 2,536 2,028	750,000	885,139[16][17]	North America; International Level 3 Peering Policy
Deutsche Telekom AG (ICSS)[18]	Germany	3320	581	?	?	DTAG Peering Details
GTT Communications, Inc. (formerly Tinet & nLayer)[19]	United States	3257 (4436)	1,576	25,000	40,000[20]	GTT Peering Policy
KPN International[21]	Netherlands	286	276	75,000	120,000[22]	KPN Peering Policy
Liberty Global[23][24]	United Kingdom[25]	6830	777	500,000	800,000[26]	Peering Principles
NTT Communications (America) (formerly Verio)[27]	Japan	2914	1,714	?	?	North America
Orange (OpenTransit)[28]	France	5511	181	?	?	OTI peering policy
PCCW Global	Hong Kong	3491	680	?	?	Peering policy
Sprint (SoftBank Group)[29]	Japan	1239	392	26,000	42,000[30]	Peering policy
Tata Communications India Limited (Acquired Teleglobe)[31]	India	6453	724	435,000	700,000[32]	Peering Policy
Telecom Italia Sparkle (Seabone)[33]	Italy	6762	482	347,967	560,000	Peering Policy
Telxius (Subsidiary of Telefónica)[34]	Spain	12956	304	40,000	65,000[35]	Peering Policy
Telia Carrier[36]	Sweden	1299	1,664	40,000	65,000[37]	TeliaSonera International Carrier Global Peering Policy
Verizon Enterprise Solutions (formerly UUNET and XO Communications)[42]	United States	701 702 703 2828	1,204 280 98 1,031	500,000	805,000[43]	Verizon UUNET Peering policy 701, 702, 703
Zayo Group (formerly AboveNet)[44]	United States	6461	1,718	122,000	196,339[45]	Zayo Peering Policy

References: https://en.wikipedia.org/wiki/Tier_1_network

A Tier 2 network is an Internet service provider which engages in the practice of peering with other networks, but which also purchases IP transit to reach some portion of the Internet.

A tier 2 ISP is a service provider who connects between tier 1 and tier 3 Internet service providers. Tier 2 providers will exchange Internet traffic through peering agreements, as well as purchase Internet transit.

Tier 2 and tier 3 internet providers are sometimes used interchangeably. A tier 2 supplier tends to find it easier to purchase transit than to work out a peering agreement with a tier 1 provider. The reason for this is the level of transit. The tier 2 provider may not have enough transit, or capabilities, for it to make sense to peer with the tier 1 provider.

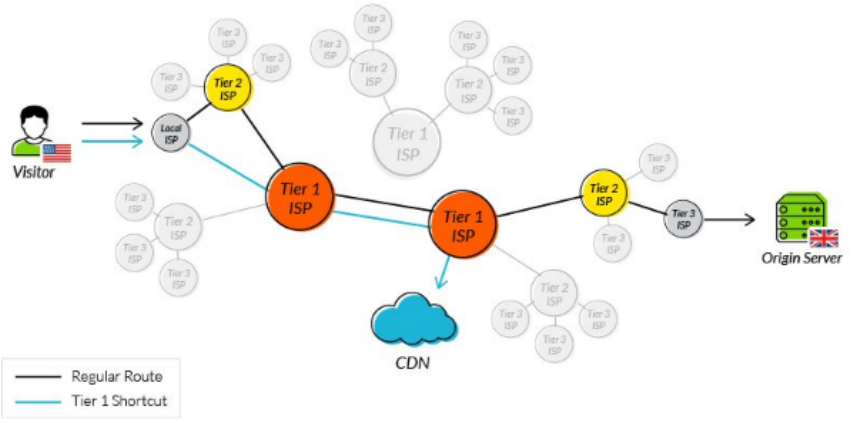
———— {Connection: peering Agreement }

List of Tier 2 providers

Name	AS Number	September 2016 degree[1][2]	Reason
Hurricane Electric	6939	4809	IPv4: Buys transit from Telia Carrier/AS1299.[3] IPv6: Does not provide IPv6 routing/connectivity to Cogent/AS174.[4]
Cogent Communications (formerly PSINet)	174	4641	IPv6: Does not provide IPv6 routing/connectivity to Google/AS15169 or Hurricane Electric/AS6939.[4][5]
Interoute	8928	552	Uses transit from GTT/AS3257.[6]
Korea Telecom	4766	492	Buys transit from Cogent/AS174, Telia Carrier/AS1299, Tata Communications/AS6453, Hurricane Electric/AS6939, Global Telecom & Technology (GTT)/AS4436, Sprint/AS1239.
China Telecom	4134/4809	151	Buys transit from Verizon/AS701.
KDDI	2516	310	Buys transit from Level3/AS3356, Tata Communications/AS6453, Verizon/AS701, Global Telecom & Technology (GTT)/AS3257, CenturyLink/AS209.
Internet Initiative Japan	2497	307	Buys transit from NTT America/AS2914, Verizon/AS701.
SK broadband	9318	484	Buys transit from Tata Communications/AS6453, Verizon/AS701, Telia Carrier/AS1299, NTT America/AS2914, China Telecom/AS4134&AS4809.
Vodafone (formerly Cable and Wireless)	1273	296	Buys transit from Level 3 Communications/AS3356, Telia Carrier/AS1299, Verizon Enterprise Solutions (formerly UUNET)/AS701, Global Telecom & Technology (GTT)/AS3257.
Vodafone (formerly ARCOR AG Germany)	3209	166	Buys transit from Level3/AS3356, Telia Carrier/AS1299, Verizon Enterprise Solutions (formerly UUNET)/AS701, nLayer Communications/AS4436, Pacnet/AS10026.[7]
Telkom Indonesia International	7713	258	Buys transit from Level3/AS3356, Cogent/AS174, Telia Carrier/AS1299, NTT America/AS2914, Telecom Italia Sparkle/AS6762, Tata Communications/AS6453.
TDC	3292	221	Buys transit from Sprint/AS1239, NTT America/AS2914.
Virgin Media	5089	192	Buys transit from GTT/AS3257, Liberty Global/AS6830.
Comcast	7922	160	Buys transit from Tata Communications/AS6453, NTT America/AS2914.
PT Mora Telematika Indonesia	23947	158	Buys transit from NTT America/AS2914, GTT/AS3257, Tata Communications/AS6453.
British Telecom	5400	155	Buys transit from Telia Carrier/AS1299 and NTT America/AS2914, Global Telecom & Technology (GTT)/AS4436.
Easynet	4589	108	Buys transit including NTT America/AS2914, GTT/AS3257.
Tele2 (formerly SWIPNet)	1257	107	Buys transit from Sprint/AS1239, Cogent/AS174.
Fibrenoire	22652	79	Buys transit from Level3/AS3356, Tata Communications/AS6453, GTT/AS3257.
Spirit Communications	2711	70	Buys transit from Level3/AS3356, Cogent/AS174, Telia Carrier/AS1299, NTT America/AS2914, GTT/AS3257, Hurricane Electric/AS6939.
FiberRing	38930	59	Buys transit from NTT America/AS2914, Telia Carrier/AS1299, Tata Communications/AS6453, Cogent/AS174.
Internap	14744	43	Buys transit from Cogent/AS174, NTT America/AS2914, XO Communications/AS2828, AboveNet/AS6461, CenturyLink/AS209, AT&T/AS7018.
KCOM Group	12390	37	Buys transit from KPN/AS286, Level 3/AS3356, Cogent/AS174, NTT America/AS2914.
Stealth Communications	8002	28	Buys transit from Tata Communications/AS6453, Cogent/AS174.
RETN	9002	620	Buys transit from Level 3/AS3356, NTT America/AS2914.

References: https://en.wikipedia.org/wiki/Tier_2_network

A tier 3 ISP is a provider who strictly purchases Internet transit. A tier 3 provider is the last mile provider who delivers Internet access to residential homes and businesses.

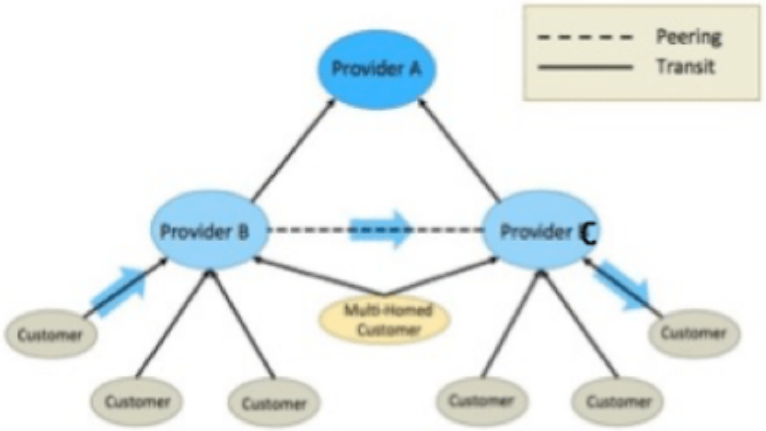


References: https://www.incapsula.com/cdn-guide/wp-content/uploads/sites/7/2018/04/ping2.jpg

Downloading documents from a server in California to a user in Germany means all three levels of ISPs have to work together. This is done through a combination of peering and paid Internet transit agreements.

BGP Peering

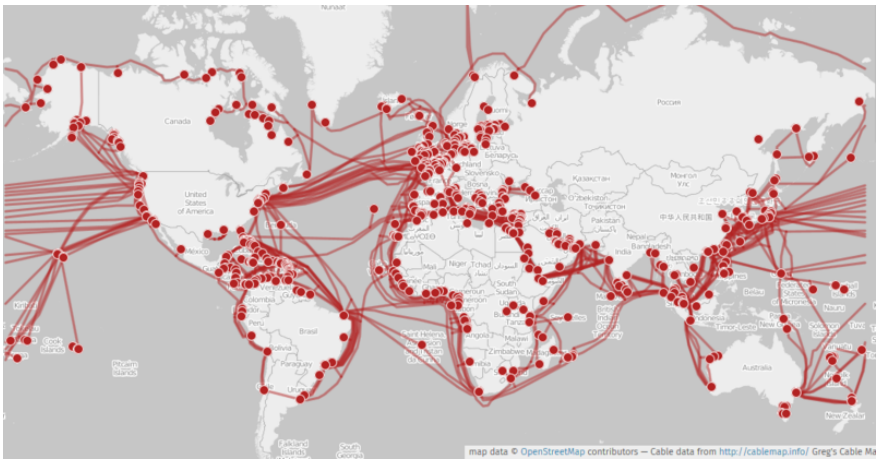
BGP Peering is an agreement between different Service Providers. It is an EBGP neighborship between different Service Providers to send BGP traffic between them without paying upstream Service Provider. To understand BGP peering, first, we must understand how networks are connected to each other on the Internet. The Internet is a collection of many individual networks, which interconnect with each other under the common goal of ensuring global reachability between any two points.

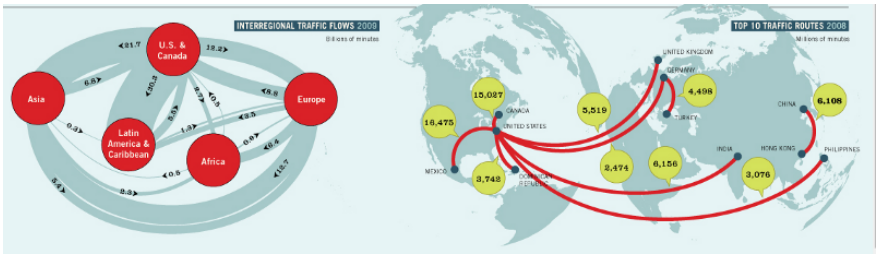


BGP Peering and Transit Links

As in the above picture, there are three primary relationships in this interconnection:

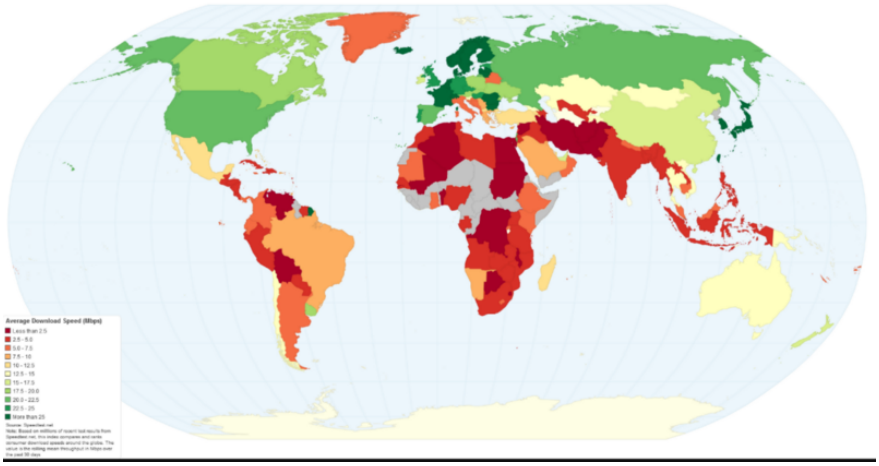
- Provider: Typically someone who is paid and has the responsibility of routing packets to/from the entire Internet.
- Customer: Typically someone who pays a provider with the expectation that their packets will be routed to/from the entire Internet.
- Peers: Two networks that get together and agree to exchange traffic between each others' networks, typically for free. There are generally two types of peering: public and private. Both will be explained in this session.





references: https://www.telegeography.com/assets/website/images/maps/global-traffic-map-2010/global-traffic-map-2010-x.jpg

References: https://www.telegeography.com/assets/website/images/maps/global-traffic-map-2010/global-traffic-map-2010-x.jpg

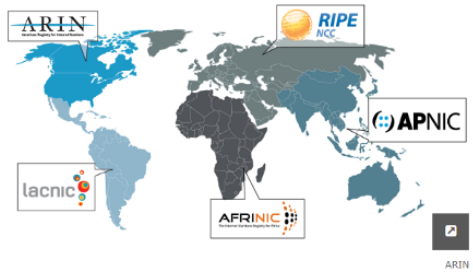


References: https://cdn3.vox-cdn.com/assets/4463835/WrLYaXB.png

13

Who controls IP addresses

For the internet to work, everyone needs a unique Internet Protocol (IP) address. To coordinate the distribution of these addresses, the internet is broken up into five zones. Each zone has been assigned hundreds of millions of IP addresses to manage. Unfortunately, the original internet architecture, called IPv4, only allows for about 4 billion addresses, and the network has nearly exhausted the supply. The problem is particularly growing in fast-growing regions like Asia. Engineers have developed a long-run solution to this problem: switching to a new internet standard called IPv6. IPv6 offers such a large number of potential addresses that the world will never run out. But adoption of IPv6 has been slow. Today, the overwhelming majority of internet traffic uses the old standard. But with few IPv4 addresses left, people joining the internet in the future will have little choice but to use IPv6.



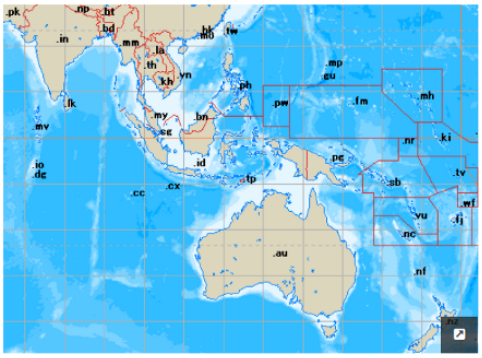
ARIN

references: https://www.vox.com/a/internet-maps

15

Some small island nations lend their domains to internet startups

Even very small countries get ccTLDs. Here's a close-up of the area around Australia and the many small island nations that have their own domain names. Some of these countries realized that they could make a lot of money if they opened their domains to foreigners. The result: popular websites like last.fm (.fm is the domain of the Federated States of Micronesia) and twitch.tv (.tv is the domain for the island nation of Tuvalu). The .io domain, assigned to the British Indian Ocean Territory, has become popular among programmers. They associate the domain with the technical term input/output and use it to create "artisinal websites." Click to see a full world map.

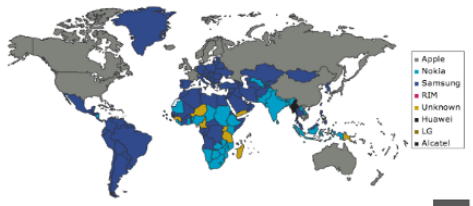


reference: https://www.vox.com/a/internet-maps

26

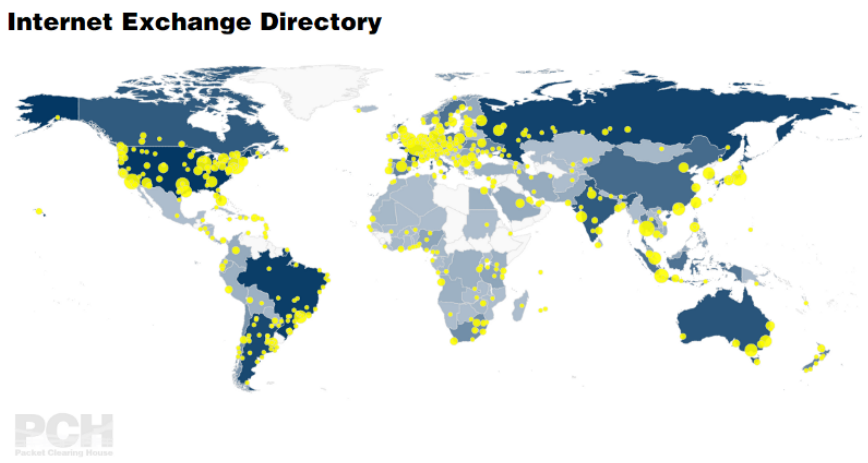
Most popular mobile phone brands

Right now, three companies dominate the global market for mobile phones, and they've largely divided the market by income. In wealthy countries, Apple's iPhone is popular. In middle-income countries, especially in Latin America, Eastern Europe, and the Middle East, Samsung devices have the lead. In poor countries, especially in sub-Saharan Africa, Nokia often dominates. These data are based on web browsing patterns, so it may not be a perfect reflection of the number of units sold by these companies.



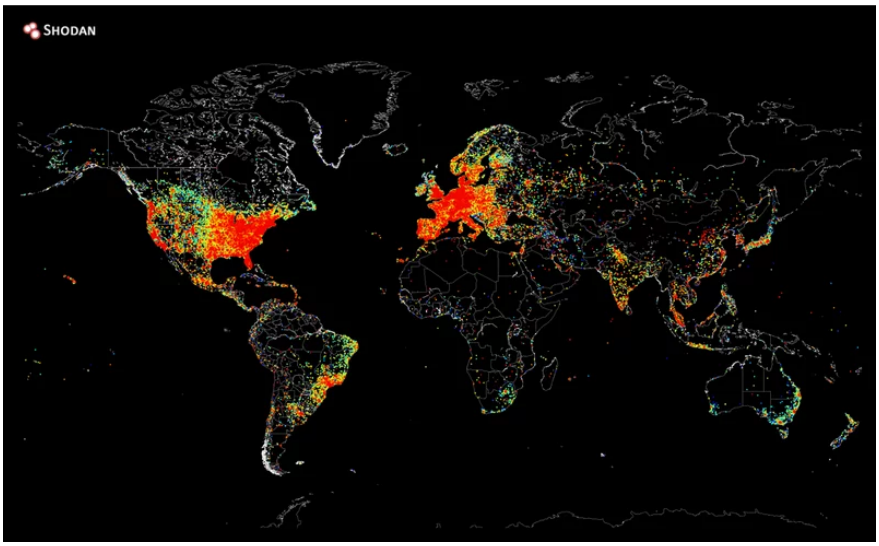
Statcounter

Reference: https://www.vox.com/a/internet-maps



References: https://www.pch.net/ixp/dir

Devices connected to the internet

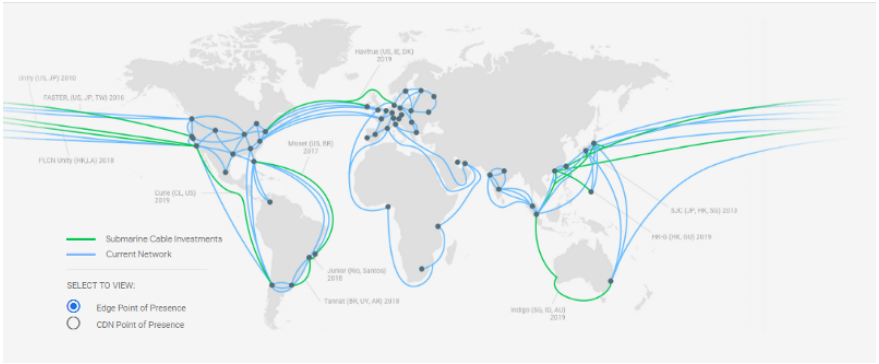


references: http://time.com/3221958/internet-map/

So, these maps will give you a holistic idea about the network traffic in the world and users in the world.

Let's get back to google.

Google is probably tier 1 or tier 2 network provider but as they only transfer traffic within Google's network they are not regarded as ISP. Google has a private global network connecting many data centers and POP(point of presence). They have



reference: https://cloud.google.com/about/locations/#network-tab

Google's private network connects their regional locations to more than 100 points of presence(POP). Google Cloud Platform uses software-defined networking and distributed systems technologies to host and deliver services around the world. Since Google has a global private network, this will help make your product global linking all the regions by the high-speed network.

1. Managing Networking for your resources
2. Worldwide Autoscaling and load balancing
3. Highly available global DNS Network
4. Fast, High availability Interconnect
5. Content Delivery Network(CDN)

Regions and Zones

When developing your application in GCP it is very important to understand regions and zones,

Resources are also regional and zonal so you must also have an idea about which resource is what before going in detail.

A region is a geographical location that is sub-divided into zones.

While few of the resources in GCP are global, others may be restricted by region or zone.

Regional resources can be used anywhere within the same region, while zonal resources can be used anywhere within the same zone. Some examples of this are:

Global Resources:

- Images
- Snapshots
- VPC Network
- Firewalls
- Routes

Regional Resources:

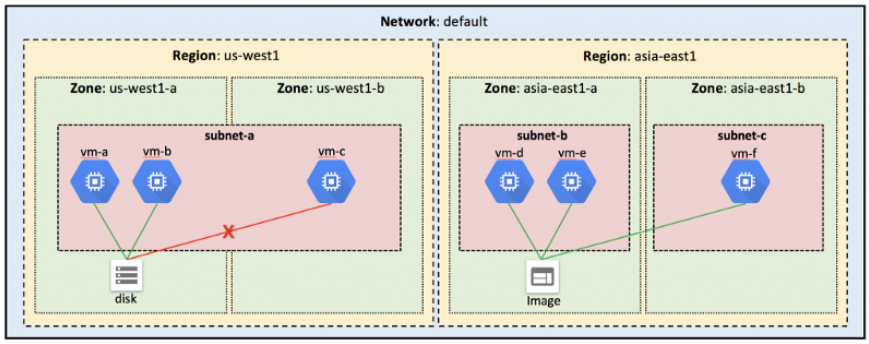
- Static external IP addresses
- Subnets

Zonal Resources:

- Instances (VMs)
- Persistent Disks

For example, I can attach a disk from one instance to another within the same zone, but I cannot do this across zones. However, since images

and snapshots are Global Resources, I can use these across zones in the same region.



reference: https://www.networkmanagementsoftware.com/google-cloud-platform-gcp-networking-fundamentals/

Managing Networking for resources

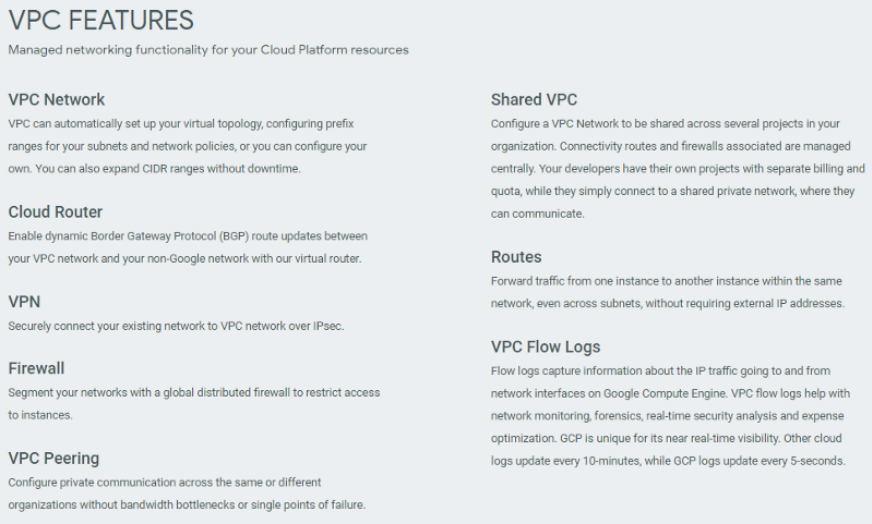
With the help of Google Virtual private cloud (VPC), you can

- provision GCP resources
- connect resources
- isolate resources

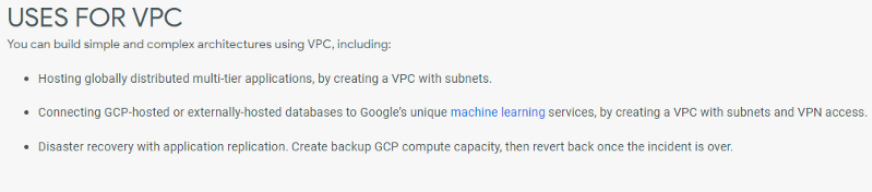
- make fine-grained policies for accessing resources and network

VPC consists of

- IP Address
- firewall
- VPN
- cloud router



References: https://cloud.google.com/vpc/



References: https://cloud.google.com/vpc/

Worldwide Autoscaling and load-balancing

Autoscaling is one of the most important aspects of Cloud in general and Google has made it even easier to scale with minimum configuration. You can scale from zero to full-throttle with Google cloud load balancing with no pre-warming required.

You can select whether to make the instance regional or multi-regional while creating it and then configure the load balancing option to make it more specific.

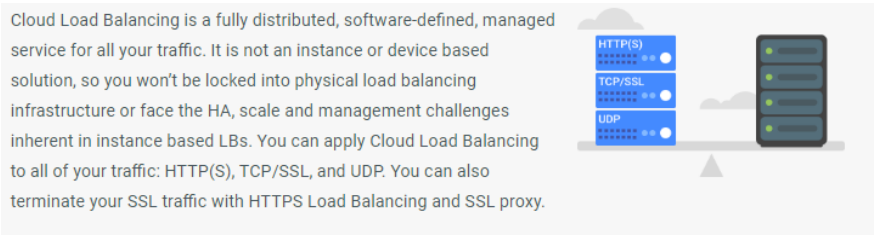
Cloud Load balancing can put resources behind a single unicast IP and scale your resources up or down with intelligent Autoscaling.

Cloud load balancing is integrated with Google cloud CDN for optimal application and content delivery.

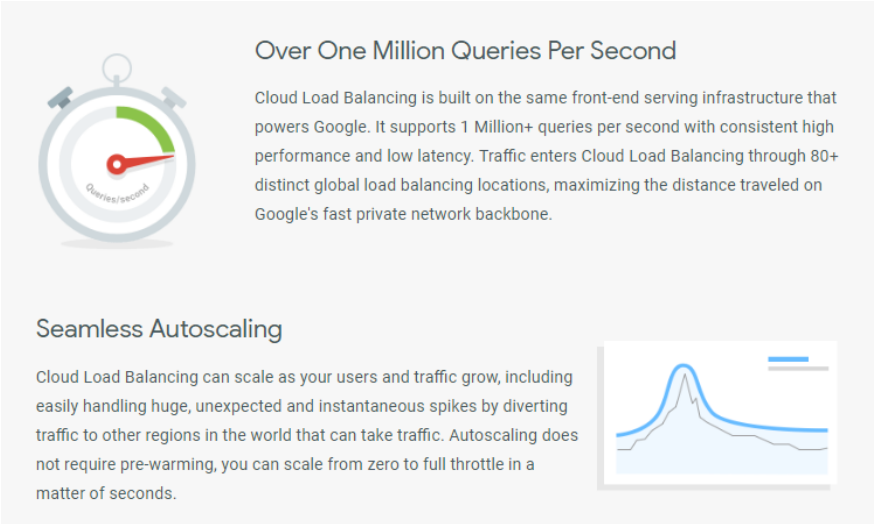
Global load balancing with single Anycast IP

With cloud load balancing a single anycast IP can be used in front-end for load balancing all backend resources around the world. It provides cross-region load balancing including automatic multi-region failover which gently moves traffic infractions if backend becomes unhealthy.

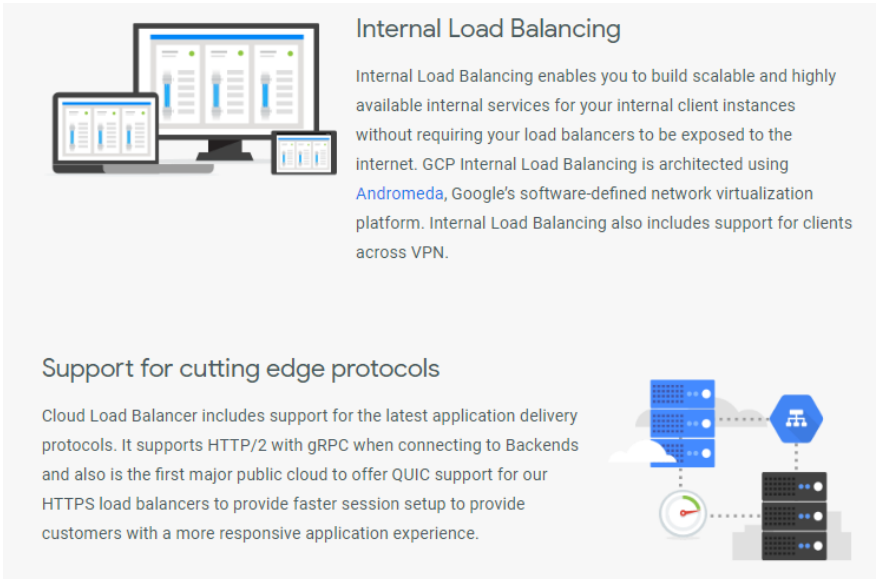
Software-defined load balancing



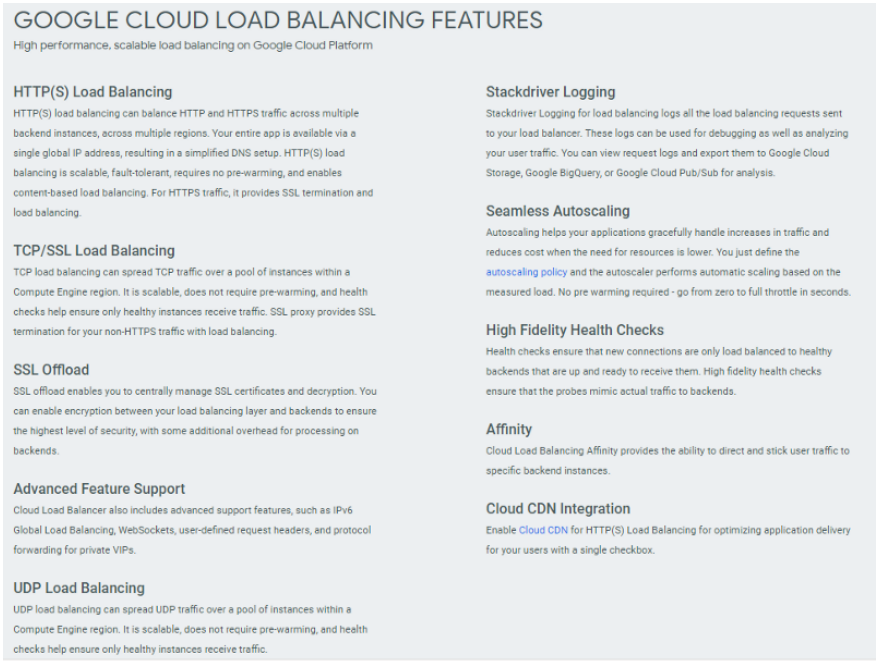
references: https://cloud.google.com/load-balancing/



references: https://cloud.google.com/load-balancing/



references: https://cloud.google.com/load-balancing/



references: https://cloud.google.com/load-balancing/

Differentiating load balancing options

Google Cloud Platform Load Balancing enables you to do the following:

- Distribute load-balanced resources in single or multiple regions
- Meet your high availability requirements

- Put your resources behind a single anycast IP address
- Scale your resources up or down with intelligent Autoscaling
- Use Cloud CDN for optimal content delivery

By using cloud load balancing with Cloud CDN you can serve as close to your users as possible that can handle 1 million queries per second.

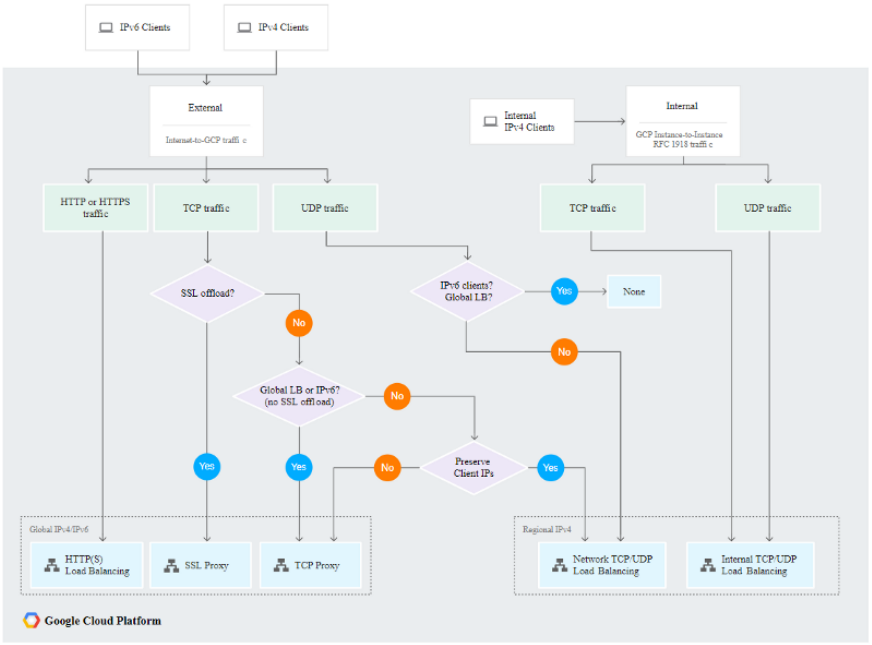
Since there are more than one load balancing options available it is very important for you to decide which one best fits your need.

It is not an instance or device based, so you do not need to manage a physical load balancing infrastructure.

The selection should be based on whether you want load balancing for

- Global or regional load balancing
- External or internal load balancing
- based on traffic type like HTTP, HTTPS, FTP

After you determine whether you need global or regional load balancing, external or internal load balancing, and what traffic type your load balancers must handle, use the following flowchart to determine which load balancers are available for your client, protocol, and network configuration.



reference: https://cloud.google.com/load-balancing/images/choose-lb.svg

Below are the types of load balances and options

Load balancer	Traffic type	Global/Regional	External/Internal	External Ports for Load Balancing
HTTP(S)	HTTP or HTTPS	Global	External	HTTP on 80 or 8080; HTTPS on 443
SSL Proxy	TCP with SSL offload	Global	External	25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, and 5222
TCP Proxy	TCP without SSL offload. Does not preserve client IP addresses	Global	External	25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
Network TCP/UDP	TCP/UDP without SSL offload. Preserves client IP addresses.	Regional	External	Any
Internal TCP/UDP	TCP or UDP	Regional	Internal	Any

reference: https://cloud.google.com/load-balancing/docs/choosing-load-balancer

Highly Available Global DNS Network

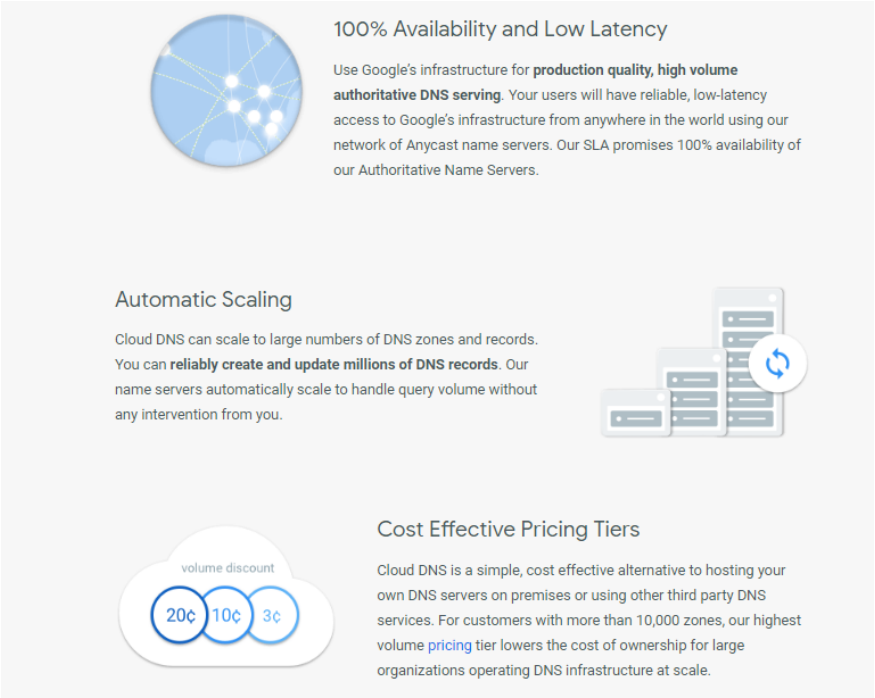
Cloud DNS

Google Cloud DNS is scalable, reliable and managed authoritative Domain Naming System service running on the same infrastructure as Google. It has low latency, high availability and is a cost-effective way to make your application and services available to the users.

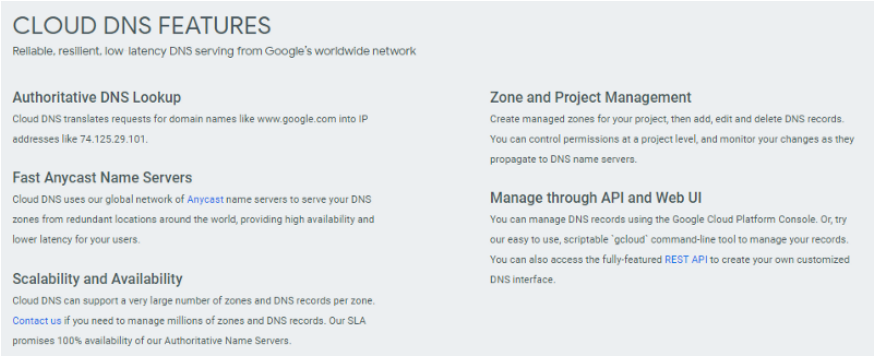
Cloud DNS translates requests for domain names like www.google.com into IP addresses like 74.125.29.101.

Cloud DNS

Cloud DNS is programmable and you can manage millions of zones and records using simple user interface, command line interface or API.



reference: https://cloud.google.com/dns/



reference: https://cloud.google.com/dns/

Configuring Cloud DNS

Before you begin

The Google Cloud DNS API requires that you create a Google Cloud DNS project and enable the Cloud DNS API.

If you are creating an application that will use the REST API, you will also need to create an OAuth 2.0 client ID.

1. If you don't already have one, sign up for a Google account.
2. Enable the Google Cloud DNS API in the GCP Console. You can choose an existing Compute Engine or App Engine project, or you can create a new project.
3. If you need to make requests to the REST API, you will need to create an OAuth 2.0 ID: Setting up OAuth 2.0.
4. Note the following information in the project that you will need to input in later steps:

- The client ID (`xxxxxx.apps.googleusercontent.com`).
- The project ID that you wish to use. You can find the ID at the top of the **Overview** page in the GCP Console. You could also ask your user to provide the project name that they want to use in your app.

If you have not run the `gcloud` command-line tool previously, you will need to run the following command to specify the project name and authenticate with the GCP Console:

```
gcloud auth login
```

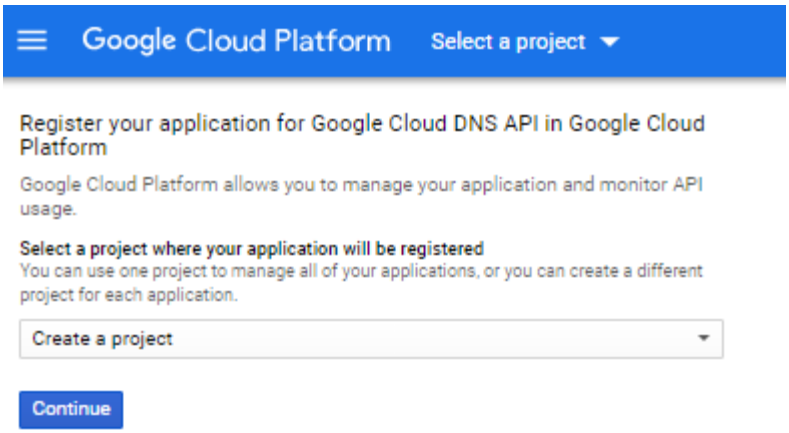
You can also specify the `--project` parameter for a command to operate against a different project for that invocation.

Working in a Cloud DNS may seem a bit different than traditional DNS because you have to understand behind the concept before doing anything in Cloud DNS.

Before we begin

Google cloud DNS API requires that you create Google Cloud DNS project and enable DNS API.

To do this, go to dashboard, select Network Services and then **Cloud DNS**



Add or link Application to use in Cloud DNS

Creating managed zones

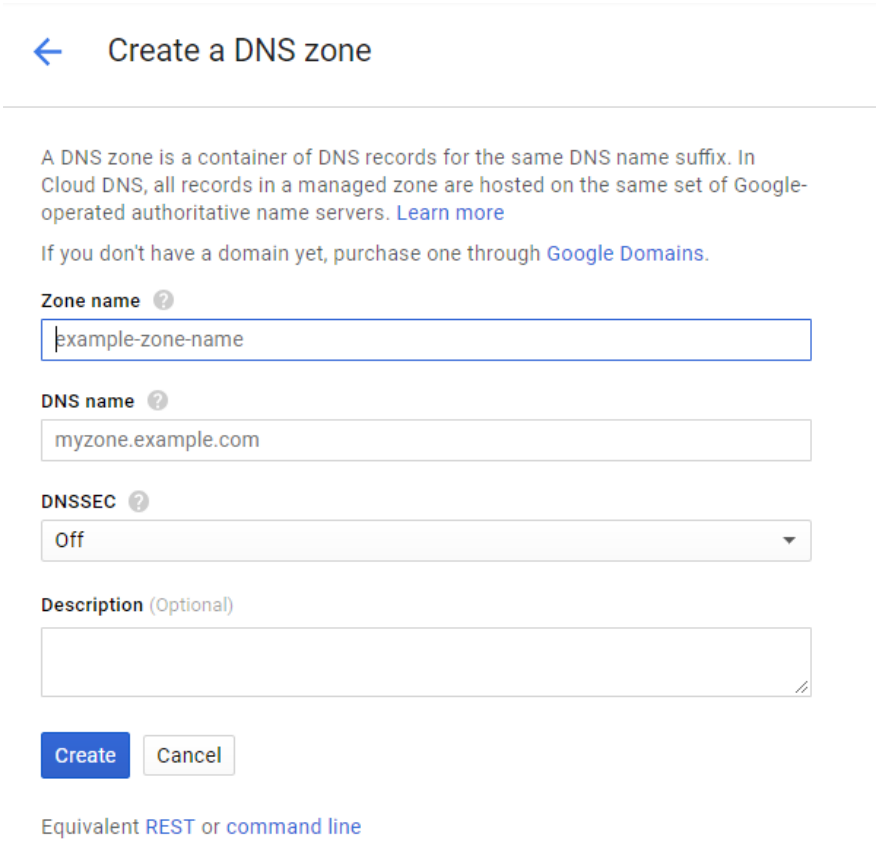
When you get started with Cloud DNS API, you will need to create a managed zone to contain DNS record. A managed zone is always connected to your cloud project.

Note that when you create a zone, the new zone won't be used until you update your domain registration, or explicitly point some resolver at, or directly query, one of your zone's name servers.

```
gcloud dns managed-zones create \
        --dns-name="example.com." \
        --description="A zone" "myzonename"
```

where `[KEY]:[VALUE]` is an arbitrary key: value pair, such as `Dept:Marketing` or `Project:project1`. The `--labels` flag is not required for this command.

In the cloud console,



Updating managed zones

Once you have created a managed zone to contain your DNS records, you may want to update some of its properties. Currently, you can only update the description and DNSSEC configuration.

To update a zone, you must provide the zone resource name (which cannot contain `.` —as opposed to the DNS name, which does) and the updated information associated with the zone:

```
gcloud dns managed-zones update --description="My zone"
"myzonename"
```

You can also go to the console and tick the options to apply to the current project.

Listing managed zones

To list all of your zones within a project:

```
gcloud dns managed-zones list
```

Getting managed zone details

To get details about your managed zone, such as if you need to look up the associated name servers:

```
gcloud dns managed-zones describe "myzonename"
```

Deleting managed zones

To delete a zone, provide the zone name to the delete command:

```
gcloud dns managed-zones delete "myzonename"
```

Note that only empty zones can be deleted. An empty managed-zone has only SOA and NS record-sets. You can easily empty a zone using the import command as follows:

```
touch empty-file
gcloud dns record-sets import -z "myzonename" --delete-all-
existing empty-file
rm empty-file
```

Adding and updating labels for managed zones

You can add labels to a managed zone, and you can remove existing labels.

Add labels when you create a managed zone

```
gcloud dns managed-zones create \
    --dns-name="example.com." \
    --labels [KEY]:[VALUE] \
    --description="A zone" "myzonename"
```

Add labels to an existing managed zone

This command adds a label to an existing managed zone.

```
gcloud dns managed-zones update \
    --labels [KEY]:[VALUE],[[KEY]:[VALUE]] \
    "myzonename"
```

Update values of label key: value pairs

This command update the `value` of an existing key:value label pair. If the `key` does not already exist, a new key: value pair is created.

```
gcloud dns managed-zones update \
    --update-labels [KEY]:[VALUE],[[KEY]:[VALUE]] \
    "myzonename"
```

Remove label key: value pairs

This command removes the specified key: value label pair(s).

```
gcloud dns managed-zones update \
    --remove-labels [KEY]:[VALUE],[[KEY]:[VALUE]] \
    "myzonename"
```

Clear all label key-value pairs

This command clears all labels.

```
gcloud dns managed-zones update \
    --clear-labels \
    "myzonename"
```

Here I will try to configure DNS server for the domain I have registered. Some of the tasks in configuring Cloud DNS are

Managing Zones

A managed zone is the container for all of your DNS records that share the same domain name, for example, `example.com` . Managed zones are automatically assigned a set of name servers when they are created to handle responding to DNS queries for that zone. A managed zone has quotas for the number of resource records that it can include.

Managing Records

Managing DNS records for the Google Cloud DNS API involves sending change requests to the API. These changes consist of additions and deletions to your resource record sets collection. You can easily send the desired changes to the API using the `import` , `export` , and `transaction` commands, as described below.

Importing and exporting record-sets

You can use `import` and copy record-sets into and out of a managed zone. The formats you can import from and export to are either BIND zone file format, or YAML records format.

To import record-sets, you use the `dns record-sets import` command. The `--zone-file-format` flag tells `import` to expect a BIND zone formatted file. If you omit this flag, `import` expects a YAML formatted records file:

```
gcloud dns record-sets import -z=examplezonename \
   --zone-file-format path-to-example-zone-file
```

Note: Some DNS implementations and providers export BIND zone files without final periods on domain name data in CNAME, MX, PTR, and other records. In zone files Google Cloud DNS follows RFC standards and interprets all domain names without a final period as relative to the DNS name of the zone, so importing the following MX records into a zone with the DNS name `example.com` results in identical (and probably undesired) records for both:

```
in.smtp              IN MX 5 gmail-smtp-in.l.google.com
in.smtp.example.com. IN MX 5 gmail-smtp-
in.l.google.com.example.com.
```

Check your zone files to ensure all names that need them to have final periods before importing them.

To export record-sets, you use the `dns record-sets export` command. Use the `--zone-file-format` flag to tell `export` to export the record-sets into a BIND zone formatted file. If you omit this flag, `export` exports the record-sets into a YAML formatted records file:

```
gcloud dns record-sets export -z=examplezonename \
   --zone-file-format example.zone
```

Migrating to Cloud DNS

Cloud DNS supports the migration of an existing DNS domain from another DNS provider to Cloud DNS. This procedure describes how to complete the necessary steps: creating a managed zone for your domain, importing your existing DNS configuration, and updating your registrar's name service records.

Create a managed zone

To migrate an existing domain, first create a managed zone to contain your DNS records. Note that when you create a zone, the new zone won't be used until you update your domain registration, explicitly

point some resolver at it, or directly query one of your zone's name servers.

To create a zone, provide the DNS zone name, a description, and a name to identify the zone:

```
gcloud dns managed-zones create --dns-name="example.com." --
description="A zone" "examplezonename"
```

Export your DNS configuration from your existing provider

Consult your provider's documentation to see how to export your zone file. Cloud DNS supports the import of zone files in BIND or YAML records format.

For example:

- For Dyn, see Download Your Zone File.
- AWS Route 53 does not support export. Instead, you can use the open source cli53 tool.

Import the record set

Once you have the exported file from your other provider, you can use the `gcloud dns record-sets import` command to import it into your managed zone.

To import record-sets, you use the `dns record-sets import` command. The `--zone-file-format` flag tells `import` to expect a BIND zone formatted file. If you omit this flag, `import` expects a YAML-formatted records file:

```
gcloud dns record-sets import -z=examplezonename --zone-
file-format path-to-example-zone-file
```

Caution: When importing record sets, if your zone file contains SOA records, you must use the `--delete-all-existing` flag to replace the SOA records provided by Cloud DNS. Otherwise, the update will fail, since the imported records will conflict with the pre-existing Cloud DNS records. **Note:** Some DNS implementations and providers export BIND zone files without final periods on domain name data in CNAME, MX, PTR, and other records. In zone files Cloud DNS follows RFC standards and interprets all domain names without a final period as relative to the DNS name of the zone, so importing the following MX records into a zone with the DNS name `example.com` results in identical (and probably undesired) records for both:

```
in.smtp              IN MX 5 gmail-smtp-in.l.google.com
    in.smtp.example.com. IN MX 5 gmail-smtp-
in.l.google.com.example.com.
```

Check your zone files to ensure all names that need them to have final periods before importing them.

Verify DNS propagation

You can use the Linux `watch` and `dig` commands to monitor and verify that your changes have been picked up by the Cloud DNS name servers. Note that `watch` and `dig` are not `gcloud` commands and are not used with the `gcloud` prefix. On other operating systems, you might need to install the `watch` and `dig` commands.

1. Look up your zone's Cloud DNS name servers:

- `gcloud dns managed-zones describe examplezonename`

You will see output that looks something like this:

- nameServers:—ns-cloud-a1.googledomains.com.—ns-cloud-a2.googledomains.com.—ns-cloud-a3.googledomains.com.—ns-cloud-a4.googledomains.com.

Check if the records are available on the name servers. Replace `your_zone_nameserver` with one of the name servers returned when you ran the previous command.

- `watch dig example.com @your_zone_nameserver`

Once you see your change, press `Ctrl-C` to exit.

The `watch` command runs the `dig` command every 2 seconds by default. You can use this command to determine when your authoritative name server picks up your change, which should happen within 120 seconds.

Update your registrar's name server records

Log into your registrar provider and change the name server records to point to the name servers you saw in the prior step. At the same time, make a note of the time to live (TTL) your registrar has set on the records. That will tell you how long you have to wait before the new name servers will begin to be used.

Wait for changes, then verify

To see the authoritative name servers for your domain on the Internet, run the following:

```
dig +short NS example.com
```

If the output shows that all changes have propagated, you're done. If not, you can check intermittently or you can automatically run the command every 2 seconds while you wait for the nameservers to change. To do that, run the following:

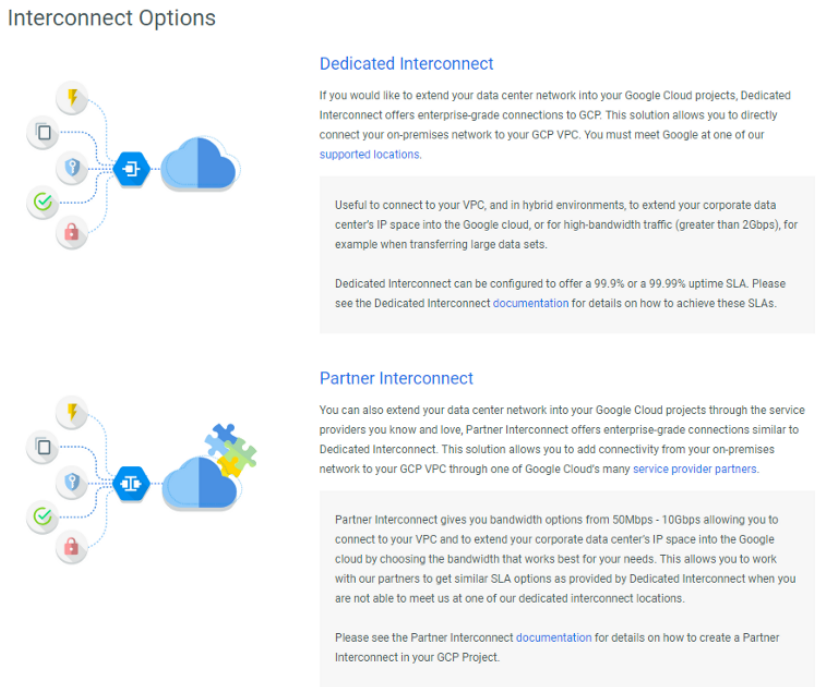
```
watch dig +short NS example.com
```

`Ctrl-C` exits the command.

Fast, High Availability Interconnect

Google Cloud Interconnect allows Cloud platform customers to connect to Google via enterprise-grade connections with higher availability and/or lower latency than their existing Internet connections. Connections are offered by Carrier Interconnect service provider partners and may offer higher SLAs than standard Internet connections. Google also supports direct connections to its network through direct peering. Customers who cannot meet Google at its peering locations, or do not meet peering requirements, may benefit from Carrier Interconnect.

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



IPsec VPN

Google Cloud VPN securely connects your on-premises network to your GCP Virtual Private Cloud (VPC) network through an IPsec VPN connection. Traffic traveling between the two networks is encrypted by one VPN gateway, then decrypted by the other VPN gateway. This protects your data as it travels over the Internet.

Useful to connect to your VPC over the public internet for low volume data connections.

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

Peering Connection Options

Direct Peering

If you require access to Google and Google Cloud properties and can satisfy Google's peering requirements, Direct Peering is the solution. Direct peering does not have an SLA.

Useful to connect directly to Google, and to save up to 67% on egress fees compared to VPN or public access over the Internet.

Carrier Peering

If you require access to Google public infrastructure and cannot satisfy Google's peering requirements, you can connect via a Carrier Peering partner. Carrier Peering does not have an SLA.

Useful if you like the benefits of Direct Peering, but are unable to meet the peering requirements without a partner.

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

Here is a side by side comparison of the Interconnect options.

CONNECTION	ACCESS TYPE	CAPACITY	COST	OTHER CONSIDERATIONS
Dedicated Interconnect				
Dedicated, direct connection to VPC networks	Internal IP addresses in RFC 1918 address space	10 Gbps for each link	Reduced egress costs, fee for each link and VLAN	Requires you to have a connection in a Google supported colocation facility, either directly or through a carrier
Partner Interconnect				
Dedicated Bandwidth, connection to VPC Network through a service provider	Internal IP Addresses in RFC 1918 address space	50Mbps - 10Gbps per connection	Reduced egress costs, fee for each VLAN attachment. For additional costs, contact your service provider	Service providers might have specific restrictions or requirements.
IPsec VPN tunnel				
Encrypted tunnel to VPC networks through the public Internet	Internal IP addresses in RFC 1918 address space	1.5-3 Gbps for each tunnel	Egress is billed the same as general network pricing + a fee for each tunnel	Requires a VPN device on your on-premises network

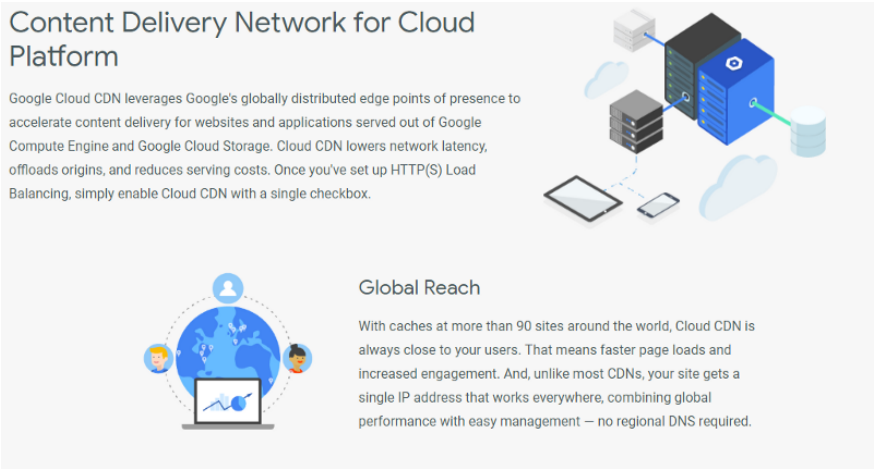
Here is a side by side comparison of the Peering options.

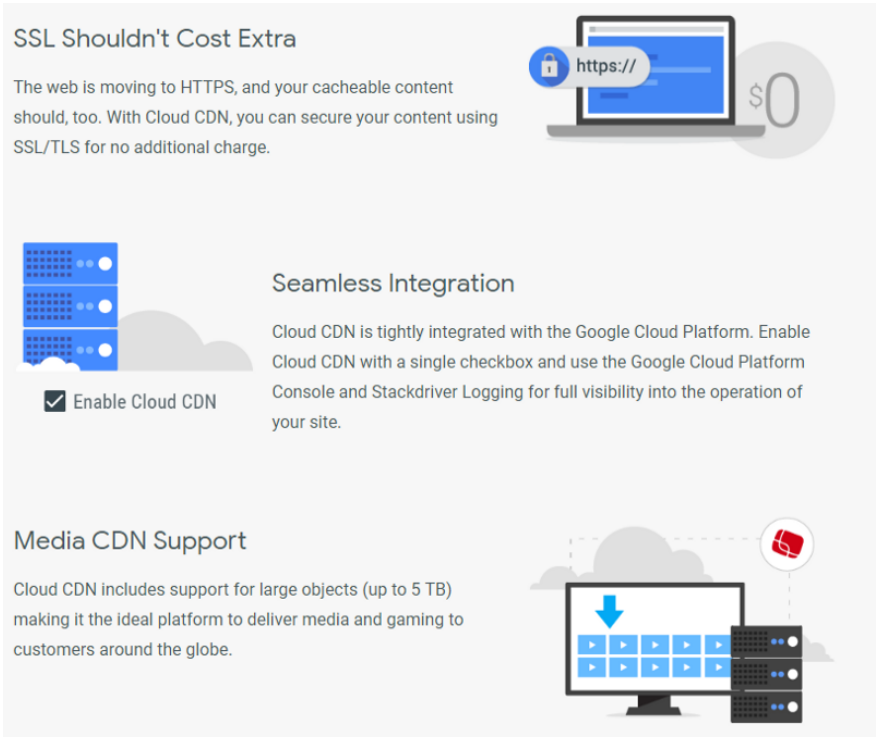
CONNECTION	ACCESS TYPE	CAPACITY	COST	OTHER CONSIDERATIONS
Direct Peering				
Dedicated, direct connection to Google's network	Public IP addresses	10 Gbps for each link	Settlement free peering, reduced cost for egress	Requires you to have a connection in a colocation facility, either directly or through a carrier provided wave service
Carrier Peering				
Peering through service provider to Google's public network (list of partners)	Public IP Addresses	Varies based on partner offering	Cost based on partner offering, reduced cost for egress	Requirements vary by partner

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

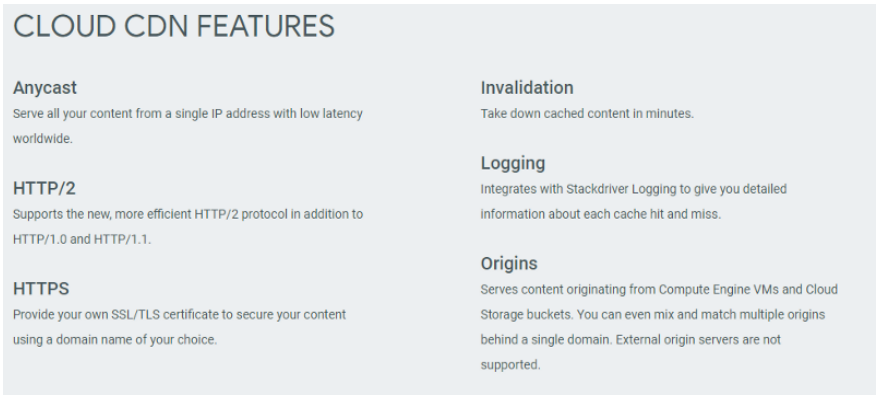
Content Delivery Network

Google Cloud CDN leverages Google's globally distributed edge caches to accelerate content delivery for websites and applications served out of Google Compute Engine. Cloud CDN lowers network latency, offloads origins, and reduces serving costs. Once you've set up HTTP(S) Load Balancing, simply enable Cloud CDN with a single checkbox.





reference: https://cloud.google.com/cdn/



https://cloud.google.com/cdn/