Google Cloud

Partner Certification Academy

Associate Cloud Engineer

The information in this presentation is classified:

Google confidential & proprietary

⚠ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

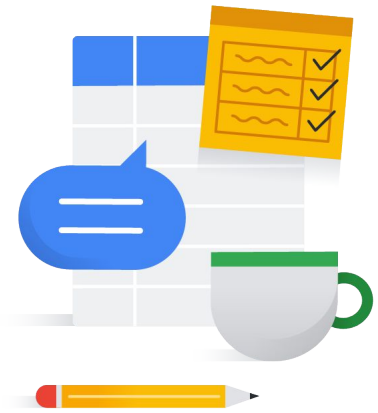
Google Cloud

Session logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.
- These slides are available in the Student Lecture section of your Qwiklabs classroom.
- The session is **not recorded.**
- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



Associate Cloud Engineer

The Google Cloud Certified

Associate Cloud Engineer exam assesses your ability to:

- Setup a cloud solution environment
- Plan and configure a cloud solution
- Deploy and implement a cloud solution
- Ensure successful operation of a cloud solution
- Configure access and security

For more information:

https://cloud.google.com/certification/cloud-engineer

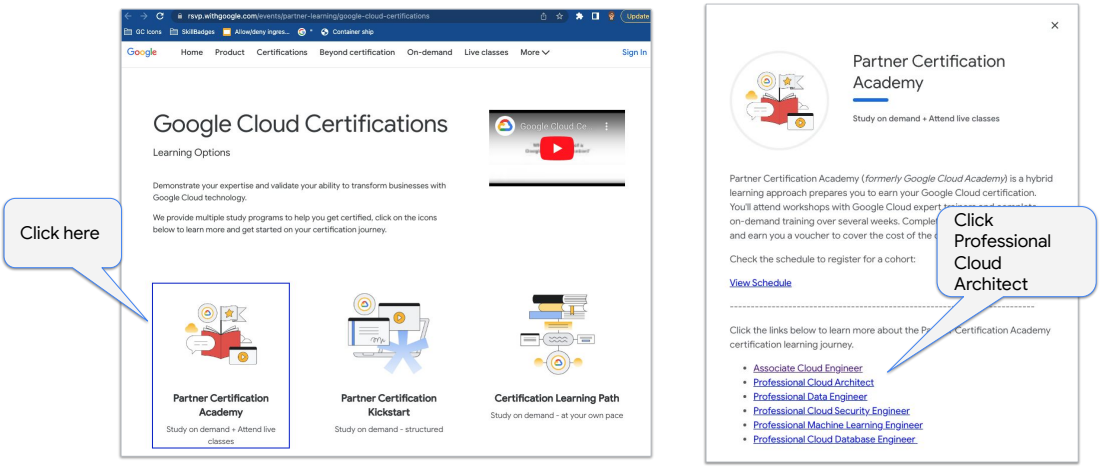
Associate Cloud Engineer
https://cloud.google.com/certification/cloud-engineer

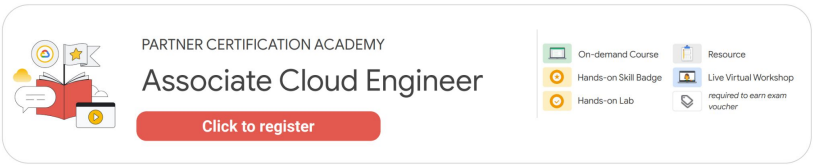
Exam Guide
https://cloud.google.com/certification/guides/cloud-engineer

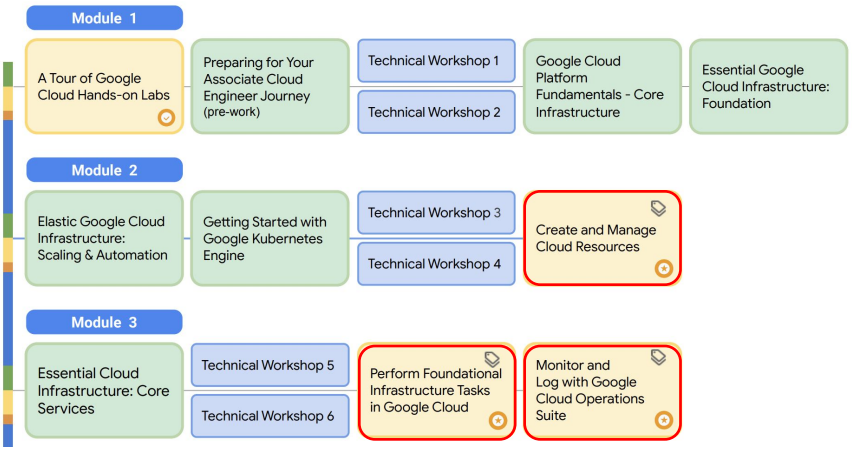
Sample Questions
https://docs.google.com/forms/d/e/1FAIpQLSfexWKtXT2OSFJ-obA4iT3GmzgiOCGvjrT9OfxilWC1yPtmfQ/viewform

Learning Path - Partner Certification Academy Website

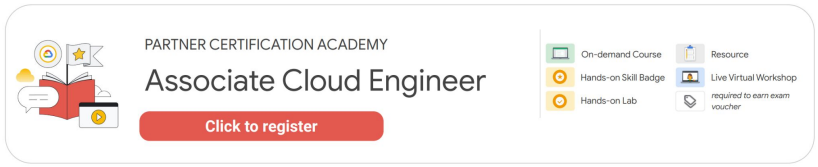
Go to: https://rsvp.withgoogle.com/events/partner-learning/google-cloud-certifications

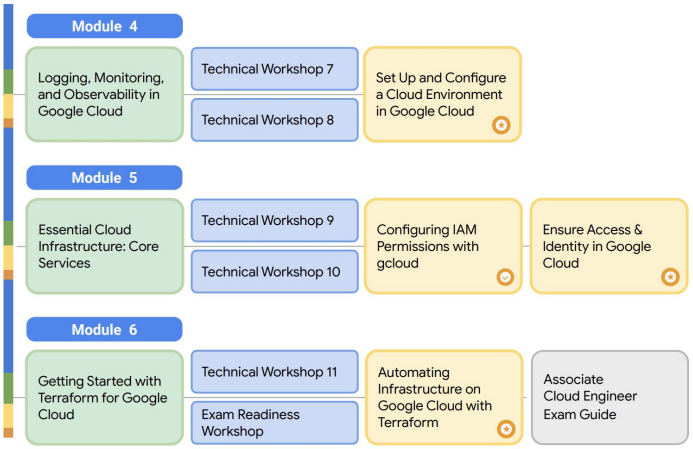






Needed for Exam Voucher





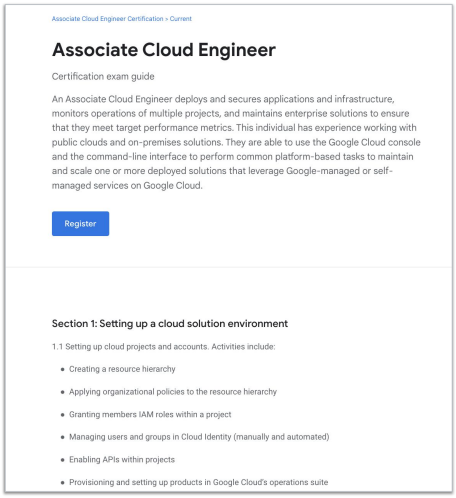
Associate Cloud Engineer (ACE) Exam Guide

Each module of this course covers Google Cloud services based on the topics in the ACE Exam Guide

The primary topics are:

- Compute Engine
- VPC Networks
- Google Kubernetes Engine
- Cloud Run, Cloud Functions and App Engine
- Cloud Storage and database options
- Resource Hierarchy/Identity and Access Management (IAM)
- Logging and Monitoring

Next discussion



Associate Cloud Engineer Certification > Current

Associate Cloud Engineer

Certification exam guide

An Associate Cloud Engineer deploys and secures applications and infrastructure, monitors operations of multiple projects, and maintains enterprise solutions to ensure that they meet target performance metrics. This individual has experience working with public clouds and on-premises solutions. They are able to use the Google Cloud console and the command-line interface to perform common platform-based tasks to maintain and scale one or more deployed solutions that leverage Google-managed or self-managed services on Google Cloud.

Register

Section 1: Setting up a cloud solution environment

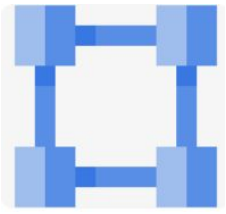
1.1 Setting up cloud projects and accounts. Activities include:

- Creating a resource hierarchy
- Applying organizational policies to the resource hierarchy
- Granting members IAM roles within a project
- Managing users and groups in Cloud Identity (manually and automated)
- Enabling APIs within projects
- Provisioning and setting up products in Google Cloud's operations suite

https://cloud.google.com/certification/guides/cloud-engineer/

VPC Network

Exam Guide Overview - VPC Network



VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

2.4 Planning and configuring network resources. Tasks include:

- 2.4.1 Differentiating load balancing options
- 2.4.2 Identifying resource locations in a network for availability
- 2.4.3 Configuring Cloud DNS

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

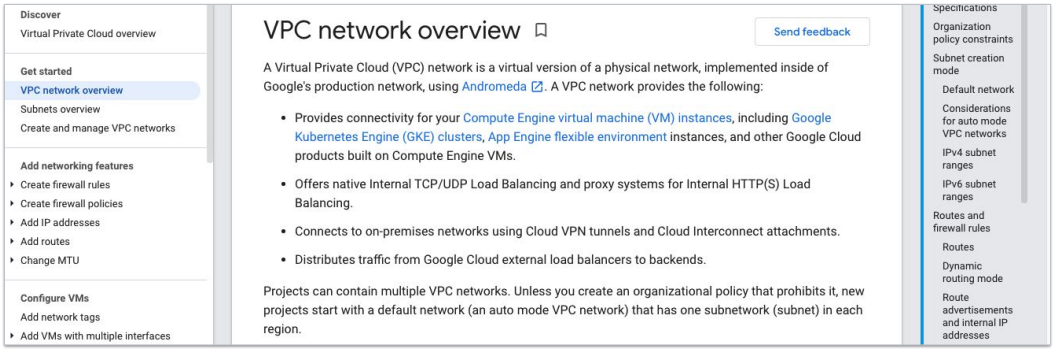
- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

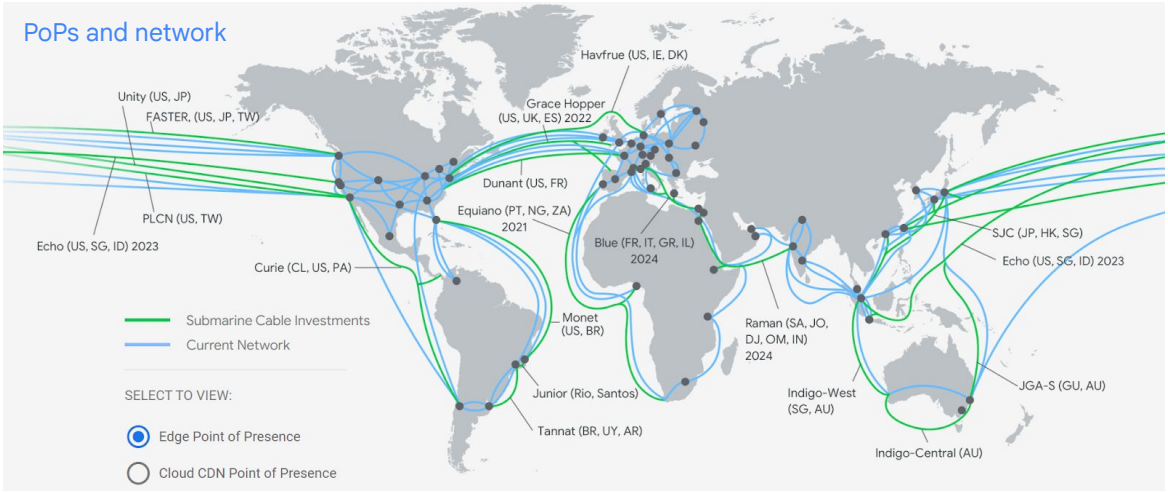
VPC network overview

- Website contains a detailed description of various VPC components



https://cloud.google.com/vpc/docs/overview

Google Cloud network



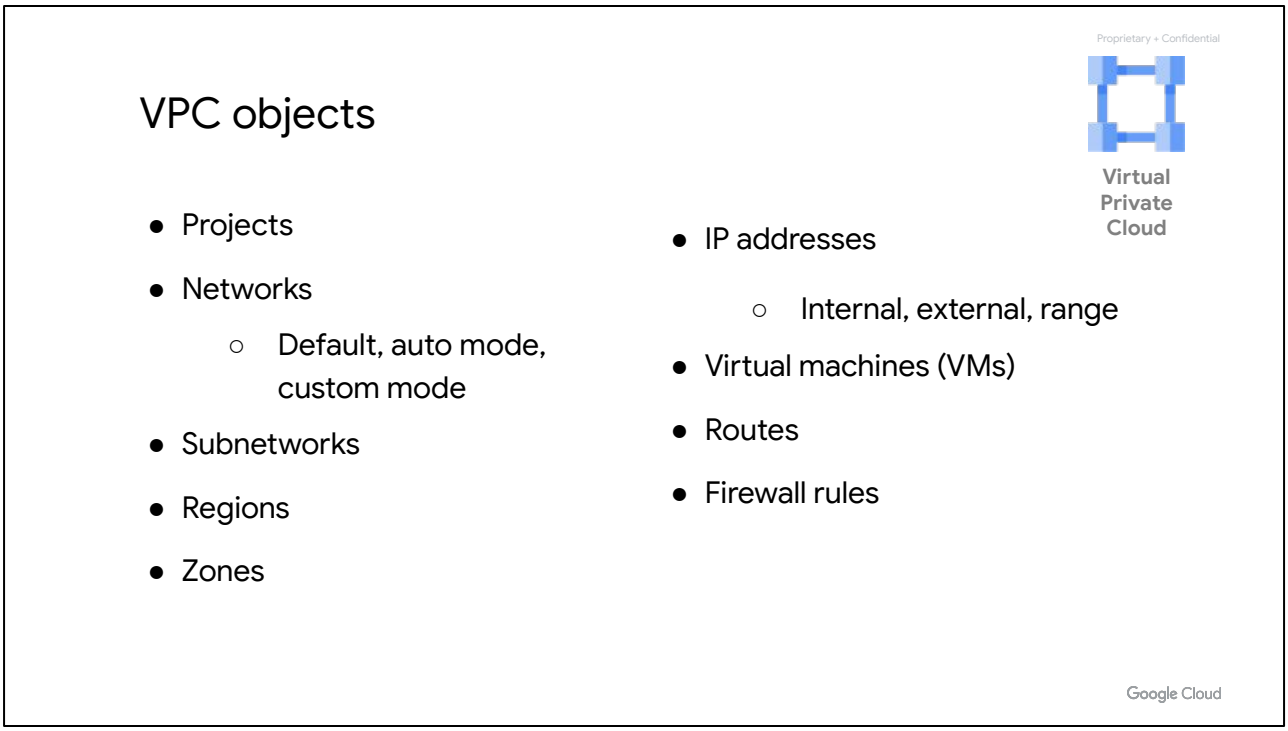
Google Cloud Regions and Zones
https://cloud.google.com/about/locations#network

What's in a name? Understanding the Google Cloud network "edge"
https://cloud.google.com/blog/products/networking/understanding-google-cloud-network-edge-points

According to some publicly available estimates, Google's network carries as much as 40% of the world's internet traffic every day. Google's network is the largest network of its kind on Earth. Google has invested billions of dollars over the years to build it.

It is designed to give customers the highest possible throughput and lowest possible latencies for their applications.

The network interconnects at more than 90 Internet exchanges and more than 170 points of presence worldwide. When an Internet user sends traffic to a Google resource, Google's edge caching nodes respond to users requests from an Edge Network location that will provide the lowest latency.



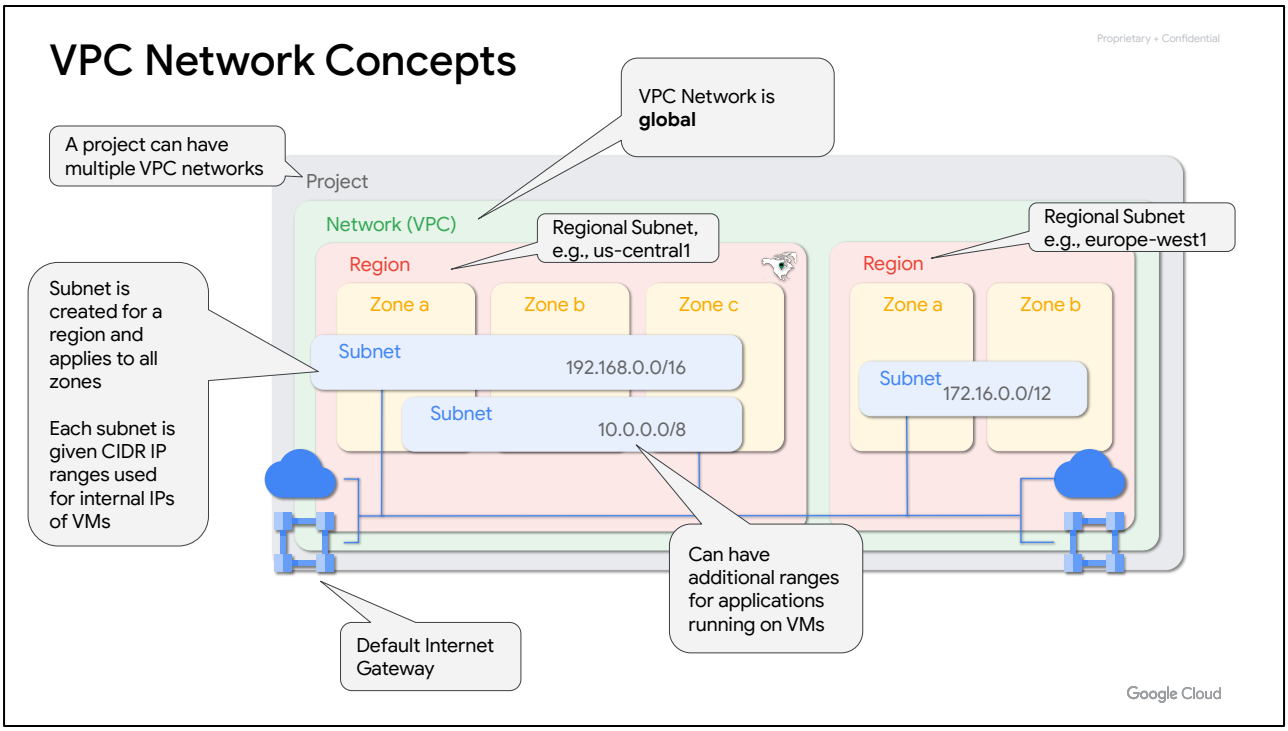
Create and modify Virtual Private Cloud (VPC) networks: https://cloud.google.com/vpc/docs/create-modify-vpc-networks

With Google Cloud, you can provision your Google Cloud resources, connect them to each other, and isolate them from each other in a Virtual Private Cloud. You can also define fine-grained networking policies within Google Cloud, and between Google Cloud and on-premises or other public clouds. Essentially, VPC is a comprehensive set of Google-managed networking objects, which we will explore in detail throughout this module.

Here is a high-level overview of these objects:

- Projects are going to encompass every single service that you use, including networks.
- Networks come in three different flavors: Default, auto mode, and custom mode.
- Subnetworks allow you to divide or segregate your environment.
- Regions and zones represent Google's data centers, and they provide continuous data protection and high availability.
- VPC provides IP addresses for internal and external use, along with granular IP address range selections.
- As for virtual machines, in this module we will focus on configuring VM

- instances from a networking perspective.
- We’ll also go over routes and firewall rules.



Nice set of Youtube videos:
https://bit.ly/34uBApk

Best practices and reference architectures for VPC design
https://cloud.google.com/architecture/best-practices-vpc-design

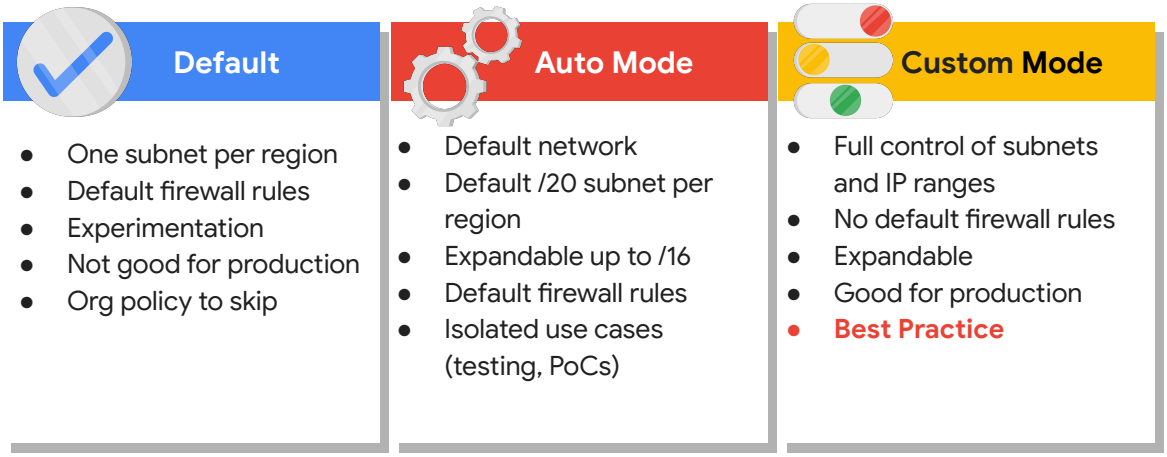
VPC networks

- Created within projects, which means there is no cross-project communication by default. More on that soon.
- Global resources, for example: VM in US can communicate with a VM in APAC
- Private RFC 1918 IP range
- Can be non RFC 1918 IP range

Subnets

- Are part of a VPC network
- Regional objects
- VMs which are zonal resources are allocated with an IP from a subnet in the same region
- Do not provide network boundaries. VMs can communicate across subnets.
- However, default firewall rules deny traffic between VMs regardless of subnets.

Subnet creation modes



Subnets overview:
https://cloud.google.com/vpc/docs/subnets

Subnet creation mode
https://cloud.google.com/vpc/docs/vpc#subnet-ranges

Create and modify Virtual Private Cloud (VPC) networks:
https://cloud.google.com/vpc/docs/using-vpc#add-subnets

Expand a primary IPv4 range:
https://cloud.google.com/vpc/docs/create-modify-vpc-networks#expand-subnet

Edit secondary IPv4 ranges:
https://cloud.google.com/vpc/docs/create-modify-vpc-networks#edit-secondary

Every project is provided with a default VPC network with preset subnets and firewall rules. Specifically, a subnet is allocated for each region with non-overlapping CIDR blocks and firewall rules that allow ingress traffic for ICMP, RDP, and SSH traffic from anywhere, as well as ingress traffic from within the default network for all protocols and ports.

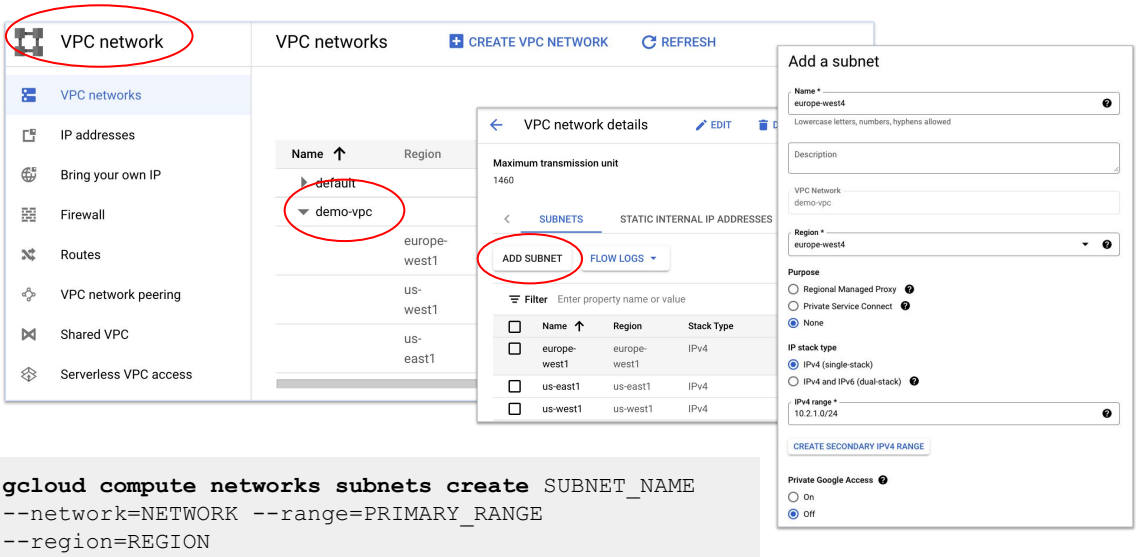
In an auto mode network, one subnet from each region is automatically created within it. The default network is actually an auto mode network. These automatically created

subnets use a set of predefined IP ranges with a /20 mask that can be expanded to /16. All of these subnets fit within the 10.128.0.0/9 CIDR block. Therefore, as new Google Cloud regions become available, new subnets in those regions are automatically added to auto mode networks using an IP range from that block.

A custom mode network does not automatically create subnets. This type of network provides you with complete control over its subnets and IP ranges. You decide which subnets to create, in regions you choose, and using IP ranges you specify within the RFC 1918 address space. These IP ranges cannot overlap between subnets of the same network.

Now, you can convert an auto mode network to a custom mode network to take advantage of the control that custom mode networks provide. However, this conversion is one way, meaning that custom mode networks cannot be changed to auto mode networks. So, carefully review the considerations for auto mode networks to help you decide which type of network meets your needs.

Adding Subnet to VPC - Console




```
gcloud compute networks subnets create SUBNET_NAME
--network=NETWORK --range=PRIMARY_RANGE
--region=REGION
```

Adding Subnet to VPC - REST API

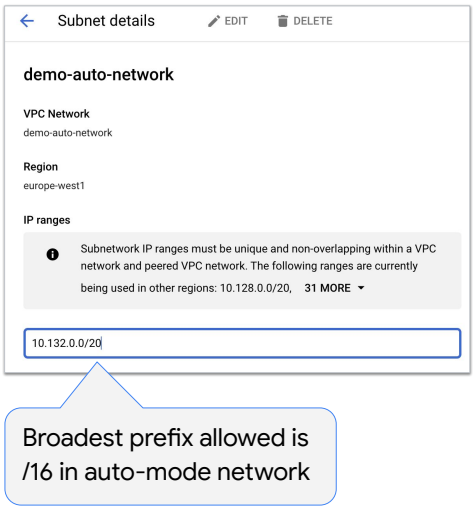
```
POST
https://www.googleapis.com/compute/v1/projects/PROJECT_ID/regions/\
REGION/subnetworks
{
    "ipCidrRange": "IP_RANGE",
    "network": "NETWORK_URL",
    "name": "SUBNET_NAME"
}
```

Many additional parameters can be passed in

Expanding a subnet

- A subnet can be expanded by modifying the subnet mask of an existing subnet
  - Can change the prefix length to a smaller number in CIDR notation.
- Broadest prefix is /16 for an automatically created subnet in an auto-mode network
- Expanding the primary IP address range can't be undone

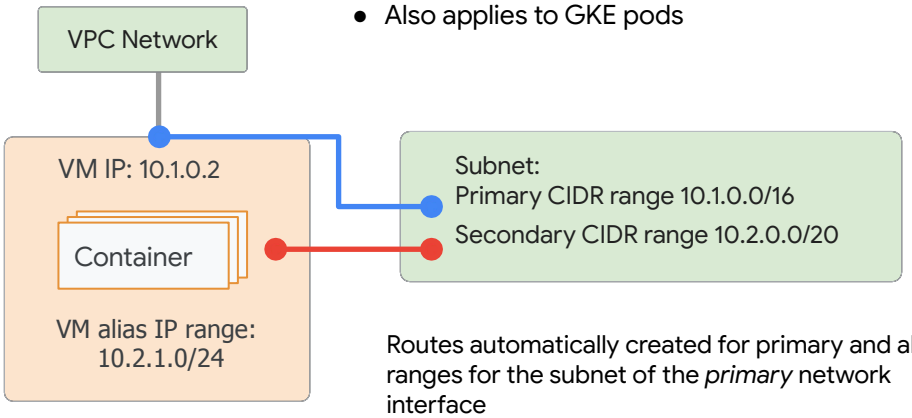
```
gcloud compute networks subnets
expand-ip-range SUBNET_NAME --region=REGION
--prefix-length=PREFIX_LENGTH
```



Broadest prefix allowed is /16 in auto-mode network

A subnet can contain a secondary range of internal IP addresses

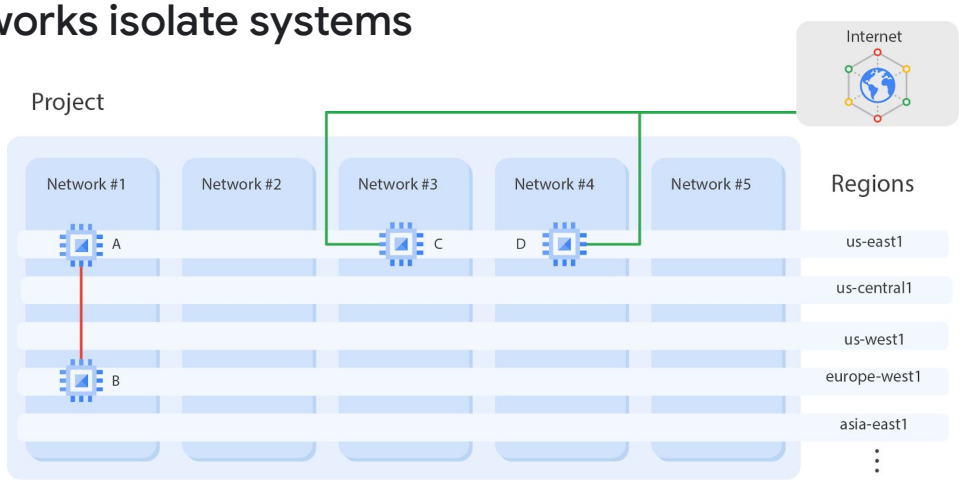
- Useful when multiple services are running on a VM and each needs its own IP address
- Also applies to GKE pods



Routes automatically created for primary and alias IP ranges for the subnet of the *primary* network interface

Alias IP overview
https://cloud.google.com/vpc/docs/alias-ip

Networks isolate systems



- **A** and **B** can communicate over internal IPs *even though they are in different regions.*
- **C** and **D** must communicate over external IPs *even though they are in the same region.*

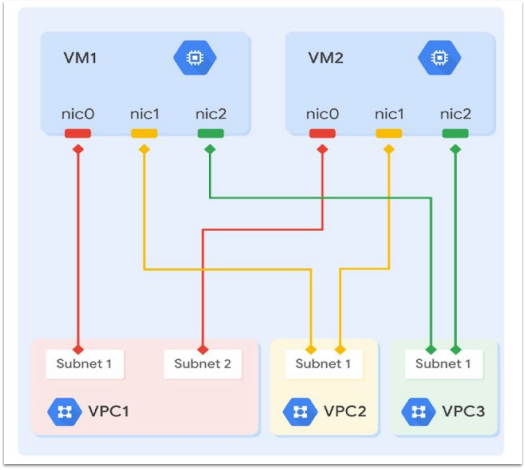
On this slide, we have an example of a project that contains 5 networks. All of these networks span multiple regions across the world, as you can see on the right.

Each network contains separate virtual machines: A, B, C, and D. Because VMs A and B are in the same network, network 1, they can communicate using their internal IP addresses, even though they are in different regions. Essentially, your virtual machines, even if they exist in different locations across the world, take advantage of Google's global fiber network. Those virtual machines appear as though they're sitting in the same rack when it comes to a network configuration protocol.

VMs C and D, however, are not in the same network. Therefore, by default, these VMs must communicate using their external IP addresses, even though they are in the same region. The traffic between VMs C and D isn’t actually touching the public internet, but is going through the Google Edge routers. This has different billing and security ramifications that we will explore later.

VMs can connect to multiple VPCs via multiple network interfaces

- VMs have multi-NIC support (8 max)
  - Each NIC must connect to a different VPC network
  - Allows communication between VPCs using private IPs
- Are other ways to accomplish private IP communication between VPCs, such as
  - VPC Peering
  - VPN
  - These will be discussed later



Creating instances with multiple network interfaces
https://cloud.google.com/vpc/docs/create-use-multiple-interfaces

Routes map traffic to destination networks

- Managed at the VPC level
- Applies to traffic egressing a VM
- Enables VMs on same network (VPC) to communicate via private IP
  - Only if it is allowed by a firewall rule
- Automatically created when a subnet is created
- Can manually create static/custom routes
  - Next hop can be: Instance IP or name, Cloud VPN, Internal TCP/UDP load balancer, default internet gateway
- Routes can be selectively applied to
  - All instances, instances with specific network tags, instances with specific service accounts
- Internet access is enabled by a **default route** (priority=1000)
  - Applies to VMs with external IPs
  - No gateway or public component needed

Routes
https://cloud.google.com/vpc/docs/routes

There are 3 kinds of routes

Subnet routes

- System-generated. Added for each subnet.
- Allows routing between subnets.
- Non Removable and non overridable.
- Exchanged with VPC Peering, and by default through Cloud Router. More on that in a later slide.
- The narrowest possible IP range, which means it cannot be overridden.

Static routes

- Considered a custom route
- Manually added by users
- Next hop can be: Instance IP or name, Cloud VPN, Internal TCP/UDP load balancer, default internet gateway

Dynamic routes

- Considered a custom route
- Added by **Cloud Router** through a BGP session
- Next hop is alway the BGP peer

Unlike other cloud providers, internet access is enabled by a default route (priority=1000). No gateway or public component is needed.

- It doesn't mean all VM's have internet access. an external IP on VM's is needed for public Internet access.
- Removable with caveats
    - A public internet route to destination of Google API's is needed for Private Google Access
    - Cloud CDN requires the default internet route

Network pricing (subject to change)

Traffic type	Price
Ingress	No charge
Egress to the same zone (internal IP address)	No charge
Egress to Google products (YouTube, Maps, Drive)	No charge
Egress to a different Google Cloud service (within same region; exceptions)	No charge
Egress between zones in the same region (per GB)	$0.01
Egress to the same zone (external IP address, per GB)	$0.01
Egress between regions within the US and Canada (per GB)	$0.01
Egress between regions, not including traffic between US regions	Varies by region

This table is from the Compute Engine documentation, and it lists the price of each traffic type.

First of all, ingress or traffic coming into Google Cloud's network is not charged, unless there is a resource such as a load balancer that is processing ingress traffic. Responses to requests count as egress and are charged.

The rest of this table lists egress or traffic leaving a virtual machine. Egress traffic to the same zone is not charged, as long as that egress is through the internal IP address of an instance. Also, egress traffic to Google products, like YouTube, Maps, Drive, or traffic to a different Google Cloud service within the same region is not charged for.

However, there is a charge for egress between zones in the same region, egress within a zone if the traffic is through the external IP address of an instance, and egress between regions.

As for the difference in egress traffic to the same zone, Compute Engine cannot determine the zone of a virtual machine through the external IP address. Therefore, this traffic is treated like egress between zones in the same region.

Also, there are some exceptions, and pricing can always change, so please refer to the documentation page.

Bring your own IP (BYOIP)

Bring your own IP (BYOIP) lets you provision and use your own public IPv4 addresses for Google Cloud resources.

After the IP addresses are imported, Google Cloud manages them in the same way as Google-provided IP addresses, with these exceptions:

- The IP addresses are available only to the customer who imported them
- There are no charges for idle or in-use IP addresses.

Bring your own IP
https://cloud.google.com/vpc/docs/bring-your-own-ip

Summary - VPC Networks

- One VPC Network must exist prior to creating a VM
  - When VMs are created, they must be assigned to a network
- A default network is created when the Compute Engine API is enabled
  - Contains a subnet for every region of Google Cloud
- Upon creation a VM is assigned an internal IP from the CIDR range assigned to the subnet in which the VM was created
  - Can optionally be given a external IP address (ephemeral or status)
- VMs on same network communicate via internal IPs
- VMs in different networks **must** communicate via external IPs
  - Unless
    - VPC Peering is enabled (discussed later)
    - VMs have multiple NICs
- To prevent access to a machine from outside its network don't give it an external IP

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, **shared VPC**)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

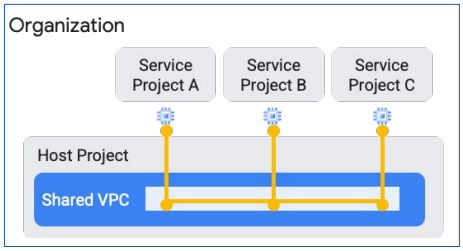
4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

A Shared VPC is created in one project, but can be shared and used by other projects

Networking specialists

- Create the VPC in the **host** project.
- Shares the VPC with other **service** projects.



Allows centralized control over network configuration

- Network admins configure subnets, firewall rules, routes, etc.
- Remove network admin rights from developers.
- Developers focus on machine creation and configuration in the shared network.
- Disable the creation of the default network using an organizational policy.

Shared VPC overview
https://cloud.google.com/vpc/docs/shared-vpc

Shared VPC
allows an organization to connect resources from multiple projects to a common VPC network so that they can communicate with each other securely and efficiently using internal IPs from that network. When you use Shared VPC, you designate a project as a host project and attach one or more other service projects to it. The VPC networks in the host project are called Shared VPC networks. Eligible resources from service projects can use subnets in the Shared VPC network. Eligible resources include Compute Engine resources, GKE clusters, and App Engine flexible instances.

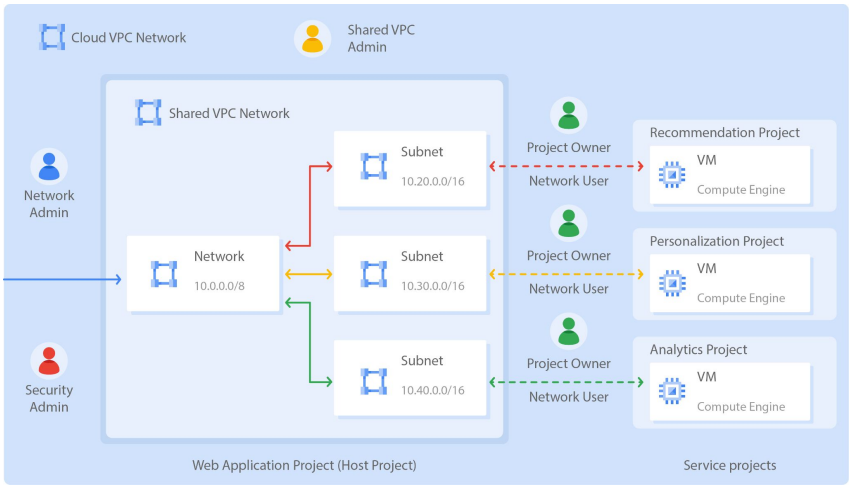
More details of eligible resources can be found here:
https://cloud.google.com/vpc/docs/shared-vpc#resources_that_can_be_attached_to_shared_vpc_networks_from_a_service_project

Shared VPC lets organization administrators delegate administrative responsibilities, such as creating and managing instances, to Service Project Admins while maintaining centralized control over network resources like subnets, routes, and firewalls. This model allows organizations to do the following:

1. Implement the security best practice of least privilege for network admin, auditing, and access control. Shared VPC admins delegate admin tasks to admins in the shared network without allowing service project admins to make

1. network-affecting changes. They can only create and manage instances that use the shared VPC.
2. Apply and enforce consistent access control policies at the network level for multiple service projects.

Shared VPC



In this diagram, the Shared VPC Admin configured the Web Application Project to be a host project with subnet-level permissions. Doing so allowed the Shared VPC Admin to selectively share subnets from the VPC network.

Next, the Shared VPC Admin attached the three service projects to the host project and gave each project owner the Network User role for the corresponding subnets. Each project owner then created VM instances from their service projects in the shared subnets. By the way, billing for those VM instances is attributed to the project where the resources are created, which are the service projects.

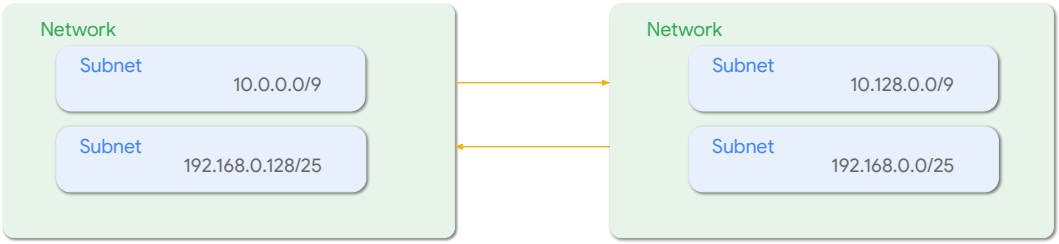
Shared VPC Admins have full control over the resources in the host project, including administration of the shared VPC network. They can optionally delegate the Network Admin and Security Admin roles for the host project. Overall, shared VPC is a centralized approach to multi-project networking because security and network policy occurs in a single designated VPC network.

For a demo on how to create VM instances in a Shared VPC network, please refer here:
https://storage.googleapis.com/cloud-training/gcpnet/student/M3_Demo_SharedVPC.mp4

VPC peering allows VMs on different VPC networks to communicate with private IPs

Different concept from Shared VPC

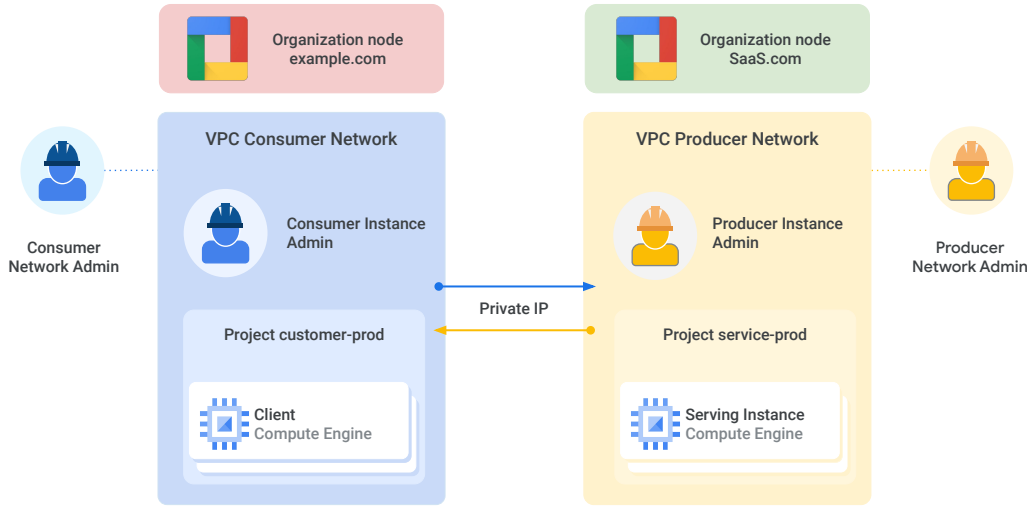
- VMs in different VPC networks cannot communicate over private IPs by default
- VPC Peering connects two VPC Networks
  - As long as there are no overlapping subnet IP ranges
  - Networks can be in the same project, different projects or different organizations
- Traffic latency within a peering group is the same as if they were the same VPC network



VPC Network Peering overview:
https://cloud.google.com/vpc/docs/vpc-peering

Using VPC Network Peering:
https://cloud.google.com/vpc/docs/using-vpc-peering

VPC peering



VPC Network Peering allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization.

Now, remember that each VPC network will have firewall rules that define what traffic is allowed or denied between the networks.

For example, in this diagram there are two organizations that represent a consumer and a producer, respectively. Each organization has its own organization node, VPC network, VM instances, Network Admin and Instance Admin. In order for VPC Network Peering to be established successfully, the Producer Network Admin needs to peer the Producer Network with the Consumer Network, and the Consumer Network Admin needs to peer the Consumer Network with the Producer Network. When both peering connections are created, the VPC Network Peering session becomes Active and routes are exchanged This allows the VM instances to communicate privately, using their internal IP addresses.

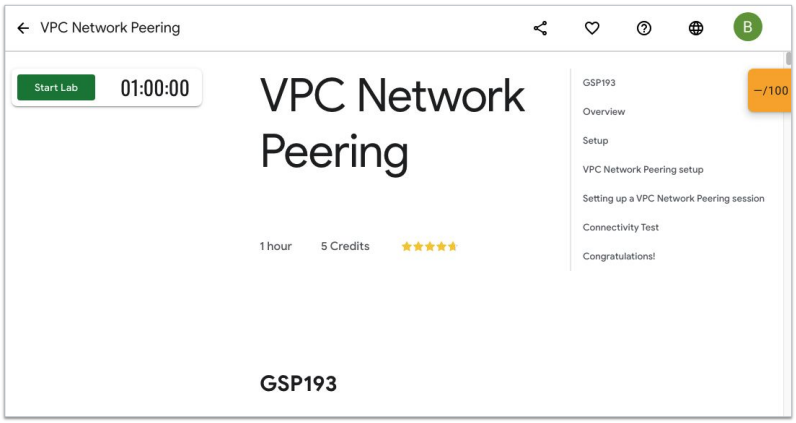
VPC Network Peering is a decentralized or distributed approach to multi-project networking, because each VPC network may remain under the control of separate administrator groups and maintains its own global firewall and routing tables. Historically, such projects would consider external IP addresses or VPNs to facilitate

private communication between VPC networks. However, VPC Network Peering does not incur the network latency, security, and cost drawbacks that are present when using external IP addresses or VPNs.

VPC peering benefits

- **Reduce latency**
  - Connecting via private IPs will have lower latency than public IPs
- **Reduce costs**
  - Google Cloud charges egress bandwidth when using public IPs to communicate
  - Peering communication is via private IPs
- **Improve Security**
  - VMs may no longer require public access

Suggested Lab (if time allows)



https://partner.cloudskillsboost.google/catalog_lab/935

Lab: VPC Network Peering
https://partner.cloudskillsboost.google/catalog_lab/935

Shared VPC vs. VPC peering

Consideration	Shared VPC	VPC Network Peering
Across Organizations	No	Yes
Within Project	No	Yes
Network Administration	Centralized	Decentralized

Now, that we've talked about Shared VPC and VPC Network Peering, let's compare both of these configurations to help you decide which is appropriate for a given situation.

If you want to configure private communication between VPC networks in different organizations, you have to use VPC Network Peering. Shared VPC only works within the same organization.

Somewhat similarly, if you want to configure private communication between VPC networks in the same project, you have to use VPC Network Peering. This doesn't mean that the networks need to be in the same project, but they *can* be, as you will explore in the upcoming lab. Shared VPC only works across projects.

Exam Guide - VPC Network

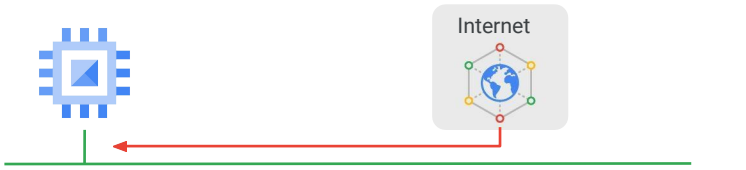
3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

VMs *must* have internal IP and *may* have external IP addresses



Internal IP

- Allocated from subnet range to VMs by DHCP
- Alternatively, can reserve (static) internal IP address

External IP

- Assigned from pool (ephemeral)
- Alternatively, can reserve (static) external IP address
- Bring Your Own IP address (BYOIP)

IP addresses
https://cloud.google.com/compute/docs/ip-addresses

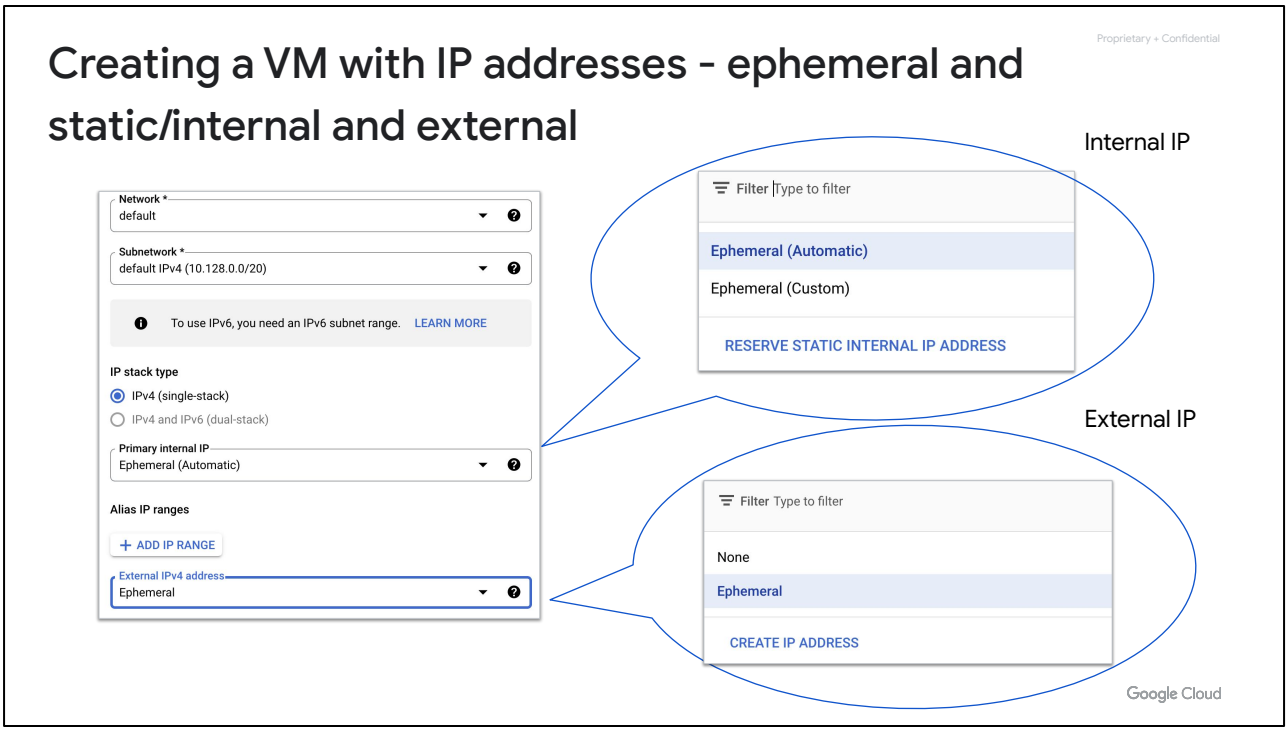
Support for IPv6 addresses:
https://cloud.google.com/vpc/docs/subnets?hl=en_US#ipv6-ranges

Static IPs are needed when a VM requires a internal or external IP address that won't change

- Can either
  - Create a new internal/external IP address and assign it to a new VM
  - Promote an existing ephemeral IP address to become static
- Static internal IPs are allocated from the range of IPs assigned to the VM's subnet
- Are reserved for the project until it is released
- Can be assigned to a new or existing instance

Note: Load balancers can also use ephemeral or static IPs

Overview:
https://cloud.google.com/vpc/docs/ip-addresses



Reserve a static external IP address
https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address

Reserving a static internal IP address
https://cloud.google.com/compute/docs/ip-addresses/reserve-static-internal-ip-address

Assign a static external IP address to a new VM instance:
https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address#assign_new_instance

Create a VM instance with a specific internal IP address:
https://cloud.google.com/compute/docs/ip-addresses/reserve-static-internal-ip-address#create_a_vm_instance_with_a_specific_internal_ip_address

Create a VM with no external IP address
https://cloud.google.com/sdk/gcloud/reference/compute/instances/create
Look at the “--no-address” flag

Internal DNS;
https://cloud.google.com/compute/docs/internal-dns

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, **Google private access**, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

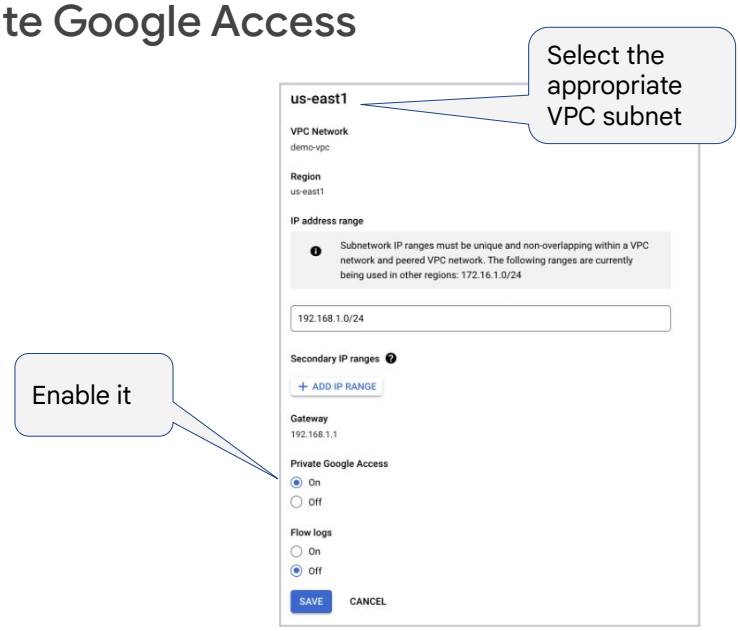
- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

Private Google Access

- Certain Google API based managed services do not use VPC Networks
  - For example
    - Cloud Functions
    - Cloud Storage
    - Cloud Run
    - Container/Artifact Registry
    - AI Platform - Video Intelligence, Translation, etc.
- They use **public** service endpoints that specifies the network address of an API service
- For VMs to reach these, they need an external IP
  - External IPs on VMs are not best practice
- Solution
  - Enable Private Google Access in the subnetwork to which the VM is attached

https://cloud.google.com/vpc/docs/private-google-access

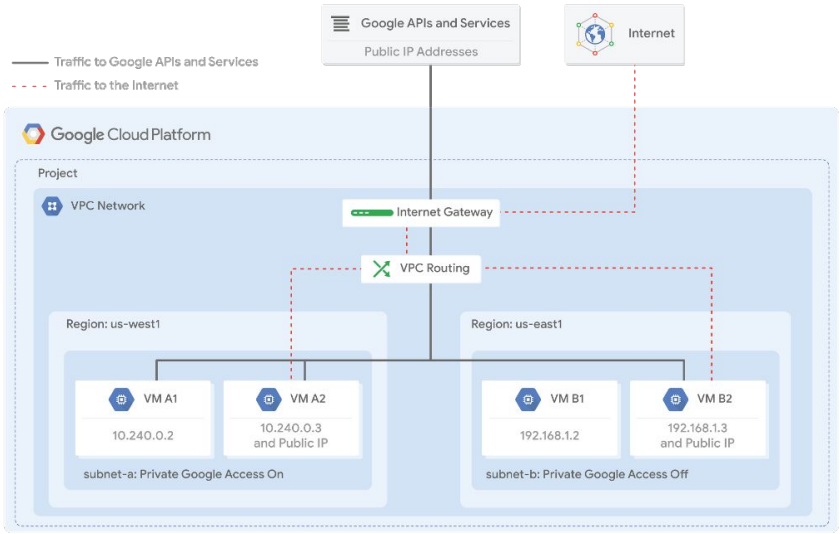
Enabling Private Google Access



Configuring Private Google Access:
https://cloud.google.com/vpc/docs/configure-private-google-access

Enabling:
https://cloud.google.com/vpc/docs/configure-private-google-access#config-pga

Private Google Access to Google APIs and services

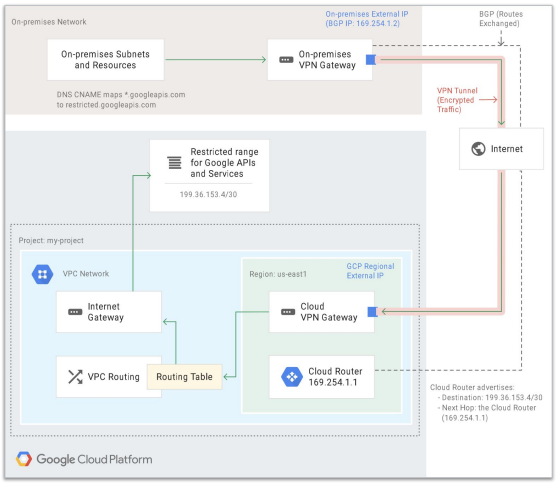


You enable Private Google Access on a subnet-by-subnet basis. As you can see in this diagram, subnet-a has Private Google Access enabled, and subnet-b has it disabled. This allows VM A1 to access Google APIs and services, even though it has no external IP address.

Private Google Access has no effect on instances that have external IP addresses. That's why VMs A2 and B2 can access Google APIs and services. The only VM that can't access those APIs and services is VM B1. This VM has no public IP address, and it is in a subnet where Google Private Access is disabled.

Private Google Access On-Prem

- Allows on-premises hosts to reach Google APIs and services using internal IPs
  - Must use Cloud VPN or Cloud Interconnect
- To enable Private Google Access for on-premises hosts,
  - Configure DNS, firewall rules, and routes in on-premises and VPC networks.
  - **No need to enable Private Google Access for any subnets in your VPC network**



Private Google Access for on-premises hosts
https://cloud.google.com/vpc/docs/private-google-access-hybrid

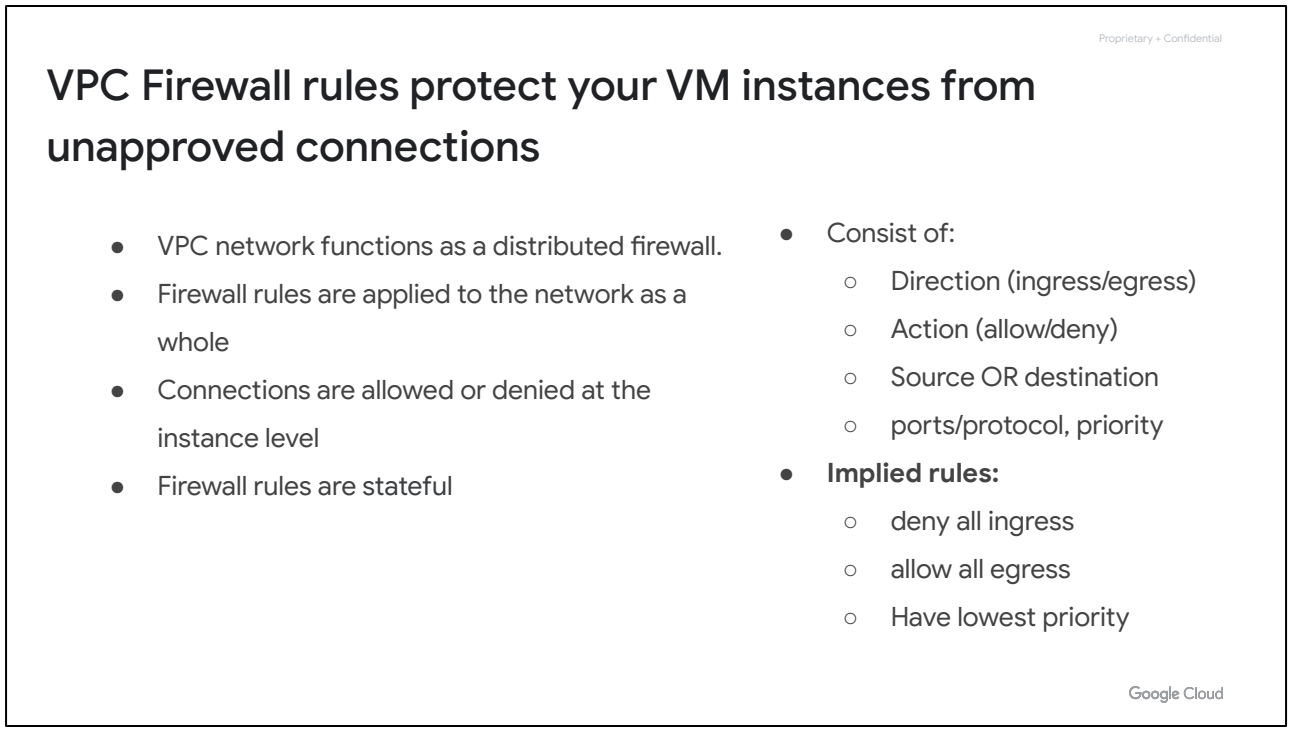
Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules



VPC Firewall rules
https://cloud.google.com/vpc/docs/firewalls

3 ways to configure robust firewall rules:
https://cloud.google.com/blog/products/gcp/three-ways-to-configure-robust-firewall-rules

Google Cloud firewall rules protect your virtual machine instances from unapproved connections, both inbound and outbound, known as ingress and egress, respectively. Essentially, every VPC network functions as a distributed firewall.

Google Cloud firewall rules provide effective protection and traffic control regardless of the operating system your instances use. Google Cloud firewall rules are defined for the VPC network as a whole, and since VPC networks can be global in Google Cloud, firewall rules are also global.

Although firewall rules are applied to the network as a whole, connections are allowed or denied at the instance level. You can think of the firewall as existing not only between your instances and other networks, but between individual instances within the same network.

Google Cloud firewall rules are stateful. This means that if a connection is allowed between a source and a target or a target and a destination, all subsequent traffic in either direction will be allowed. In other words, firewall rules allow bidirectional

communication once a session is established.
Also, if for some reason, all firewall rules in a network are deleted, there is still an implied "Deny all" ingress rule and an implied "Allow all" egress rule for the network.

All VPCs have implied firewall rules

Implied IPv4/IPv6 firewall rules are present in all VPC networks

- Implied allow egress rule
  - Lets any instance send traffic to any destination
- Implied deny ingress rule
  - Protects all instances by blocking incoming connections to them
- Override them with your own firewall rules (if desired)

Implied IPv4 firewall rules are present in all VPC networks, regardless of how the networks are created, and whether they are auto mode or custom mode VPC networks. The default network has the same implied rules.

- **Implied IPv4 allow egress rule.** An egress rule whose action is allow, destination is 0.0.0.0/0, and priority is the lowest possible (65535) lets any instance send traffic to any destination, except for traffic blocked by Google Cloud.

- **Implied IPv4 deny ingress rule.** An ingress rule whose action is deny, source is 0.0.0.0/0, and priority is the lowest possible (65535) protects all instances by blocking incoming connections to them. A higher priority rule might allow incoming access.

If IPv6 is enabled, the VPC network also has these two implied rules:

- **Implied IPv6 allow egress rule.** An egress rule whose action is allow, destination is ::/0, and priority is the lowest possible (65535) lets any instance send traffic to any destination, except for traffic blocked by Google Cloud. A higher priority firewall rule may restrict outbound access. Internet access is allowed if no other firewall rules deny outbound traffic and if the instance has an external IP address.

- **Implied IPv6 deny ingress rule.** An ingress rule whose action is deny, source

- is ::/0, and priority is the lowest possible (65535) protects all instances by blocking incoming connections to them. A higher priority rule might allow incoming access.

The implied rules *cannot* be removed, but they have the lowest possible priorities.

For more information on implied rules check out the link in the speaker notes.

- **Link:** cloud.google.com/vpc/docs/firewalls#default_firewall_rules

Default VPCs have additional allow rules

Rule	Description
`default-allow-internal`	Allows ingress connections for all protocols and ports among instances within the VPC network
`default-allow-ssh`	Allows port 22 - secure shell (ssh) access
`default-allow-rdp`	Allows port 3389 - remote desktop protocol (RDP) access
`default-allow-icmp`	Allows ICMP traffic

In Google Cloud, all projects get a default VPC created automatically. In addition to the implied rules, the default VPC network is pre-populated with firewall rules that allow incoming, or ingress, traffic to instances.The first rule is default-allow-internal which allows ingress connections for all protocols and ports among instances within the VPC network. It effectively permits incoming connections to VM instances from others in the same network.

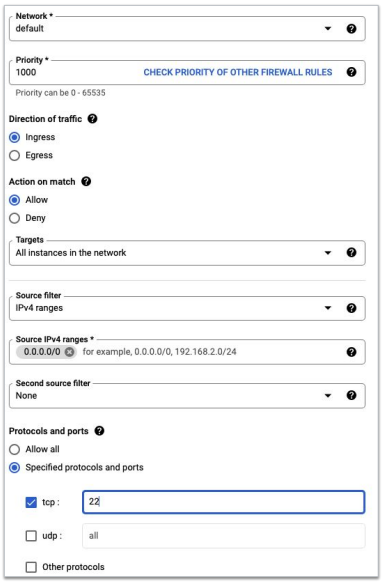
The other three rules in the default network are default-allow-ssh, default-allow-rdp and default-allow-icmp.These rules allow port 22 - secure shell (ssh), port 3389 - remote desktop protocol (RDP), and ICMP traffic respectively, from any source IP address to any instance in the VPC network.

All of these rules have the second-to-lowest priority of 65534.

As you may have noticed some of these rules can be a little dangerous. These rules can (and should) be deleted or modified as necessary.

Creating Firewall Rules

- When creating rules, specify
  - Source
    - Could be the internet (0.0.0.0/0 IP range)
    - Individual or ranges of IPv4 or IPv6 addresses
    - Could be VMs with specific network tags or service accounts
  - Target - Defines which VMs the rule applies to
    - All instances in the network
    - VMs with specific network tags
    - VM's with service accounts



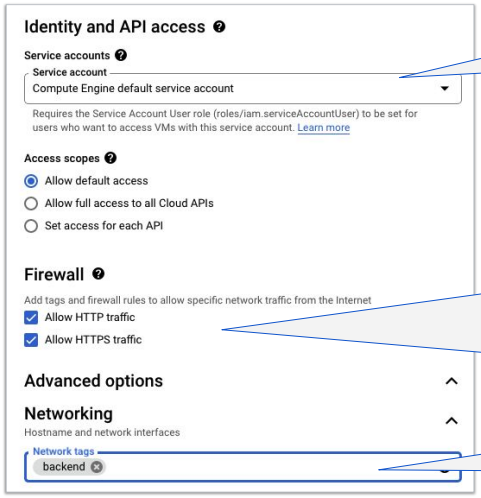
VPC firewall rules overview:
https://cloud.google.com/vpc/docs/firewalls

Using firewall rules:
https://cloud.google.com/vpc/docs/using-firewalls

Configure firewall rules for common use cases:
https://cloud.google.com/vpc/docs/using-firewalls#rules-for-common-use-cases

3 ways to configure robust firewall rules:
https://cloud.google.com/blog/products/gcp/three-ways-to-configure-robust-firewall-rules

Console - Creating a VM with network tags and a service account



Set the Service Account here

Developer needs a specific IAM role to attach a service account to a resource

Clicking one (or both) of these boxes results in automatically generated firewall rule(s) that apply to VMs with the network tags:

- allow-http
- allow-https

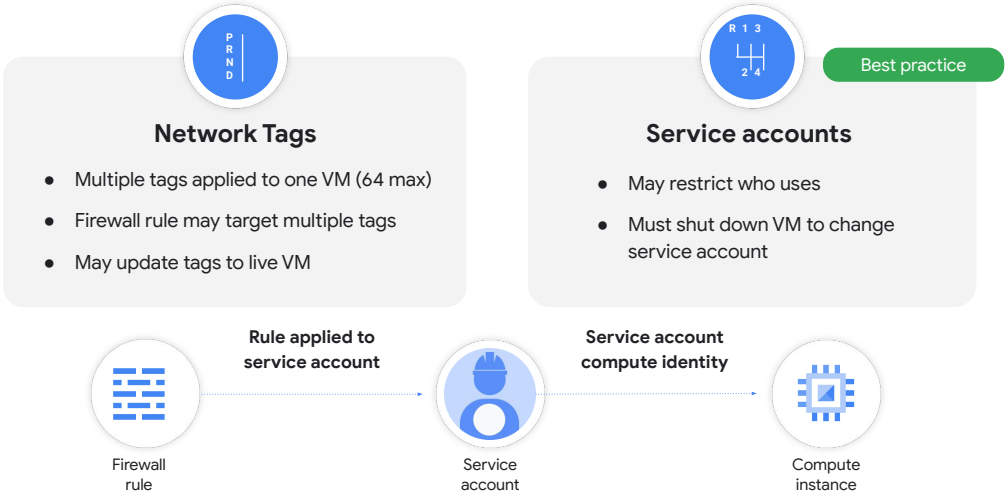
These tags are automatically added to the VM upon creation.

Can manually add network tags. One VM can have multiple tags

Configuring network tags:
https://cloud.google.com/vpc/docs/add-remove-network-tags

Adding, viewing, and removing tags:
https://cloud.google.com/vpc/docs/add-remove-network-tags#adding_viewing_and_removing_tags

Creating firewall rules - tags vs service accounts



- While some firewall rules may apply to all VM instances, there will be others that you want to apply selected instances.
- For example, you may have
  - A front end Web application running in a Managed Instance Group, that should only be accessible via an external Load balancer proxy
  - And a Backend app that should only be accessible on specific ports by the Web application instances
- This can be done based on either network tags or service accounts. Which one should you use?

Network Tags vs Service Accounts

Network Tags

Are very flexible

- Tags can be updated while a VM is running, while a service account cannot
- A VM can have multiple tags, while a VM can have one service account only
- FW rules match if >= 1 tag matchesTags

Potential con:

- Not possible to control who can apply a specific network tag.
- Example: A HTTP ingress allow rule exists that applies to instances with 'webapp-http-allow' network tag
- Anyone with Compute Engine Admin IAM role can create VM's with this network tag.

```
gcloud compute firewall-rules create web-logdata \
    --network logging-network \
    --allow TCP:443 \
    --source-tags web-production \
    --target-tags log-data
```

Network Tags vs Service Accounts

Network Tags

Are very flexible

- Tags can be updated while a VM is running, while a service account cannot
- A VM can have multiple tags, while a VM can have one service account only
- FW rules match if >= 1 tag matchesTags

Potential con:

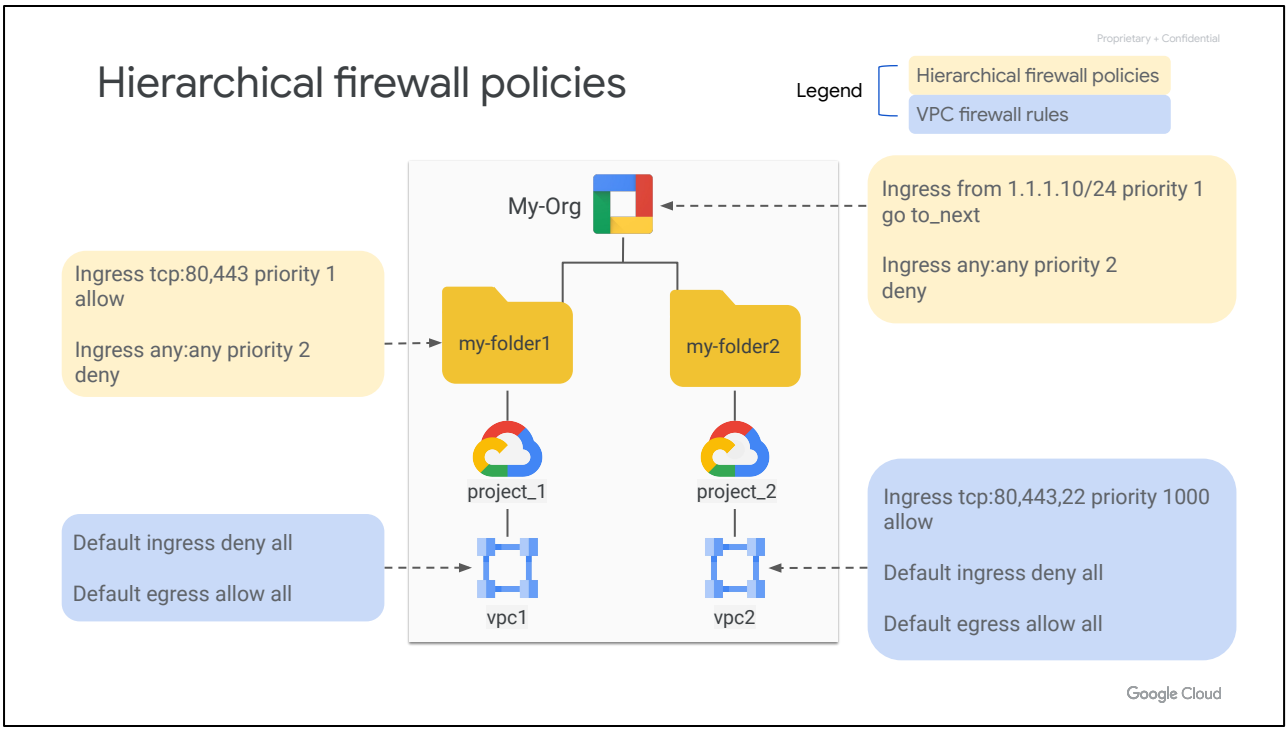
- Not possible to control who can apply a specific network tag.
- Example: A HTTP ingress allow rule exists that applies to instances with 'webapp-http-allow' network tag
- Anyone with Compute Engine Admin IAM role can create VM's with this network tag.

Service accounts

Best practice from a security viewpoint

- Can control who can assign a given service account to instances
- Allows strict control over the instances a given rule will apply to.
- Goes hand in hand with IAM best practice - creating service accounts with minimal privileges for different applications and their components.

```
gcloud compute firewall-rules create frontend-to-backend \
--direction=INGRESS --network=default --allow TCP:443 \
--source-service-accounts=frontend@proj.iam.gserviceaccount.com \
--target-service-accounts=backend@proj.iam.gserviceaccount.com
```



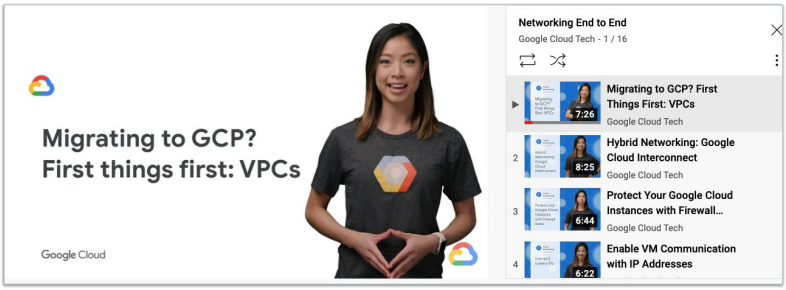
Managing cloud firewalls at scale with new Hierarchical Firewall Policies
https://cloud.google.com/blog/products/identity-security/new-google-cloud-hierarchical-firewall-policies

Hierarchical firewall policies let you create and enforce a consistent firewall policy across your organization. You can assign hierarchical firewall policies to the organization as a whole or to individual folders. These policies contain rules that can explicitly deny or allow connections, as do Virtual Private Cloud (VPC) firewall rules. In addition, hierarchical firewall policy rules can delegate evaluation to lower-level policies or VPC network firewall rules with a goto_next action. Lower-level rules cannot override a rule from a higher place in the resource hierarchy. This lets organization-wide admins manage critical firewall rules in one place.

By default, all hierarchical firewall policy rules apply to all VMs in all projects under the organization or folder where the policy is associated. However, you can restrict which VMs get a given rule by specifying a target network or target service account. The levels of the hierarchy at which firewall rules can now be applied are represented in the diagram, shown here. The yellow boxes near the top represent hierarchical firewall policies, while the blue boxes at the bottom represent VPC firewall rules.

Google Cloud VPC - Youtube Videos

Covers many of the same topics mentioned here today



https://bit.ly/34uBApk

Youtube videos covering VPC topics
https://bit.ly/34uBApk

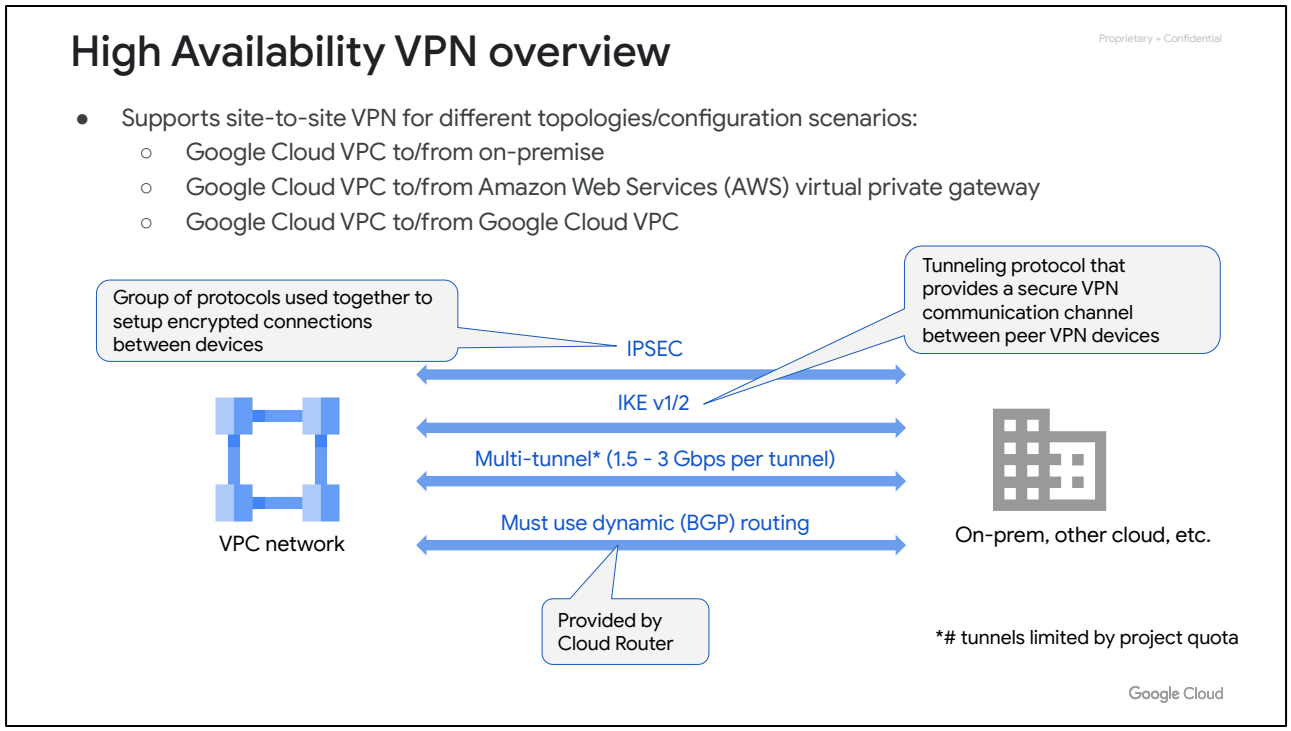
Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules



Cloud VPN overview:
https://cloud.google.com/network-connectivity/docs/vpn/concepts/overview

Cloud VPN documentation:
https://cloud.google.com/network-connectivity/docs/vpn

On Google Cloud, dynamic routing can be established using Cloud Router. It exchanges network topology information through Border Gateway Protocol (BGP). Cloud Router advertises subnets from its VPC network to another router or gateway via BGP.

HA VPN is a high availability Cloud VPN solution that lets you securely connect your on-premises network to your Virtual Private Cloud (VPC) network through an IPsec VPN connection in a single region. HA VPN provides an SLA of 99.99% service availability. To guarantee a 99.99% availability SLA for HA VPN connections, you must properly configure two or four tunnels from your HA VPN gateway to your peer VPN gateway or to another HA VPN gateway.

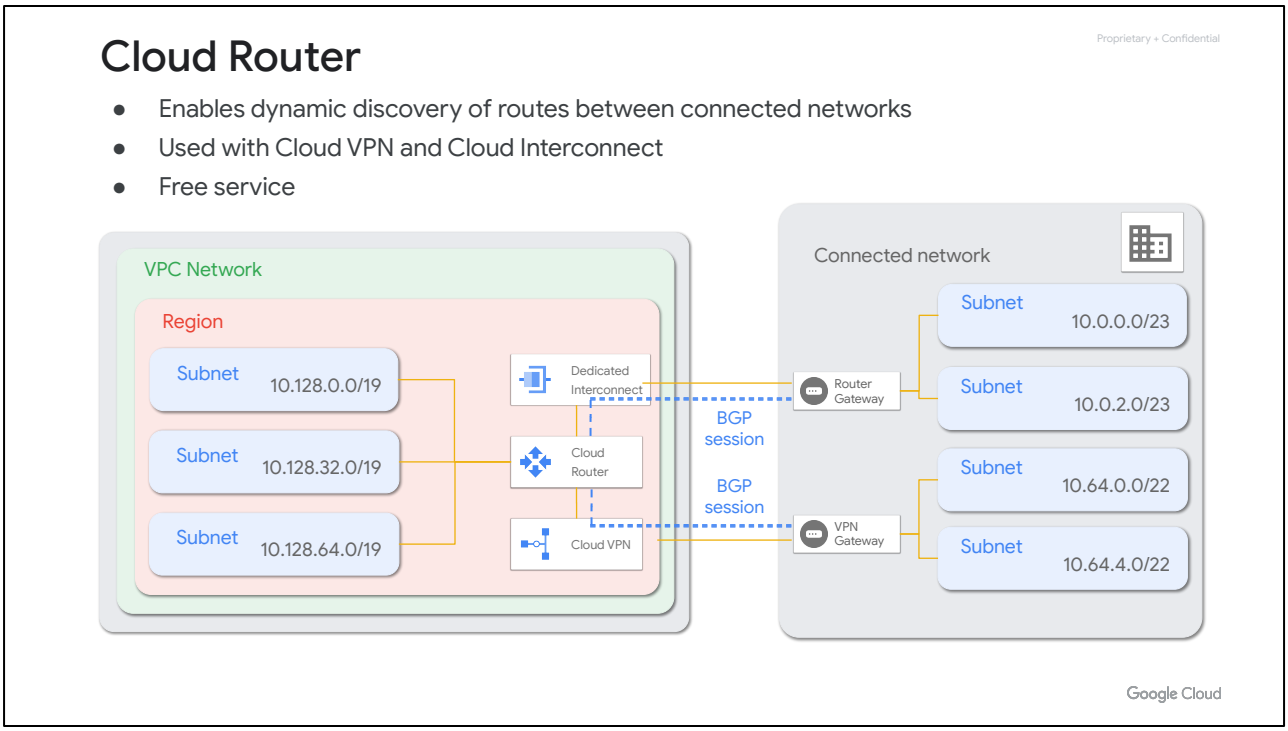
When you create an HA VPN gateway, Google Cloud automatically chooses two external IP addresses, one for each of its fixed number of two interfaces. Each IP address is automatically chosen from a unique address pool to support high availability.

Each of the HA VPN gateway interfaces supports multiple tunnels. You can also

create multiple HA VPN gateways. When you delete the HA VPN gateway, Google Cloud releases the IP addresses for reuse. You can configure an HA VPN gateway with only one active interface and one external IP address; however, this configuration does not provide a 99.99% service availability SLA. VPN tunnels connected to HA VPN gateways must use dynamic (BGP) routing. Depending on the way that you configure route priorities for HA VPN tunnels, you can create an active/active or active/passive routing configuration.

HA VPN supports site-to-site VPN in one of the following recommended topologies or configuration scenarios:

- An HA VPN gateway to peer VPN devices
- An HA VPN gateway to an Amazon Web Services (AWS) virtual private gateway
- Two HA VPN gateways connected to each other



Cloud Router overview:
https://cloud.google.com/network-connectivity/docs/router/concepts/overview

Cloud Router pricing
https://cloud.google.com/network-connectivity/docs/router/pricing

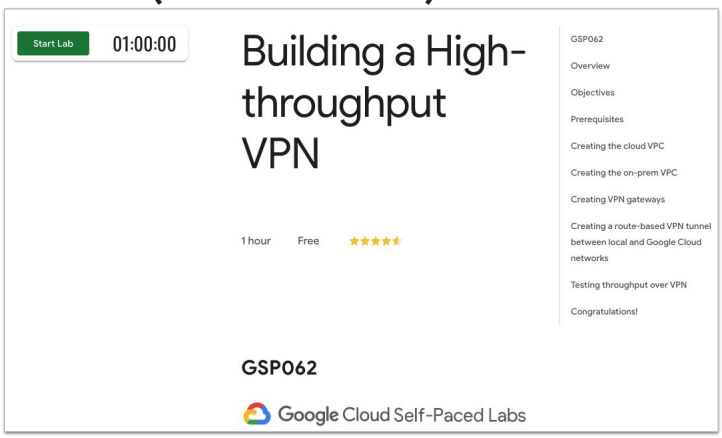
Cloud Router is a fully distributed and managed Google Cloud service that uses the Border Gateway Protocol (BGP) to advertise IP address ranges. It programs custom dynamic routes based on the BGP advertisements that it receives from a peer. Instead of a physical device or appliance, each Cloud Router consists of software tasks that act as BGP speakers and responders. A Cloud Router also serves as the control plane for Cloud NAT. Cloud Router provides BGP services for the following Google Cloud products:

- Dedicated Interconnect
- Partner Interconnect
- Cloud VPN, specifically HA VPN
- Router appliance (part of Network Connectivity Center)

Key points:

- An important **building block** of **Cloud VPN** and **Interconnect**
- **Managed service**
- **Regional**
- **Control plane** only. Not part of the data path.
- **Exchanges routes** between your VPC and your on-premises network via **BGP**
- **Dynamic routing** options
    - Regional: Shares routes **only for subnets** in the region **where Cloud Router is provisioned**
    - Global: Shares routes for **all subnets** in the VPC
- **Route advertisement** (on BGP session or Cloud Router level)
    - **Default**: Advertises subnets according to the selected routing option
    - **Custom:** Customise which subnets and IP ranges to advertise, for example:
        - when you want to **skip advertising specific subnets**
        - when you want to **advertise subnets in different BGP sessions** for **separation purposes**
- **Scaling:**
    - Note that **dynamic routes** learned by Cloud Router are l**imited to 100 by default**.
    - The **routes** dynamically announced by **on-premises to Google Cloud** should be **summarized to avoid** hitting this **quota**.

Suggested Lab (if time allows)



https://partner.cloudskillsboost.google/catalog_lab/620

https://partner.cloudskillsboost.google/catalog_lab/620

Exam Guide - VPC Network

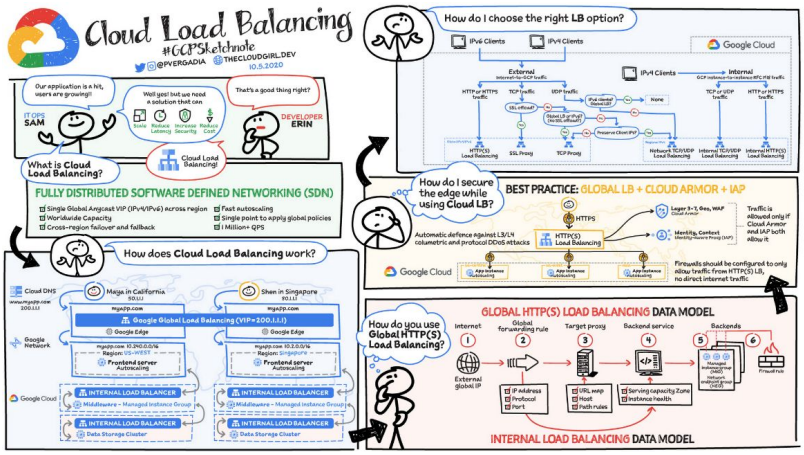
3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

Cloud Load Balancing



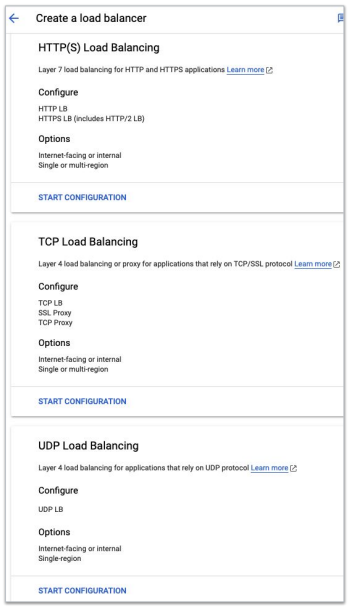
[What is Cloud Load Balancing?]

Video: https://www.youtube.com/watch?v=h8EqM6Xt3MA

Corresponding blog
https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-load-balancing

Load balancing Features

- High performance
- Fully distributed and software defined
- Anycast IP
- Regional/zonal spillover and failover
- Intelligent backend autoscaling and health checks



Cloud Load Balancing
https://cloud.google.com/load-balancing

Choosing a load balancer (contains decision tree):
https://cloud.google.com/load-balancing/docs/choosing-load-balancer

Cloud Load Balancing overview:
https://cloud.google.com/load-balancing/docs/load-balancing-overview

What is Cloud Load Balancing?
https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-load-balancing

YouTube video:
https://www.youtube.com/watch?v=h8EqM6Xt3MA&t=160s

Load Balancing docs:
https://cloud.google.com/load-balancing/docs

Internal HTTP(S) load balancing:
https://cloud.google.com/load-balancing/docs/l7-internal

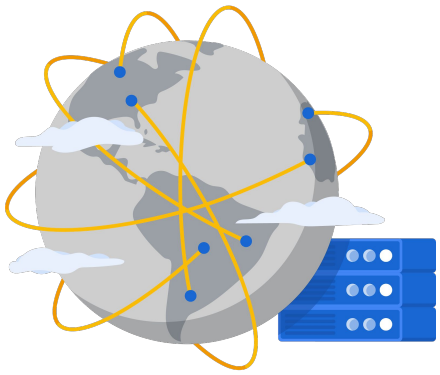
External TCP/UDP Network Load Balancing overview:

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

- Load balancers are **software defined and distributed**, which means they are **highly performant** and not bottlenecked by a single appliance
- They are **tightly integrated with Compute Engine and GKE** to allow for **intelligent autoscaling** based on various metrics, and **health checks**, so we can route traffic to the **healthy instances and locations.**
- Load balancers are available based on **geo scope** (regional / global), **network tier** (premium / standard) and **proxy or pass-through**
- **Pass-through** proxies means client IP is preserved. **Proxy** means the client IP is not preserved, and a different connection is established between the LB and the backend instances.
- **High-level review**
    - **Internal**
        - **Regional** and require **premium network tier**
        - **L4 (TCP/UDP)** and **L7 (HTTP/s)**
    - **External**
        - Supports both **network tiers**
            - **Proxy** load balancers (TCP/SSL/HTTPs) are available as **global resources** by **directing traffic** to **healthy backends** closest to the end-user. For that, they make use of **Google global network infrastructure**, and accordingly require **Premium Tier.**
            - They are also supported with **standard tier**, in which case they effectively function as **regional load balancers**
        - **HTTPs** load balancer easily integrates with **Cloud CDN** and **Cloud Armor (WAF)**
        - The **Network Load Balancer** provides a L4 (TCP/UDP) regional load balancer that is pass-through

HTTP(S) load balancing

- Global or regional (internal or external) load balancing
- Anycast IP address
- HTTP on port 80 or 8080
- HTTPs on port 443
- IPv4 or IPv6
- Autoscaling
- URL maps



HTTP(S) Load Balancing

Google Cloud's HTTP(S) load balancing provides global load balancing for HTTP(S) requests destined for your instances. This means that your applications are available to your customers at a single anycast IP address, which simplifies your DNS setup. HTTP(S) load balancing balances HTTP and HTTPS traffic across multiple backend instances and across multiple regions.

HTTP requests are load balanced on port 80 or 8080, and HTTPS requests are load balanced on port 443.

This load balancer supports both IPv4 and IPv6 clients, is scalable, requires no pre-warming, and enables content-based and cross-region load balancing.

You can configure URL maps that route some URLs to one set of instances and route other URLs to other instances. Requests are generally routed to the instance group that is closest to the user. If the closest instance group does not have sufficient capacity, the request is sent to the next closest instance group that does have capacity.

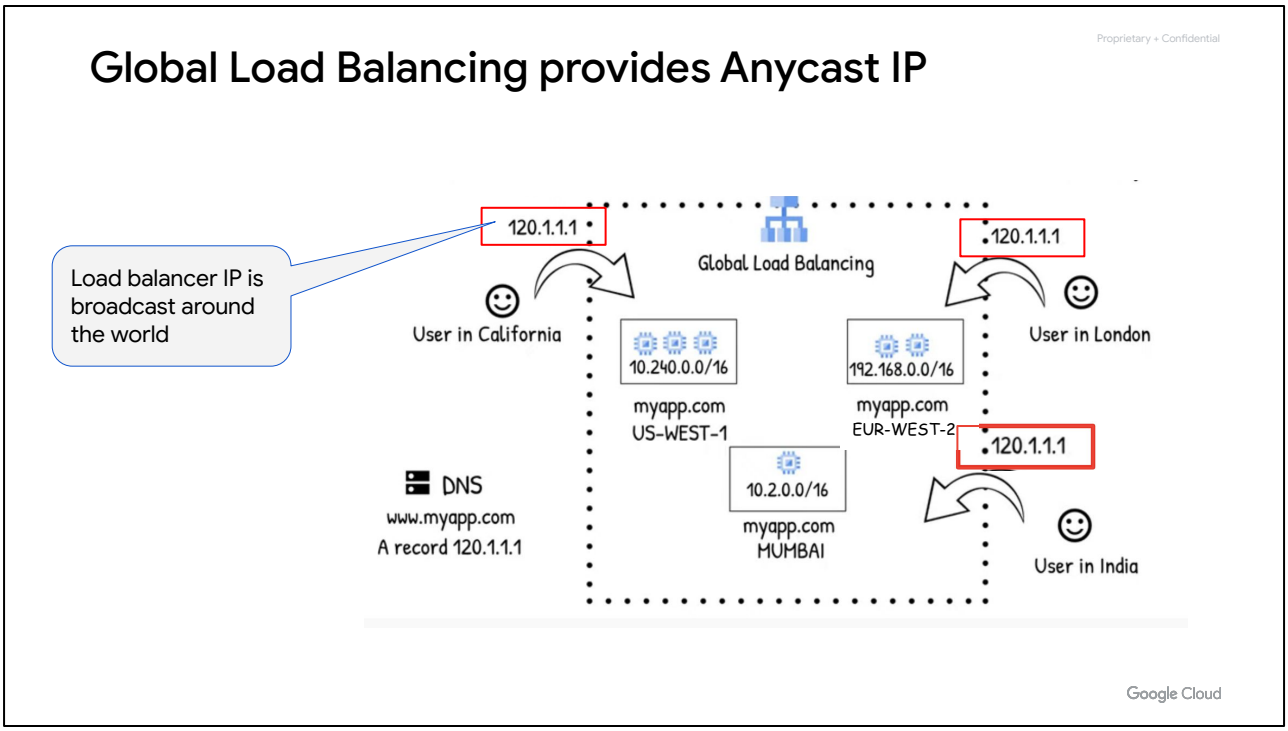


Image from video:
https://www.youtube.com/watch?v=0fQr7TRhnnU&list=PLTWE_Imu2InBzuPmOcgAYP7U80a87cpJd

Corresponding blog:
https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-load-balancing

Regional/Zonal Spillover

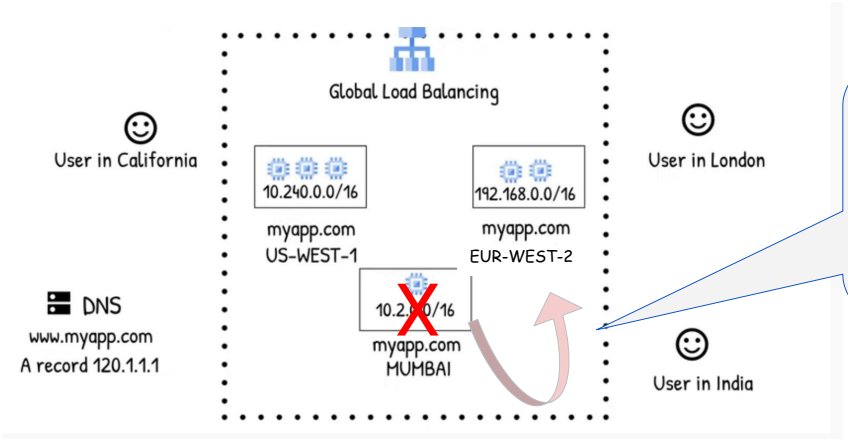


Image from video:
https://www.youtube.com/watch?v=0fQr7TRhnnU&list=PLTWE_Imu2InBzuPmOcgAYP7U80a87cpJd

Corresponding blog:
https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-load-balancing

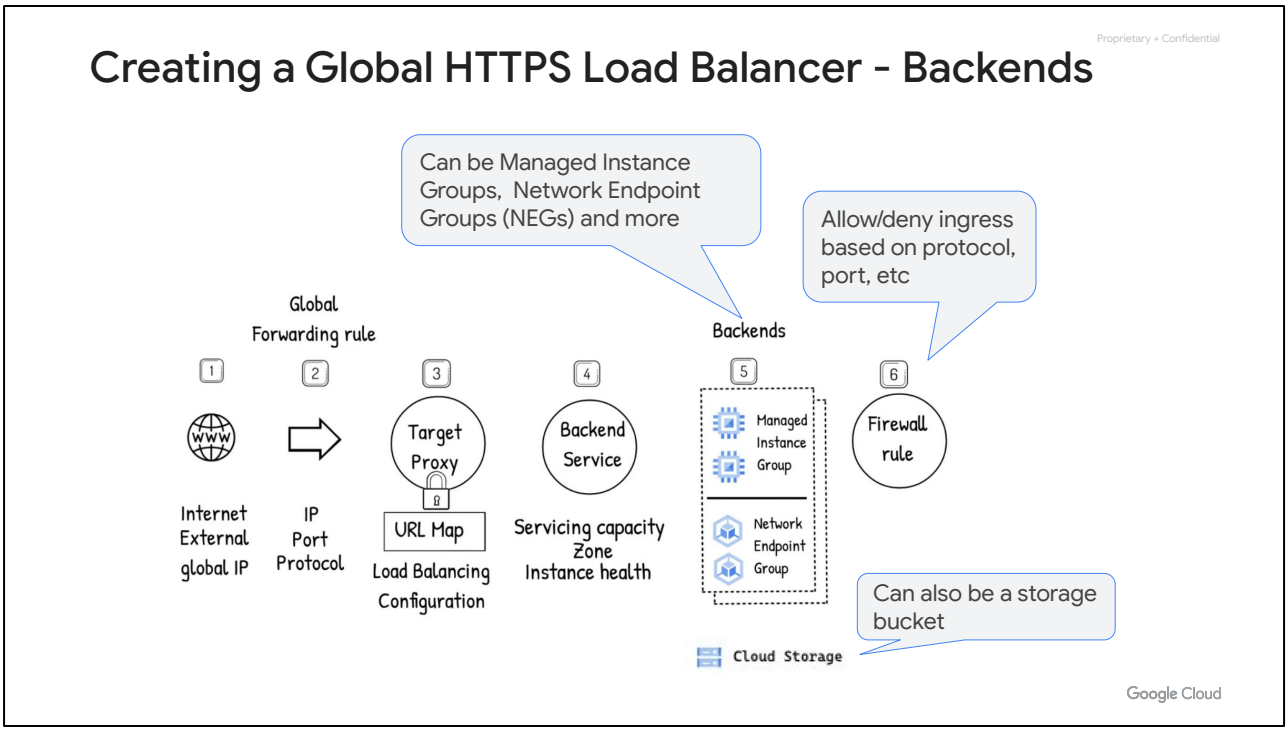


Image from video:
https://www.youtube.com/watch?v=0fQr7TRhnnU&list=PLTWE_Imu2InBzuPmOcgAYP7U80a87cpJd

Corresponding blog:
https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-load-balancing

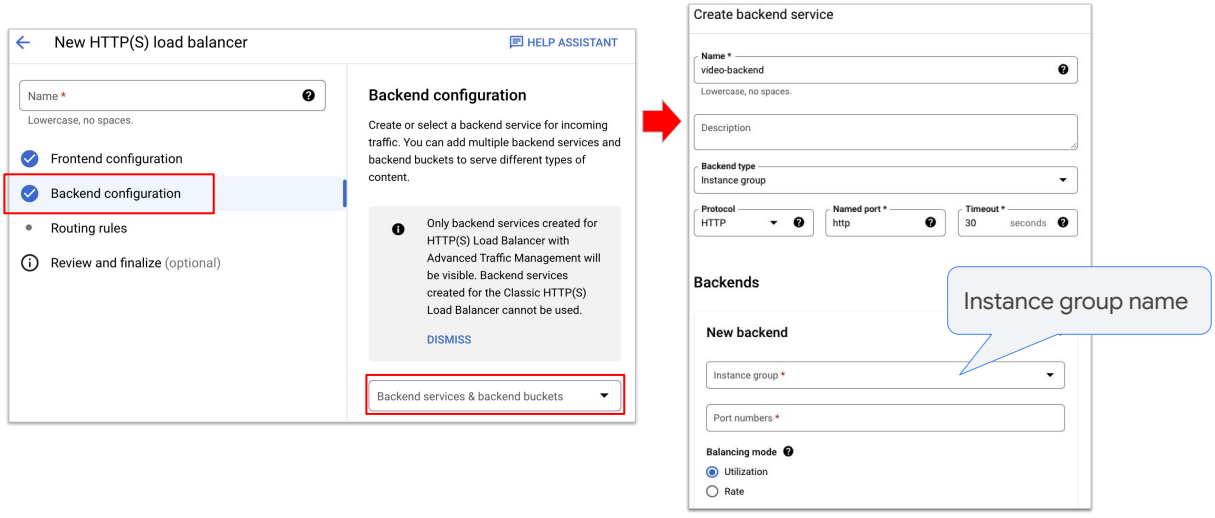
Backend options

- Backend - one or more endpoints that receive traffic from a load balancer
- Options include
  - Instance groups
  - Cloud Storage buckets
  - Network endpoint groups (NEG)
    - A group of backend endpoints or services
    - Common use case: Services running in containers
  - Zonal NEG
    - A group of VMs in the same network, subnet and zone
  - Serverless NEG
    - Cloud Run, Cloud Function, App Engine, API Gateway
  - Hybrid connectivity NEG
    - Routing traffic to an on-premises location or another cloud

Backend services overview

Backend services overview
https://cloud.google.com/load-balancing/docs/backend-service

Setting up an Instance Group backend configuration



Instance group backend services
https://cloud.google.com/load-balancing/docs/backend-service#instance_groups

Some backend services require health checks

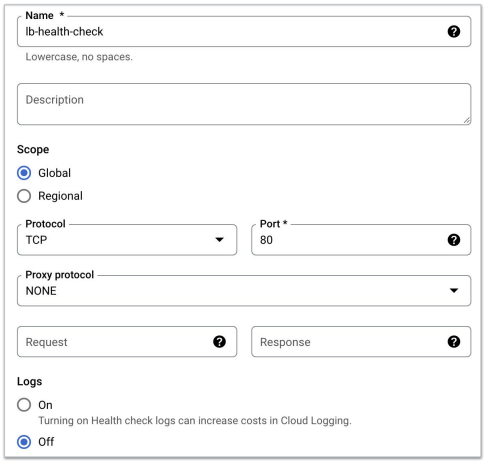
- Instance groups or zonal NEGs backend services must have an associated health check
  - Serverless NEG or an internet NEG backend services must **not** reference a health check
- HTTP(S) Load Balancer uses health checks to determine if a particular backend instance should receive traffic
  - An instance marked as “unhealthy” will not receive traffic
  - Managed Instance Groups also use health checks, but for a different purpose
    - If an instance is reported an unhealthy, it will be removed and replaced with a healthy instance

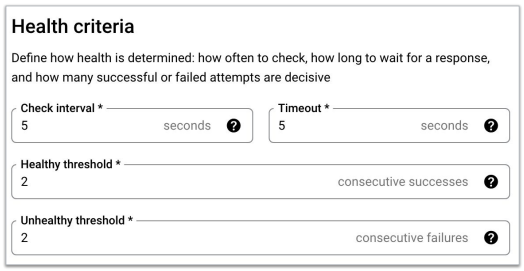
https://cloud.google.com/load-balancing/docs/backend-service#health-checks

Health checks
https://cloud.google.com/load-balancing/docs/backend-service#health-checks

Setting up a Health Check for an Instance Group Backend

- Used by the load balancer to determine whether to send traffic to a particular VM



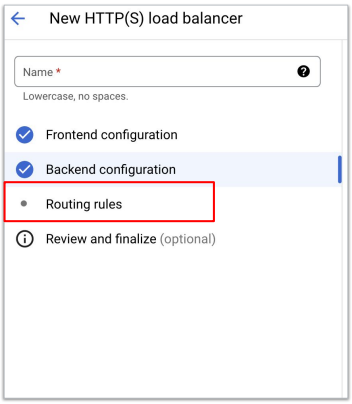


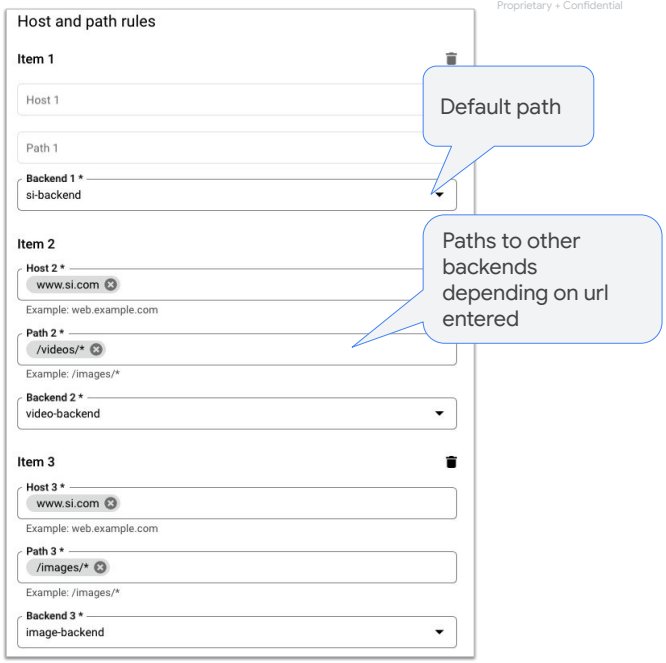

```
gcloud compute health-checks create http
backend-basic-check --port=80 --no-enable-logging
--check-interval=5 --timeout=5
--unhealthy-threshold=3 --healthy-threshold=2
```


Note: The settings shown are for illustrative purposes only. The actual configuration needs to be planned and tested.

Setting up Routing rules

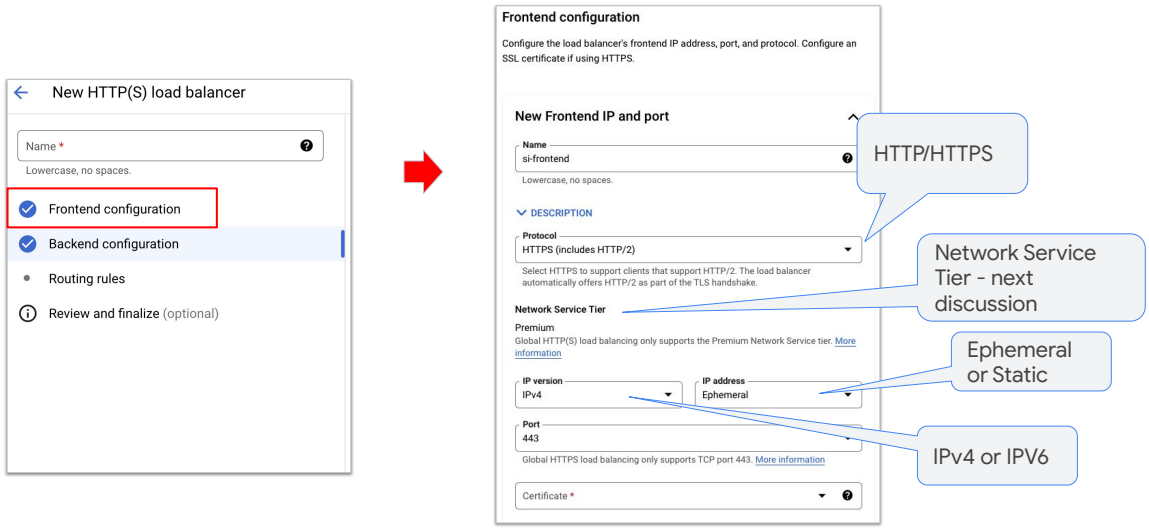
- Paths determines which backend service receives the traffic





URL maps overview
https://cloud.google.com/load-balancing/docs/url-map-concepts

Setting up the Frontend configuration



URL maps overview
https://cloud.google.com/load-balancing/docs/url-map-concepts

Network Service Tiers

Optimize your network for performance or cost

- **Premium**
  - Delivers traffic on Google's premium backbone
  - Optimized for performance, higher cost
  - For services needing global availability
  - Global HTTP(S) load balancing supports this tier only
  - Backed by SLA
- **Standard**
  - Delivers traffic using regular ISP networks
  - Optimized for cost, lower performance
  - For services hosted within a single region
  - No SLA

Premium
Network Tier

Standard
Network Tier

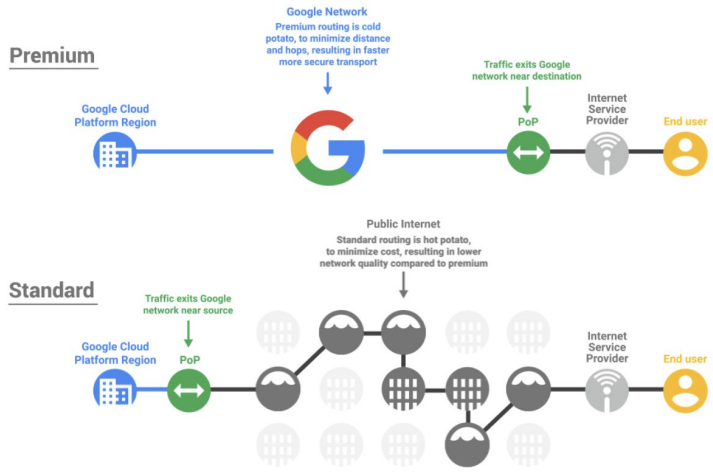
Network Service Tiers

Network Service Tiers
https://cloud.google.com/network-tiers

Network Service Tiers overview
https://cloud.google.com/network-tiers/docs/overview

Google Cloud networking in depth: Understanding Network Service Tiers (May 15, 2019)
https://cloud.google.com/blog/products/networking/google-cloud-networking-in-depth-understanding-network-service-tiers

Network Service Tiers



Premium Tier
https://cloud.google.com/network-tiers/docs/overview#premium_tier

Premium Tier delivers traffic from external systems to Google Cloud resources by using Google's low latency, highly reliable global network. This network consists of an extensive private fiber network with over 100 points of presence (PoPs) around the globe. This network is designed to tolerate multiple failures and disruptions while still delivering traffic.

Premium Tier supports both regional external IP addresses and global external IP addresses for VM instances and load balancers. All global external IP addresses must use Premium Tier. Applications that require high performance and availability, such as those that use HTTP(S), TCP proxy, and SSL proxy load balancers with backends in more than one region, require Premium Tier. Premium Tier is ideal for customers with users in multiple locations worldwide who need the best network performance and reliability.

Standard Tier
https://cloud.google.com/network-tiers/docs/overview#standard_tier

Standard Tier delivers traffic from external systems to Google Cloud resources by routing it over the internet. It leverages the double redundancy of Google's network only up to the point where Google's data center connects to a peering PoP. Packets that leave Google's network are delivered using the public internet and are subject to the reliability of intervening transit providers and ISPs. Standard Tier provides network

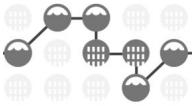
quality and reliability comparable to that of other cloud providers.

Standard Tier is priced lower than Premium Tier because traffic from systems on the internet is routed over transit (ISP) networks before being sent to VMs in your VPC network or regional Cloud Storage buckets. Standard Tier outbound traffic normally exits Google's network from the same region used by the sending VM or Cloud Storage bucket, regardless of its destination. In rare cases, such as during a network event, traffic might not be able to travel out the closest exit and might be sent out another exit, perhaps in another region.

Standard Tier offers a lower-cost alternative for the following use cases:

- You have applications that are not latency or performance sensitive.
- You're deploying VM instances or using Cloud Storage that can all be within a single region.

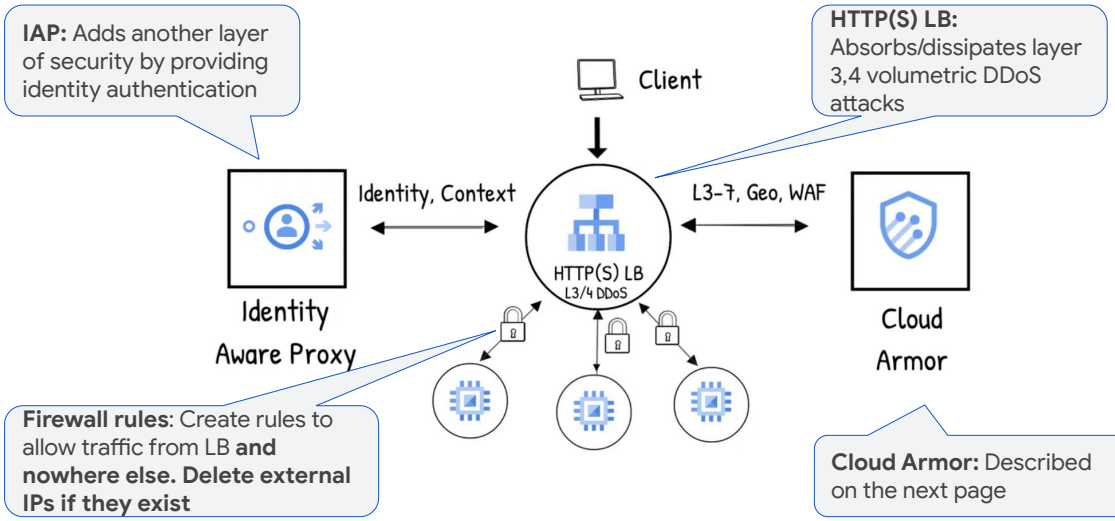
Network Service Tier summary

	Premium	Standard
Choice	 **Global LB**	 **Regional LB**
	Performance-optimized	Cost-optimized
Why?	• Performance, Security, Reliability over Google's global private network • Unique to Google Cloud	• Lower Price (~24–33% cheaper) • Lower performance than Premium but comparable to major public clouds
When?	• Recommended Tier • Default for all workloads	• For cost-sensitive workloads • For performance and outbound costs comparable to other public clouds • For non-critical workloads in single region

Ask yourself whether high performance or lower cost is most important for your workload or resource. The Premium Tier is the clear choice for performance. If cost is your main consideration, remember that the Standard Tier has other restrictions in addition to network performance. If you want to deploy your backends or have users in multiple regions, but don't want to use the public internet over Google's network for inter-continental and cross region traffic, you want to choose the Premium Tier. Also, if you want Global Load Balancing or Cloud CDN, you need to use the Premium Tier.

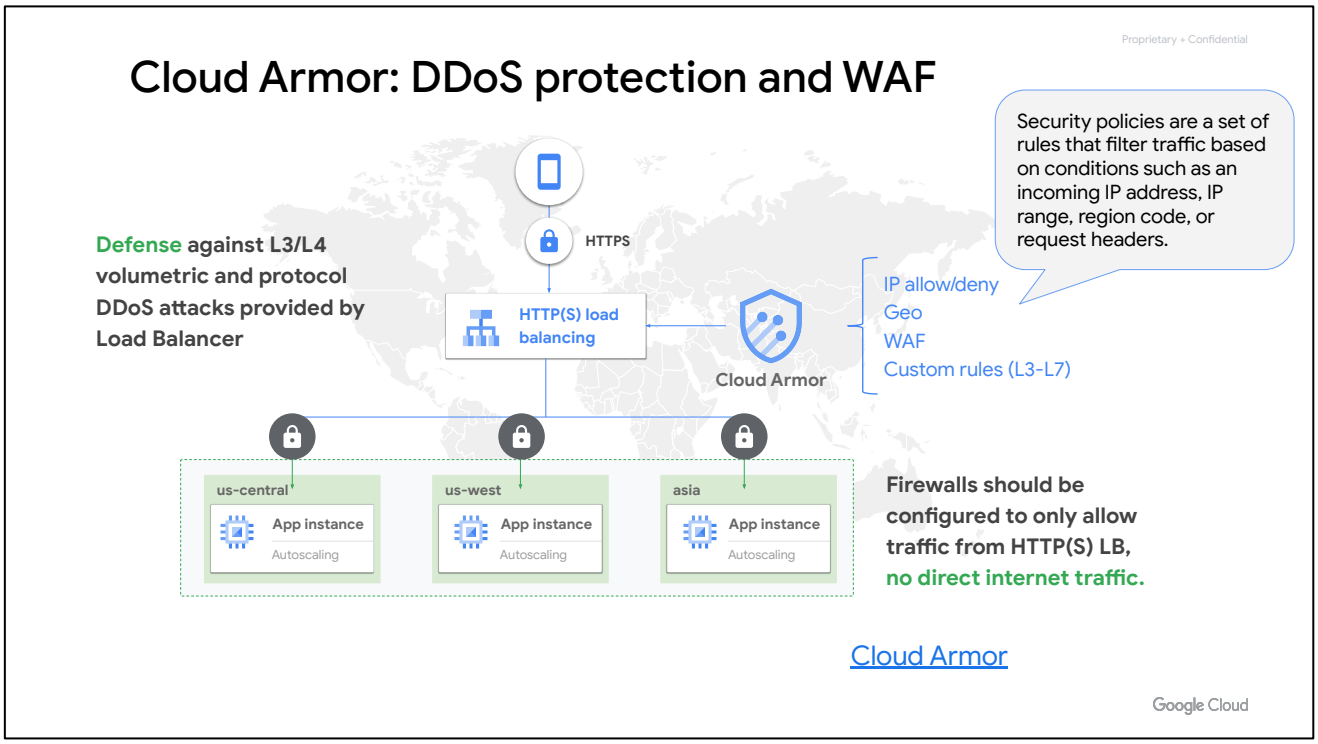
Otherwise, the Standard Tier is a great choice if you don't need any of those services and are okay using the public internet instead of Google's network.

Cloud Load Balancing Attack Protection Options



WAF: Web application firewall

What is a volumetric attack? A volumetric attack sends a high amount of traffic, or request packets, to a targeted network in an effort to overwhelm its bandwidth capabilities. These attacks work to flood the target in the hopes of slowing or stopping their services.



Google Cloud Armor
https://cloud.google.com/armor

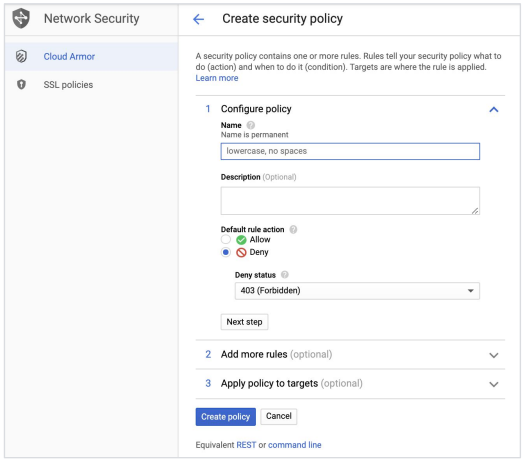
Key points:

- When using **External HTTP(s) LB,** you **benefit** from an unrivaled level of **L3/L4 DDoS protection** from a wide variety of attack types.
- **Cloud Armor**
  - **Managed WAF solution**
  - Runs **distributed over Google's edge**
  - Integrates with **External HTTP(s) Load Balancer**
  - **Provides**
    - **Layer 7 and Application layer** protection
    - **Access controls** to backend services based on **IP and Geo**
    - **Security policies** based on **L3 to L7 request and client attributes**
    - **Request throttling**
    - **Pre-configured** rules (XSS and SQLi based on OWASP Modsecurity)
    - **Real time telemetry** in the form of **Cloud Operations logs** containing Cloud Armor's decisions on a per-request basis, as well as a **monitoring dashboard** that gives granular views of

        - allow, denied, or previewed traffic.
    - **Security policies and IP deny list** are **not supported with Cloud CDN**

Use Google Cloud Armor to create network security policies

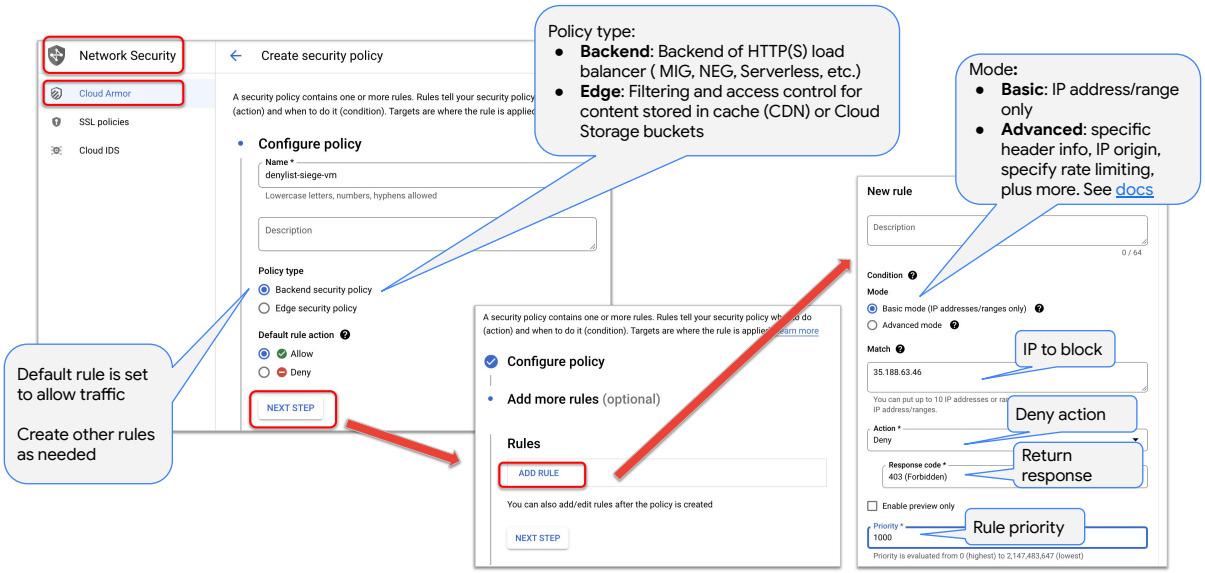
- Can allow or deny access to your Google Cloud resources using IP addresses or ranges.
- Create allow lists to allow known addresses.
- Create deny lists to block known attackers.



For additional features over built-in DDoS, such as IPv4 and IPv6 allow or deny, and defense against application-aware attacks such as cross-site scripting and SQL injection, Google offers Google Cloud Armor, which works in conjunction with global HTTP/HTTPS load balancing and enables you to deploy and customize defenses for your internet-facing applications. It's based on the same technologies and global infrastructure that we use to protect Google services like Search, Gmail, and YouTube.

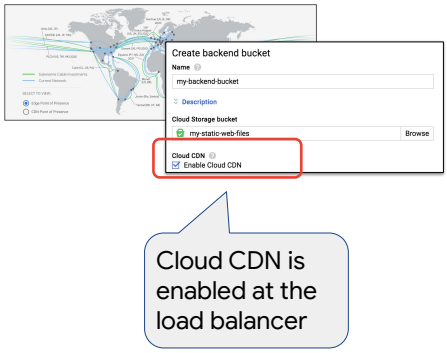
Google Cloud Armor security policies enable the access or denial of HTTP(S) requests to load balancers at the Google Cloud edge as close as possible to the source of incoming traffic. This prevents unwelcome traffic from consuming resources or entering the VPC networks.

Cloud Armor example - Deny access to specific IP



Google Cloud provides two Content Delivery Network (CDN) options

- Cloud CDN caches regularly accessed static content
  - Optimized for serving static web assets such as CSS, JavaScript, or non-dynamic HTML files
- Media CDN is a media delivery platform
  - Optimized for high-throughput egress workloads such as streaming video and large file downloads
- Both use Google's globally distributed Edge locations to **cache content** close to your users
- **170+ locations**, with **single IP** across multiple regions



Choose a CDN product

The load balancer can also be configured to leverage Google's content delivery network, called Cloud CDN.

By enabling Cloud CDN, you can cache content all over the world, closer to the user.

Introducing Media CDN—the modern extensible platform for delivering immersive experiences
https://cloud.google.com/blog/products/networking/introducing-media-cdn/

Choose a CDN product
https://cloud.google.com/media-cdn/docs/choose-cdn-product

Exam Guide - VPC Network

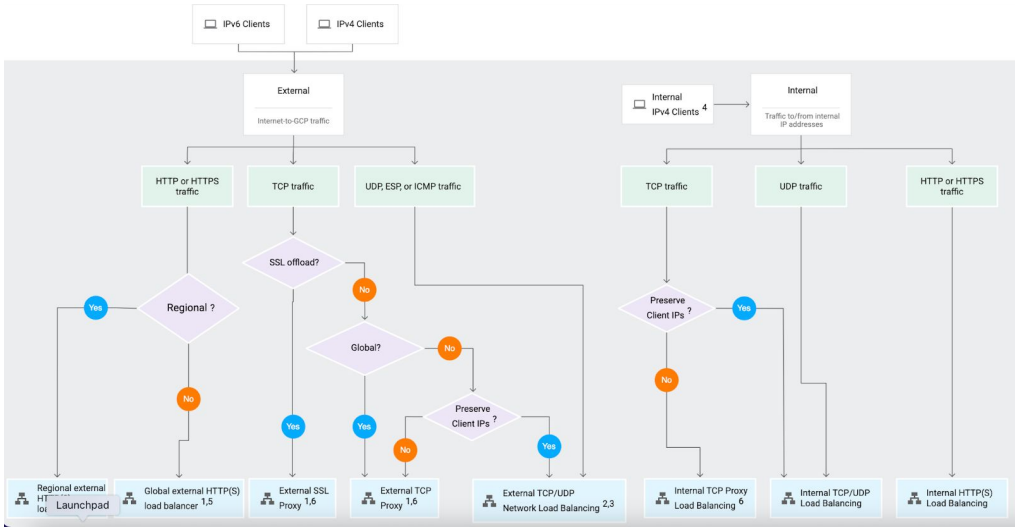
2.4 Planning and configuring network resources. Tasks include:

- 2.4.1 Differentiating load balancing options
- 2.4.2 Identifying resource locations in a network for availability
- 2.4.3 Configuring Cloud DNS

How do you decide on the type of load balancer to use?

- Based on use case
- The right option depends on
  - Whether the traffic is Internal or external
  - The type of traffic, e.g. HTTP(S), TCP, UDP, etc
- Use global load balancing when
  - Backends are distributed across multiple regions
    - Provides a single anycast IP address, and supports IPv6 addresses
- Use regional load balancing when
  - Backends are in single region
  - Only require IPv4 termination
- Refer to Choosing a Load Balancer

Load Balancer Decision Tree



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

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

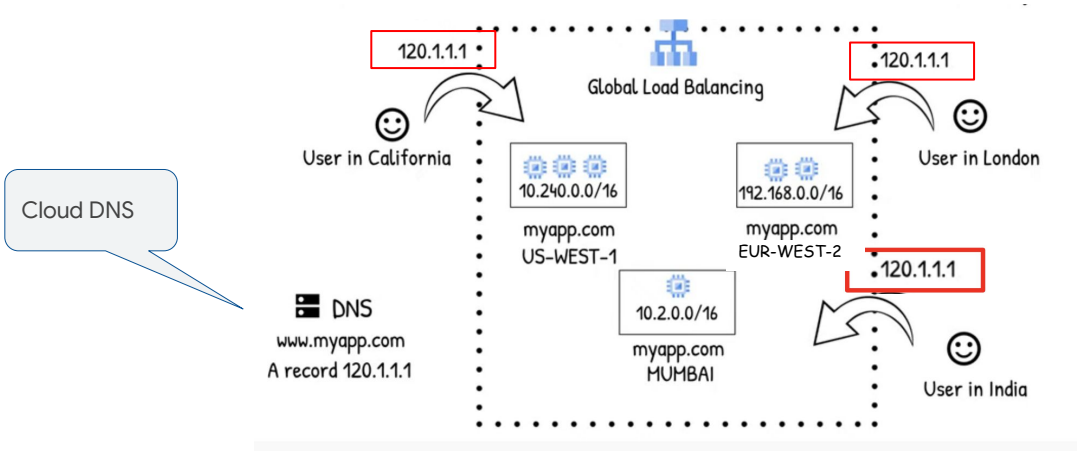
- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with Cloud DNS, CloudNAT, Load Balancers and firewall rules

Exam Guide - VPC Network

2.4 Planning and configuring network resources. Tasks include:

- 2.4.1 Differentiating load balancing options
- 2.4.2 Identifying resource locations in a network for availability
- 2.4.3 Configuring Cloud DNS

Cloud DNS can be used to map Load Balancer IP to domain name



https://cloud.google.com/dns

Cloud DNS

- Managed DNS service that runs on the same infrastructure as Google
- Low latency, high availability, and cost-effective way to make applications and services available to users
- The DNS information you publish is served from redundant locations around the world.
- Cloud DNS is programmable. You can publish and manage millions of DNS zones and records using the Google Cloud console, the command-line interface, or the API.

Cloud DNS explained!:
https://cloud.google.com/blog/topics/developers-practitioners/cloud-dns-explained

Google Cloud offers **Cloud DNS** to help the world find them. It's a managed DNS service that runs 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 Google Cloud console, the command-line interface, or the API.

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

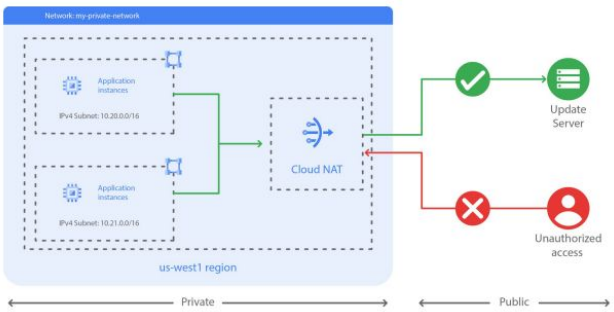
- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

Cloud NAT

- Lets certain resources without external IP addresses create outbound connections to the internet



Cloud NAT overview
https://cloud.google.com/nat/docs/overview

Use with Compute Engine:
https://cloud.google.com/nat/docs/gce-example

Use with GKE:
https://cloud.google.com/nat/docs/gke-example

Suggested Lab: Implement Private Google Access and Cloud NAT:
https://partner.cloudskillsboost.google/course_sessions/1138516/labs/209945

Now, as a general security best practice, I recommend using only assigning internal IP addresses to your VM instances wherever possible. Cloud NAT is Google's managed network address translation service. It lets you provision your application instances without public IP addresses, while also allowing them to access the internet in a controlled and efficient manner. This means your private instances can access the internet for updates, patching, configuration management, and more. In this diagram, Cloud NAT enables two private instances to access an update server on the internet, which is referred to as outbound NAT. However, Cloud NAT does not implement inbound NAT. In other words, hosts outside your VPC network cannot directly access any of the private instances behind the Cloud NAT gateway. This helps you keep your VPC networks isolated and secure.

Exam Guide - VPC Network

3.5 Deploying and implementing networking resources. Tasks include:

- 3.5.1 Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- 3.5.2 Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- 3.5.3 Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- 3.5.4 Creating a VPN between a Google VPC and an external network using Cloud VPN
- 3.5.5 Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

4.5 Managing networking resources. Tasks include:

- 4.5.1 Adding a subnet to an existing VPC
- 4.5.2 Expanding a subnet to have more IP addresses
- 4.5.3 Reserving static external or internal IP addresses
- 4.5.4 Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

Exam Guide - Monitoring and Logging


Cloud Logging


Cloud Monitoring


Cloud Profiler


Cloud Trace


Error Reporting


Cloud Debugger

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Google Cloud operations suite documentation (Includes tutorials):
https://cloud.google.com/stackdriver/docs

How To guides:
https://cloud.google.com/monitoring/docs/how-to

- **Cloud Operations** is Google Cloud's **fully managed native logging and monitoring tool**
- **Monitoring**
  - Default monitoring for all resources, with plugins for many third party tools
  - Uptime checks for groups or specific resources
- **Logging**
  - **All logs are captured in Cloud Logging**
  - Search and filter
  - Derive metrics from logs, use to create dashboard or autoscale instances
  - Export for retention and leveraging other tools
- **Performance** - Highly scalable and performant
- **Multi-cloud** - Support Google Cloud and AWS. Can monitor on-premises using a partner solution

Cloud Monitoring dashboards can visualize utilization and network traffic

- Collects metrics, events, and metadata from
  - Google Cloud,
  - Amazon Web Services (AWS)
  - Application instrumentation
- Generates insights via dashboards, charts, and alerts
- Applications can generate custom metrics

Cloud Monitoring

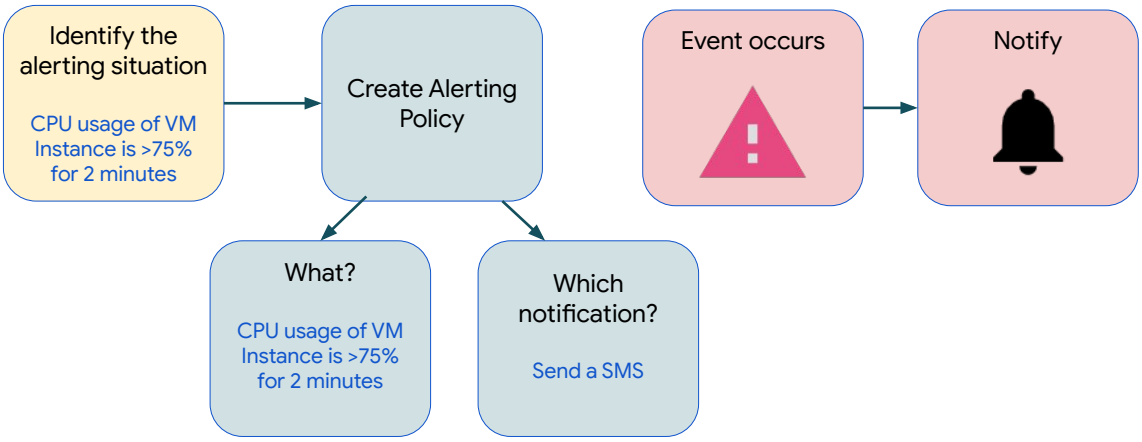
Cloud Monitoring
https://cloud.google.com/monitoring

Cloud Monitoring allows you to create custom dashboards that contain charts of the metrics that you want to monitor. For example, you can create charts that display your instances' CPU utilization, the packets or bytes sent and received by those instances, and the packets or bytes dropped by the firewall of those instances.

In other words, charts provide visibility into the utilization and network traffic of your VM instances, as shown on this slide. These charts can be customized with filters to remove noise, groups to reduce the number of time series, and aggregates to group multiple time series together.

For a full list of supported metrics, please refer to the documentation:
https://cloud.google.com/monitoring/api/metrics_gcp

Creating a Cloud Monitoring Alert



Introduction to alerting:
https://cloud.google.com/monitoring/alerts

How to add an alerting policy:
https://cloud.google.com/monitoring/alerts#types-of-policies

Create metric-based alert policies
https://cloud.google.com/monitoring/alerts/using-alerting-ui

Step 1: Choose a metric for which you want to receive an alert
Step 2: Create a Alert Policy. Specify the specific metric over what time period, and specify how you want to receive the alert (email, sms, etc.)

When the condition occurs, the notification will be sent

Alerting policies can notify you of certain conditions

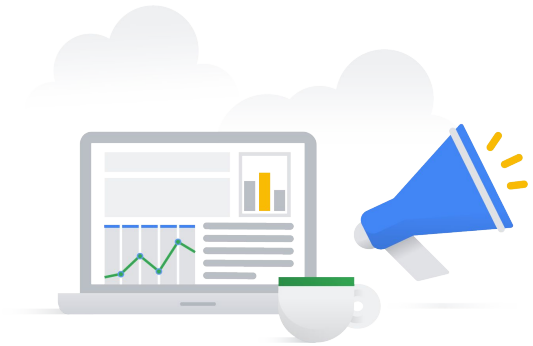
Now, although charts are extremely useful, they can only provide insight while someone is looking at them. But what if your server goes down in the middle of the night or over the weekend? Do you expect someone to always look at dashboards to determine whether your servers are available or have enough capacity or bandwidth? If not, you want to create alerting policies that notify you when specific conditions are met.

For example, as shown on this slide, you can create an alerting policy when the network egress of your VM instance goes above a certain threshold for a specific timeframe. When this condition is met, you or someone else can be automatically notified through email, SMS, or other channels in order to troubleshoot this issue.

Creating Alerting Policies

Create notifications in response to a policy that exceeds some condition

1. Create a condition that determines when some metric exceeds some value for a specified period of time.
2. Specify a notification – can be a text, email, webhook, Google Cloud Mobile app, Slack, Pub/Sub and others.
3. Can add additional documentation to the notification.
4. Name the policy.



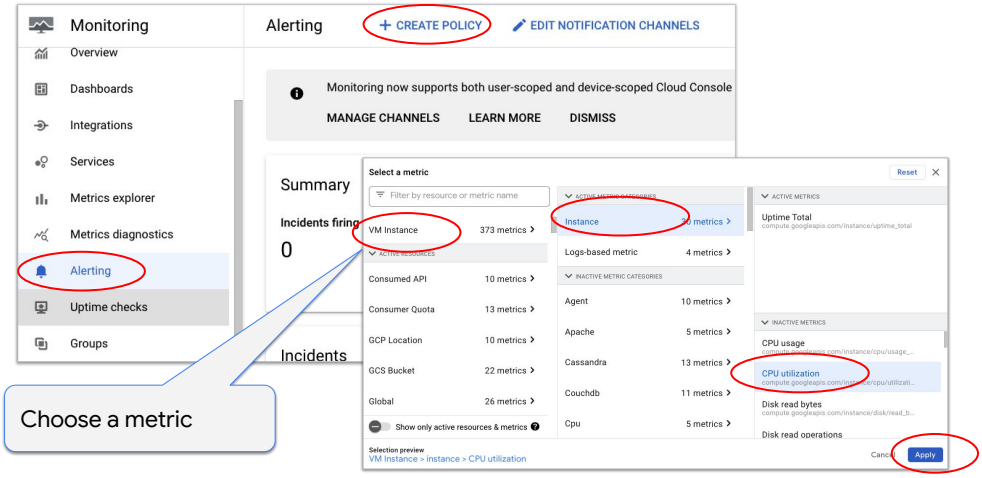
How to add an alerting policy:
https://cloud.google.com/monitoring/alerts#types-of-policies

An alerting policy allows you to respond to anomalies in your system. For example, if your system starts to get an unusually high number of requests, you might want to be notified of a potential denial of service attack. Or, if a VM is not working, you can have the system recreate it.

To create an alert, you define a condition that determines if some metric is above or below some value for some period of time.

When the condition is met, you respond with a notification. Notifications include emails, texts, and webhooks. Use an email or text to notify a human. Use a webhook to run a program in response to the anomaly.

Cloud Monitoring Alerts - Console and CLI




```
gcloud alpha monitoring policies create
--policy-from-file="rising-cpu-usage.json"
```


Cloud Monitoring
https://cloud.google.com/monitoring

Create metric-based alert policies
https://cloud.google.com/monitoring/alerts/using-alerting-ui

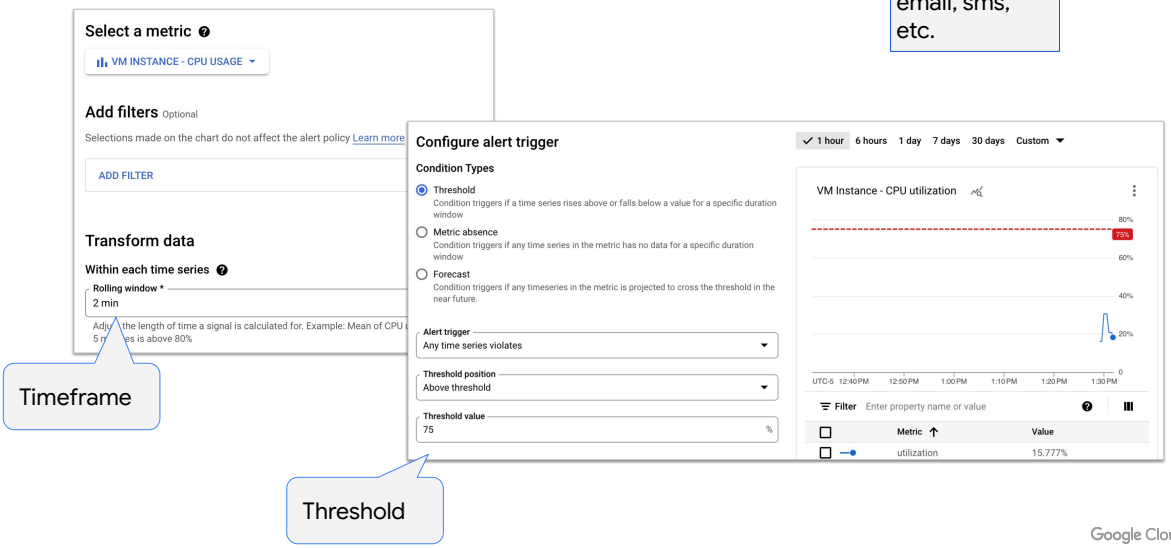
Cloud Monitoring allows you to create custom dashboards that contain charts of the metrics that you want to monitor. For example, you can create charts that display your instances' CPU utilization, the packets or bytes sent and received by those instances, and the packets or bytes dropped by the firewall of those instances.

In other words, charts provide visibility into the utilization and network traffic of your VM instances, as shown on this slide. These charts can be customized with filters to remove noise, groups to reduce the number of time series, and aggregates to group multiple time series together.

For a full list of supported metrics, please refer to the documentation:
https://cloud.google.com/monitoring/api/metrics_gcp

Cloud Monitoring Alerts (continued)

Not shown is selecting the alert type, e.g., email, sms, etc.



Timeframe

Threshold

Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Custom metrics

- Built-in metrics can provide information on backend latency or disk usage, for example
- Custom metrics (application-specific metrics) can provide information built-in metrics cannot
  - For example, the count of number of users logged into the application
- Custom metrics are captured by using an API
  - Google recommends open-source OpenCensus for metric selection
    - Provides a way to create custom metrics, add data to those metrics, and export the metrics to Cloud Monitoring
- Can also create metrics based on the content of log entries
  - For example, number of log entries containing a particular message

Custom metrics
https://cloud.google.com/monitoring/custom-metrics

OpenCensus:
https://opencensus.io/

Create custom metrics with OpenCensus:
https://cloud.google.com/monitoring/custom-metrics/open-census

Create custom metrics with the API:
https://cloud.google.com/monitoring/custom-metrics/creating-metrics

Log-based metrics (Youtube):
https://www.youtube.com/watch?v=YiwT1cxpRDQ

Log based metrics:
https://cloud.google.com/logging/docs/logs-based-metrics

Suggested Lab: Reporting Application Metrics into Cloud Monitoring (if time allows)

- Uses OpenCensus to record metrics
- Discusses other instrumentation methods

https://partner.cloudskillsboost.google/catalog_lab/1047

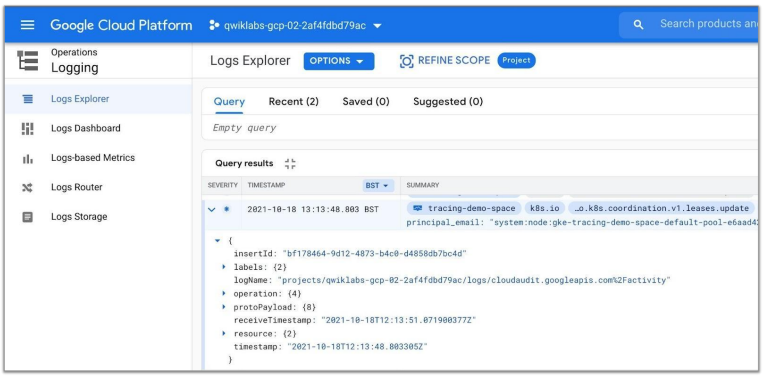
Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Cloud Logging collects logs from admin, system and application activity

- Data is available from over 150 common application components, on-premises systems, and hybrid cloud systems
- Store, search, analyze, monitor, and alert on logging data from Google Cloud and Amazon Web Services



Cloud Logging pricing for Cloud Admins: How to approach it & save cost

Cloud Logging pricing for Cloud Admins: How to approach it & save cost
https://cloud.google.com/blog/topics/cost-management/how-to-approach-cloud-logging-pricing-for-cloud-admins

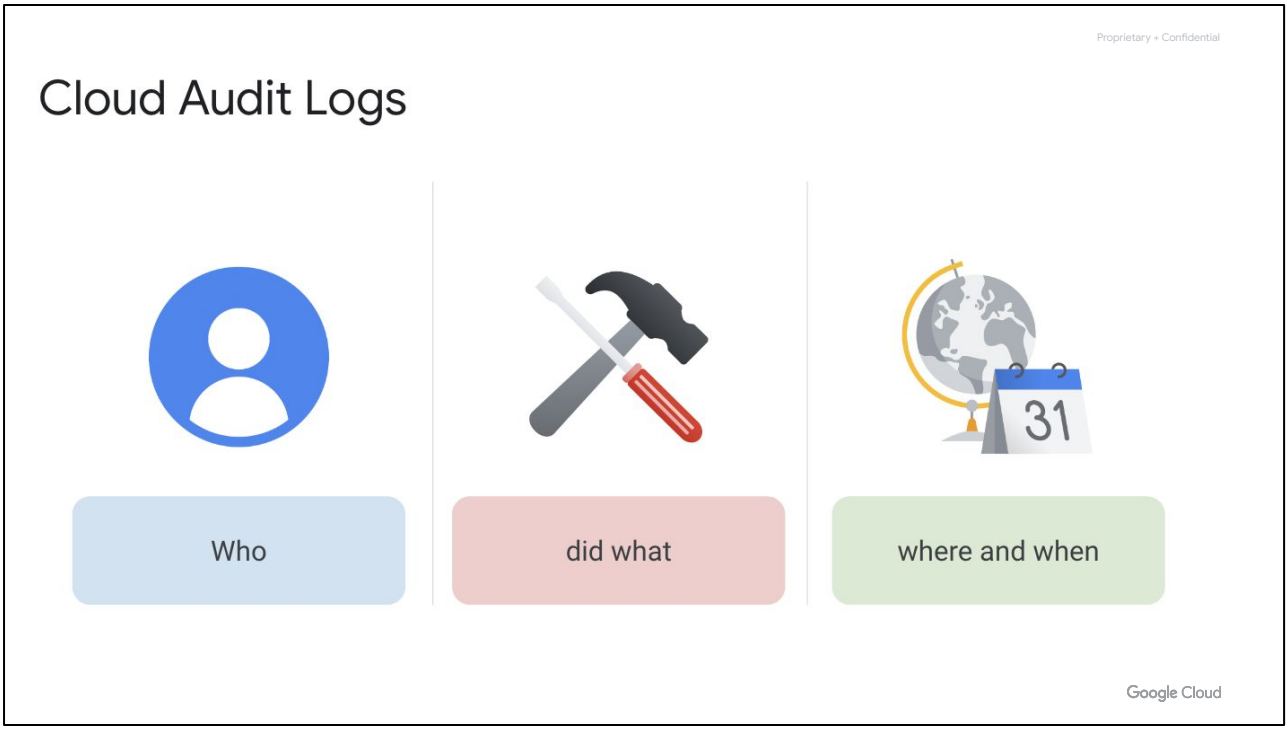
Logs Explorer features summary:
https://cloud.google.com/logging/docs/view/logs-explorer-summary

Cloud Logging
https://cloud.google.com/logging/docs

Log data compliance:
https://services.google.com/fh/files/misc/whitepaper_data_governance_logs_how_to.pdf

Using the API to view logs:
https://cloud.google.com/logging/docs/reference/v2/rest/v2/entries/list

Using the CLI to view logs:
https://cloud.google.com/logging/docs/reference/tools/gcloud-logging



Resource Manager audit logging information
https://cloud.google.com/resource-manager/docs/audit-logging

Cloud Audit Logs overview
https://cloud.google.com/logging/docs/audit

Google Cloud services with audit logs
https://cloud.google.com/logging/docs/audit/services

Cloud Audit Logs records Google Cloud account activity, including actions performed with the Google Cloud Console, command line tools, APIs, and AWS services.

Cloud Audit Logs helps you answer the questions of: Who did what, where, and when within your Google Cloud projects.

Several audit logs exist

Log	Default setting	Description	Example
Admin Activity	Always enabled	Record admin actions that modify resource configuration or metadata	IAM grants, VM creation/deletion
System Events	Always enabled	Compute Engine system events	Live migration
Data Access	Off by default; Enabled per service	Activities that create, modify or read user-provided data	In Cloud SQL: listing databases names, creating users, creating backups, etc.
Policy Denied	Always enabled, but can exclude these logs from ingestion via filters	Recorded when a principle is denied access due to a security policy violation	Someone tries to create a Cloud Storage bucket in a project in which that role is denied

Cloud Audit Logs maintains four audit logs for each project, folder, and organization: Admin Activity logs, System Events, Data Access logs and Policy Denied logs.

Admin Activity audit logs contain log entries for API calls or other administrative actions that modify the configuration or metadata of resources. For example, the logs record when VM instances and App Engine applications are created and when permissions are changed.

Admin Activity audit logs are always enabled by default and there is no charge for your Admin Activity audit logs.

System Event audit logs contain log entries for when Compute Engine performs a system event. For example, each live migration is recorded as a system event. System Event audit logs are always enabled and there is no charge for your System Event audit logs.

Data Access audit logs record API calls that create, modify, or read user-provided data. These logs are disabled by default because they can be quite large. Enabling the logs might result in your project being charged for the additional logs usage.

Note that Data Access audit logs do not record data-access operations on resources that are publicly shared.

Log retention varies by log type

Log	Default retention period	Retention period modifiable?	Chargeable?
Admin Activity	400 days	No	No
System Events	400 days	No	No
Data Access	30 days	Yes	Yes*
Policy Denied	30 days	Yes	Yes*

*Can eliminate unwanted entries via filtering. Can also route data elsewhere via sinks.

Types of audit logs
https://cloud.google.com/logging/docs/audit#types

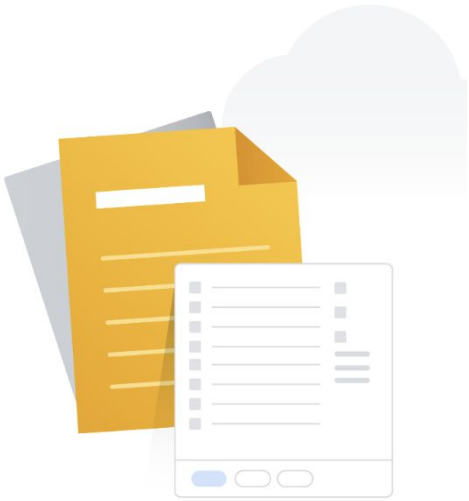
Best practices for Cloud Audit Logs
https://cloud.google.com/logging/docs/audit/best-practices

Individual audit log entries are kept for a specified length of time and are then deleted. Admin activity and system events are kept for 400 days, while data access logs are kept for 30 days by default, but the duration can be changed. For longer retention, export audit log entries from Cloud Logging and keep them for as long as you wish.

Viewing audit logs

Log can be viewed in:

- Project's Activity page
- Cloud Logging
- Cloud Logging API
- Cloud SDK



You can view audit log entries in your project's Activity page, in the Logs Explorer, in the Cloud Logging API, and in the Cloud SDK. You can also export audit log entries to Cloud Logging, Pub/Sub, or Cloud Storage.

To view the logs, you must have the IAM roles Logging/Logs Viewer for Admin Activity logs and Logging/Private Logs Viewer for Data Access logs. For more information on Cloud Logging roles, see Access Control.

Analyzing Audit Logs

- Advanced logs filters
- BigQuery
- Third-party analysis tools

When exporting logs, log filters can be defined to limit which log events are exported. This can help reduce the amount of data exported and reduce the scope of the data.

When you export to BigQuery datasets, Cloud Logging creates dated tables to hold the exported log entries. The data can then be analyzed using BigQuery.

Log files can also be exported to Pub/Sub and to storage buckets, which facilitates analyzing the data with third-party tools.

Monitoring & Logging Agents for Compute Engine

- Automatically stream logs and metrics from third-party applications, system logs, and your code to the Operations logs and Monitoring
- Install on VMs running Linux or Windows
  - AWS EC2 VMs also supported
  - Not required for App Engine or Kubernetes clusters

Ops Agent Overview

Monitoring and Logging agents
Legacy method: Installing the Cloud Logging agent on individual VMs:
https://cloud.google.com/logging/docs/agent/logging/installation

Legacy method: Installing the Cloud Monitoring agent on individual VMs:
https://cloud.google.com/monitoring/agent/monitoring/installation

Currently recommended method: Ops Agent overview:
https://cloud.google.com/stackdriver/docs/solutions/agents/ops-agent

If you're using Kubernetes, Cloud Functions, or App Engine to deploy your applications, then all messages sent to standard out end up in the Operations logs automatically. When using virtual machines though, you will need to add the Logging Agent when you create the VM.

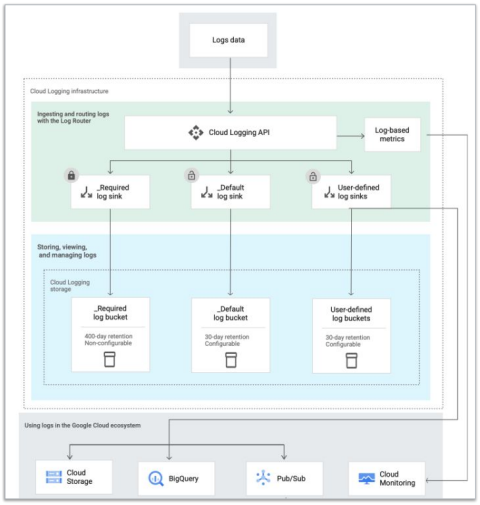
Once the Logging Agent is added, any messages you log from your application will go to Operations. Also, the logs for third-party applications like Apache web server will be aggregated into the Operations logs.

The Logging Agent can also be used on VMs running in AWS EC2.

Just add a couple lines of code to your startup script.

See the documentation for details.

High level look at log routing



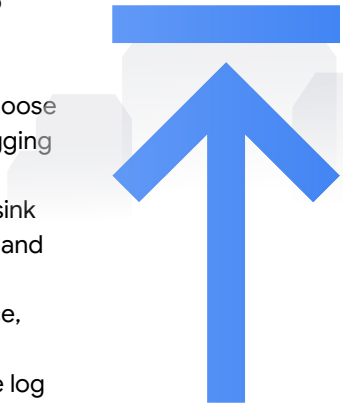
Routing and storage overview

Routing and storage overview:
https://cloud.google.com/logging/docs/routing/overview

Log Sinks

Provides a method to retain logs for longer periods of time or to stream logs to other applications.

- To export specific logs write a filter that selects them, and choose a destination in Cloud Storage, BigQuery, Pub/Sub, Cloud Logging or Splunk
  - The filter and destination are held in an object called a sink
    - Can be created at the organisation, folder, project, and billing account level
- A sink can only export logs that belong to its parent resource, e.g. project/folder.
  - When a log comes in that matches a filter, a copy of the log is written to the export destination.



Sinks

Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
https://cloud.google.com/logging/docs/routing/overview

Scenarios for exporting Cloud Logging data - Splunk
https://cloud.google.com/architecture/exporting-stackdriver-logging-for-splunk

Log sinks:
https://cloud.google.com/logging/docs/export/configure_export_v2

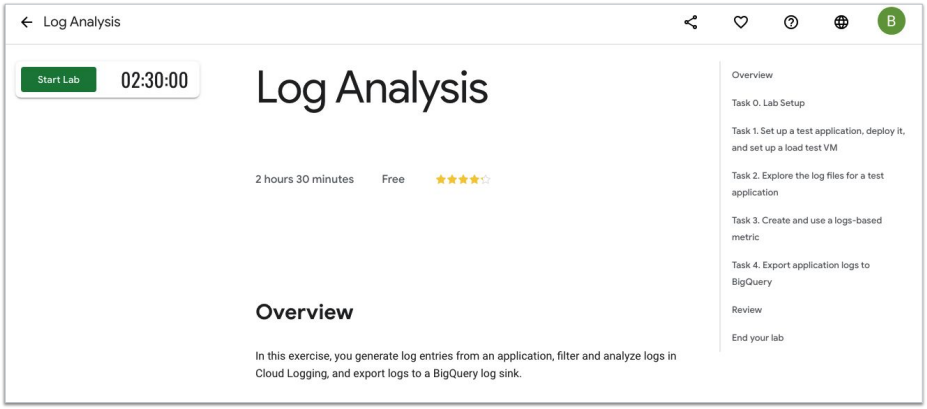
Configure and manage sinks:
https://cloud.google.com/logging/docs/export/configure_export_v2

Configure aggregated sinks:
https://cloud.google.com/logging/docs/export/aggregated_sinks

View logs in sink destinations:
https://cloud.google.com/logging/docs/export/using_exported_logs

BigQuery schema for logs:
https://cloud.google.com/logging/docs/export/bigquery

Suggested Lab: Log Analytics (if time allows)



https://partner.cloudskillsboost.google/catalog_lab/2649

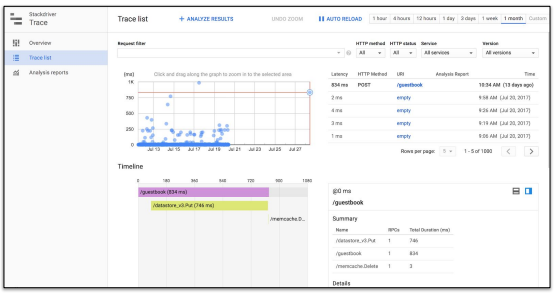
Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Cloud Trace

- Collects latency data from applications
- Useful for finding performance bottlenecks and detecting issues in near-real time
- Displays requests along with their timings
- Provides information on
  - How long it takes applications to handle incoming requests from users or other applications
  - How long it takes to complete operations like RPC calls performed when handling the requests



Viewing trace details

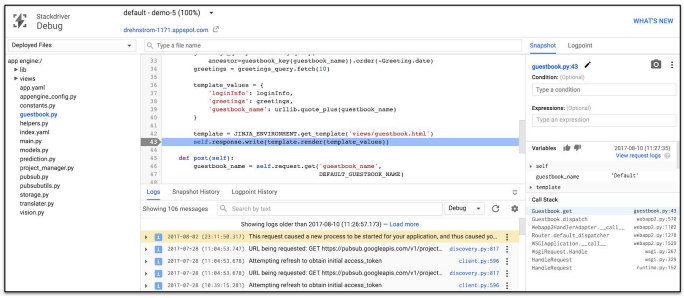
Cloud Trace
https://cloud.google.com/trace

Viewing trace details:
https://cloud.google.com/trace/docs/viewing-details

Setting up Cloud Trace:
https://cloud.google.com/trace/docs/setup

Cloud Debugger*

- Watchpoints can be added to code of deployed applications
- Creates a snapshot of the running app when a breakpoint is hit
  - Snapshot includes variable contents and the call stack
- Can also inject debug logpoints into a running application



Depreciated. Suggested replacement is Snapshot Debugger

Product overview:
https://cloud.google.com/debugger/docs

Snapshots:
https://cloud.google.com/debugger/docs/using/snapshots#snapshot

Logpoints:
https://cloud.google.com/debugger/docs/using/logpoints

Setting up Cloud Debugger:
https://cloud.google.com/debugger/docs/setup

Sometimes a program works fine on the developer's machine and the test server, but fails when you put it into production. This can be a hard problem to track down and debugging the code on your own machine won't help because it works on your own machine.

Google Cloud includes a cloud-based debugger. You can set log points in your code When your app hits a log point, a snapshot of it will be created. You can then inspect variable values and things to help track down the bug.

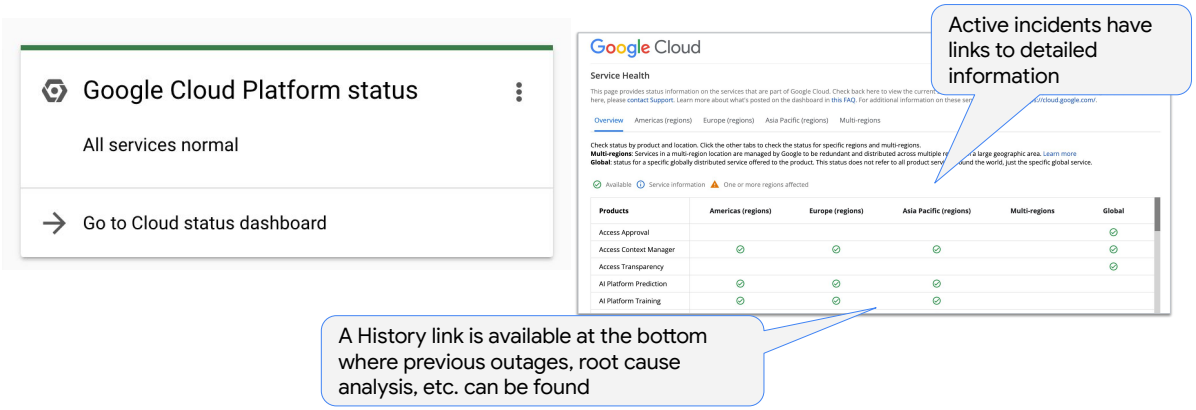
Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status

Viewing Google Cloud Platform status

- Status information can be found in the dashboard of every project
  - Click “Go to Cloud status dashboard” to receive a detailed view



Service Health:
https://status.cloud.google.com/

Incidents and the Google Cloud Service Health Dashboard:
https://cloud.google.com/support/docs/dashboard

Exam Guide - Monitoring and Logging

4.6 Monitoring and logging. Tasks include:

- 4.6.1 Creating Cloud Monitoring alerts based on resource metrics
- 4.6.2 Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- 4.6.3 Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- 4.6.4 Configuring log routers
- 4.6.5 Viewing and filtering logs in Cloud Logging
- 4.6.6 Viewing specific log message details in Cloud Logging
- 4.6.7 Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- 4.6.8 Viewing Google Cloud status


Google Cloud