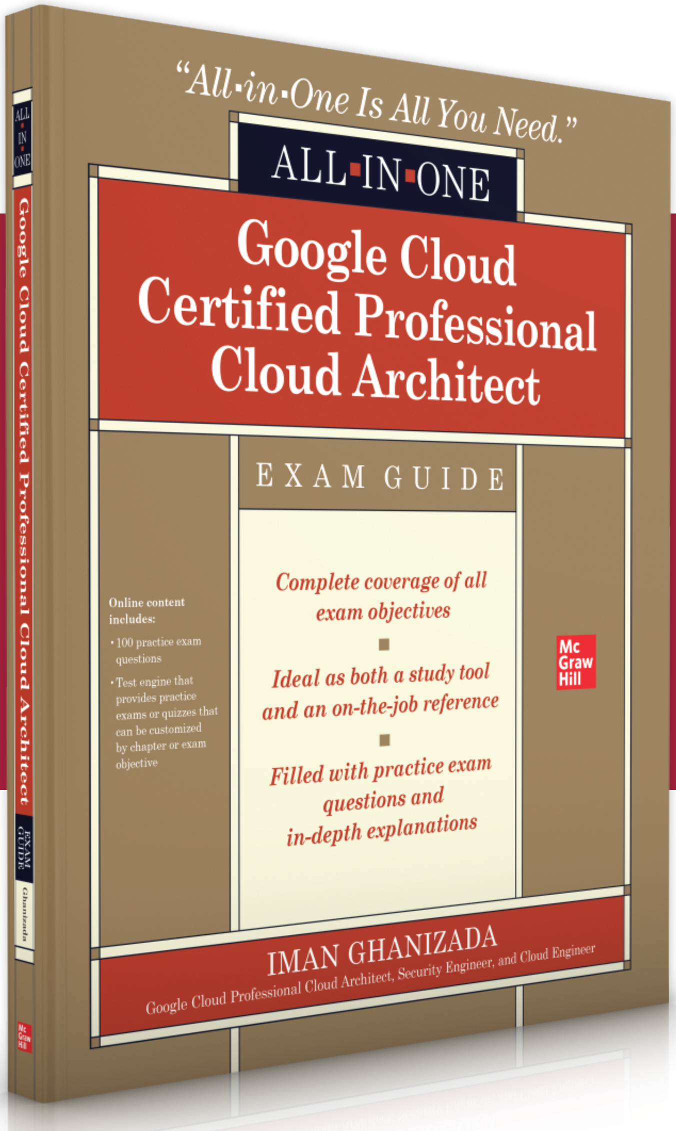
ALL-IN-ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE


"All-in-One Is All You Need."
ALL-IN-ONE
Google Cloud Certified Professional Cloud Architect
EXAM GUIDE
Complete coverage of all exam objectives
Ideal as both a study tool and an on-the-job reference
Filled with practice exam questions and in-depth explanations
Online content includes:
• 100 practice exam questions
• Test engine that provides practice exams or quizzes that can be customized by chapter or exam objective
IMAN GHANIZADA
Google Cloud Professional Cloud Architect, Security Engineer, and Cloud Engineer

Sample Chapter

CHAPTER 6:

Networking

LEARN MORE

BUY NOW

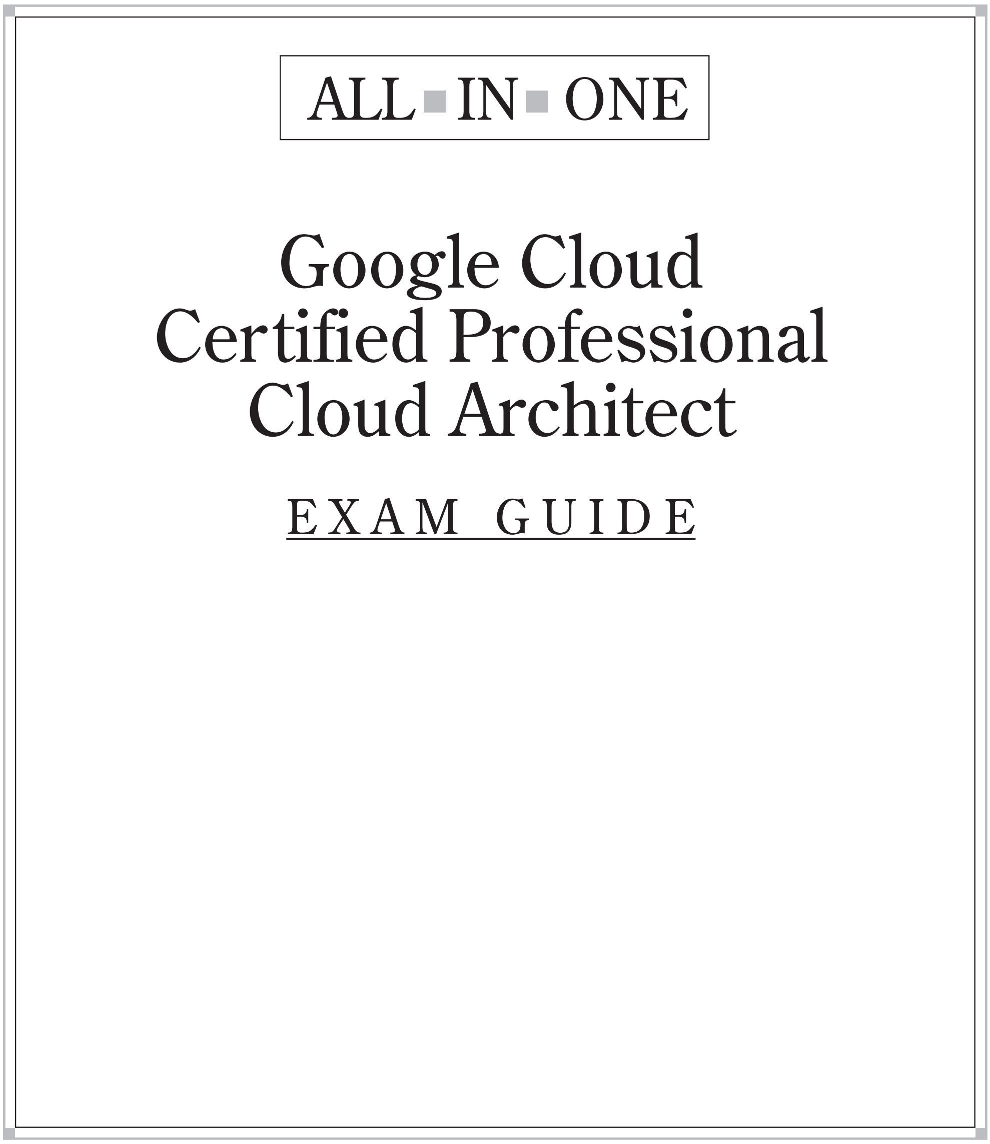
Because learning changes everything.®

mhprofessional.com

ALL■IN■ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE



ALL■IN■ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE

LEARN MORE

BUY NOW

ABOUT THE AUTHOR

Iman Ghanizada is a founder, author, and cloud computing authority residing in Los Angeles. At 28, he's an accomplished young technology leader, providing executive vision and strategy around industry-wide security challenges as a Security Solutions Manager at Google Cloud. Previously, he helped Fortune 50 global business executives transform their organizations securely in the cloud at Google Cloud, Capital One, and more. He has 10+ years of experience in the cloud and holds 14 technical certifications, including the Google Cloud Professional Cloud Architect, Professional Security Engineer, CISSP, and several more.

Iman is the founder of TheCertsGuy.com—a blog site intended to provide technologists easy insight into achieving certificates and growing in their careers. He is a proud Hokie, holding a B.S. in Business Information Technology at Virginia Tech.

As a first-generation Afghan-American, Iman seeks to amplify and accelerate the growth of underrepresented communities in their professional lives. He strongly believes that helping others grow and provide for their families is the ultimate form of fulfillment.

He is also an avid gamer and, in his own words, if he were to hit the jackpot and retire, he'd probably start retirement by drinking energy drinks, eating donuts and pizza, and playing all the games he wishes he had time for now.

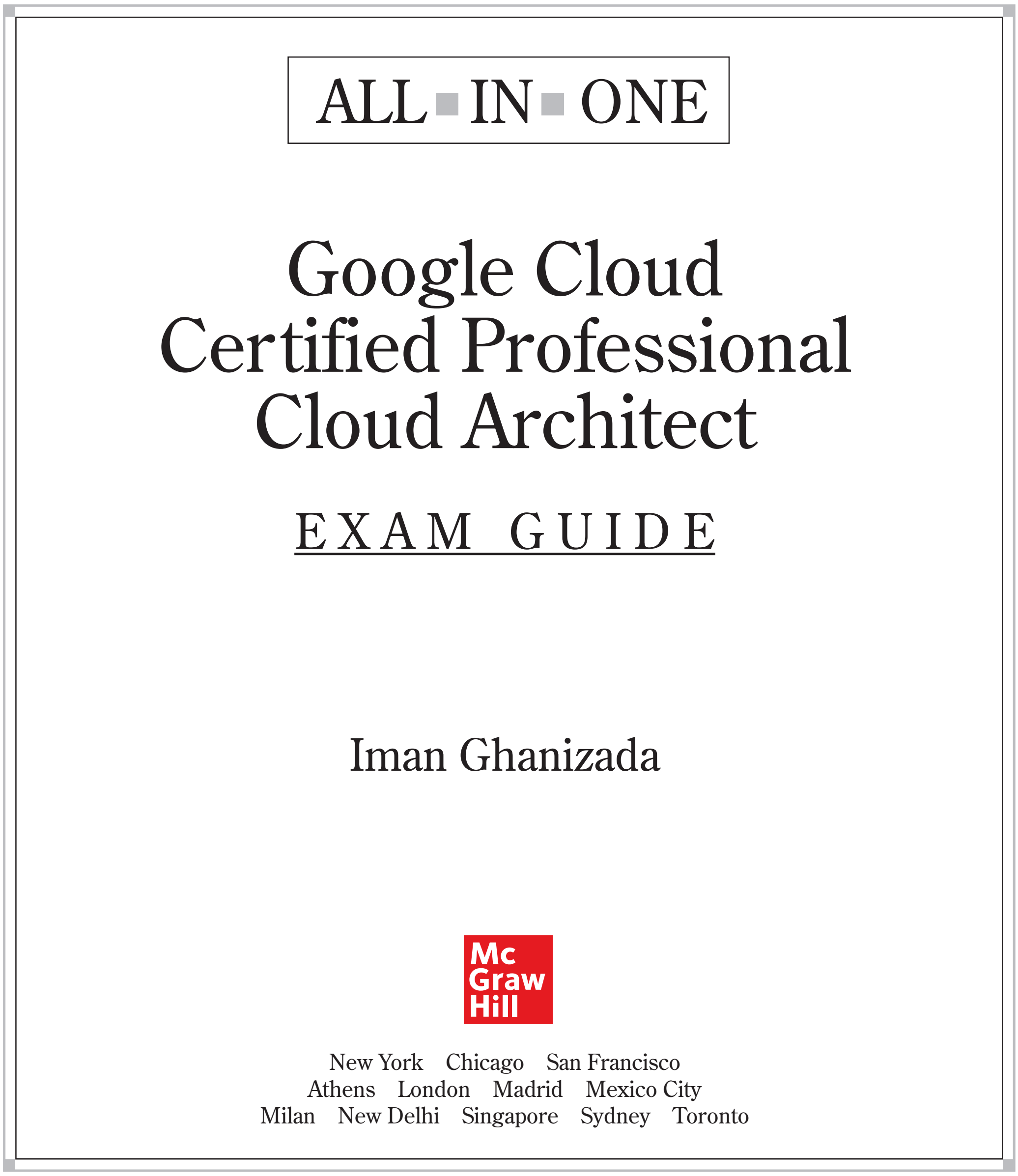
About the Technical Editor

Richard Foltak is VP, Head of Cloud for Dito (ditoweb.com, a Google Premier Partner). Richard focuses on enriching Dito clients' business value streams in embracing and optimizing leading cloud technologies within their practices. Richard holds a Bachelor of Engineering degree and an MBA, along with numerous industry certifications, including those in Infrastructure Architecture, Data Engineering, Data Analytics, Machine Learning, DevOps, Networking, Cyber Security, IT Governance, and ITIL 4. His professional background includes being chief architect at Deloitte Consulting, distinguished architect at Verizon Data, and senior tech leader at Cisco Systems.

ALL-IN-ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE



ALL ■ IN ■ ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE

Iman Ghanizada

New York Chicago San Francisco
Athens London Madrid Mexico City
Milan New Delhi Singapore Sydney Toronto

McGraw Hill is an independent entity from Google Cloud and is not affiliated with Google Cloud in any manner. This publication may be used in assisting students to prepare for the Google Cloud Associate Cloud Engineer Exam. Neither Google Cloud nor McGraw Hill warrants that use of this publication will ensure passing any exam.

LEARN MORE

BUY NOW

Because learning changes everything.®

mhprofessional.com

CHAPTER 6

Networking

In this chapter, we'll cover

- How network constructs are built in Google Cloud Platform
- The various options for connecting to your cloud, including where and when they should be used
- Best security principles to defend your network against internal and external attackers
- A story about networks in a post-apocalyptic world

Evidence shows that the earliest human beings existed more 3 million years ago. These human beings had no concept of a network. Evolution had not enough time to grow and develop their brains to think about things like language and connections, and they were unable to think beyond their instincts for staying alive—find food and eat food or you'll die. About 2.5 million years ago, the first networks were formed as evolution began to progress, as humans realized that having other people around would be more resourceful and could help increase the probability of living another day. It took 2.5 million years to get to where we are today, after humans iteratively improved their capabilities by developing more complex thought patterns, understanding the necessities of life, learning how to communicate, developing technology, and then starting to learn empathy.

It has been a fascinating journey. Networking has enabled humans to progress in virtually every single aspect of our human development. It's easy to see how important it is to be resourceful and build connections in our lives. The "do it all alone" mind-set doesn't scale—not in personal development or from a technology standpoint.

As with human history, in the early history of computing, there was no such thing as a network. It wasn't until 1969 that the first form of computer-to-computer connection was born with the creation of ARPANET, which enabled the successful transmission of the word "login" from one computer to another. ARPANET was the precursor to the Internet. By enabling computers to communicate with one another, we unleashed a major evolution in technology. This technology has since been scaled exponentially by leveraging as many resources (or computers) as these networks could support to share data and iterate.

The Internet, in my opinion, is the single most impactful human invention in the past 3-plus–million years. With the advent of the Internet, we took networking from being confined to knowledge and data that could be mined in a local vicinity, to sharing and iterating data and knowledge collaboratively with billions of individuals and who-knows-how-many machines around the world, and even beyond. Think about life before the Internet. How did you source your knowledge? From the radio? From TV? From the nearest library? Internet networking has enabled you to connect with people all over the world and learn about their cultures, their beliefs, and their research. And it's networking that enables you to watch memes endlessly on Instagram, sharing them directly with Grandma at the click of a button.

Networking has evolved so much since the first network in 1969. In traditional computing, we used the castle-and-moat philosophy of protecting our organizations. Today in the cloud, we can rapidly deploy and delete an endless amount of castles and moats on demand, connected as necessary and when necessary, to spread resources for efficiency and to diversify risk. Safe perimeters can be logically extended across the world to provide instantaneous communications and resourcefulness across all of our castles. If we are even more advanced, we can follow a zero trust philosophy that assumes that nobody in the castle is a safe actor. This new paradigm gives us the power to leverage resources at massive scale (infinite scale is not correct, however, because there are only so many data centers that exist or will exist in the world) to solve the most complex of problems.

Be forewarned that this is a long and dense chapter. I recommend loading YouTube as you read so you can watch some videos if you need clarification on some topics—visual knowledge can be much more attainable at times. We're going to dive into all things networking.

Take off your on-premises hats and don't try to apply every traditional concept here. With the cloud, networking can operate in an entirely new and fundamental way that differs from traditional environments. Finally, security is still a feature that needs to be baked into every single design consideration. As you ponder how you're going to design your network in the cloud, wear an attacker's hat and think about how you can break into this architecture. That will help you right-size security patterns and controls into the way you design your architecture.

Networking Deep Dive

Networking in Google Cloud, or in the cloud in general, operates from a fundamentally different point of view than networking in traditional environments. Amazon Web Services (AWS) has been in the cloud space since around 2008 and has commanded a large market share because of its presence and timing on the market. Google's massive global infrastructure has been in place to serve content to billions of users worldwide. This move to the cloud was a no-brainer for Google, because all they had to do was externalize their infrastructure. Think about Google's massive global content delivery network that serves billions of users around the world with YouTube, Google Search, and Gmail; sharing this already established infrastructure to the world makes perfect sense.

Google's Global Network

Google Cloud's worldwide network serves content to billions of users, from internal applications for employees, to Google-developed public applications such as Gmail and YouTube, to customer applications built on their own infrastructure such as Spotify.

Hundreds of thousands of miles of fiber-optic cables connect this entire backbone of data center regions across land and under oceans. To date, Google has data centers and networking sites spanning over a total of 22 regions, 67 zones, 140 edge points of presence, and more than 800 global cache edge nodes. These numbers grow every month as Google is in a constant state of expansion because of the rapid growth of its business-to-consumer (B2C) technologies as well as its cloud business.

An *edge point of presence* (POP) is a location where Google connects its network to the rest of the Internet via peering. *Edge nodes*, also known as content delivery network (CDN) POPs, are points at which content can be cached and served locally to end users. The user journey starts when a user opens an application built on Google's infrastructure, and then their user request is routed to an edge network location that will provide the lowest latency. The edge network receives the request and passes it to the nearest Google data center, and the data center generates a response optimized for that user that can come from the data center, an edge POP, and edge nodes.

This entire infrastructure supports Google Cloud's custom Jupiter network fabric and Andromeda virtual network stack. The *Jupiter network fabric* is Google's system of networking hardware, represented as a fabric that provides Google with a tremendous amount of bandwidth and scale, delivering more than 1 petabit per second (Pbps) of total bisection bandwidth. This is enough capacity for 100,000 servers to exchange data at 10 Gbps each. To visualize the depth of this bandwidth and scale, the network could read the entire contents of the Library of Congress in less than 1/10th of a second. Insane networking throughput! *Andromeda* is Google's software-defined networking (SDN) stack that provides an abstraction on top of all of the underlying networking and data center hardware for Google and its cloud tenants to conduct business securely, privately, and efficiently. Google continually iterates its hardware and SDN—for example, the company saw a 3.3× latency improvement from virtual machine–to–virtual machine network latency from its release of Andromeda 2.1 over Andromeda 2.0. With this SDN stack, Google can offer endless cloud networking possibilities for GCP customers. Google is able to provide the most complex and innovative networking solutions as its world-class engineers continue developing new technologies in GCP.

EXAM TIP Google's SDN and network fabric are all behind-the-scenes features made available to customers of GCP. You will not see anything about them in the exam, but it is quite fascinating to know how Google Cloud runs and operates its technology stack, especially if you are trusting GCP with your business.

Encryption in Transit

Encryption in transit is an important topic for cloud customers. Google encrypts and authenticates all network data in transit at one or more network layers when that data flows outside physical boundaries not controlled by Google; data in transit within GCP is not necessarily encrypted, but it is generally authenticated. This usually triggers some questions for security professionals, as they would imagine that in a multitenant cloud environment, they'd like all their data to be encrypted in transit and protected against packet sniffing or Address Resolution Protocol (ARP) cache poisoning attacks across the entire network, end to end.

Although Google encrypts all traffic outside its boundaries, Google employs many other security controls within its boundaries to ensure that customer data is private and protected from access by other Cloud customers. There are a few things to unpack here. The *Google Front End* (GFE) is a reverse proxy that protects the backend Google services. When a user sends traffic to Google, it hits the nearest edge node that routes the traffic through the GFE; then the GFE authenticates the user, assures integrity, and employs encryption in transit by default using Transport Layer Security (TLS)—specifically, BoringSSL, which is an open source version of OpenSSL. Once you are proxied by the GFE, you are under the purview of GCP.

Once your data is inside Google, varying levels of protection are available, depending on whether or not the data goes outside of the physical boundaries protected by Google. When data goes outside of that physical boundary, the network is semi-trusted, and Google enforces authentication and encryption in transit for that connection. For application layer, or layer 7 security, Google uses Google Remote Procedure Calls (gRPCs) from service to service that are authenticated, tamper-evident (maintains integrity), and encrypted in transit using Advanced Encryption Standard (AES) for calls that leave the physical boundary. Within Google's physical boundaries, data is authenticated and assured to be tamper-evident. Google attests to rigorous internal security controls and maintains the highest level of audit and compliance for them in order to prevent insider threats. Moreover, based on the inner workings of the software-defined network, there is no way that tenants sharing a physical host would be able to listen to other tenants' traffic. You still have the ability to employ additional secure measures on the application layer if you want or if your compliance and regulations require it, but for most organizations, using the default encryption in transit is sufficient.

Network Tiers

Since Google owns its entire network end to end, it is able to offer its customers the concept of *network service tiers*. Google offers two network service tiers: a default Premium network service tier and Standard network service tier. The *Premium tier* is the default setting for GCP customers, offering users access to high-performance networking using Google's entire global network, as described previously. In the Premium tier, Google uses *cold potato routing*, a form of network traffic routing in which Google will hold onto all network packets through the entire life cycle until they reach their destination. Once inbound traffic reaches Google's POP, Google will use cold potato routing to hold onto packets until they reach their destination. Outbound traffic will be routed through Google's network until it gets to an edge POP nearest to the user.

The *Standard tier* is a more cost-effective, lower-performance network that does not offer access to some features of Cloud networking (such as the ability to use a global load balancer) to save money. In the Standard tier, Google uses a *hot potato routing* method, whereby Google will offload your network traffic as fast as possible and hand it off to the public Internet to save you money.

NOTE In cold potato routing (Premium tier), Google will hold onto your network packets until they get as close to the user as possible. In hot potato routing (Standard tier), Google will offload your packets as close to the source, not the user, as possible so that the public Internet can handle the rest, which means you save some money on GCP.

Standard tier offers much lower performance and is less secure than Premium tier, and as a result, most GCP users do not use the Standard networking tier. Actually, the Premium tier is unique to GCP, because Google has its own massive private global network. Other cloud providers do not have a global network of fiber-optic cables connecting their data centers, so they have to use hot potato routing. This is one of the major reasons why GCP is a more secure cloud by default.

Virtual Private Cloud, Subnets, Regions, and Zones

We discussed all the things outside of your purview in Google Cloud—the magic behind the scenes. Now let's get into some details of the network constructs within your GCP environment.

Virtual Private Clouds

The fundamental network in the cloud is a *virtual private cloud* (VPC), a virtual version of a physical network built on Google's software-defined network stack, Andromeda. A VPC provides connectivity for your virtual machines (VMs), Google Kubernetes Engine (GKE) clusters, App Engine flex environment instances, and any other products that are built on Google Compute Engine (GCE) VMs. VPCs offer native internal TCP/UDP load balancing and proxy systems for internal HTTP(s) load balancing, as well as the ability to leverage a variety of connectivity options, from VPN tunnels to Cloud Interconnect attachments that connect to your on-premises environments. VPCs are global resources and are not associated with a particular region or zone; they are built inside of projects and by default do not permit cross-project communication.

Subnets

You can create your VPC in two ways, and your choice is determined by the way you create your subnets, which we'll dive into in the next paragraph. A *subnet* is a subnetwork, or a logical subdivision of your RFC 1918 IP space. An *RFC 1918 IP range* is simply a private network that uses both IPv4 and IPv6 specifications to define the usable private IP addresses in your network. In layman's terms, your RFC 1918 IP range is just your private network. Think about all the internal applications and devices that are hosted in your organization; they all have IP addresses on your RFC 1918 IP range (your private IP space) that is associated with your subnets. Subnets exist to make your network infrastructure more efficient, more manageable, and more secure. Subnets are also regional resources, whereas VPCs are global resources. Think about a company like Google.

Can you guess how many devices and servers those nerds have? All of those devices and servers need a unique name (an IP address) to talk to one another, and you can't shout at Habib the Database Server and expect a timely response in a single room full of a billion other servers.

You can't have a VPC network without at least one subnet in it, so you can either use auto mode VPC networks or custom mode VPC networks to provision your network. An *auto mode VPC network* is the default network that is created when you create a project. In this configuration, each region automatically gets a default /20 subnet created in it. *Custom mode VPC networks* do not come with any subnets, giving the network administrator full control to define the subnets and IP ranges before the network is usable. Custom mode VPC networks are the more likely choice of provisioning a network, especially because most enterprises are building their infrastructure with code. They do not come with any firewall rules by default. Many companies want to extend their private network from on-premises to the cloud, or between clouds. When you're defining your subnets, you want to be cognizant of things like the private IP ranges of your other private networks to avoid conflicts and right-sizing your subnets to make them most efficient. It's also very important to avoid overlapping IP addresses, and leveraging custom mode VPC networks enables you to have that full end-to-end control over your subnet creation. Auto mode networks are good if you just need to build and tear down a quick environment to do a proof of concept (POC), for example.

EXAM TIP Remember that VPC networks are global resources and subnets are regional resources. This design is one of the more significant advantages to using GCP over other clouds, because the global end-to-end physical network is fully owned by Google.

Regions and Zones

You probably already know the difference between a region and a zone if you're studying for this certification, but if not, let's do a quick refresher. A *region* is a collection of zones. A *zone* is an isolated location within a region. For example, in Northern Virginia, the us-east4 region is located in Ashburn and consists of us-east4-a, us-east4-b, and us-east4-c zones—which are typically individual data centers within the Ashburn, Virginia, geographical location. When you're building critical applications, you want to avoid any single points of failure, which could result from building applications in one zone or even building in one region. As you do with security, you imagine the worst will happen, and that will give you the north star for how to build your applications and networks to be fault tolerant, redundant, and highly available. Outages occur, and you should never believe otherwise; this is why service level agreements (SLAs) are so important. But there can be further outages beyond those anticipated in the SLA, for which you'd typically be eligible for financial credits from Google Cloud or whatever is determined as compensation in your SLA.

Two Outrageous Outages

A few years ago, in 2017, AWS experienced a major outage of its Amazon Simple Storage Service (S3) environment in Northern Virginia, which is commonly referred to as the "backbone of the Internet." Although S3 provides storage, similar to Google Cloud Storage (GCS), many other elements of web applications depend on S3 to surface data. This outage took down Medium, Slack, Reddit, and even the AWS cloud health status dashboard. In total, the four-hour outage resulted in a loss of approximately $150 million to the aggregate of the S&P 500 companies that were affected.

There was another outage caused by extreme winds in Northern Virginia later that year. I was sitting in a testing center across from Capital One in McLean, Virginia, taking the—wait for it—AWS Solutions Architect Associate certification exam. The wind took out power for the whole testing center, and there was another massive AWS outage. Luckily, the testing center providers had designed a strong disaster recovery system, and 25 minutes later, everything came back up with the answers still intact. That was a really tough exam.

For the context of your networks, VPCs are global resources, but the subnets themselves are regional objects. Your subnets can be assigned to one or multiple zones, but you cannot assign a subnet across multiple regions. The region you select for a resource determines the subnets that it can use. Let's say, for example, that you create a VM instance in the us-east4 zone. You can assign it an IP address only within the subnets that span across that zone. In Figure 6-1, you can see an example of a custom mode VPC network with three subnets in two regions.

Subnet Ranges and IP Addressing

You should be familiar with *Classless Inter-Domain Routing* (CIDR) and subnetting by now, but don't stress the nitty-gritty details of subnet ranges for the exam, because it's unlikely that you'll see questions about calculating subnet ranges and whatnot. CIDR is the guiding standard for Internet Protocol (IP) that determines the unique IP addresses for your networks and devices. These IP ranges are typically referred to as *CIDR blocks*. GCP has certain ranges that are restricted for its Google APIs, Google services, and other things.

When you create a subnet, you must define its primary IP address range, and you can optionally add a secondary IP range that is used only by alias IP ranges. Both IP ranges for all subnets in a VPC network must be a unique and valid CIDR block. For your RFC 1918 address space, it's the same as it would be in your on-premises environment, where you can use a 10.0.0.0/8, 172.168.0.0/12, or 192.168.0.0/16 IP range. The primary internal addresses for VM instances, internal load balancers, and internal protocol forwarding comes from the subnets' primary range.

ALL■IN■ONE

Google Cloud Certified Professional Cloud Architect

EXAM GUIDE

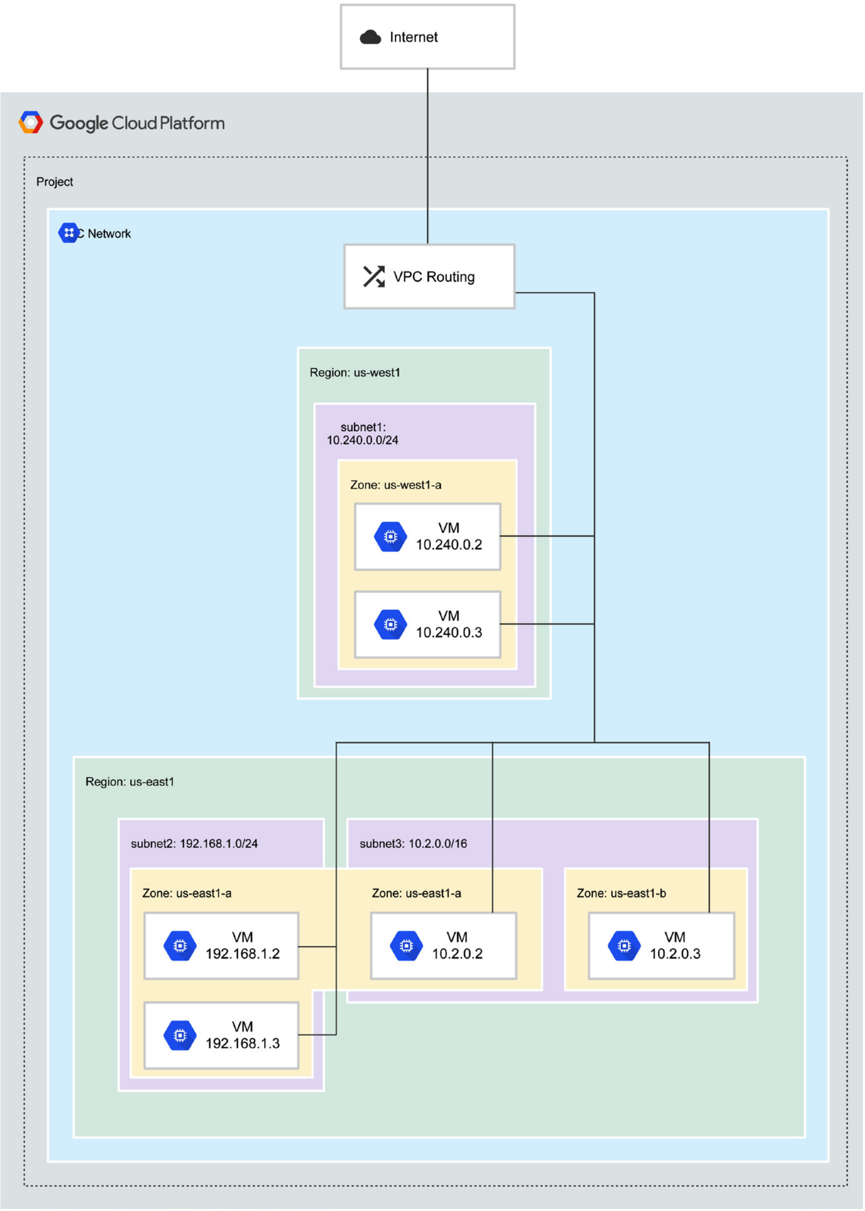


Figure 6-1 Custom mode VPC network with three subnets

External IP addresses are accessible to the public Internet. These addresses are not on your RFC 1918 IP address range. You can use external IPs on VMs and external load balancers. If you want your VM to be directly accessible from the outside (which you should avoid at all costs unless you have a need for public-facing endpoints), you would request a public IP as part of your VM creation. Google will then assign your VM a public IP address from its pool of available external IP addresses.

Keep in mind that externally accessible IPs are one of the biggest threat vectors for malicious actors, so you should be very aware and deliberate regarding what you're exposing to the outside world and how you're exposing it. Although there are many incredible Internet visitors who deserve praise for using the Internet ethically, there are an equal number of not-so-nice humans who are actively seeking out vulnerable people and technologies to exploit to gain a reward—even if that reward is simply bragging rights on 4chan. There are many use cases for external IPs—game servers, web applications, other consumer-facing applications, and more. However, you'll avoid much pain by simply asking this question: Who needs access to these machines and how can I reduce or avoid attack risks from those who don't?

To Expose, or Not to Expose

Following are some considerations to ponder when you're deciding whether or not to use an external IP:

- If you want to open an internal instance of Jira to your internal users who are working remotely, you don't need to expose it to the Internet.
- If you want your contractors to access an HVAC system application remotely, you don't need to expose it to the Internet.
- If a bunch of VMs need to perform updates from the Internet, you don't need to expose them to the Internet directly. Use network address translation (NAT) instead.
- If you need public users to access a web application, you should serve static content through your content delivery network (CDN) and put your application behind an external load balancer and/or a web application firewall (WAF), such as Cloud Armor.
- If users need to access a production game server, use public IPs and put them behind a proper security stack.
- If there are API calls to and from your network, rather than using public IPs, put them behind an API gateway and leverage whitelisting and proper firewall rules to allow communication.

(continued)

- If you are evaluating a marketplace application in a sandbox project that uses a public IP address as part of its deployment, consider adding a firewall rule to allow traffic only from known IP sources (such as your company's external IP CIDR block or the IP address assigned by your home service provider).
- Finally, if you absolutely need to expose a VM to the Internet and you can't properly secure that machine (and you should assume that you can't), then isolate your VM from the rest of your network. Basically, if machines need direct access to the Internet, you should limit the blast radius to those minimum resources. Don't let a breach of these VMs extend beyond those resources.

There are many ways to expose an application to the public without granting users direct access to the application instance itself. Adding layers of abstraction such as a load balancer, a CDN for static content, or an API gateway always adds an additional layer of security to the underlying application.

We'll dive into network security best practices later in the chapter, but for now, note that everything is always easier in theory. In practice, when you're facing large and complex migrations and hard deadlines, you'll have to play a role in prioritizing the most important controls to protect your enterprise and to minimize risk. As a cloud architect, you must clearly gather the requirements for a desired network pattern and maintain a set of approved patterns, documented clearly alongside a risk assessment and clear usage criteria that is accessible to your users internally.

Routes and Firewall Rules

When it comes to routing, you should know the simple difference between ingress and egress traffic. From a routing efficiency, security, and cost perspective, knowing how to optimize your network routes is key to building a well-architected network. *Ingress traffic* refers to packets that have a destination inside of your network boundary. *Egress traffic* refers to packets that originate inside of your network boundary but have a destination outside of your boundary. In this context, your network boundary is not necessarily Google's data centers. You can have many network boundaries, depending on how you set up your VPC. Here are a few examples:

- Egress traffic between zones in the same region on your RFC 1918 address space.
- Egress traffic between regions within the United States and Canada, or other continents, on your RFC 1918 address space.
- Egress traffic between GCP and on-premises RFC 1918 CIDR blocks.
- Egress traffic between regions across continents on your RFC 1918 address space.
- Egress traffic through external IP addresses.
- Internet egress.

Ingress traffic is free, whereas egress traffic has costs associated with it, depending on the usage and the services being used. Quite often during the sales cycle, enterprises and Google Cloud sales teams are gathering networking usage details to estimate the cost of traffic routing. It's not the ingress traffic that costs ISPs a lot of money; it's always the egress traffic. On Google Cloud, it's very much the same. Imagine all of the traffic that flows between data centers, between regions, and between continents—somebody has to pay that bill, and when your meme-generator company has petabytes of internal traffic flowing across the world, it's not going to be cheap!

Routes define paths for egress traffic. Think of your network like a combination of highways and roads sprawling across the country. The roads themselves (like physical network cables) don't know where drivers are going. Luckily, however, signs on the road tell drivers which exit and paths to take if they are going toward a specific destination. Network routes act like the signs on the road, helping a packet get from one place to another. Routes define how individual packets will travel across the network, depending on where they are going. This can get rather complicated because, unlike modern roads, these routes know about only the next exit junction. They don't see "the big picture." So, if you are going toward X, Y, or Z, they'll tell you to get off this road at this exit; otherwise, you should stay on this road to the next exit. With modern networking, these signs can be either *static* (hardcoded) or *dynamic* (updated behind the scenes). The trouble with networking, and what drives architects, developers, and cloud operators crazy at times, is that when your car/packet runs into a dead end, it is eliminated/dropped. Talk about a bad road trip! Networking concepts can get very deep and go way beyond the requirements for this certification. However, it is very important that you understand that if you don't set up the signs on the road correctly, your drivers will get lost and will be terminated! Thankfully, configuring networks on public cloud platforms are, relatively speaking, much less painful than what your typical Cisco Certified Internetworking Expert has to deal with.

In GCP, there are two categories of routes: system-generated routes and custom routes. Every new network starts with two types of system-generated routes: a default route and a subnet route. The *default route* is a system-generated route that defines a path for traffic that meets Internet access criteria to leave the VPC. Now this is important. What does it mean to leave the VPC in your architecture? Within GCP, it is assumed that you will exit to the Internet through GCP resources such as a cloud NAT or Internet gateway. But many enterprises, as a security policy, do not want GCP resources to have their own Internet traffic routes. Such access to the Internet adds points to the attack surface area. Any one of these Internet access points could be used to launch a direct attack on all GCP networked resources and even on-premises resources. So instead, such access is often routed to explicit destinations such as back to the on-premises network (which already protects Internet access) or to a dedicated environment such as a DMZ (demilitarized zone) within GCP to provide a single point of entry/exit to protect properly.

For the sake of simplicity, let's assume that your VPC projects are isolated and self-contained. For these environments, the *default route* criteria would have traffic that has to do the following:

- Have a default Internet gateway route that provides a path to the Internet
- Have firewall rules that must allow egress traffic from the instance
- Have an external IP or be able to use NAT

The *subnet route* is created for each of the IP ranges associated with a subnet, with each having at least one subnet route for its primary IP range. These routes define paths for traffic to reach instances that use the subnets. This extends across your entire RFC 1918 private IP network (on GCP, on premises, multi-cloud, and so on). Let's go back to the road concept. Think of subnet CIDR ranges as the most granular level that road signs understand. Inside an individual subnet, all the "homes" can easily reference one another by their home address (IP address). But when they need to "drive to another home" outside their current subnet, the network needs to find the best road/path to the destination egress address. It does this by using the destination subnet CIDR. From a network engineering perspective, as your network gets more complicated, this is the configuration you need to ensure is correct to enable individual subnets across your network to communicate with one another. If your machines are unable to communicate across multiple subnets, you likely have a misconfiguration in your subnet routes or a firewall misconfiguration.

Right-sizing your subnets is incredibly important when you're designing your network. You can always expand IP addresses on subnets, but you can't shrink and repurpose the IP ranges elsewhere. There is a finite amount of addressable IPs in IPv4, and if you're expected to build only 1000 homes in your neighborhood, you don't want to provision a subnet range for 100,000 homes. Right-size your subnets to the best of your ability, because you can always expand them and cross that bridge when you get there; but you cannot do the opposite!

TIP Once you create a primary IP range for your subnet, you can't modify it on GCP. If you need to add IPs to your subnet, you first need to configure alias IP ranges and then use those as secondary IP ranges for your subnet. This is how you can expand your subnet IPs on GCP.

Firewall rules are your first perimeter boundary to protect yourself to and from the evil world. (Treat the world as evil. I don't care how many nice, wholesome videos you saw on Reddit today. In 2020, a 17-year-old kid hacked Twitter. This is unrelated to firewalls, but the message is the same: trust nobody! This is the Internet, where everybody gets pwned.) Firewall rules are leveraged to allow or deny traffic to and from VPC networks. When a rule is created, you specify a VPC network and a set of components that defines what the rule does. You can target certain types of traffic based on protocols, ports, sources, and destinations.

One of the items you'll see on the test on a question or two will have something to do with network tags. *Network tags* are strings that are used to make firewall rules and routes applicable to specific VM instances; they are added to the tags field in any resource, such as a compute engine instance or instance templates. Here are a few examples of when to leverage network tags:

- You want to make a firewall rule apply to specific instances by using target tags and source tags.
- You want to make a route applicable to specific instances by using a tag.

EXAM TIP You may see a scenario in which a web application has VMs running inside of a VPC, and you're asked to restrict traffic between the instances to specific paths and ports that you authorize. The application autoscales, so you can't route based on a static IP address. In this case, you'd use firewall rules to authorize traffic based on network tags that are attached to your VMs.

Private Access

A VPC is your virtual private cloud. It is private, where you manage IP addressing and your entire infrastructure. Google Cloud uses many public API endpoints for its managed services such as Google Cloud Storage and BigQuery. If you have an instance that needs to communicate with these services, you would need to have an external IP address to communicate with the resources outside of its network. This is where Private Google Access and Private Service Access come into play.

Private Google Access

Private Google Access enables instances without external IP addresses to access resources outside of their network inside Google Cloud. This was created to avoid security concerns for sensitive workloads where you did not want to have to assign your VM an external IP address and have it communicate over the Internet to communicate with Google Cloud's managed services. You can extend this access through your on-premises network if you have a VPN or an interconnect setup.

Private Google Access enables you to use Google managed services across your corporate intranet without having to access the Internet and exposing your GCP resources to Internet attackers. Remember that RFC 1918 is your friend from a security perspective. The fewer public IP addresses you are dealing with, the lower your external attack surface area. Private Google Access can be used to maintain a high security posture using private IP addressing within your network, while leveraging Google services and resources that live outside it but still inside Google Cloud. Google Cloud acts as a sort of DMZ in that you can leverage these services to bring assets (such as data files into GCS, public repositories in BigQuery, Git repositories in Cloud Source Repositories, and so on) into your ecosystem without actually exposing your network to the outside world. This significantly reduces your exposure to a network attack. You don't always need to access the Internet from your environment to keep it up-to-date. You can have the Internet come to you via Google Cloud services and then access those resources indirectly through Private Google Access.

Imagine a scenario in which you need to install software on your VMs, and the content is located in an on-premises file server, but your organization does not permit or have connectivity (VPN or Interconnect) between your Google Cloud environment and your on-premises network. Moreover, your Google environment needs to install this software securely without accessing the Internet as per security requirements. You can enable this capability securely using Private Google Access. You can establish a secure TLS session from your on-premises environment to GCS when you need to upload your files using **gsutil**. Then you can set up Private Google Access on the GCP project's subnet that your VM is located on, and just use the **gsutil** tool to download your content to the VM from GCS on GCP. It's like a secret backroad to Google Cloud public endpoints, and Google acts as your security guard.

Private Service Access

Private Service Access enables you to connect to Google and third-party services that are located on other VPC networks owned by Google or third parties. So imagine a company that offers a service to you, hosted on its internal network on GCP, and you want to connect to that service without having to go over the public Internet. This is where you'd want to leverage Private Service Access. This sounds a bit more risky, but Google has many controls to ensure privacy between tenants to prevent security incidents. Still, you should assess each offering with the lens of your organization's risk tolerance. This security model is similar to how AWS implemented a dedicated GovCloud for government agencies wanting a more secure cloud. The idea is that by never leaving the Google Cloud network and using trusted and certified third parties, you are able to leverage best-of-breed Software as a Service (SaaS) services or partner services without having to host these applications yourself in your own cloud.

Cross-Project Communication

We discussed in Chapter 4 (Resource Management) that it's a best practice to follow a "many projects" approach. Oftentimes, you'll need to communicate between VPCs that are used on separate projects. Even if you created two VPCs on the same project, they won't be able to communicate with each other by default. You've got options for how you want to enable cross-project communication. You can use these communication approaches beside one another, as some network complexities will require multiple communication mechanisms between VPCs. In this section, we'll discuss the usage patterns for a shared VPC, when to use VPC peering, and when you want to leverage a VPN gateway between VPCs.

Shared VPC

Some enterprises want strict control over all of the happenings inside of the cloud, and other organizations want to enable full developer freedom. Shared VPC provides a happy medium. It enables an organization to connect resources from multiple projects to a common VPC so that they can communicate securely on the same RFC 1918 IP space while still having a great amount of freedom and ownership over their projects. In the shared VPC model, a host project essentially becomes your control plane, and service projects give your developers freedom to build while still being confined to the constraints defined in the control plane. Your host project is where your VPC gets provisioned, including all of the networking elements. Your service projects are attached to the host project, where you can provision resources and allocate them internal IP addresses from the shared VPC.

In a shared VPC model, billing is attributed to the service project. Think about a large enterprise with a variety of revenue-generating business units that wants to use Google Cloud, but the company wants all of its business units to adhere to the organization's governing model. In such a case, you'd want to use a shared VPC, so that each business

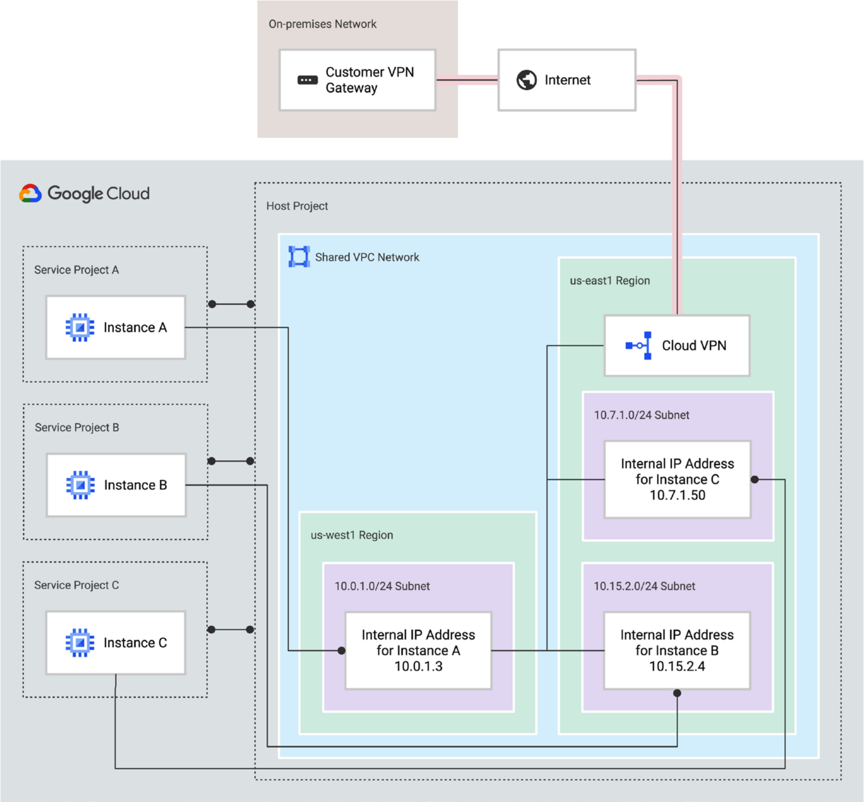


Figure 6-2 Sample shared VPC network

unit owns a portion of the bill and has more freedom to develop and build, and your centralized network and security administrators define the policies that govern the entire enterprise. In Figure 6-2, you can see an example of a shared VPC deployment.

You can have multiple shared VPCs inside a host project, but a service project can be connected only to one host project. This is becoming a much more common pattern, especially with large enterprise organizations, as they are federating the cost of cloud computing to their teams who want to leverage the cloud. In most enterprises, if any of the businesses want to migrate, they can administer most of their infrastructure, communicate to other business units securely, and also pay their portion of their bill themselves.

NOTE Shared VPC is typically referred to as "XPN" in the API.

VPC Peering

VPC peering enables two separately managed VPCs to communicate with each other. These connections are nontransitive, meaning that multiple VPCs cannot communicate through a common VPC unless they are explicitly peered with each other. This is great for organizations that federate the governance and ownership of their projects to their teams to own their project and VPCs fully, but their projects still need to communicate with one another. From a performance perspective, there are no latency issues, as peering a VPC would provide the same performance they'd experience if they were on the same VPC network. In Figure 6-3, you can see a case of an ingress firewall that allows traffic only from certain source IPs in a peered network.

Cloud VPN

You can leverage the cloud VPN to connect VPC networks in hybrid environments. Imagine that you have a network in AWS and you'd like to build a network pattern that connects your GCP VPC to your AWS VPC environment. Cloud VPN is a transitive IPSec VPN tunnel. This is a more scalable solution, but it requires that you manage your VPN tunnels and network configurations. There is also limited throughput—you only get 1.5 Gbps over the Internet and 3 Gbps over direct peering or between GCP projects.

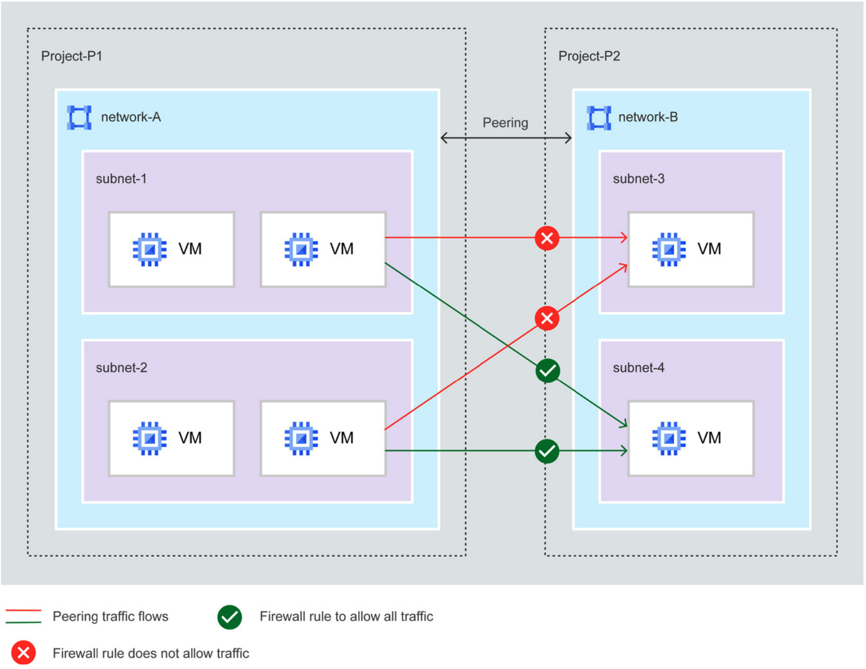


Figure 6-3 Firewall with VPC network peering

It is likely that you'll see improvements to this technology in the future, with Google Cloud and other cloud providers offering VPN solutions that significantly improve throughput.

EXAM TIP Know the difference between transitive and nontransitive peering. If, for example, VPC-A is connected to VPC-B using VPC peering, and VPC-B is connected to VPC-C via peering, VPC-A will still explicitly have to be connected to VPC-C to communicate—hence it being nontransitive. In the transitive model, as long as routes and rules are properly configured, you can communicate across to any other connected networks, which is typically the pattern desired in a hub-and-spoke model.

Cloud DNS

If you have a web server running your startup's website with an IP address of 72.44.923.12, good luck trying to get anyone beyond you and your co-founder to memorize that address. The Domain Name System (DNS) solves this by translating domain names to IP addresses so your web browser can load Internet resources. It's the phonebook of the Internet. You should have a strong understanding of DNS by this point in your career, so we'll skip the basics. (If you don't, I recommend that you brush up on how DNS works online.)

EXAM TIP You likely won't see anything too complex on the test about DNS, except maybe some questions about troubleshooting, where the answers could have to do with incorrect DNS records or setup.

In the world of the cloud, with hybrid environments consisting of multiple clouds and on-premises environments, DNS records for internal resources typically need to be accessed across environments. Traditionally, in on-premises environments, these DNS records are manually administered using an authoritative DNS server. When you run DNS in Google Cloud, it's important to be familiar with a few concepts:

- **Internal DNS** This service automatically creates DNS records for VMs and internal load balancers on GCE with a fully qualified domain name.
- **Cloud DNS** This managed service provides ultra–low-latency and highly available DNS zone serving with 100-percent SLA. It can act as an authoritative DNS server for public zones that have Internet visibility or private zones that are visible only within an internal network.
- **Public DNS** This is a Google service, not Google Cloud, offering an open, recursive DNS resolver—you've probably seen 8.8.8.8 and 8.8.4.4 in your technology life at some point.

It's possible that an organization may want to hold off migrating their DNS service to Cloud DNS for public zones but may want to use Cloud DNS just for their private zones. If you're using Cloud DNS for your internal-facing DNS records, you'll need to configure private zones to manage all of the internal records. The VPC must be in the same project and authorized to use the private zones in order for this to work properly, and if you're working across projects, you'll need to configure DNS peering. DNS peering enables you to query private zones across VPC networks without having to connect the networks. This is especially useful in a hub-and-spoke model, where you'd want to forward queries from a hub project on GCP to your on-premises environment. Let's think about that for a second.

- Imagine you have a hub-and-spoke model for your company, Covfefe, where your hub (VPC-A) is a VPC that is connected to your on-premises environment via an interconnect.
- Your spokes are separate VPC networks (VPC-B and VPC-C) that are peered to your hub (VPC-A).
- Since peering is nontransitive, you can't automatically resolve DNS queries across VPC-A to the on-premises environment.
- You'd want to leverage DNS peering to peer your private DNS zones on VPC-B and VPC-C to your peering zone (gcp.covfefe.com) on VPC-A. Then you could use a DNS forwarding zone to forward those queries to your on-premises (onpremise.covfefe.com) environment.

As mentioned earlier, don't worry too much about diving deeply into DNS if you're really focused on passing the exam. Just think about what could happen in the event of a missing record—what types of errors you'd run into. Could the lack of a DNS record or an improper zone configuration be the answer to your problem?

Connectivity to Your Cloud

Most organizations are not cloud-native. Most are running either their own data centers or are working with third-party providers that provide the organization space to own its physical infrastructure (the not-so-cloud cloud). As you can imagine, if you've already been running technology for so many years, it isn't easy to jump ship and rebuild or migrate everything to Google Cloud. For many organizations, including cloud-native ones, there are always aspects of connectivity requirements between data centers, other cloud providers, and Google Cloud that come into play. It's very rare to see an organization put all its eggs in one basket, and that's not a best practice from a risk-management standpoint. So oftentimes you need connectivity to provide critical services access to others across this multi-cloud or hybrid environment.

Imagine, for example, that you're running a machine learning environment that is processing data and needs to ingest streaming data from an on-premises application in a secure fashion. How would you set up this ingestion pipeline securely without having proper connectivity in place between your on-premises data center and GCP? As a cloud architect, if you're building these network pipes, you'll want to gather some information about the bandwidth, SLA, and redundancy requirements. You'll want to know the use case, where your data centers are located, and how to leverage the right connections in the most operationally efficient and cost-efficient way. In this section, we'll talk about using a VPN, using the various Cloud Interconnect options, and peering your networks with GCP. But, first, let's discuss the most important foundational element here, which is Cloud Router.

Cloud Router

Cloud Router is a managed service that dynamically exchanges routes between your VPC and your on-premises network using the Border Gateway Protocol (BGP) via a Dedicated Interconnect or Cloud VPN. This router is the foundational element of using the Cloud VPN and Cloud Interconnect.

BGP is a very complex exterior gateway path-vector routing protocol that advertises routing and reachability information among autonomous systems on the Internet. Virtually the whole Internet runs on BGP. There are two flavors of BGP you should be aware of—external BGP (eBGP) and internal BGP (iBGP).

Let's imagine we have five large companies that have their own networks full of computing systems: Froogle, Tastebook, Netzeroflix, Bamazon, and Microdelicate. The collection of each of their public networks is considered an independent autonomous system (AS). eBGP enables each of those autonomous systems to advertise to its connected peers and say, "Hey world, this is Froogle, I'm right here! This is my Public IP network." eBGP lets all of these systems announce themselves to the world as their assigned AS number and what IP ranges their networks own. eBGP then propagates these details across the global network, with each node calculating best paths to optimize traffic routing with this new information. eBGP is the mechanism that enables the Internet to propagate public IP addresses across the globe. iBGP, on the other hand, is a mechanism to provide detailed private IP network information within an AS. So when you connect two separate ASs with a solution like Cloud VPN, their private network details are exchanged using iBGP. Obviously, Froogle can't access Tastebook's internal networks, because these are only aware of each other's public IP network via eBGP.

So in the context of the cloud, the Cloud Router provides a managed service that exchanges private network routes between your VPCs and your on-premises networks using iBGP. You can do this on a regional level by sharing routes only for subnets in the region where you have leveraged a Cloud Router, or you can leverage global routing to share the routes for all of your subnets inside of your VPC. When these routes are advertised, by default, they advertise subnets according to the regional or global routing option. If you want more control, you can use custom advertising to decide which IP ranges and subnets to advertise. Imagine that Froogle and Bamazon partnered up and ran a multiorganizational infrastructure with shared resources across the cloud, and you have to connect to your on-premises environment but you don't want to share Bamazon's subnets—this is where you'd use custom advertising. In Figure 6-4, you can see an example of what a regional dynamic route would look like.

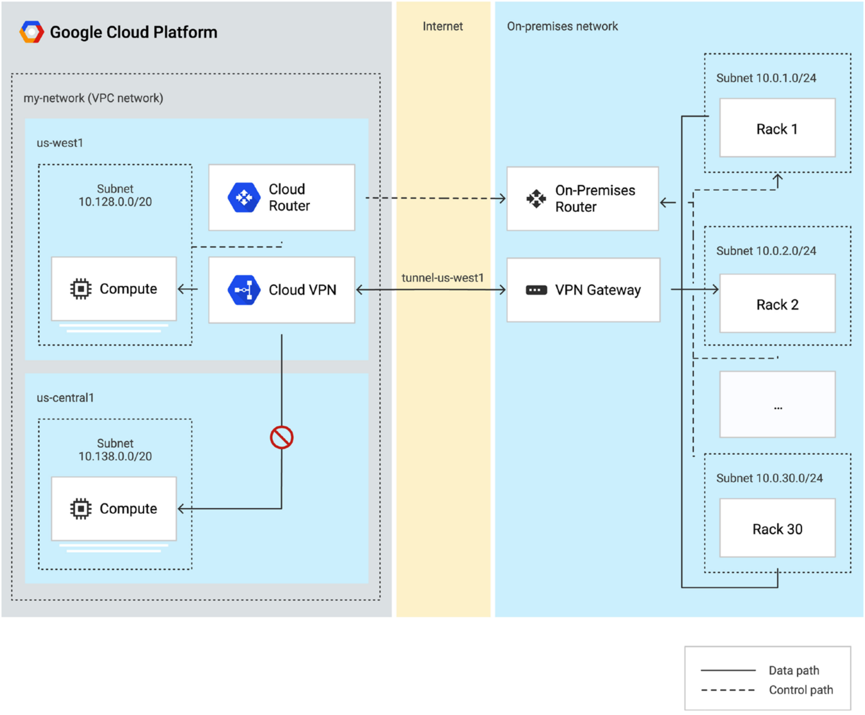


Figure 6-4 Cloud Router with regional dynamic routing

Cloud VPN

I would imagine that by this point in your career, you know what a VPN is. If you don't, that's alright; there's always next year. Kidding. Let's do a five-second refresher. A VPN lets you create a secure tunnel over an unsecure channel to avoid people sniffing and snooping on your packets. Cloud VPN is a solution that enables you to connect any of your peer networks to your VPC securely through an IPSec-encrypted tunnel. IPSec, Internet Protocol Security, is a secure network protocol that authenticates and encrypts the packets of data between two endpoints. This includes data that flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host).

You can use Cloud VPN to secure access from other clouds or your on-premises environment to GCP. But it has its limitations, particularly around bandwidth, because it supports only 1.5 to 3 Gbps per tunnel, depending on your peering location. There are two types of Cloud VPNs: Classic and High Availability (HA). Classic VPN supports static routing in addition to dynamic routing. As you can imagine, if you're using static routing, you've got to have an external IP and interface for each VPN gateway you set up. There's really no reason to use Classic VPN. If you don't have any need for static routing, leverage the HA VPN. With the HA VPN, you can do dynamic routing and you can connect as many on-premises VPNs to your HA VPN gateway without having all of the overhead of managing several VPN gateways in GCP. Many companies have not adopted the BeyondCorp zero trust access model and still have a need to use Cloud VPN to secure their on-premises traffic to GCP; sometimes it's easier and more cost-effective to spin up a Cloud VPN if it meets your use cases to secure traffic between sites as well, rather than having to provision an interconnect.

Cloud Interconnect

You know you're a born engineer when you have a story about how you got so tired of your awful Wi-Fi, especially after ISPs started doing multimedia-over-coax in their router/modem combos. Since you were too lazy to bridge your custom-flashed DD-WRT router, you instead decided to run a 100-foot cable through the crevices of your house just so that your Internet friends wouldn't hate you for lagging everyone up in a StarCraft match. Cloud Interconnect is similar to this but better, because it offers a low-latency, highly available network connection to transfer data and connect your RFC 1918 address space between your on-premises environments and GCP VPCs. Cloud Interconnect offers two options for extending your on-premises network: Dedicated Interconnect and Partner Interconnect. *Dedicated Interconnect* provides a direct, dedicated physical connection between your on-premises network and GCP. *Partner Interconnect* provides a connection between your on-premises and VPC networks through a supported service provider.

Dedicated Interconnect

Dedicated Interconnect is a dedicated line that connects your on-premises environment to your VPC network, enabling your private RFC 1918 space to communicate while bypassing the Google Front End (GFE). Dedicated Interconnect can use either 10 or 100 Gbps lines (commonly referred to as network pipes) that provide a 99.9 percent or 99.99 percent SLA and can attach to multiple VPCs by creating a VLAN with VLAN attachments in your Cloud Router. This line is dedicated to your traffic only, so it is not encrypted. Whenever security architects hear "not encrypted," they panic, so let's backtrack a bit to see exactly how this is set up.

First, Dedicated Interconnect is not available everywhere, because your physical network will need to meet Google's network in a colocation facility. Inside of Google's colocation facility, you'll be providing your own routing equipment that must support a 10G or 100G circuit and some other requirements. This is called a *cross connect*, because it connects your on-premises router (essentially your on-premises network) to Google's peering edge network inside of the colocation facility via a dedicated 10G link, a 100G link, or a bundle of links—this is your Dedicated Interconnect. You would then use interconnect attachments (VLAN attachments) to create an 802.1q VLAN to your Cloud Router that can be attached to multiple VPCs. Each attachment can be connected to a single VPC only, and only within the regions that the colocation facility serves; you can see those details online. Here are the essential details:

1. You bring your own trusted hardware into a colocation facility owned by Google that extends your on-premises network into Google's colocation facility.
2. From there, you cross-connect to a Google peering edge via a Dedicated Interconnect line(s) that Google reserves for you.
3. From this Dedicated Interconnect, you create interconnect attachments to configure a VLAN and associate it with a Cloud Router in the regions that the physical colocation facility supports, bypassing the GFE entirely, and avoiding any multitenant concerns.
4. Your Cloud Router is then attached to multiple VPCs.
5. At this point, you essentially have a full extension of your private RFC 1918 into Google Cloud. I hope you properly defined your VPCs' RFC 1918 IP ranges and subnets to avoid overlapping and conflicts with your internal network. Otherwise, you'll need to solve that problem now.

In Figure 6-5, you can visualize this configuration. You can see that at no point do you have to worry about any other cloud tenants or malicious users sniffing your unencrypted traffic, as long as you trust Google enough—and I doubt you'd be hosting your data with them if you didn't. Some organizations need to encrypt traffic among themselves because of compliance regulations. For this use case, you can use any third-party IPSec VPN to encrypt traffic inside of your interconnect. You can't use Cloud VPN, though. Imagine, for example, that your company is trying to build a backup replica of its on-premises MySQL database on GCP. It's a large 25TB database and updates multiple times a day, and it requires RFC 1918 space. Cloud VPN won't have enough bandwidth here. You'll need to use a Dedicated Interconnect.

When you use a Cloud Interconnect, traffic between your on-premises network and your VPC network does not traverse the public Internet, and that means fewer network

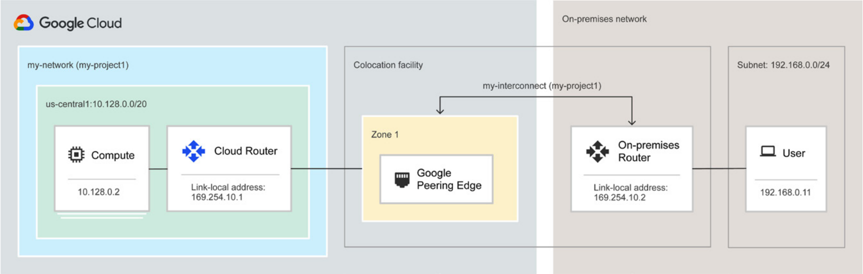


Figure 6-5 Sample Google Dedicated Interconnect configuration

hops and fewer points of failure where your traffic might get dropped or distributed. You also won't need to use a NAT or VPN to reach internal IP addresses. Using Cloud Interconnect is a great way to reduce your egress costs, as you still have to worry about egress costs when you use a Cloud VPN. Also, don't forget Private Google Access, because you can still use it in conjunction with Cloud Interconnect so that your on-premises hosts use internal IPs to reach Google APIs and services. If you don't use Private Google Access, your hosts will still be leveraging TLS when they're accessing Google APIs and services through the GFE.

Partner Interconnect

You get the gist of how the interconnect works. So I'll keep this section short and sweet. Dedicated Interconnect costs more, and it isn't available unless you are geographically near a Google peering edge. You also don't need to spend the money for a 10G Dedicated Interconnect if you only need bandwidth between 50 Mbps and 10 Gbps. So it can be more cost-efficient. Partner Interconnects are available in a lot more parts of the world. With this type of connection, instead of connecting your on-premises network to a Google peering edge, you connect your on-premises router to a partner peering edge that is connected to a Google peering edge, which then connects to your Cloud Router via the same attachment mechanism described in the last paragraph.

EXAM TIP Don't forget the bandwidth constraints of the various connectivity options. Cloud VPN only supports up to 3 Gbps per tunnel, Partner Interconnect supports up to 10 Gbps, and Dedicated Interconnect supports up to 100 Gbps. If you get a question on the exam about speed, privacy, and connecting between on-premises to GCP—you know what to do.

Cloud Load Balancing

One of the major benefits of using Google Cloud over other cloud solutions is Google's fully owned global infrastructure. Other cloud solutions offer regional infrastructure. With Google Cloud, this plays a significant role for load balancing. Think about it: Google has had to load-balance global traffic for billions of users using Gmail, YouTube, and Google Search worldwide for 20 years now. They're experts! The cloud uses the same infrastructure that serves Google and the world.

Cloud Load Balancing is a fully distributed, high-performance, software-defined, managed load balancing service that dynamically distributes user traffic across your infrastructure to reduce performance and availability issues. Offering a variety of load balancers, with global load balancing you get a single anycast IP that fronts all your backend instances across the world, including multiregion failover. There also is software-defined load balancing services that enable you to apply load balancing to your HTTP(S), TCP/SSL, and UDP traffic. You can also terminate your SSL traffic with an SSL proxy and HTTPS load balancing. Internal load balancing enables you to build highly available internal services for your internal instances without requiring any load balancers to be exposed to the Internet.

Imagine, for example, that you have a website serving up millions of users or clients worldwide, and on the backend of the website, you have many VMs that are autoscaling to meet user demand. It's great for the VMs to autoscale on user demand, but how do you actually distribute the load between those servers? Imagine the servers keep running at full capacity and spinning up new ones to support additional users, but every time a server maxes out, it crashes. This is where load balancing comes into play: it helps spread the load to the backend evenly based on capacity and the health of the servers. You'd much rather have 20 servers running at 50 percent instead of 10 servers running at 100 percent, potentially bottlenecking your application and causing reliability issues.

NOTE Remember that reliability is the most important objective for businesses! How do you plan on running a business if your service is having availability issues? Reliability is one of the biggest reasons for customer churn.

Overview

There are many types of load balancers with varying categories of configuration options. In Table 6-1, I've outlined the various types of load balancers. Take a look at the table and get familiar with the current list.

External load balancers balance the load of external users who reach your applications from the Internet. As described earlier, this can be for TCP/UDP, HTTP(S), or SSL traffic. The TCP/UDP external load balancer is a pass-through load balancer and is commonly referred to as a network load balancer. Pass-through means that the client retains its IP address instead of getting proxied by the server (or load balancer in this case).

Load Balancer Type	Traffic Type	Internal or External	Regional or Global	Supported Network Tiers	Proxy or Pass-through
Internal TCP/UDP	TCP or UDP	Internal	Regional	Premium only	Pass-through
Internal HTTP(S)	HTTP or HTTPS	Internal	Regional	Premium only	Proxy
TCP/UDP Network	TCP or UDP	External	Regional	Premium or Standard	Pass-through
TCP Proxy	TCP	External	Global*	Premium or Standard	Proxy
SSL Proxy	SSL	External	Global*	Premium or Standard	Proxy
External HTTP(S)	HTTP or HTTPS	External	Global*	Premium or Standard	Proxy

*Global in Premium tier only; otherwise, they are effectively regional in Standard tier.

Table 6-1 Types of Load Balancing Options in Google Cloud

Recall the MountKirk Games case study described in Chapter 1. Their CTO is looking to provide low-latency load balancing across to their users worldwide, and they want to get rid of their physical servers and move their game servers to VMs. Think about a game like Fortnite; what kind of load balancer would you use to distribute traffic worldwide across your VMs?

Internal load balancers distribute traffic to instances inside of GCP. Your load balancers periodically need to check on the instances to see how they're doing before the load balancers shovel more traffic onto their instances. These load balancer *health checks* determine whether the backends, such as instance groups, properly respond to traffic. They say, "Hey, instance group, are you dead yet?" and the instance group will respond, "I'm alive, buddy" or as you can imagine, dead instance groups provide no response. Based on a few details of how fast instance groups respond and how many times they respond successfully, the load balancer keeps an accurate tab on how they're doing. In order for these health checks to work properly, there needs to be a firewall rule that enables the load balancer health checks to reach the instances in the instance group. Otherwise, you can get HTTP 502 and 503 error codes, timeouts, and other issues. The health check may also tell the VM to restart itself if the instance groups continue to fail.

TIP When it comes to health checks, you need to ensure that the firewall rules are properly set up, and it's a best practice to check health and serve traffic on the same port. If you spin up an instance group behind an HTTP(S) load balancer and you notice that the VM instances are being terminated and relaunched every minute, it's probably not your VMs that are broken; you're missing a firewall rule.

Cloud CDN

Cloud Content Delivery Network (CDN) is a fast, reliable web and video content delivery network with global scale and reach through Google's global infrastructure. It provides edge caches peered with nearly every major ISP worldwide, and it leverages an anycast architecture to give you a single global IP address for global distribution. A lot of organizations use Cloudflare or Akamai for their CDN, which can all integrate with Cloud CDN. You'd want to use this to cache content from instances and storage buckets. How do you think companies like YouTube, Instagram, and TikTok are able to serve you photos and videos in near real time across the world? Global content caching is incredibly powerful and powers much of the content on the Internet today.

Network Security

With the old castle-and-moat belief system in the trash, and the post-apocalyptic architecture scenario that you'll see in the next section, you should ground yourself with the many common fundamental network security principles as well as new principles you can follow to secure your network. In this new shared responsibility model, depending on whether you use Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or

Software as a Service (SaaS) solutions, you're benefiting from a varying level of default security that Google provides. But you also need to establish strong security controls in many other areas. It just so happens that networking is up there as one of the most important areas in the cloud, including from a security perspective, hence this dense chapter. (Honestly, this chapter is really designed to prepare you for the MCAT and the GRE at the same time, while you're preparing to self-nominate for a Nobel Peace Prize.) Let's talk about network security.

Network Security Principles

I will preface this by saying that when you're involved in a highly complex implementation, it's very important that you understand the core principles of designing a secure network. However, when you're reading a certification book, it sounds a lot easier than it is in real life. Oftentimes, you'll run into highly sophisticated projects that have incredibly meticulous requirements that you'll continually discover, with engineering challenges that prevent you from applying certain controls and best practices. So although you need to understand security very well, you should also work alongside your information security team to ensure that they're doing risk assessments and providing advice for every bit of architecture work that is going into your GCP environment. For that reason, we won't dive too deep into the most advanced network security controls, and we'll focus instead on the foundational principles of designing a secure network. This is subject to change in complexity based on the industry you work with, specific compliance and security requirements you're beholden to, and your organization's overall risk posture.

If you've read any security book, you know about the *CIA triad*, the most foundational information security model, which refers to *confidentiality*, *integrity*, and *availability*. This basic model will give you guidance when you're designing any system in the cloud. It's also important that you know the *shared responsibility matrix*, because Google will provide protection to many areas in the cloud based on IaaS, PaaS, or SaaS, and you'll need to step up. It's a partnership, so don't slack on securing your network. Most cloud security breaches you hear about in the news happen because of a mistake that the cloud tenant made. Very rarely are they caused by failures of cloud providers.

The principle of *confidentiality* refers to protecting your sensitive and private data from unauthorized access. If someone sees your data but shouldn't see your data, even if they see it at the wrong time, that's a breach of this principle. The principle of *integrity* refers to the protection of data against modification from an unauthorized party. Think of hashing algorithms like SHA1 and MD5, calculating a hash, and appending it to a message so that the recipient of the message knows that it has not been tampered with. The principle of *availability* refers to ensuring that authorized users are able to access your service or data when needed. Availability is not binary—whether something works or doesn't. When you think of availability, especially around networks, not only do security mechanisms protect against availability, but fundamental issues with how you've designed your cloud with anti-patterns can also prevent it from scaling. It can also be affected by attacks against your cloud provider or outages.

When you think about the concept of prevention, detection, and response with respect to networks, you and your cloud provider both have roles to play in the shared responsibility matrix. The way you design your network, the network patterns that are risk-assessed and approved, the exceptions with mitigating controls, and all the configurations associated with it are preventative controls. Setting up your network according to best practices, as I've outlined in this chapter, is all preventative work—it will prevent bad behavior.

You'll also need both your platform folks and security operations folks to be able to detect faults and malicious activity. For the networking folks, their detection is centered around ensuring the availability of a system, improving the performance of a system, and tangentially detecting security issues. For the security operations folks, their role is purely focused on detecting malicious activity. This could include attacks against availability such as DDoS attempts, unauthorized network traffic patterns, open firewall rules, and much more. Having all your network logs exported to both teams monitoring systems and also having visibility over misconfigurations will provide this level of visibility to detect faults and malicious activity.

Lastly, from a response aspect, both teams will need to define what is needed to recover from an event. If there is malicious activity going on inside of a project and your security operations team identified it, what actions are taken to validate that this is an incident and then contain the activity? Should the teams have privileged access to perform actions, or should they have to work with the project owners? Shadow IT continues to be a problem, especially when the security operations folks know little about the engineering behind the network but everything about detection and response. So the platform teams continue to roll over them.

My recommendation is to understand the shared responsibility matrix and your organization's security requirements and to ground yourself in these principles when you're designing your network. That will be a great start. Take the initiative when nobody else does to talk about network security requirements during a meeting and find the right folks who you can partner with to help protect your business. A breach can result in loss of customers, leading to layoffs, or shutting down an entire product. It's in everyone's best interest to take part in designing a secure cloud. Also, infrastructure as code is king. When you have documented configurations adhering to sound governing principles, a process to build and deploy, and a mechanism to prevent misconfiguration, you are less likely to encounter many network security issues in the cloud. Google Cloud does a phenomenal job of doing most of the work in defending against sophisticated network attacks. Think about how these principles have applied to the previous chapters, organization design, identity, and identity and access management (IAM), and also consider these same principles as you continue throughout the book.

Google Front End

We discussed a bit about the GFE earlier in the chapter. The GFE is a smart reverse-proxy that is on a global control plane in more than 80 locations around the world, all interconnected through Google's global network. When a user tries to access an address on Google's infrastructure, the GFE authenticates the user and employs TLS to ensure confidentiality and integrity. The load balancing algorithm is applied at the GFE servers to find an optimal backend, and then the connections are terminated at the GFE. After the traffic is proxied and GFE has determined an optimal backend, it will leverage a gRPC call to send the request to the backend. You get DoS protection from the GFE and DDoS protection from the GLB. It's basically a traffic cop that decides who to route where in an orderly fashion.

Firewalls

Firewalls are among the most important elements of network security. The purpose of your firewall is to monitor and control incoming and outgoing network traffic based on rules you determine. Nowadays, there are next-gen firewalls that provide capabilities beyond port/protocol rule-based traffic matching and do L7 (layer 7, or deep packet) inspection, intrusion prevention, and more intelligence-based services. A significant amount of incidents occur because of misconfigured firewalls. We've all heard or seen cases of this. Even the most talented folks can make the mistake of improperly opening up a firewall. So the more you can automate this and get human hands off of making firewall changes, the better your life will be.

Managing firewall rules can be cumbersome, especially with large organizations. It's almost impossible to manage firewall rules manually in a clean, organized fashion when you're scaling and growing many teams, many applications, and many solutions. Anyone who works on a traditional enterprise knows that there is a massive risk register full of old firewall rules that need to be deleted. Most of the cloud-native companies look to manage firewall rules in an automated fashion by employing tools like Terraform to do everything as infrastructure as code (IaC), creating a central policy repository for approved firewall patterns, and allowing developers to do pull requests when they want to leverage an approved pattern. This enables organizations to keep tight governance controls over what rules are approved in the repository, to detect anything that violates the policy repo, and to update and delete rules in a more centrally governed fashion. It's similar to our conversation about network patterns—all network patterns should include required firewall rules so that security architecture and information security can review the risk, tag the pattern accordingly, and keep it centrally stored so developers don't need to bug security every time they want to reuse a pattern and so security can use tools to automate the configuration enforcement and detection.

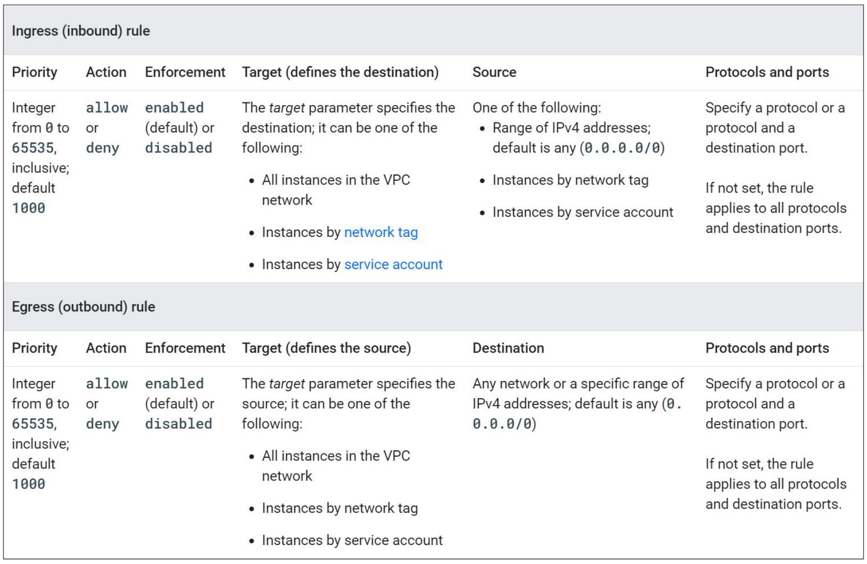
Web application firewalls are L7 application layer firewalls focused on protecting web applications by filtering and monitoring HTTP traffic between a web app and the Internet. The WAF technology should be implemented in front of any of your external IPs, without a doubt. Don't forget that most cybersecurity attacks happen on external IPs, as you're exposing your technology to all of the 1337 h4x0rs in the world. The issue is that configuring and managing WAF rules is a lot more complicated than just firewall rules. Cloud Armor is Google Cloud's solution to WAF. WAF will protect against common attacks such as cross-site forgery (CSRF), cross-site-scripting (XSS), SQL injections, and others.

CAUTION Do not expose your public IPs without adequate security to protect them from the world!

VPC Firewall Rules

You use *VPC firewall rules* to allow or deny connections to or from your VMs based on a configuration that you specify. These rules are always enforced. In GCP, every VPC network functions as a distributed firewall, where rules are set at the network level and connections are allowed or denied on a per-instance basis. You can set rules between your instances and other networks and also between instances within your network. Firewall rules can apply to an ingress or egress connection, but not both. The only action you can take is to allow or deny. (I wish “New phone… Who dis?” was an option. That would be nice if you’re testing a rule.) You must select a VPC when you’re creating a firewall rule, because they’re distributed firewalls across VPCs. You can’t share a rule among VPC networks; you’d have to define a separate rule in other VPC networks. When it comes to shared VPC, your firewall rules are centralized and managed by the host project. Firewall rules are *stateful*, meaning that after a session is established, bidirectional traffic flow will be permitted. A few components of a firewall rule are described in Figure 6-6.

Most of the items in the figure are pretty straightforward, but let’s rehash a few. Priority is sorted from 0 to 65535 in ascending importance, where 0 is the highest priority.



Ingress (inbound) rule					
Priority	Action	Enforcement	Target (defines the destination)	Source	Protocols and ports
Integer from `0` to `65535`, inclusive; default `1000`	`allow` or `deny`	`enabled` (default) or `disabled`	The *target* parameter specifies the destination; it can be one of the following: • All instances in the VPC network • Instances by network tag • Instances by service account	One of the following: • Range of IPv4 addresses; default is any (`0.0.0.0/0`) • Instances by network tag • Instances by service account	Specify a protocol or a protocol and a destination port. If not set, the rule applies to all protocols and destination ports.
Egress (outbound) rule					
Priority	Action	Enforcement	Target (defines the source)	Destination	Protocols and ports
Integer from `0` to `65535`, inclusive; default `1000`	`allow` or `deny`	`enabled` (default) or `disabled`	The *target* parameter specifies the source; it can be one of the following: • All instances in the VPC network • Instances by network tag • Instances by service account	Any network or a specific range of IPv4 addresses; default is any (`0.0.0.0/0`)	Specify a protocol or a protocol and a destination port. If not set, the rule applies to all protocols and destination ports.

Figure 6-6 The components of a firewall rule

You can target the destination based on all instances in a VPC, or you can use network tags, or you can apply it to instances by service account. If you use network tags, you have to create and manage the tags, which may make it easier to apply firewall settings. If you target instances by service account, then when a cloud application scales up or down and new VMs are added and removed, you won't have to make any firewall modifications, which may drastically simplify managing IP address–based firewall rules. You still need to specify a protocol and port.

You need to consider a few *implied rules*, which are firewall rules that are built in to VPC Firewall by default; you cannot change them, but if you need to make a pattern that goes against the implied rule, you can give it a higher priority and it will take precedence. Implied rules were created to prevent a default, new project from being exposed to the world. There are two rules:

- **Implicit Ingress Deny** This rule will deny any ingress traffic to your VPC by default. You don't want to open up your VPC to the outside world unless you really need to!
- **Implicit Egress Allow** This rule will allow your instances inside your VPC to send traffic to any destination if it means Internet access criteria (that is, it has an external IP or uses NAT).

Google is also releasing a new feature called *hierarchical firewall policies*, which solve a lot of challenges when it comes to building more consistent, yet granular firewall policies and applying them at various parts of the resource hierarchy. With hierarchical firewall policies, you can assign firewall policies at the organization level node or to individual folder nodes. Managing firewall rules can be tough to centralize and approve in a streamlined fashion, even through Terraform. Using hierarchical firewall rules enables you to give your teams more freedom to open up firewalls as they need to, but you can still apply critical policies to allow or deny certain high-risk traffic. All that said, do not open your firewall accidentally to the world!

TIP If you are running a multitier web application and you need to determine the direction in which traffic should flow, add tags to each tier and set up associated firewall rules to allow the desired traffic flow.

Simple Solution to a Major Misconfiguration

There are times that your users will want to deploy prepackaged solutions from Google Marketplace or simply deploy a proof-of-concept (POC) application whose installation instructions recommend the use of a public IP address. We have all seen this, and it is often the leading cause of successful attacks within public cloud networks. Although we will mention more hardened solutions for this later, when it comes to firewall rules, what can you do to minimize risk? A very successful strategy in dealing with those use cases is to enforce firewall rules that follow a least privilege configuration approach, even in sandbox environments. Basically, the way to achieve this is simply never to permit ingress "allow 0.0.0.0/0" in your firewall configuration! This will save you a ton of hurt! Sometimes that means allowing only your corporate public IP CIDR block access to your resources, or your test users' home IP addresses, or the entire RFC 1918 range. It's just rarely the case that any self-managed resource you deploy in a public cloud should ever need to be exposed to the entire Internet directly via its public IP address. So why are you allowing users to configure their firewalls as such?

Cloud Armor

Cloud Armor is a managed WAF that integrates directly with external HTTP(S) load balancers for any applications that you are exposing to the world through an external IP. As mentioned earlier, you should not expose your external IPs directly to the world. Instead, use an external HTTP(S) load balancer and leverage Cloud Armor to provide WAF capabilities to protect against attacks beyond just the default DDoS mitigation that you get with the external load balancers. In fact, Cloud Armor provides both L7 application security and layer 3/4–enhanced DDoS protection in the form of WAF capabilities via preconfigured rules, custom rules, and access control restrictions that are attached to your external HTTP(S) load balancer by a security policy. When you have a Cloud Armor security policy on your external HTTP(S) load balancer and external traffic hits the GFE when a user is going through the proxying process, your Cloud Armor policies are assessed against that user as well.

Here are some of the key things to note around Cloud Armor WAF:

- IP-based and geographic requirements can improve access to backend services.
- Preconfigured rules provide protection against XSS, SQL injection, local file inclusion, remote file inclusion, and remote code execution attacks. These rules are based on the OWASP ModSecurity Core Rule Set.
- Custom rules can be written through a Common Expression Language (CEL) format.
- You can preview the effect of a rule before enforcing it.
- All logs are logged against your external load balancer, and your security team will want these.

CAUTION Do not open your external IP addresses to the world without proper protection! Put them behind an external load balancer, and leverage Cloud Armor or a WAF of your choice to protect against attackers.

Cloud NAT

Cloud NAT is a managed network address translation service that enables VMs and GKE clusters to connect to the Internet without having external IP addresses. Yep, the key theme of this chapter is to minimize your threat surface and avoid using external IP addresses where possible! With Cloud NAT, outbound communication is permitted from an instance that is only on your RFC 1918 space, and inbound communication is permitted only when it's a response to a request. So imagine, for example, that you have a bunch of servers that need to reach out to a public address to request updates, such as a Windows Update service. You can use Cloud NAT to enable all of your internal instances to do their updates without having to give them external IP addresses.

VPC Service Controls

We talked about how you have your internal RFC 1918 space, but Google offers managed services that are hosted on their public API endpoints. These particular managed services, such as BigQuery and Google Cloud Storage, do not sit inside of your VPC network. If you're using Private Google Access to connect to these services through the secure backchannel route that Google provides, that's great! But how are you able to prevent your own users or compromised users from exfiltrating data when it's outside of your VPC?

This is where *VPC Service Controls* come into play. They enable security administrators to define a security perimeter around managed services, such as GCS and BigQuery, to mitigate data exfiltration risks and keep data private inside of your defined perimeter. When you create this service perimeter, it effectively isolates resources of multitenant services and constrains the data from these services to fall under the enforcement boundaries that you create to protect your VPC. VPC Service Controls are awesome, but they are quite complex when you're implementing them, so be patient and deliberate with your use case here.

NOTE Mukesh Khattar, security consultant at Google Cloud, has put together a phenomenal blog series on Medium titled "Mitigating Data Exfiltration Risks in GCP Using VPC Service Controls." This is an amazing reference guide. The link is in the "Additional References" section at the end of the chapter.

Identity-Aware Proxy

Google follows a zero trust access model called BeyondCorp that shifts access controls from the network perimeter to individual users and devices, challenging users through a highly sophisticated authentication and authorization mechanism. This enables employees, contractors, and other users to work more securely from virtually anywhere in the world without the need for a traditional VPN. Identity Aware Proxy is one of the building blocks to building a zero trust model. *Identity-Aware Proxy (IAP)* is a mechanism to control access to your cloud-based and on-premises applications and VMs on GCP that uses identity and context to determine whether a user should be granted access.

In 2021 and beyond, we're in a world where users want to be able to work wherever they are, and dealing with managing a VPN involves way too much overhead. We still want to be able to trust our users and ensure that they are securely accessing internal resources. The most advanced organizations are all headed this route, so if you're ahead of the curve here, this will greatly simplify user productivity.

The secret sauce of BeyondCorp is the integration of Cloud Identity with Identity-Aware Proxy. Cloud Identity allows for managing user identities along with the endpoint devices they use such as iPads, iPhones, laptops, and so on. When you use an endpoint device to connect to IAP, the proxy authenticates and authorizes your identity, your device, and the context of the session (where you are coming from, what time of day, device security posture, and so on). Based on that information, IAP identifies explicitly what set of internal resources you are allowed to access at that moment. So, unlike the perimeter-protection solution that VPN offers, IAP allows for in-depth protection of all your resources inside your private network with context-aware access.

Network Logging

It's critical that all of your network logs are required to be enabled at the organization level, are output to your key monitoring services, and are actively being monitored by your application and security teams to ensure that anything that shouldn't be happening in your environment is in fact not happening. You need these logs to follow the CIA triad principles. We're going to do a deeper dive into each type of log in Chapter 10, but, for now, just note the importance of viewing all your telemetry data and preventing it from being tampered with.

Explain Like I'm 5 (ELI5)

The castle-and-moat analogy of computing is old news. After 2021, anything goes, so here's a better story to tell. Let's have some fun! Disclaimer: There is no underlying meaning here, it's intentionally bizarre, enjoy the read, and tag me if it helped break down the complex concepts of networking in GCP!

In January 2020, the passing of the late, great basketball giant Kobe Bryant began a year filled with a series of disastrous world events. As 2020 progressed, we faced political warfare, the COVID-19 virus, a rising income inequality gap, unprecedented racial tension, record-breaking Tesla stock surges, and a devastating explosion in Beirut, while humans continue to ignore the abnormal trajectory of world climate and increasingly severe weather patterns that are signaling the beginning of the end. Increasing ocean temperatures continue to kill marine wildlife, which triggers food shortages in the entire global ecosystem. Year upon year, as temperatures continue to fluctuate beyond their normal millennial thresholds, they cause weather pattern changes that typically took thousands of years to appear in just a hundred years. Predictions indicate that the Arctic will see ice-free summers by 2050.

(continued)

Now let's do some time-traveling into the future. The traditional form of government has just collapsed, along with the Federal Reserve, because traditional resources such as oil and gold became seemingly extinct. The remaining Earth societies have survived on the last source of energy, which is now dependent on precious metals. Since Google, Amazon, and Microsoft spent decades mining these metals for the purpose of amassing data centers next to everyone and their babushka, they have become the New World Order. Bitcoin is the standard form of currency, and the new monetary system is owned by those three companies, as they have used their massive computing power to mine the last of the remaining Bitcoin. Jeff Bezos, Elon Musk, and Bill Gates moved to Mars, where they beam TikTok streams of their perfect lives back to the inhabitants of Earth. We are back to serfdom in a post-apocalyptic world, and tribes begin to form and need places to live and re-create society.

Most of the pre-apocalyptic cities that still exist are barren, desolate wastelands, with remnants of buildings and towns (on-premises environments) haphazardly running operations to the best of their abilities. You're the leader (cloud architect) of your tribe (your company), and you're deciding where to build your encampment. Three world superpowers (Google Cloud, AWS, and Microsoft Azure) all provide a safe space for tribes to live and innovate, each with its own pros and cons. You've decided to lead your tribe to build an encampment on GCP, as you believed that the services GCP provides in its shared responsibility matrix are the most secure, cost-effective, and scalable for your tribe's needs.

You are offered an empty plot of land and are allowed to do whatever your tribe wants to do (for a fee, of course). On your new land, you build a town (VPC) inside of a geographical boundary (project) that has sections (subnets) for housing and businesses (resources). These sections are categorized based on their proximity to one another, within a neighborhood (zone) or within a postal code (region). The addresses for the homes (RFC 1918 private IPs) are known only to the inhabitants of the town. Businesses (external IPs) have addresses as well, but they have stricter requirements for allowing customers in, since they have precious goods. The roads were paved by the federal government/superpower (physical network), and you have to put up the traffic signs that provide directions to navigate the town (routes); otherwise, your tribe members won't know where it's safe to go. They don't want to fall off a cliff (dropped packets) accidentally!

You have only a finite amount of resources for your citizens to build homes and businesses, so you need to plan your zones properly to ensure that your community can grow efficiently (right-sizing subnets). In this neighborhood, you don't allow anyone to leave as they want. They're all quarantined inside the neighborhood (no public IPs), as mandated by your state government (your company's governance policies). The federal government/superpower has provided a safe mechanism for you to communicate with other neighborhoods without having to worry about any other tribes interfering. If you want to communicate with their federal resources (Google Cloud APIs), you would normally have to leave the government boundaries and go through their front door (through the Internet). Even though they will ensure that you know the secret handshake (TLS), you'd much rather go directly to their resources without all the border security guards constantly patting you down. So they offer a secure back-channel for you to go directly to their federal resources (Private Google Access) while you're still within their boundaries.

On the perimeter, security guards have a list of who is allowed to come in and leave (firewall rules). While individuals go through Customs and Border Patrol (GFE), there are traffic cops (external load balancer) who show these people the right way. Even within your local boundaries, you've hired security guards and traffic cops (internal load balancer) to ensure that people inside and across your neighborhoods are going only where they're allowed to go (internal firewall rules). It's a cold world out there, and you can't even trust everyone inside of your own neighborhood. Who knows, someone could be a mole or could be compromised! If you want to market your business to the outside world, you aren't just going to pave a road to the outside. When the wandering *Mad Max*–esque freaks notice that some Customs and Border Patrol agents aren't protecting your road, you hire a special detective (Cloud Armor WAF) and advertise your business through Customs and Border Patrol and the traffic police.

You believe you did a great job setting up and running your old city (on-premises environment), and you intend to migrate all of the resources you had there to your new city, but you are afraid of doing this over the public roads. The federal government/superpower (Google Cloud) said they'll pave an expressway toll road (Dedicated Interconnect) just for you if you can meet them in their colocation facility. You trust your citizens in the old city enough to let them work directly with the folks in your new city, so you extend the boundaries of your old city to the colocation facility and agree to have this private toll road set up.

But wait! Things are heating up! Someone just launched a massive distributed attack against you. Luckily, the federal government/superpower is protecting your borders! You also notice that one of your businesses is using 200-percent more energy than usual. So you inspect your traffic logs (VPC flow logs) and see that something funky is going on. The business isn't just seeing a lot more customers, it is actually exfiltrating your intellectual property! Someone went rogue and social engineered your security guard to open up a path to the outside world, and now an attacker has made it to the inside! Immediately you start the incident response process. You analyze this behavior, shut the business down, close your borders, arrest the perpetrators, and then reconvene with your police force (Security and Governance) to do a "lessons learned review" and figure out how to prevent this from occurring again. Luckily, you're able to recover and get back to normal, and you followed best practices in securing your town and realize you shouldn't take these security topics lightly. The good news is that everyone in your city has a chip implanted in their body, so if you need everyone to abide by a new law, all you have to do is click a button and deploy it (centralized governance).

(continued)

Ultimately, your tribe is growing at a rapid pace, and the federal government/superpower has your back as they continue beefing up their innovative offerings and security. You realize that this model makes so much more sense from a total cost of ownership perspective. All your money is no longer tied up in capital expenditures. You've been able to achieve feats that would never have been possible in your old city—so much so, that you've decided to deprecate your old city entirely and build a beautiful new world the way you envision it. Your citizens are happy, your government is empowering you, and your tribe has learned how to sustain and thrive. You wake up one morning and hear the birds chirping, looking out the window to a beautiful vista of farmland and nature. You feel safe and secure. You have a growing population of beautiful new families, shelter, water, food, and technology, all built on a strong, secure, and efficient network that you architected—and world peace is finally achieved.

Chapter Review

In this chapter, we discussed the importance of networking and dove into the deep subject areas of networking in Google Cloud. Networking is the most complex and important topic of cloud computing, as the entire cloud is a massive global system of networks designed to give your organization a cutting edge in building the most innovative, cost-efficient, and operationally effective solutions for your customers worldwide. We started with the history of networking and why it has played the most pivotal role in the growth of society. Then we went into a technical deep dive into Google's global networking concepts. Google is the only cloud provider to offer a full end-to-end global network at scale, and that gives their cloud the competitive edge over other clouds to offer things like cold potato routing, global load balancing, encryption in transit by default, and encryption at rest by default.

We talked about the fundamental networking constructs and that your Virtual Private Cloud is a virtual form of your data center, hosted in Google's software-defined global network. We talked about subnets, routes, IP addressing, and the importance of your RFC 1918 address space and why it's so critical to minimize your threat surface by avoiding unsecured external IP addresses. Private Access offerings are a competitive offering by Google to provide a more secure, advanced transportation method for communicating with Google Cloud's public API endpoints, from which most of their managed services operate. We discussed the shared VPC model and when it's used, including some of the best practices around using VPCs and/or shared VPCs.

Next, we discussed the various ways you can provide connectivity to your cloud. In an increasingly multi-cloud and hybrid-cloud world, it's very important to know how you can connect all of your intellectual property across the world in a safe, fast, and secure fashion. We also went into load balancing, discussing how to handle load effectively to provide the highest availability objectives to your customers. Remember that reliability is the most important metric in the world; you have no business if your product is not available.

We dove into the key elements of network security, including some of the foundational security elements that should be applied across any cloud architecture, such as the CIA triad. Confidentiality, integrity, and availability are the fundamental principles that you need to ground yourself with when you are designing any architecture in the cloud (or anywhere for that matter).

Lastly, we had some fun and imagined a story analogy of networking and cloud computing in an alternate universe and post-apocalyptic world. It's okay to have some fun learning how networks operate in GCP without being bored down into the weeds. Try making a fun story in your next meeting! And don't forget the history and power of networking. From the earliest form of a network to where we are today, networking with one another is what created the beautiful world we live in. Building strong, positive relationships with those around us and extending our networks to uplift one another will get us through to our next human evolution. Keep on being a positive force in the world, and keep on finding ways to leverage technology to improve the lives of your loved ones and the ones who need it most.

Additional References

If you'd like more information about the topics discussed in this chapter, check out these sources:

- **VPC Overview** https://cloud.google.com/vpc/docs/overview
- **Best Practices and Reference Architectures for VPC Design** https://cloud.google.com/solutions/best-practices-vpc-design
- **Mukesh Khattar - Mitigating Data Exfiltration Risks in GCP Using VPC Service Controls** https://medium.com/google-cloud/mitigating-data-exfiltration-risks-in-gcp-using-vpc-service-controls-part-1-82e2b440197

Questions

1. You're an engineer troubleshooting a network issue. Network traffic between one Compute Engine instance and another instance is being dropped. What is likely the cause?

   A. A firewall rule that the instances depended on was deleted.

   B. The instances are on a default network and nobody created additional firewall rules.

   C. Network bandwidth is low.

   D. Your TCP/IP settings are improper.

2. You're working at a top eSports company that has a PostgreSQL database on-premises that handles user authentication to their training platform. You'd like to build a backup replica of this database on GCP. The database is only about 10TB, and there are frequent large updates. Replication requires private address space communication, and you're looking for a secure solution. What networking approach would you use?

A. Use a Dedicated Interconnect to enable your on-premises traffic to have a steady and high-bandwidth line directly to your GCP environment.

B. Use a Google Cloud VPN connected to your data center.

C. Use a Compute Engine instance with a VPN installed connected to the data center.

D. Use NetZero or AOL. (Seriously, don't you miss the good ol' days before Facebook?)

3. Your security team discovered a rogue network that was created in your project. This network has a GCE instance with an SSH port open to the world. They're working on identifying where this network originated from. Where should they look?

A. Look in the Cloud Monitoring console in your project.

B. Connect to the VM and look at the system logs to identify who logged in to the environment.

C. Look through the logs in the console and specify GCE Network as the logging section.

D. Look at the Cloud Audit logs and determine which user accessed this environment the most. Then interrogate that user.

4. Your organization needs a private connection between its Compute Engine instances and on-premises data center. You require at least a 20 Gbps connection and want to follow the best approach. What type of connection should you set up?

A. Create a Virtual Private Cloud (VPC) and use a Dedicated Interconnect to connect it your on-premises environment.

B. Create a Virtual Private Cloud (VPC) and use a Partner Interconnect to connect it your on-premises environment.

C. Create a Virtual Private Cloud (VPC) and use a Cloud VPN tunnel to connect it your on-premises environment.

D. Use a Cloud CDN to store content to the nodes closest to your on-premises data center and connect it from there.

5. You're running several Compute Engine instances and want to restrict communication between the instances to the paths and ports you authorize. You don't want to rely on static IP addresses or subnets, because you are working on a web app that is intended to autoscale. How should you restrict communications?

A. Use firewall rules based on network tags and attach them to the instances.

B. Use service accounts to delineate this traffic.

C. Use Cloud DNS and allow connections only from authorized host names.

D. Create separate VPC networks and authorize traffic only between those VPCs as needed.

6. You have an application running on a Compute Engine instance in a single VPC spread across two regions. This application needs to communicate over VPN to an on-premises network. How would you deploy the VPN?

A. Expose the VPC on-premises using VPC sharing and proper access controls.

B. Create a global VPN gateway and create a VPN tunnel for each region to your on-premises environment.

C. Use VPC Network Peering to peer the VPC to your on-premises network.

D. Deploy a Cloud VPN gateway in each region and ensure there is at least one VPN tunnel to the on-premises peering gateway.

7. You have a few VMs that need to be able to reach the Internet to do yum updates. But you don't want to expose the VMs to the world. What should you do?

A. Assign the VMs a public IP address, but put IP blacklisting in place to prevent anyone from accessing it.

B. Create one server that can download all of the updates and store them locally in your cloud. Then have your VMs update from that server.

C. Set up a Cloud NAT gateway so that your VMs can send outbound packets to the Internet and receive the updates.

D. Properly configure your DNS settings so that the network domain name is accessible by the Internet.

8. You'd like to set up a system to enable your development teams to do firewalling in a much more automated fashion. Historically, you had to create firewall rules manually for different traffic patterns for each development team environment and go through a review for every firewall rule creation. What would be a great way to solve this challenge for your enterprise in the most operationally efficient and secure manner?

A. Create a firewall rule review process that streamlines the rule approval from your information security team(s); then task your developers to continue creating the requests as needed.

B. Use an infrastructure as code solution such as Terraform and build a system where there is a policy repository for existing firewall rules from which development teams can do pull requests to use in their environments. Also, create a mechanism to vet and approve new network patterns as needed into that policy repository.

C. Provision each development team leader with enough access to manage the firewall in GCP so that you don't have to do this yourself.

D. Deploy a WAF on your external IP addresses, and you should be good to go.

9. You have an application on VMs serving web traffic and you noticed that after you configured the VMs as a backend to an HTTP(S) load balancer, the VMs are being terminated and relaunched. What is most likely the best way to deal with the issue?

A. Ensure that a firewall rule exists to enable the health check on the load balancer to reach the instances in the instance group.

B. Assign a public IP to the instances and configure your firewall rules appropriately.

C. Use network tags on each instance to route traffic from the instances to the load balancer.

D. Allow the source traffic on the instances to reach the load balancer by using firewall rules.

10. A shared VPC is:

A. A network construct that connects resources from multiple service projects to a common host project that contains the VPC network, so that they can communicate with each other securely and efficiently using private RFC 1918 address space.

B. A network construct that connects resources from multiple projects to a common VPC network, so that they can communicate with each other securely and efficiently using public address space.

C. A network construct that connects resources from multiple organizations to a common VPC network, so that they can communicate with each other securely and efficiently using private RFC 1918 address space.

D. A network construct that connects resources from multiple projects to a group of VPC networks, so that they can communicate with each other securely and efficiently using private RFC 1918 address space.

Answers

1. A. From all the answers here, the most likely one is that a firewall rule was accidentally deleted or intentionally deleted because someone didn't understand the dependencies of the rule. Check to see if the instances are permitted to communicate with each other.

2. A. You need direct access to your private network in the cloud for RFC 1918 communication between your on-premises environment and GCP. You also need to transfer large amounts of data frequently, so being bottlenecked by the limitations of a VPN is not useful here. The best approach is to use a Dedicated Interconnect so you have a strong pipe, a private line, and continuous integration into your private RFC 1918 address space.

3. C. Your goal is to identify when a network was created. Look through the logs in the console and search for a Create Insert; it'll display the JSON code string that contains the e-mail used to create the network. Your security team can use this as part of their investigation to determine the origin and root cause.

4. A. The best approach is to use a Dedicated Interconnect, because you're looking to establish a 20 Gbps private connection. Cost is not a constraint in this question.

5. A. Network tags enable you to make firewall rules and routes applicable to specific VM instances without having to rely on static IP addresses.

6. D. VPC Peering is allowed only between VPCs, not from a VPC in the cloud to your on-premises network. The answer here is to deploy a VPN gateway in each region and deploy at least one tunnel to your on-premises environment, so that instances in other regions can use the tunnel for egress traffic and you can meet your high-availability requirements.

7. C. A Cloud NAT gateway will allow your internal IP addresses to reach outbound addresses without needing an external IP. This is helpful for things like updates.

8. B. The goal here is to give your developers more freedom to build and deploy faster and to automate as much as you can, while eliminating sources of bottlenecks. By using a system like Terraform and having a firewall rule policy repository with approved network access patterns, you can securely give your developers the freedom to open firewall rules as needed, while also ensuring that only approved access patterns are in place.

9. A. You are getting the appropriate web response from each instance, but the instances keep getting terminated and relaunched. The network pattern between the load balancer health checks and the instances may not be properly configured.

10. A. A shared VPC is used when you have multiple projects that need to leverage one VPC network so that they can communicate securely and efficiently without having to deal with too much network configuration between VPCs and other network constructs.