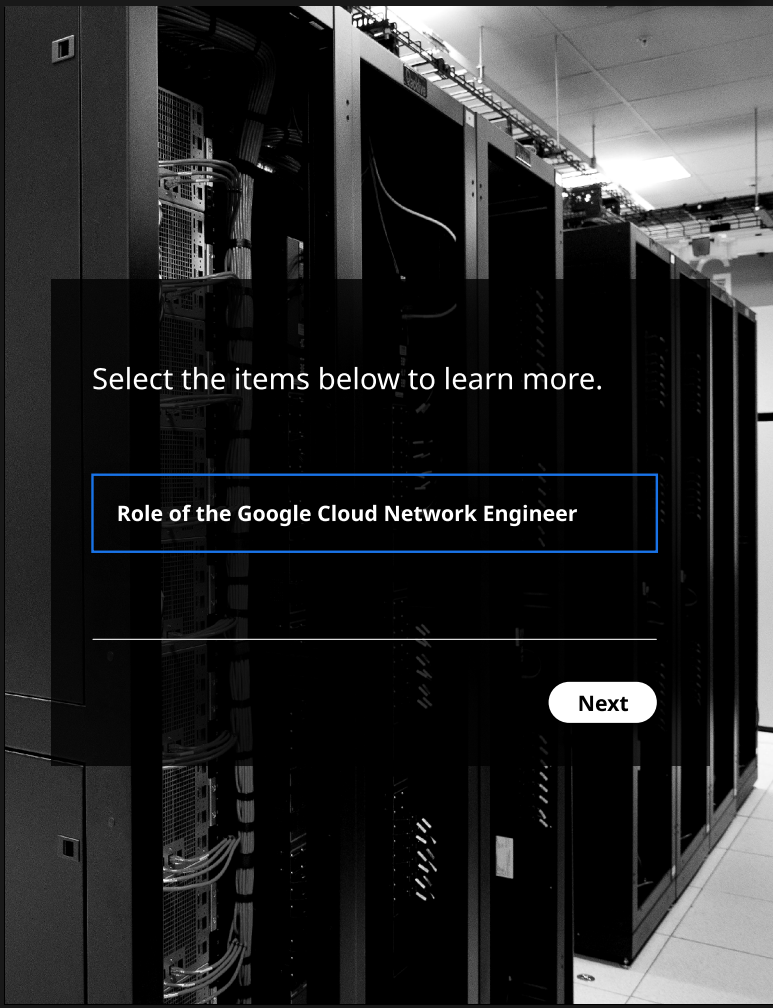
Google Cloud Certified Professional Cloud Network Engineer

Course Navigation

Linux Academy



Select the items below to learn more.

Role of the Google Cloud Network Engineer

Next

Role of the Google Cloud Network Engineer

A professional Cloud Network Engineer implements and manages network architectures in Google Cloud Platform (GCP). By leveraging experience implementing **VPCs**, **hybrid connectivity**, **network services**, and **security** for established network architectures, this individual ensures successful cloud implementations using the command line interface or the Google Cloud Platform Console .

Primary Responsibilities

- Design, plan and prototype a GCP Network.
- Implement a GCP Virtual Private Cloud (VPC).
- Configure network services.
- Implement hybrid interconnectivity.
- Implement network security.

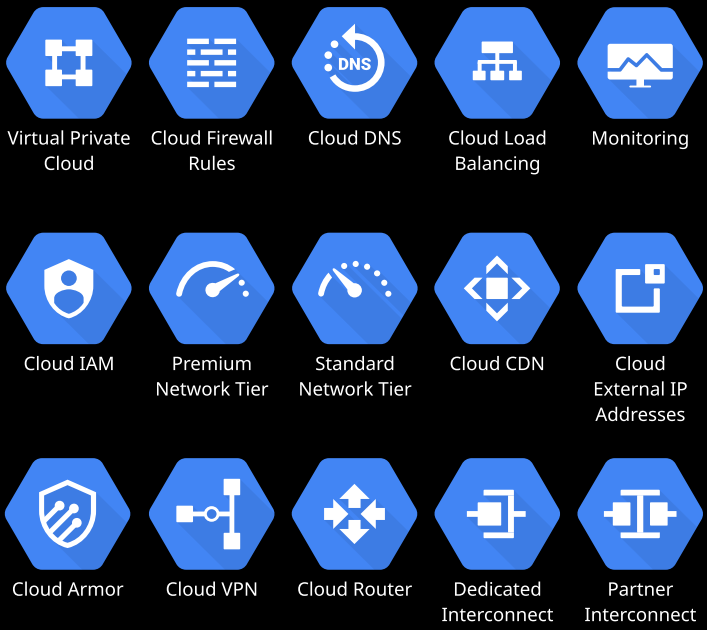
Next

Back to Main

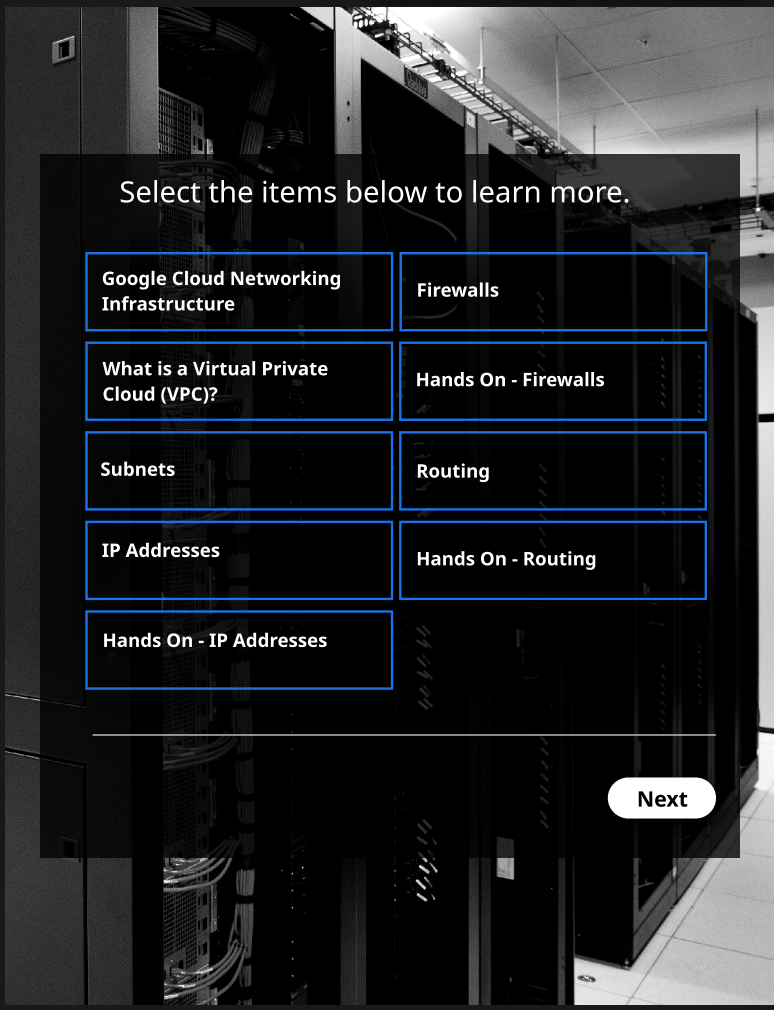
What This Means in Practice...

- Planning your network:
  - "Measure twice, cut once"
  - Who needs access?
  - Separation of duties
  - Subnet ranges
  - Interconnectivity
  - What does GCP do differently?
- Implementing (creating) all network components:
  - Subnets
  - Firewall rules
  - Routes
  - Private/public resources
- Increasing network/resource availability:
  - Load balancing
  - Managed instance groups
  - CDN
- Extending your network:
  - Hybrid connectivity
    - VPN, Interconnect, Peering
  - Sharing between projects
    - Shared VPC, Network Peering
- Securing network resources:
  - Firewalls
  - IAM
  - Cloud Armor
  - SSH connection methods
- Monitoring your network:
  - Stackdriver Monitoring and Logging
  - VPC Flow Logs

We have a lot of ground to cover — let's get to it!



Back



Select the items below to learn more.

- Google Cloud Networking Infrastructure
- Firewalls
- What is a Virtual Private Cloud (VPC)?
- Hands On - Firewalls
- Subnets
- Routing
- IP Addresses
- Hands On - Routing
- Hands On - IP Addresses

Next

Starting with the Fundamentals:

Concepts such as VPCs, subnets, firewalls, etc., will be the foundation for all other topics in the rest of this course.

Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

The Power of the Network

- Simply put, Google has one of the most powerful and robust networking infrastructures on the planet.
- It is necessary to support their own apps (multiple apps with 1 billion+ users each).
- The same network that powers Google also powers our GCP resources.

Breaking the Network Down

- Regions
- Zones
- Edge Points of Presence (POP)

Back Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Regions

- Independent geographic areas that host GCP data centers.
- At the moment, **20** regions are available worldwide, and growing.
- Typically consists of **3 or more zones**.
- Examples: us-central1, europe-west4, asia-east2

Back Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Zones

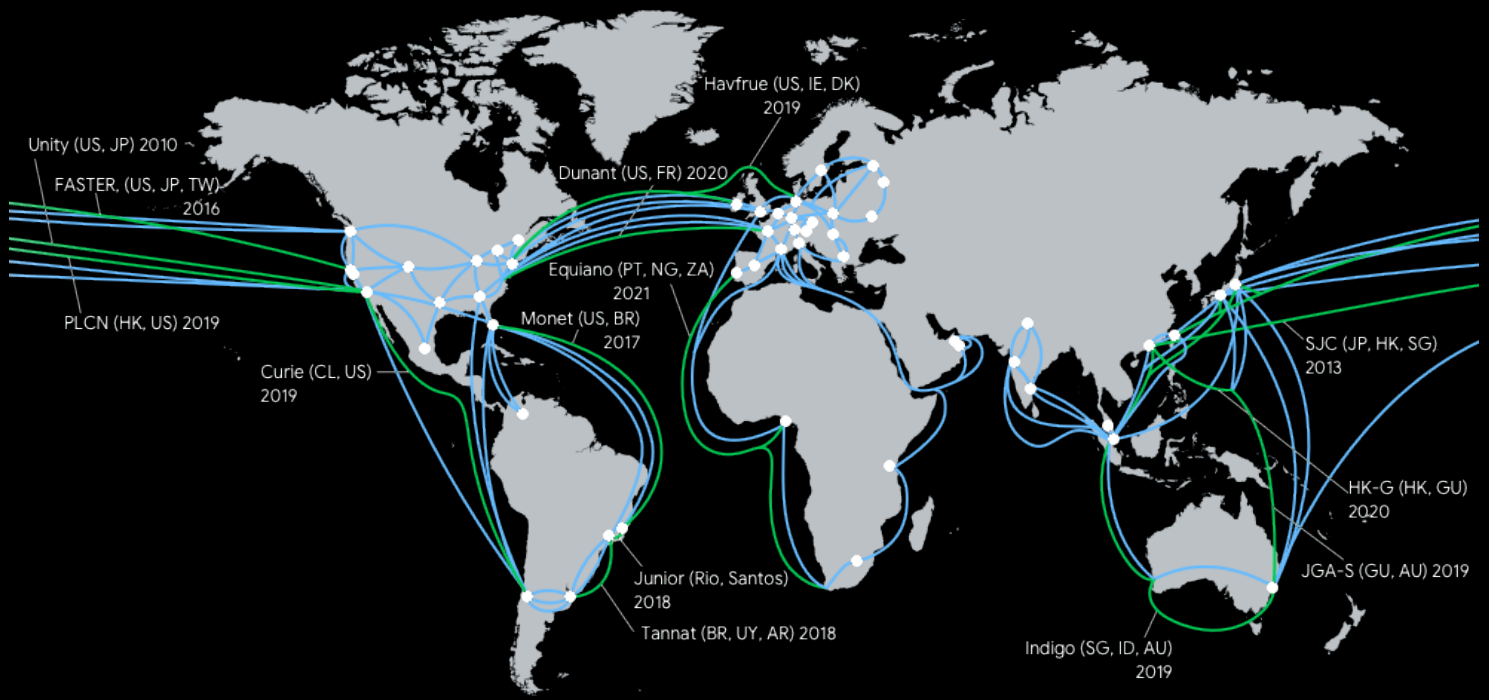
- Deployment areas for GCP resources within a region.
  - Multiple individual data center buildings in the geographical region.
- Typically **3 or more per region**.
- Considered a "single resource failure domain."
  - For fault tolerance, it is best to deploy applications across multiple zones (and regions, where applicable).
- Examples: us-central1-a, us-central1-b, asia-east2-a

Region	Zones	Location
asia-east1	a, b, c	Changhua County, Taiwan
asia-east2	a, b, c	Hong Kong
asia-northeast1	a, b, c	Tokyo, Japan
asia-northeast2	a, b, c	Osaka, Japan
asia-south1	a, b, c	Mumbai, India
asia-southeast1	a, b, c	Jurong West, Singapore
australia-southeast1	a, b, c	Sydney, Australia
europe-north1	a, b, c	Hamina, Finland
europe-west1	b, c, d	St. Ghislain, Belgium
europe-west2	a, b, c	London, England, UK
europe-west3	a, b, c	Frankfurt, Germany
europe-west4	a, b, c	Eemshaven, Netherlands
europe-west6	a, b, c	Zürich, Switzerland
northamerica-northeast1	a, b, c	Montréal, Québec, Canada
southamerica-east1	a, b, c	Osasco (São Paulo), Brazil
us-central1	a, b, c, f	Council Bluffs, Iowa, USA
us-east1	b, c, d	Moncks Corner, South Carolina, USA
us-east4	a, b, c	Ashburn, Northern Virginia, USA
us-west1	a, b, c	The Dalles, Oregon, USA
us-west2	a, b, c	Los Angeles, California, USA

Back Next

Edge Point of Presence (POP)

- Where Google's network connects to the rest of the Internet.
  - "Interconnects with other networks"
- Over 130 exchange points exist around the world.



Back Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

So What's the Point?

- GCP network is **global** in scope, and the **default** mode of operations.
- All traffic between regions (and within POP network) is on Google's private network
  - i.e., a **global private network** (never touches the public Internet).
  - Result: better security, routing, and performance.
- GCP networking resources privately communicate all over the world by default.

Back

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

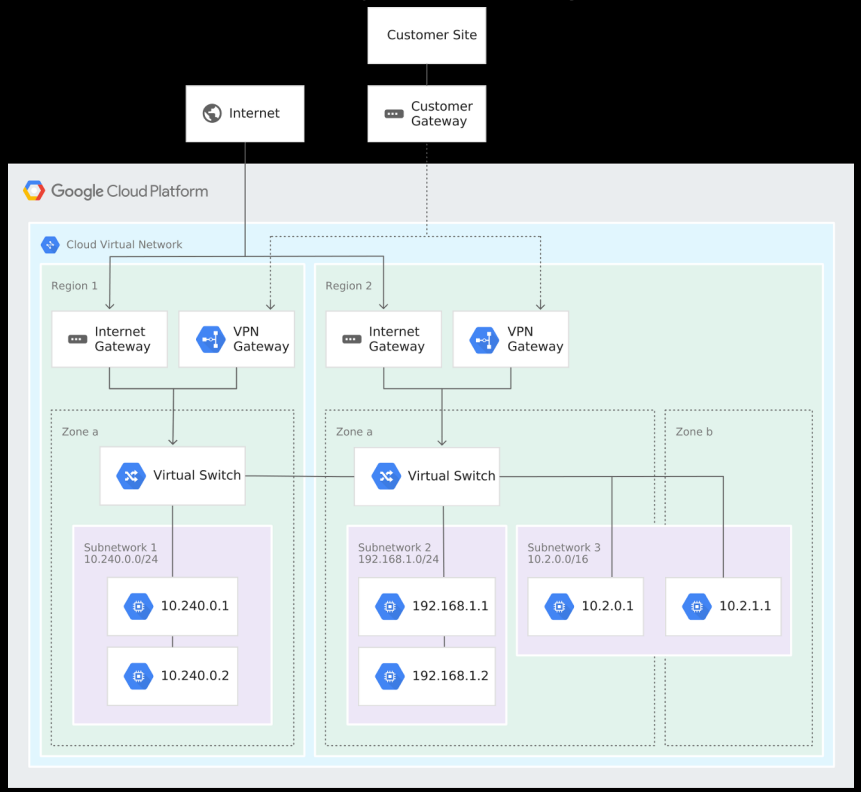
Back to Main

Big Picture Facts

- **Central foundation** of all other networking functions on GCP
- VPC = **Software Defined Network (SDN)**
  - Traditional network = multiple hardware components (routers, servers, switches, load balancers, firewall devices, device configurations, etc.)
  - Hardware management is abstracted away
  - Removes maintenance and overhead
  - Rapidly customize and scale services
  - Traditional networking concepts apply
    - Firewalls, routes, load balancing, subnets, DNS, etc.
- Global (multi-regional) communications space, private communication among resources
  - **RFC 1918** - Private (internal) networking and IP addressing standard
    - Internal/Private IP addressing — not exposed to public Internet
- Hybrid networking with on-premises networks that have interconnect options
- Can configure private (internal-only) access to other GCP resources
- Incoming (ingress) traffic is free and outgoing (egress) traffic has a cost

Next

Example VPC Diagram (with Hybrid Networking)



Back Next

Course Navigation

Core Concepts
Section 2

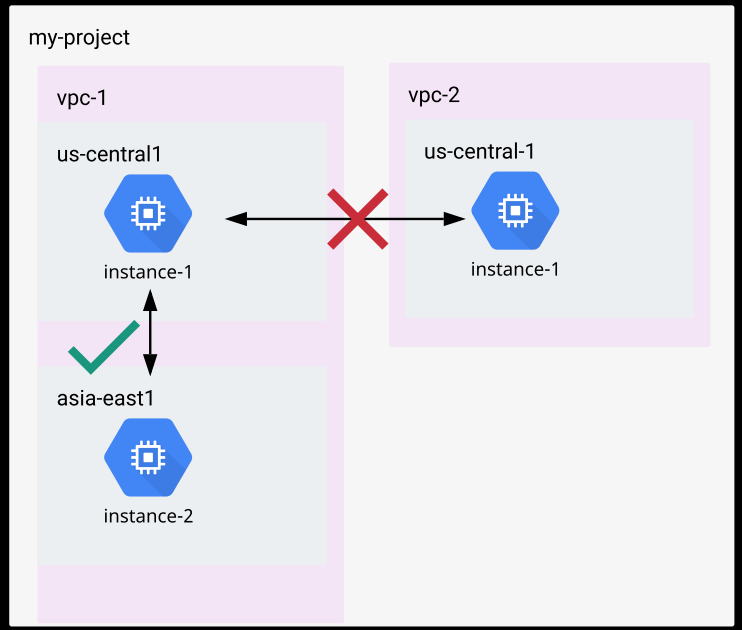
GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

VPCs and Projects

- Projects:
  - Primary resource and billing isolation construct
  - Hold one or more VPCs per project
  - VPC exists within a single project (with shared VPCs being an exception)
  - By default, can have up to five separate VPCs per project (is increased via quota management)
- Projects separate users, whereas VPCs separate systems



Back

Subnets on GCP VPCs

- VPCs do not come with an associated IP range (must create subnets)
- Subnet = a logical network partition
  - Private IP ranges
    - RFC 1918 private IP ranges (10.x.x.x/172.16.x.x/192.168.x.x)
  - Multiple “subnetworks” inside of a larger single network
  - **Subnetting** = dividing network address space to match an organization’s internal network needs
  - On GCP — designated using CIDR notation for network/host division
    - Example: ‘subnet-a’ = 10.0.1.0/10

Next

Course Navigation

Core Concepts
Section 2

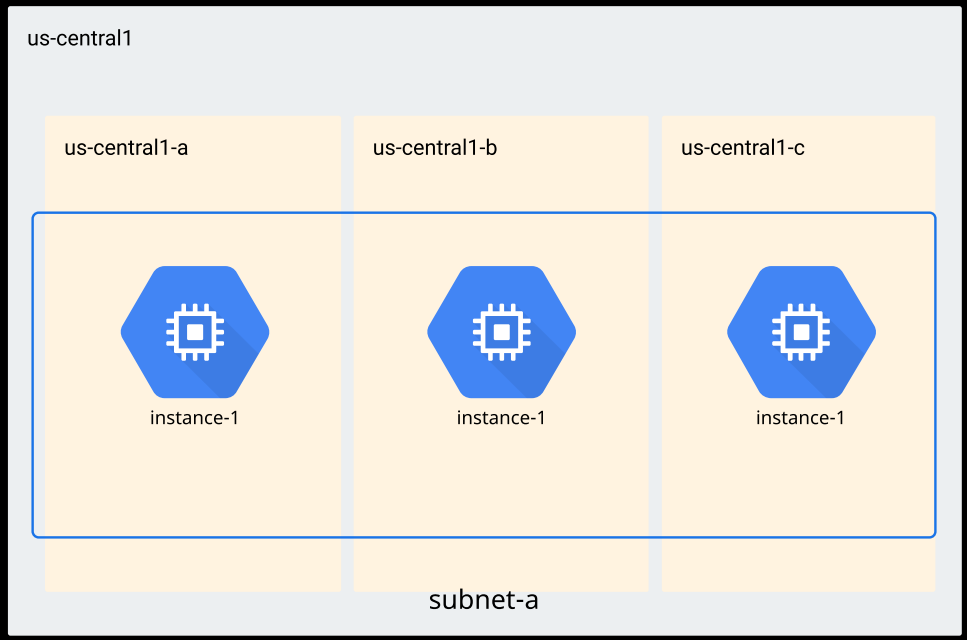
GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

VPC Structure - Subnets/IP ranges

- VPC can have one or more subnets
- Subnet = region based
  - Subnet can span zones in same region



Back
Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

GCP Subnet Modes

- **Default, Auto Mode, Custom**
- **Default** = created with every new GCP Project
  - Auto-mode network + pre-made firewall rules
- **Auto Mode Network** = automatically created subnet for each region
  - One subnet for every region
    - Subnet range of 10.x.x.x/20 per region
    - Get up and working quickly
  - Can manually add additional subnets or convert to custom mode
  - Why use auto mode?
    - Easy to set up and use
    - Predefined IP ranges don't overlap with each other
  - Why not to use auto mode?
    - Not as flexible as custom mode
    - Don't need subnet for each region
    - Connecting two different VPCs (VPN/network peering) = overlapping subnets
    - Often not suitable for production networks
- **Custom Mode Network**
  - No subnets automatically created - "blank slate"
  - Much more flexible
  - "Build Your Own Network"
- VPC mode conversions - one way only
  - Can covert auto mode to custom mode, but not vice versa

Back Next

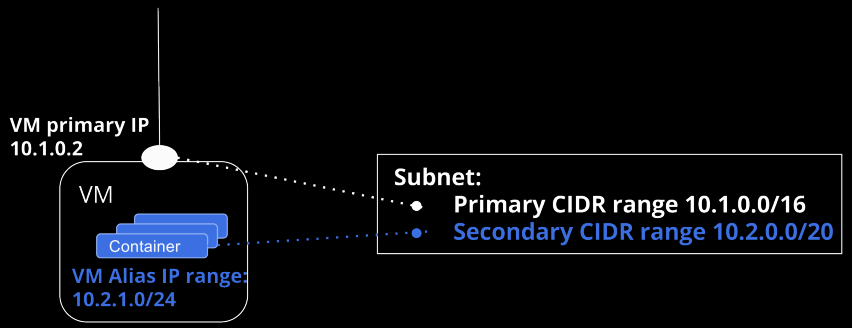
Reserved IP Addresses

- Like traditional networks, subnets have the first and last two IPs in range reserved
- First: network address
- Second: default gateway
- Second to last: future use address
- Last: broadcast address

Subnet - 10.1.2.0/24	
10.1.2.0	Network
10.1.2.1	Default Gateway
10.1.2.254	Future Use
10.1.2.255	Broadcast

Address Ranges

- **Primary Address Range**
  - Any private RFC 1918 CIDR block
  - VM primary internal IP addresses, alias addresses, internal load balancers
- **Secondary Address Range**
  - Only for alias IP ranges
  - Useful for containers or multiple VMs on a single machine
-



Back Next

Back to Main

Linux Academy

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Beware Overlapping Subnet Ranges!

- Just like traditional networks, you cannot have two subnet ranges overlap
- Considerations:
  - Subnets in same VPC
  - Subnets in multiple peered VPCs
    - Two auto-mode VPCs cannot be peered
  - Subnets in external, interconnected networks
  - Larger subnet range (e.g. /10 address) can conflict with smaller range subnets that fall in the same range

Expanding Subnets

- If your subnet range is too small, you can expand it
  - Expands only, cannot shrink
  - Example: Expand from /20 to /16 subnet

Back Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

VPC Hands On Demo

What we will cover:

- View default network
- Create auto mode network
  - Convert to custom mode network
- Create custom mode network with custom subnets
- Expand a subnet
- Throughout demo - explore Google Cloud SDK commands

Anatomy of a gcloud command

```
gcloud compute networks create my-custom-network \
  --subnet-mode=custom
```

What we are doing

Details of how it will be done

```
gcloud compute networks subnets create subnet-a \
  --network=my-custom-network \
  --region=us-central1 \
  --range=10.1.2.0/24
```

What we are doing

Details of how it will be done

Back

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

IP Addresses on for GCP Resources

- Every VM needs an IP address to communicate
- Must have at least one internal address, external address is optional

Internal Addresses

- Assigned from internal DHCP pool
  - Ephemeral remains even when stopped
- Can assign static internal address
  - Within assigned subnet range
- Internal DNS format (FQDN): **(instance-name).c.(project-id).internal**
  - Example: 'my-instance.c.test-project.internal'

External Addresses - Types

- **Ephemeral** - temporary
  - Assigned when resource created/running
  - Released when stopped/deleted
- **Reserved (Static)** - preserved through stoppage
  - Bound to specific region (e.g., us-central1)
  - Billed when not attached to VM
- Fun fact: VM doesn't know external IP - mapped to internal IP
- No default public DNS, need to use DNS services like Cloud DNS

Next

Course Navigation

Core Concepts
Section 2

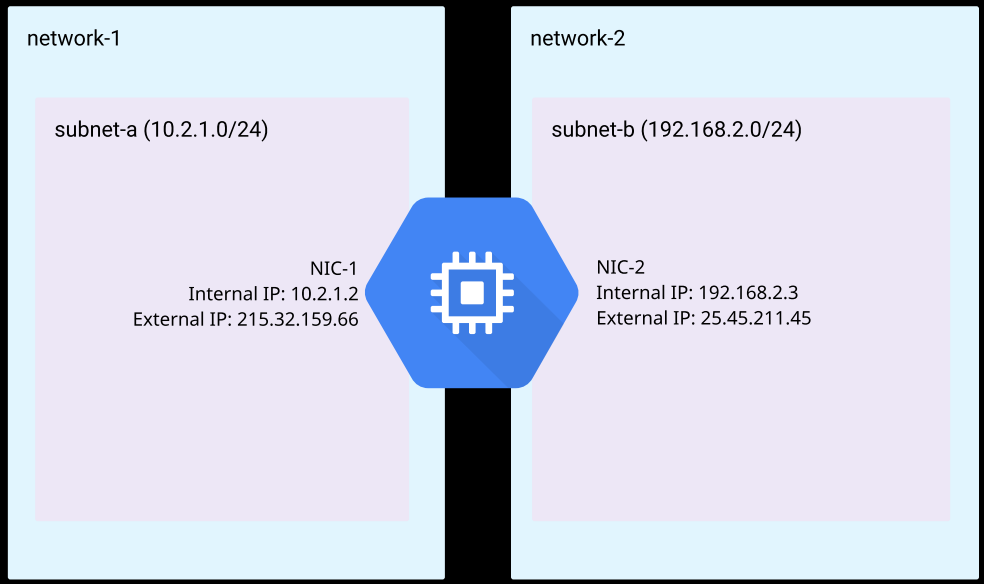
GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Multiple Network Interface Controllers (NICs)

- VM can have exactly one internal + external IP per VPC
- VM can have multiple network interfaces (one per VPC), with one internal/external per NIC
- In other words, a single VM can be on multiple VPCs, with single internal/external address for each VPC



Back
Next

Why Multiple IPs/Interfaces?

- Instance as network appliance
  - Web intrusion/firewall
  - WAN optimization
- Applications require traffic separation
  - Data plane traffic from management plane traffic

Multiple NIC Limitations

- Must set up on instance creation
- 1 NIC per network
- Cannot overlap subnet ranges
- Cannot delete interface
- Up to 8 NICs total
  - <= 2 CPUs = 2 NICs
  - > 2 CPUs = 1 NIC per CPU (up to 8)

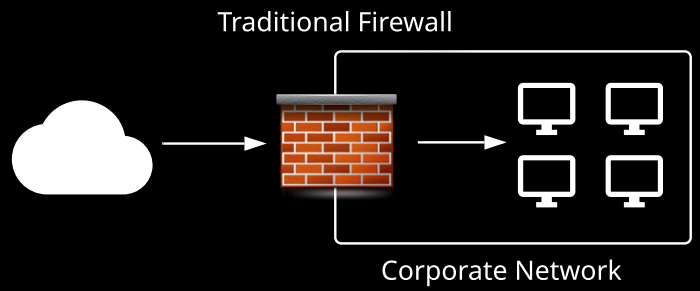
Back

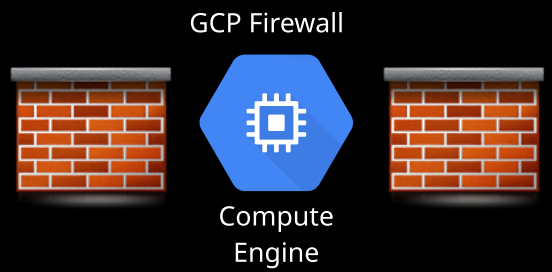
What We Will Cover

- Create an instance
  - View network settings
  - View ephemeral external IP behavior
- Reserve an external IP address
- Create two custom VPC's and a single two NIC instance
- Command to set up two VPC environments provided in lesson description

Firewall Basics

- Allow/deny traffic to and from instances
  - Based on configuration
- Manage both inbound (ingress) and outbound (egress) traffic
- Defined at network (VPC) level, but enforced for each instance





Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Firewall Rules

- Rules manage external access and also access between internal resources
- Implied 'deny all' ingress
- Implied 'allow all' egress

Firewall components:

Direction	Ingress (Incoming) or Egress (Outgoing)
Target	GCP Resources the rule applies to: Entire Network, Target Tags, Service Account
Source/Destination Filter	Incoming Sources (Ingress) or Outgoing Destinations (Egress) that rule applies to
Action on Match	Allow or Deny
Protocol/Port	Protocols and Ports that are allowed/denied
Priority	Priority to give overlapping/conflicting rules Lower number "wins" (has higher priority)

Network Tags

- Instance-level, granular enforcement
- Apply tag to instance
- Rule is enforced on only tagged instances and not entire network

Back
Next

Course Navigation

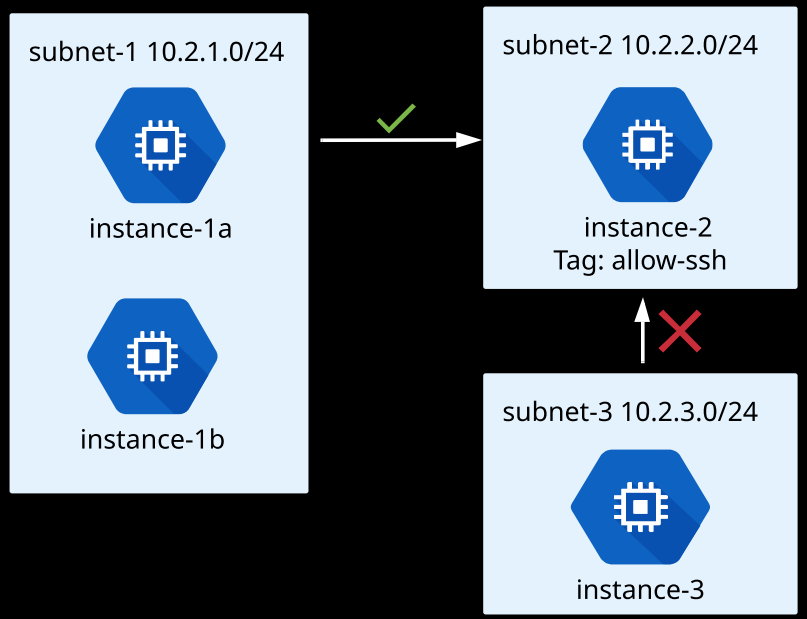
Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

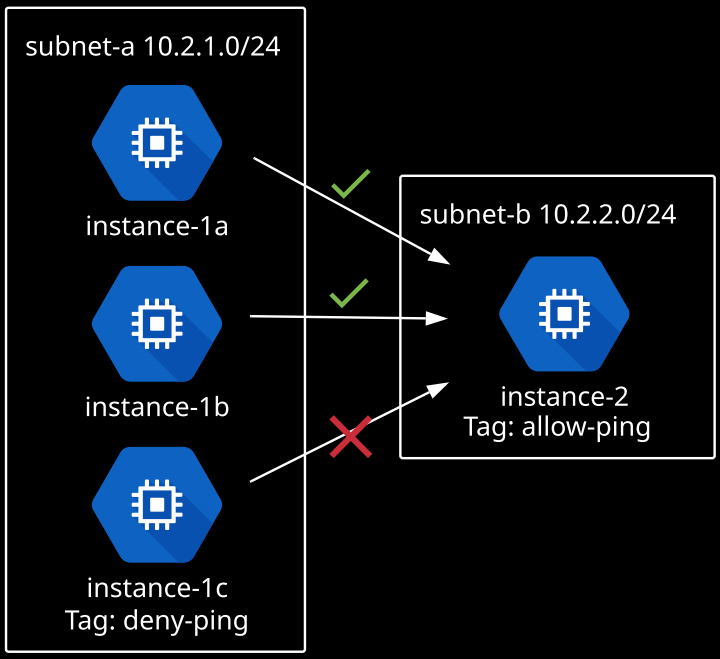
- Firewall rule:
  - Allow port 22 access
  - Target: network tag - 'allow-ssh'
  - Source filter: 10.2.1.0/24
- Result:
  - Instances in subnet-1 can SSH to instance-2, but not instance-3 outside of subnet-1



Back

What We Will Cover

- Create a VPC with two subnets, with instances placed as described below
  - instance-2 will have `allow-ping` network tag applied
  - instance-1c will have `deny-ping` network tag applied
- Create a firewall rule to allow SSH (TCP:22) access to entire network
- Create a firewall rule to allow ICMP (ping) to instances with `allow-ping` tag from all instances in subnet-a
- Create a second rule to *deny* ICMP from instance-1c and giving it higher priority
- An attempt to ping instance-2 from all three instances in subnet-a
  - All should work except for instance-1c
- Create egress rule to deny ALL ping (ICMP) traffic from instance-1c



Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Routing on GCP

- Defines paths of network traffic from a VPC resource (i.e., an instance) to other destinations
  - Each route has a single **destination** + single **next hop**
  - Both within and outside of the VPC
  - Allows VPC resources to communicate with each other
- Most GCP services automatically create **system-generated** routes
  - Created and removed for you (as needed)
- Traffic needs to match firewall rules to be delivered
- Many options similar to firewall rules
  - Priority: smaller number has a higher priority
  - Apply to entire network or network tagged instances

System-Generated Routes

- **Default Route**
  - Default route out of your VPC to the external Internet (0.0.0.0/0)
  - Provides pathing for **Private Google Access**
  - Can be deleted to isolate VPC from Internet, or replace with custom route
- **Subnet Route**
  - Define destination path for each VPC subnet ("everything finds everything")
  - Cannot be deleted
  - Cannot create custom routes that are more specific than subnet route destinations
    - To restrict traffic between subnet/resources, use firewall rules

Custom Routes - What You Create

- **Static and Dynamic Routes**
  - Routes you create, or get created by Cloud Router
  - Configure your own destination (next hop)

Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

When to Use Custom Routes

- Manually created network appliance
  - NAT Gateway, Load Balancer, Firewall
  - Per Google, consider GCP managed services instead

Instance as "Next Hop"

- Must enable IP forwarding on instance
  - Must configure on instance creation

- private-instance routes outbound Internet gateway traffic through nat-gateway
- nat-gateway has IP Forwarding enabled
- Destination = 0.0.0.0/0
- Next Hop = nat-gateway
- Route applies only to "no-ip" tagged instances

Back

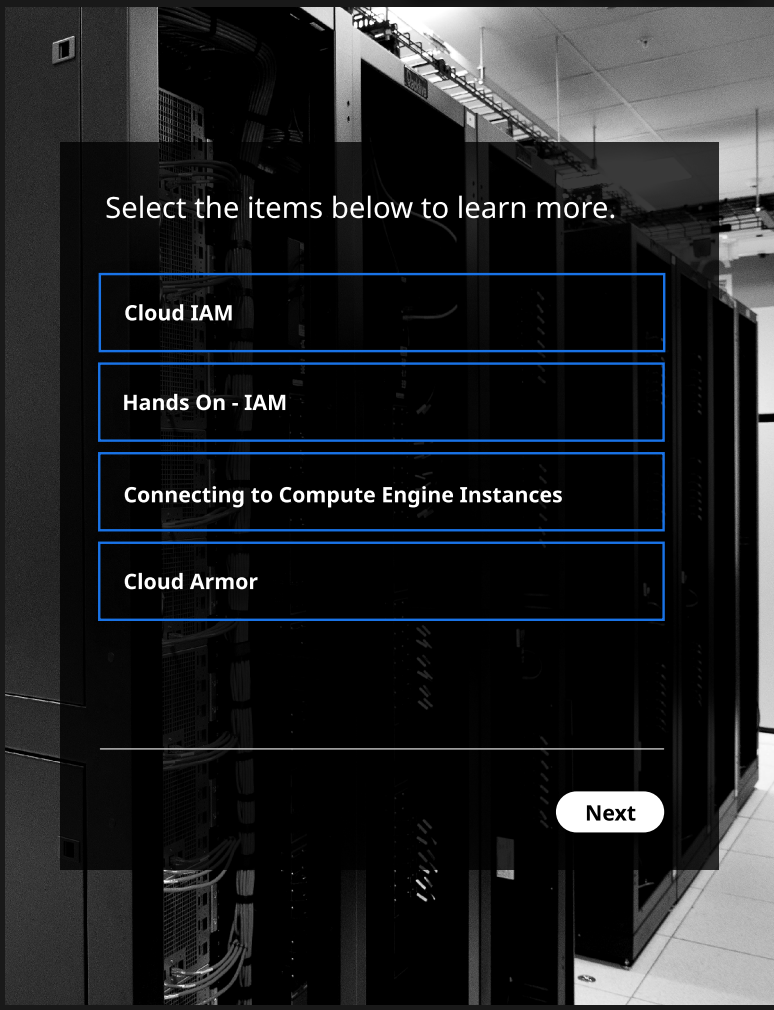
What We Are Doing

- Creating a custom route for a NAT Gateway
- Put other section concepts into practice as well
- This time, hands on demo will use **gcloud commands** for setup
  - Web console for custom route for context
  - All commands in lesson description

The Steps

- Delete default VPC (optional but recommended to clean up views)
- Create custom mode VPC
- Create subnet
- Create firewall rules
- Create NAT Gateway instance
  - Enable IP Forwarding
- Create private instance
  - Apply "no-ip" network tag
- Create custom route to forward traffic through NAT Gateway
  - Only apply to "no-ip" tagged instances

- private-instance routes outbound Internet gateway traffic through nat-gateway
- nat-gateway has IP Forwarding enabled
- Destination = 0.0.0.0/0
- Next Hop = nat-gateway
- Route applies only to "no-ip" tagged instances



Select the items below to learn more.

- Cloud IAM
- Hands On - IAM
- Connecting to Compute Engine Instances
- Cloud Armor

Next

Topics for this section

- Cloud IAM
- Connection methods to Linux instances (SSH)
- Cloud Armor

Security of VPC Resources is Multi-layered

- **Cloud IAM:** Restrict **who** can access resources - regardless of location
  - **Projects** are primary separator
- **Firewalls** restrict by traffic type and location - regardless of who
  - HTTP, SSH, RDP
  - From entire Internet or single location
- **Restricting Connection Methods** provide granular control of who can connect to Linux instances
  - Internal and External users?
  - Manage own keys or let Google do it for you?
- **Cloud Armor** prevents DDoS attacks and other malicious traffic
  - Block malicious traffic at the edge from even reaching your resources

Next

Course Navigation

Core Concepts
Section 2

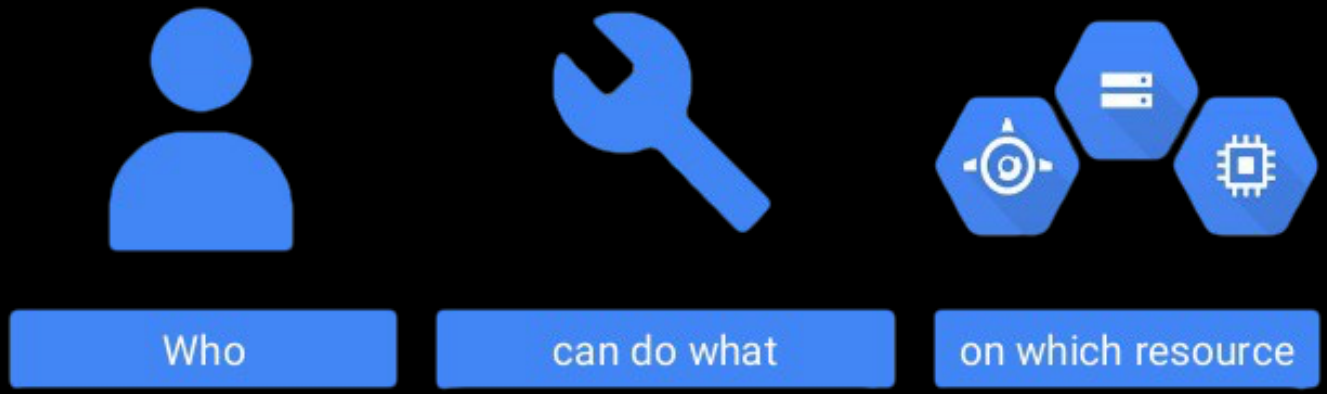
GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

What is Cloud Identity and Access Management (IAM)?

- Technical definition
  - With Cloud IAM, you can grant granular access to specific GCP resources and prevent unwanted access to other resources. Cloud IAM lets you adopt the security principle of least privilege, so you grant only the necessary access to your resources.
- Simple breakdown
  - Who = **Member**
  - Can do what = **Role**
  - On which resource = All **resources** on GCP



Back
Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Members - "Who"

- End Users and Applications/VMs
- **End Users = Google Account**
  - Personal Gmail or Google connected account
    - matt@gmail.com
  - G Suite/Cloud Identity organization (primary course focus)
    - matt@linuxacademy.com
  - Google Group = collection of individual Google and Service accounts
    - my-group@mycompany.com
- **Applications/VMs = Service Account**
  - Given to applications/code run on GCP
  - Not reliant on end user remaining part of organization
  - Email address format
    - 300965514785-compute@developer.gserviceaccount.com

Back Next

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Roles - "Can do what - with which resource"

- Bundles of permissions assigned to members ("who")
- Permissions are not directly assigned, but bundled into roles, which are directly assigned

Role - Compute Network User

compute.networks.get	compute.interconnects.get
compute.networks.list	compute.interconnects.use
compute.networks.use	compute.regions.get
compute.addresses.get	compute.regions.list
compute.firewalls.list	...

Back Next

Course Navigation

Core Concepts
Section 2

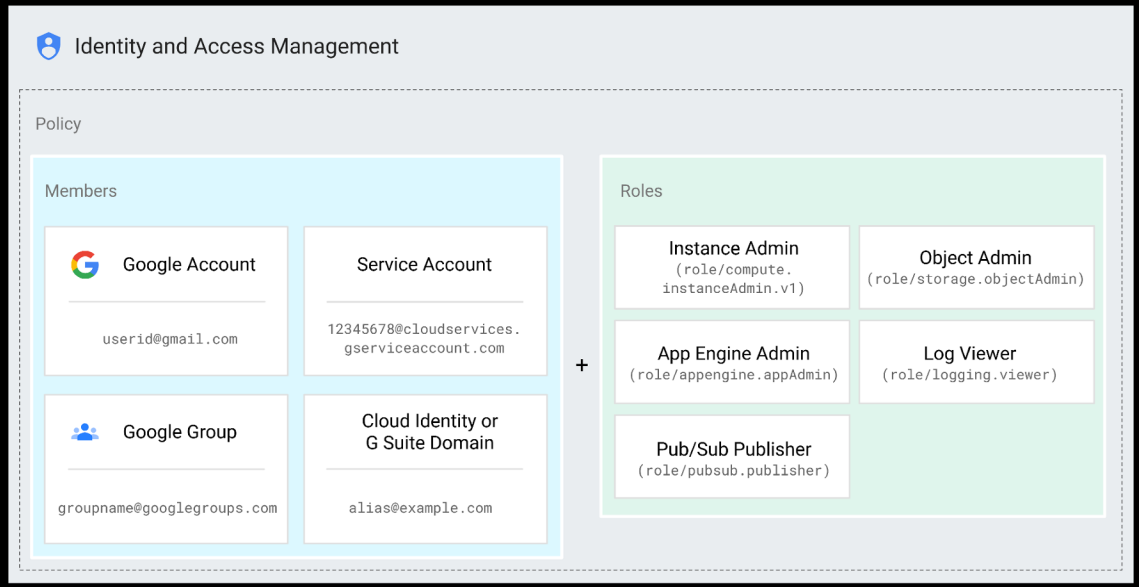
GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Role Types

- **Primitive**, **Predefined**, **Custom**
- **Primitive**: Project-wide
  - Owner: Full access + edit roles/billing
  - Editor: Full access without ability to edit roles/billing
  - Viewer :Read-only access
- **Predefined**: Granular, per-service/resource permissions
  - Example: Compute Network Admin = Only admin rights to network resources
  - Example: Storage Admin for bucket "gs://my-bucket" - admin rights only to single bucket
- **Custom**: Even more granular
  - Mix and match your exact permissions needed



Back Next

Course Navigation

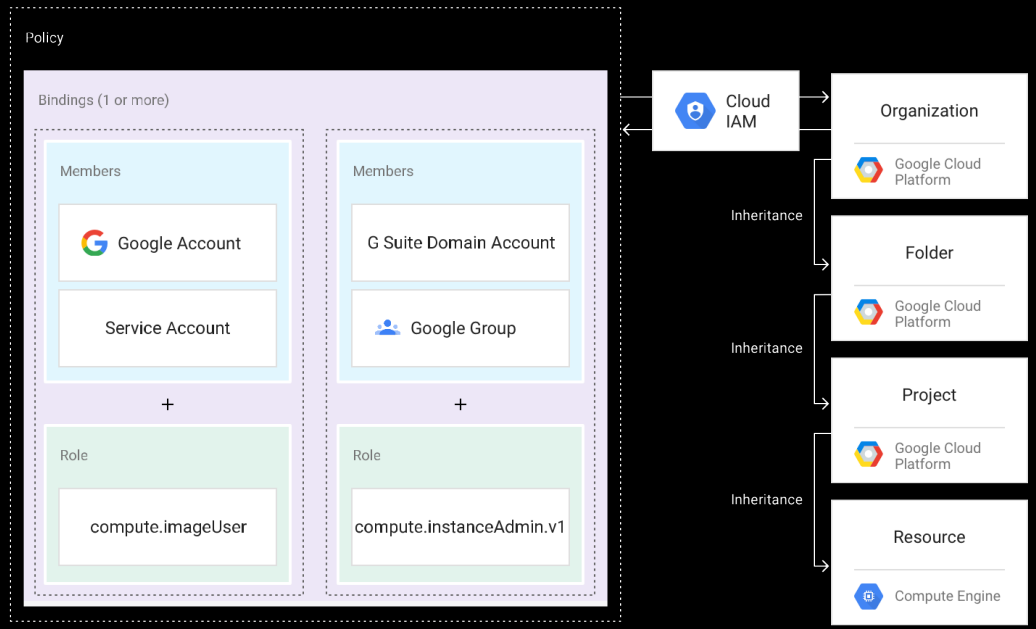
Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

IAM Policy

- How roles are granted to users
- Collection of statements that state who has access to what resources
- Policy is attached to a resource, which enforces access control
- **Policy Hierarchy**: Policies enforced from higher levels and down. More permissive higher-level role is applied to lower level



Back Next

Back to Main

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Networking Perspective

- Networking roles under Compute Engine API
- Networking roles are applied **project-wide** - across all VPCs in project
- Compute Admin role: Full access to all Compute, Networking, and Network Security (Firewall, SSL certs) services
- Other relevant Compute/Networking roles:

Network Admin	Create, modify, delete all network resources EXCEPT firewall rules and SSL certificates
Security Admin	Create, modify, delete security items Firewall rules, SSL certificates/policies Cannot create network resources
Network User	Use Network resources Cannot create/delete
Network Viewer	Read-only access to networking resources
Compute Instance Admin	Create, modify, delete all GCE resources

Back
Next

Back to Main

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

Back to Main

Editing IAM Policy via Command Line

- Two methods:
  - Edit entire exported policy
  - Direct additions/removals
- Edit exported policy
  - Export policy (JSON or YAML format)
  - Edit policy
  - Apply edited policy in place of old one
- Direct update
  - gcloud command - quickly grant or remove a role without editing entire policy
  - `gcloud [GROUP] add(or remove)-iam-policy-binding [RESOURCE] --member user:[EMAIL] --role [ROLE_ID]`
  - Example: `gcloud projects add-iam-policy-binding project-123 --member user:matt@linuxacademy.com --role roles/owner`

Back

What we will cover

- Viewing and adding roles
- View permissions attached to roles
- Create custom role
- Edit roles using command line

Editing roles with command line

- **Multiple updates - edit entire policy**
  - Getting/downloading current policy is optional, can always set fresh policy without downloading old policy first
  - **Get** (download) copy of policy
    - `gcloud projects get-iam-policy [PROJECT_ID] --format [FORMAT] > [FILE-PATH]`
    - JSON or YAML format
  - Add/edit/remove members in downloaded policy file
  - **Set** new policy by applying edited one
    - `gcloud projects set-iam-policy [PROJECT_ID] [FILEPATH]`
    - Provide file path to file you just edited
- **Directly add/remove single IAM role**
  - Does not affect other assigned roles in project
  - `gcloud projects add-iam-policy-binding [RESOURCE] --member user:[EMAIL] --role [ROLE_ID]`
  - `gcloud projects remove-iam-policy-binding [RESOURCE] --member user:[EMAIL] --role [ROLE_ID]`
  - For organization-level changes, swap out `gcloud organizations` for `gcloud projects`

Connecting to GCE Instances

- Must be able to connect to GCE instances
- Windows and Linux instance's connection authorization handled differently

Windows

- Connect over RDP (TCP:3389)
- Relatively simple. Credentials/user accounts are handled entirely by local instance
  - Independent from Cloud IAM roles
- Cloud IAM relevance = Compute Instance Admin role to reset password

Linux

- Two connection methods
  - Google-managed
    - Tied to **Cloud IAM** roles
    - Connect via Google Cloud SDK
      - `gcloud compute ssh`
  - Manage own SSH public/private keys
    - Independent from IAM

Next

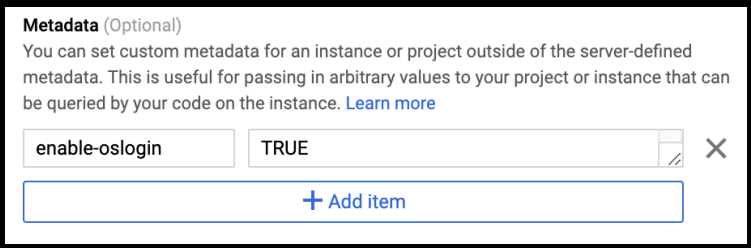
Back to Main

Google-Managed Method

- Don't have to create and store own SSH keys
- gcloud command for all scenarios
  - `gcloud compute ssh (instance-name) --(zone)`

IAM Roles to Connect

- **Compute Instance Admin**
  - May be too broad
    - Full permissions to create/delete GCE resources
    - Connect to all instances in project
- **OS Login role**
  - Only able to connect to instances
  - Allows per-instance or project-wide access
    - Enable OS Login in **metadata**
    - Can apply to entire project or individual instances
  - Required IAM roles to connect - **OS Login** and **Service Account** User
  - External (GCP) users - use **OS Login External User** role
    - Allows GCP accounts outside your organization to connect



Manual Key Management

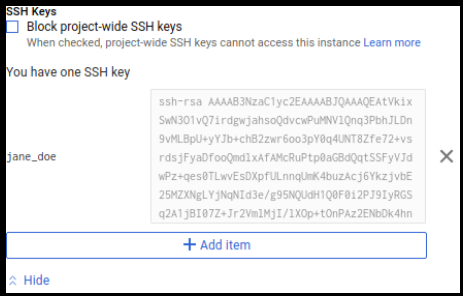
- Officially not recommended, but still an option
- Independent of IAM roles/Google Cloud SDK
- You are responsible for SSH key management
- Manually generate private/public keys
  - Add public keys to either project or per-instance metadata

Why use manual key management?

- Connect with third-party tools that don't integrate with GCP
- Allow connection by users outside of project

Generate Private/Public Key

Add Public Key to Instance/Project Metadata



Back

Course Navigation

Core Concepts
Section 2

GCP Networking Fundamentals
Google Cloud Networking Infrastructure
What is a Virtual Private Cloud (VPC)?
Subnets
IP Addresses
Hands On - IP Addresses
Firewalls
Hands On - Firewalls
Routing
Hands On - Routing
Securing Your VPC Networks
Cloud IAM
Hands On - IAM
Connecting to Compute Engine Instances
Cloud Armor

Growing Your Network
Section 3

What is Cloud Armor?

- Protect services from denial of service and other web attacks
- "Armor" for your applications
- Paired with load balancers - more details later in course
  - Enforced at Edge POP's - deny malicious traffic close to the source

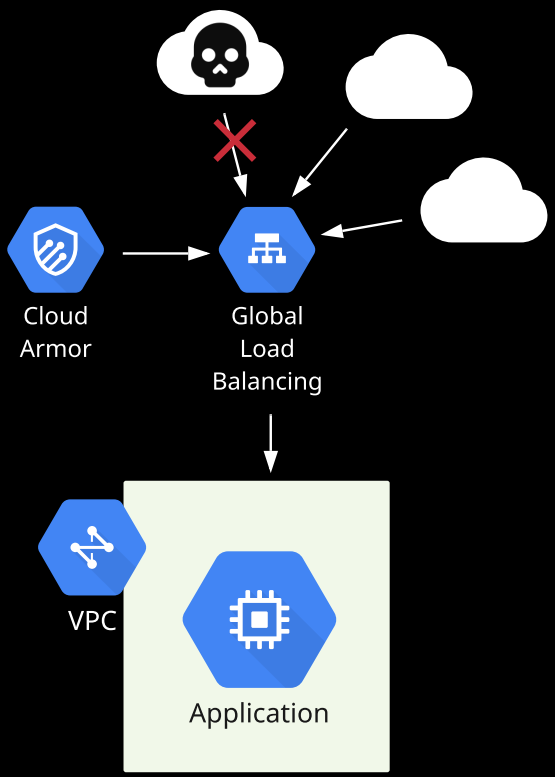
Features

- Some features available now, others coming in the future
- IP Whitelist/Blacklist controls
  - Allow access from some IP's and deny others
- Distributed Denial of Service (DDoS) attacks
- SQL Injection - Cross Site Scripting
- Geo-based Access Control
  - Enforce access based on geographic location

Next

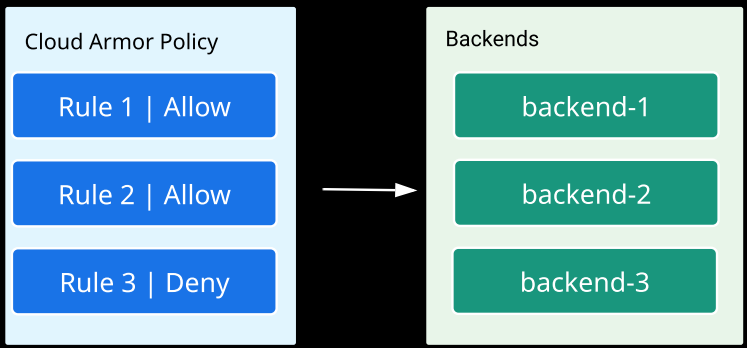
Back to Main

Enforcement at Edge POP's



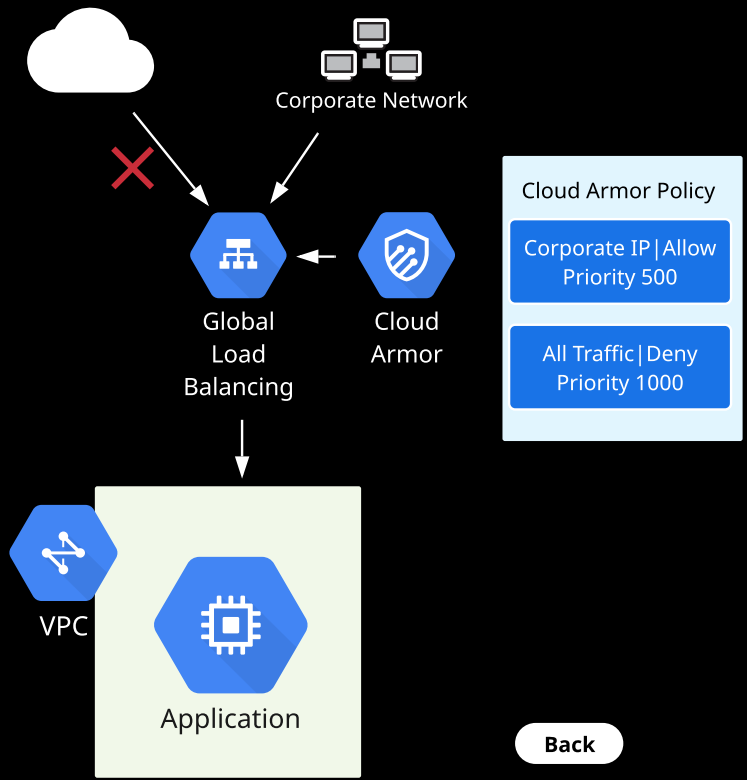
How it works

- Create a policy with 1+ rules
  - Supply IP range(s) to apply rule to
  - On rule/IP match: Allow/Deny traffic
    - Overlap rules with different priorities
  - Apply policy to Targets
    - Target = Load balanced backend services



Use case - Restrict traffic to corporate network

- Application requires public IP, but you only want traffic from corporate network to reach application
- Deny all traffic except for corporate IP range



Back

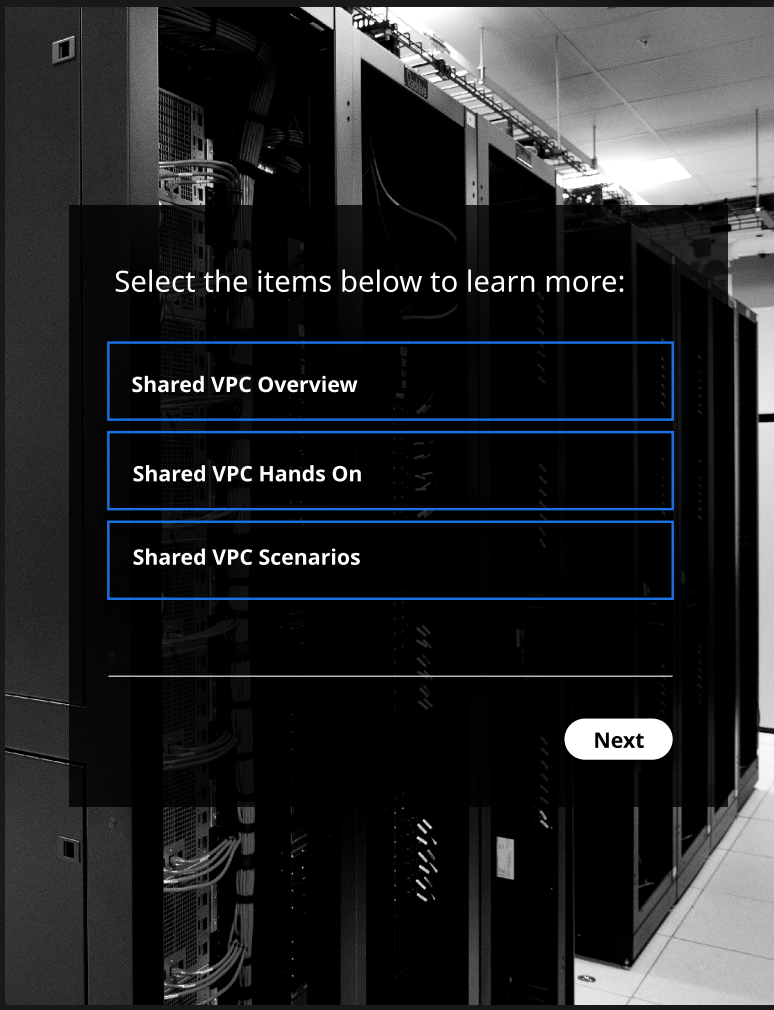
Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hands On
Shared VPC Scenarios

VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing



Select the items below to learn more:

Shared VPC Overview

Shared VPC Hands On

Shared VPC Scenarios

Next

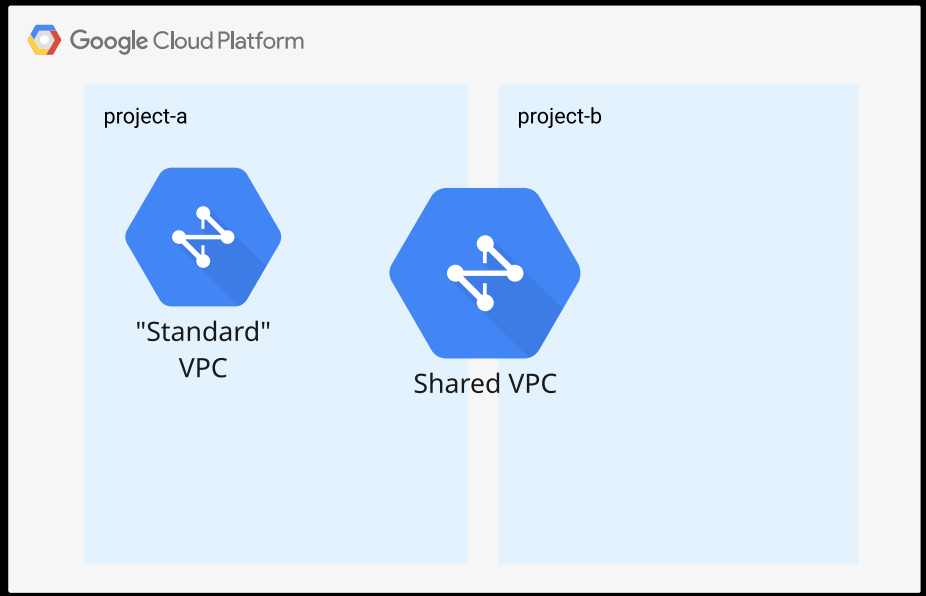
Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hands On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

What is a Shared VPC?

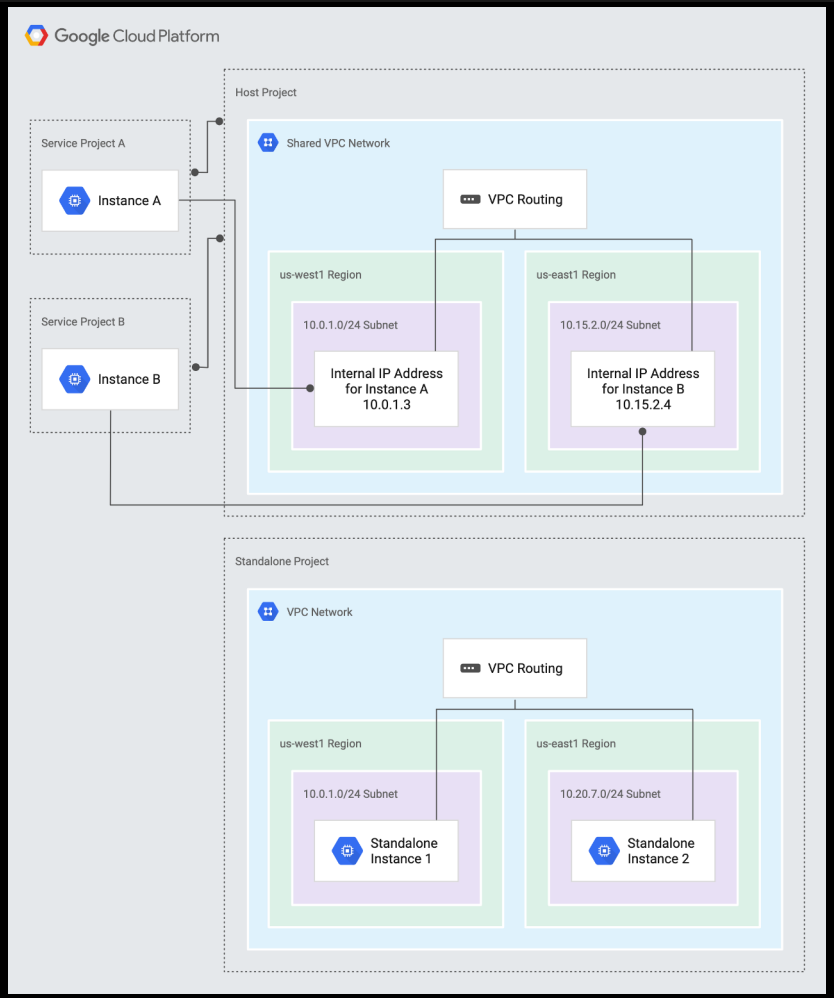
- So far in this course, VPCs have been in a single project
- Need may exist to enable private network communication and access across projects
- Shared VPC shares a VPC across multiple projects within an organization
  - i.e., Cross-Project Networking – also its former name



Next

Concepts and Terminology

- **Host project**: Project hosting the shared VPC
- **Service project**: Project with permission to shared VPC
  - Shared VPC projects can be controlled by different departments
  - Ownership of resources in shared VPC maintained by project
- **Standalone project**: Project not using shared VPC (the default)



Back Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hands On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Shared VPC IAM Roles

- Creating Shared VPC requires organization or folder-level roles
- Organization/Folder Administrator:
  - Assign Shared VPC Admin roles
  - Create organization policy to prevent project deletion

Shared VPC Admin	Service Project Admin
• Assigned at organization or folder layer • Enables Shared VPC for host project • Attaches service projects • Assigns access to subnets shared by Shared VPC (Network User)	• Owner/Editor/Compute Instance Admin/Network User of Service Project • Assignment of **Network User** role in Host Project - allows Service Project users access to Host Project network/subnets • Can assign per shared subnet • Discover and use shared VPC assets

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hands On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Why separate projects to begin with?

- Why not place everything in the same project?
- Separation of projects for access control and billing
  - ...but still need access to same VPC resources
- Projects are primary method of separating access and billing

Considerations

- Only within **single GCP organization**
- Service project can only link to single host project
- Project cannot be both host and service project
- Existing projects can use shared VPC but existing instances cannot
- Reserved (static) IP addresses associated with (and billed to) project that reserved it

Resources that can use shared VPC

- Compute Engine Instances
- Compute Engine Instance Templates
- Compute Engine Instance Groups
- Google Kubernetes Engine clusters
- Internal IP Addresses
- Internal DNS
- Cloud DNS Private Zones
- Load Balancing

Back

Back to Main
Next Topics

Linux Academy

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Hands-On Guideposts

- Enable VPC admin account - org level
- View custom VPC in host project
- Enable shared VPC
- Share some subnet(s), but not all
- Specify project name(s) (attach service project)
- Select users as service project admins
  - Separate which subnets users have access to
- Go to service project, view shared VPC settings
- Create compute engine instance, attach to shared VPC
  - Can create instance in production subnet, but not development

Course Navigation

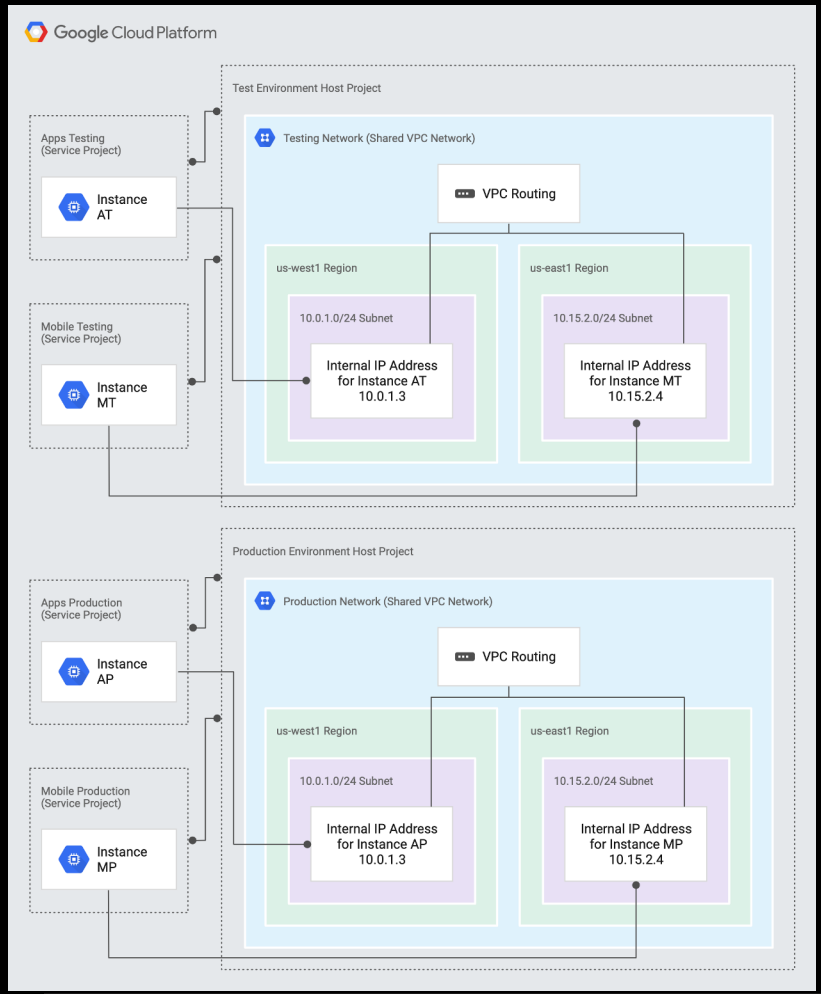
Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Testing and Production Environments

- Same subnet CIDR range and internal IP for testing and production equivalents

Next



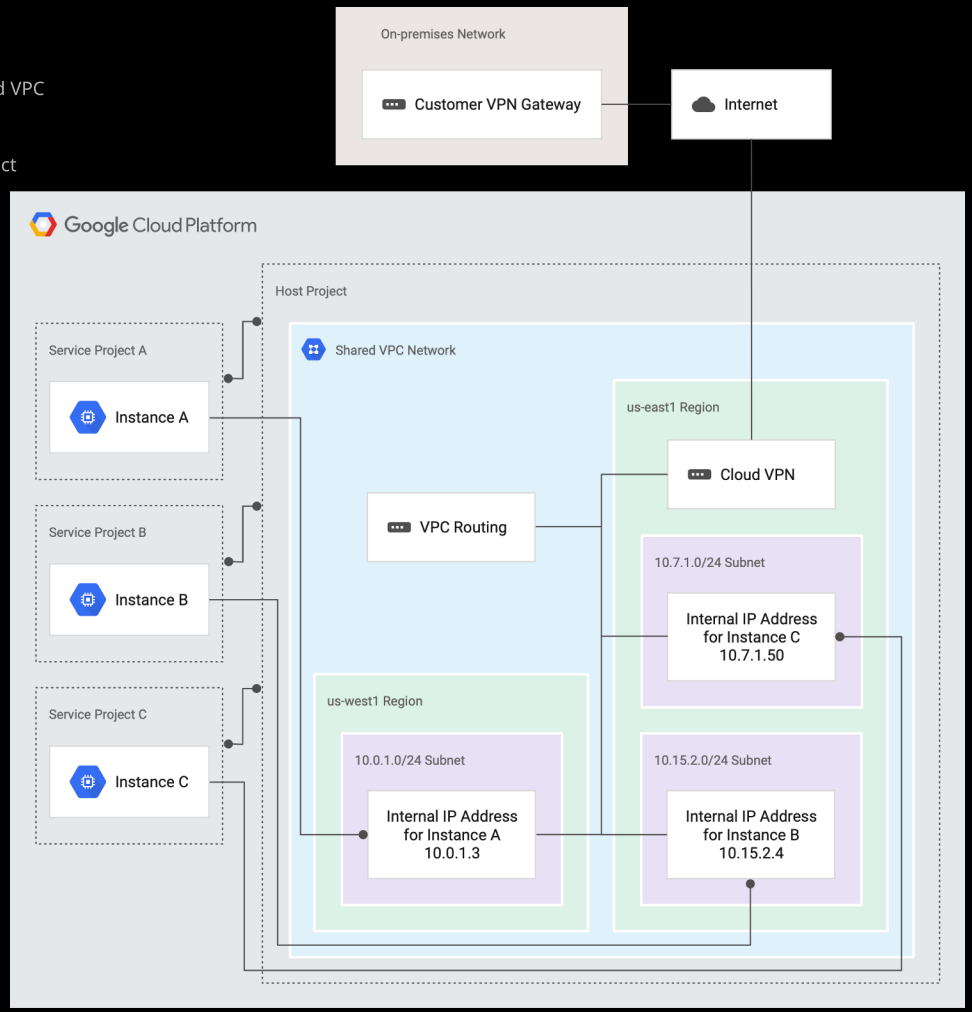
Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Hybrid Cloud

- VPN gateway connection to single shared VPC
- Resources worked on by different teams
- Access restricted by different projects
- Interconnect/VPN connects to Host Project



Back
Next

Back to Main
Next Topics

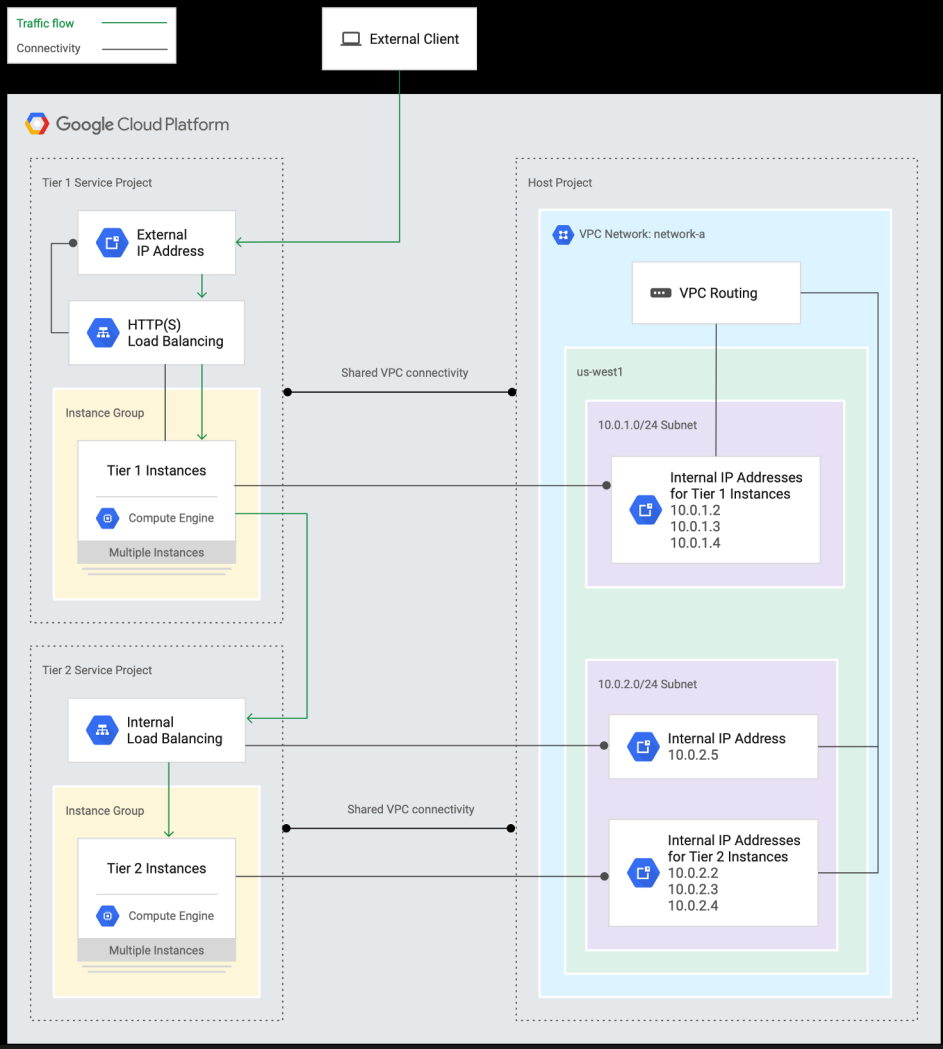
Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Two Tier Web Application

- Again, team access to specific tier separated by projects (same for billing)



Back Next

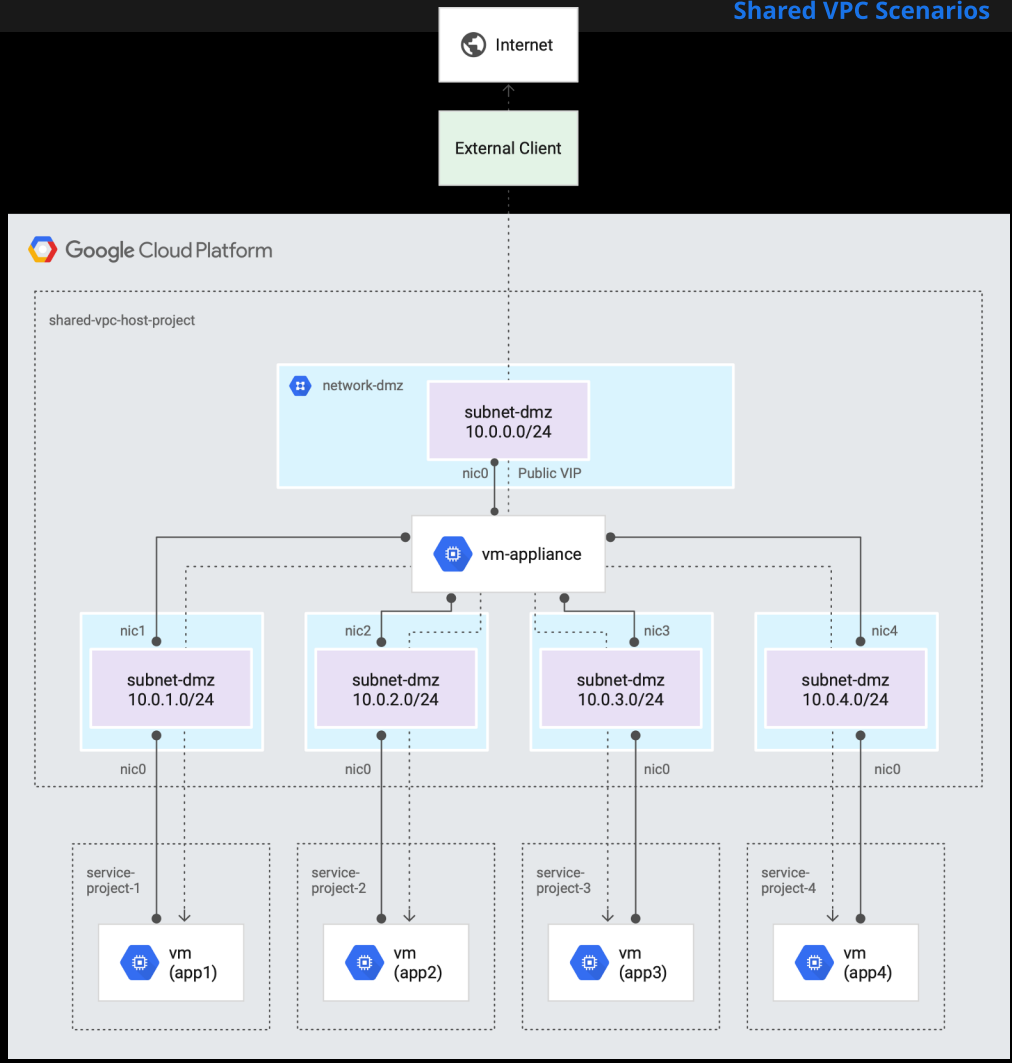
Course Navigation

Growing Your Network
Section 3

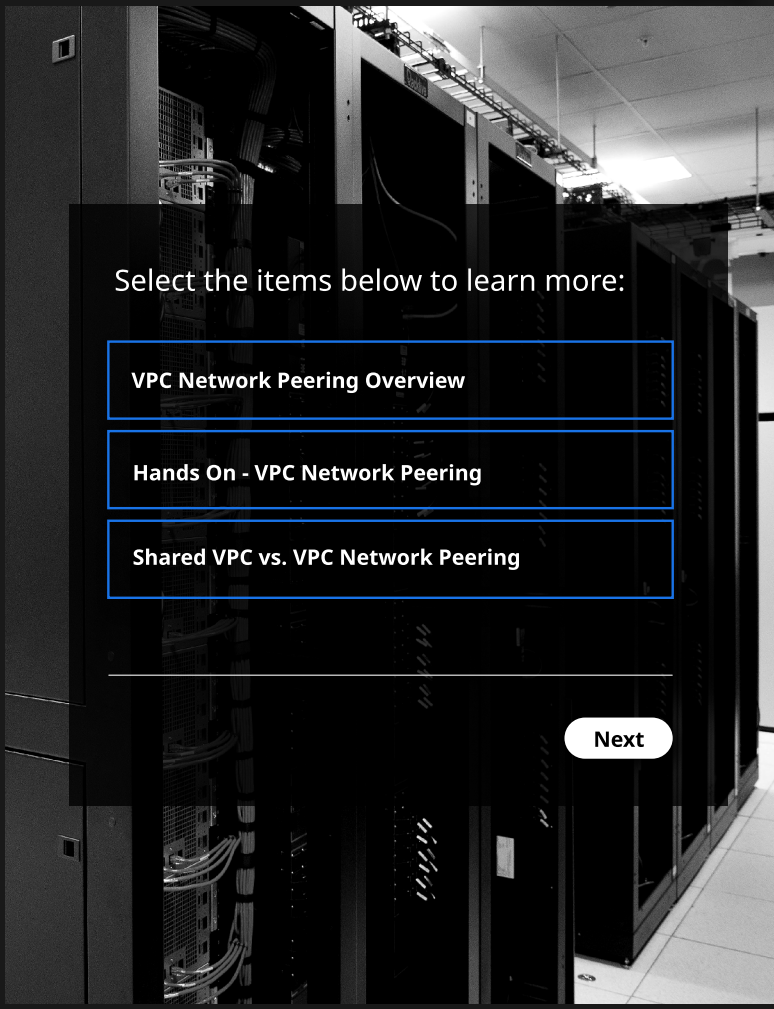
Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Multi-NIC Network Appliance

- Multiple VPCs (some shared) in same host project
- Multi-NIC appliance in host project
- Service project instances custom routes:
  - Tagged instances
  - vm-appliance as next hop
  - 0.0.0.0/0 as destination



Back



Select the items below to learn more:

VPC Network Peering Overview

Hands On - VPC Network Peering

Shared VPC vs. VPC Network Peering

Next

Course Navigation

Growing Your Network
Section 3

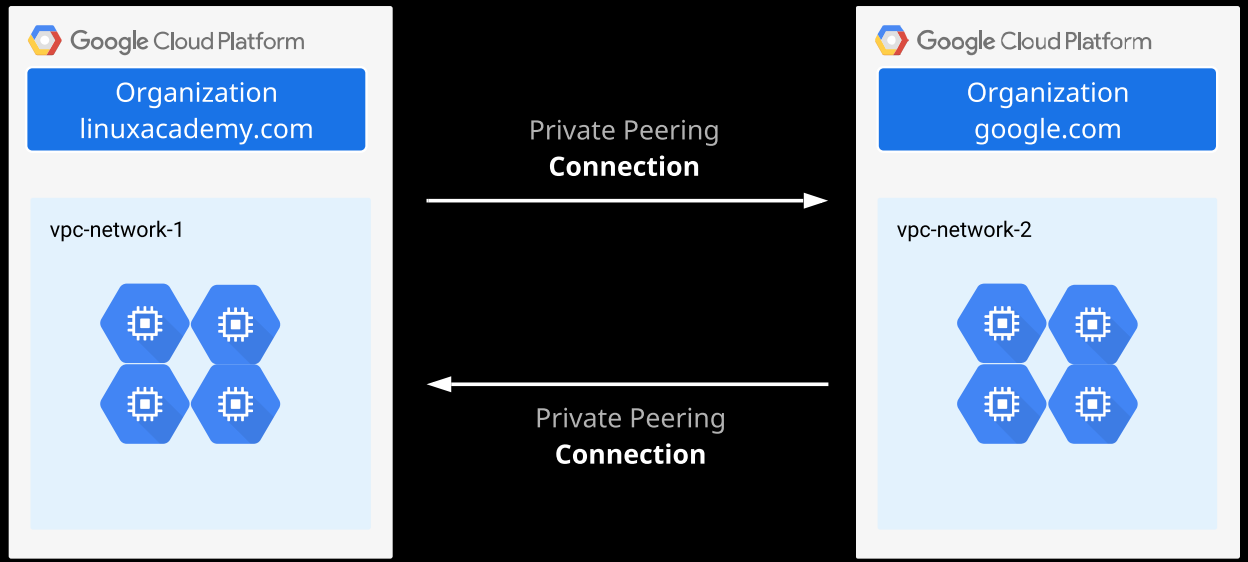
Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

What is VPC Network Peering?

- Not the same as **direct/carrier peering.**
- Private (RFC 1918) connectivity across two Google Cloud VPC networks.
  - Regardless of same project or organization.
  - Standalone VPCs, shared VPCs, same organization, different organizations — doesn't matter!
- Only requirement is that they are **GCP VPC** (no other cloud platform or on-premises).
- Traffic stays on Google's network and doesn't touch the public internet.

Use Cases

- Software as a Service ecosystems in **GCP**
  - Make services available privately across VPCs and across organizations.
- Large organizations with multiple domains can peer



Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

But Wait — Why Not Just Use VPN?

- Lower latency compared to **VPN.**
  - All peering traffic stays in Google's network.
- Security: Traffic to exposed to public internet.
- Cost: VPN = egress traffic charges, even in same zone.
  - Peering = no egress traffic in **same zone.**
- Simplicity: A lot easier to set up.

Operational Details

- Works with **Compute Engine, GKE, and App Engine Flex.**
- Administratively separate — not centrally managed.
  - Set up independently on each end. Peering connection only established after both sides match.
  - Either side can remove the peering connection at any time.
  - Routes, firewalls, and other traffic management tools administered on their own networks.

Back
Next

Back to Main
Next Topics

Course Navigation

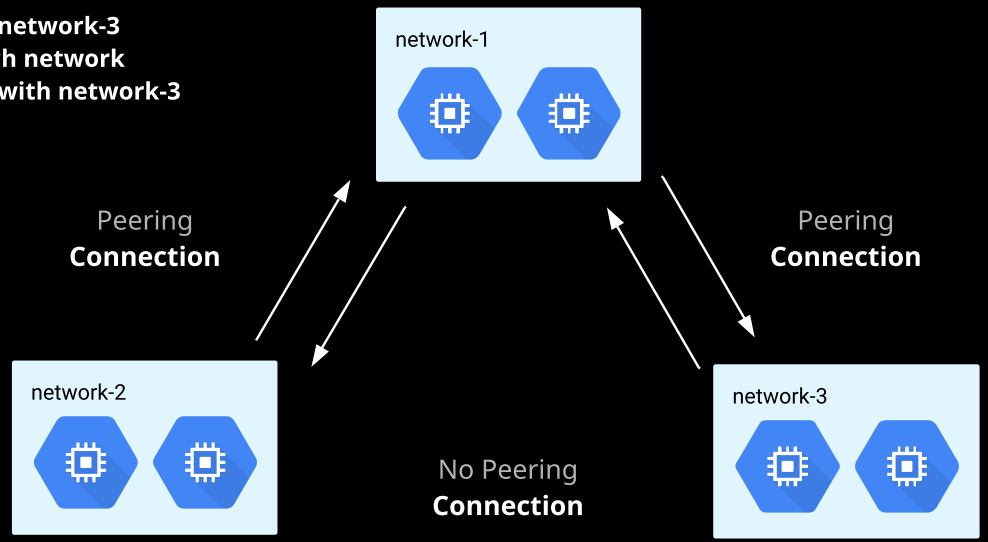
Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Constraints/Restrictions

- CIDR range for one network's subnet cannot overlap with static route on peered network.
  - i.e., no overlapping subnets
- GCP VPCs only.
- Cannot restrict which subnets are shared/routed. Everything connects to everything (subnet-wise).
  - Use firewall rules to restrict traffic.
- Network tag/service account filters do not carry over to peered network.
  - e.g., firewall rules
- Firewall rules not exchanged.
- No transitive peering.

- **network-1 peered with both network-2 and network-3**
- **network-1 communicates privately with each network**
- **network-2 CANNOT communicate privately with network-3**



Back Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

IAM Roles — Create/Delete Network Peering Connections

- Network Admin, Project Owner/Editor, or higher

Use Case Scenarios

- Connect two different **GCP domains' VPCs.**
  - Need private connectivity between both.
- Multiple development teams **need communal private network space.**
  - Using existing resources — do not want to recreate existing VMs.

Back

Course Navigation

Growing Your Network
Section 3

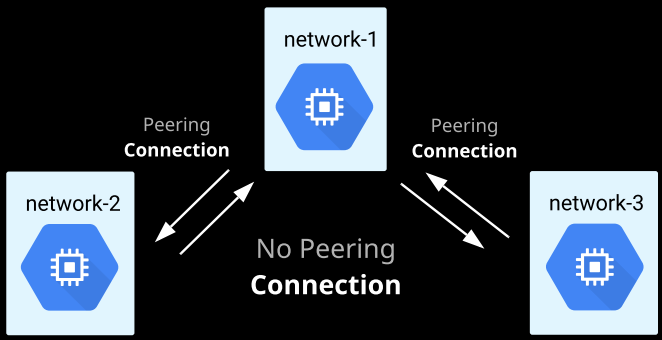
Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Hands-On Guideposts

- Command to download/run setup script in lesson description + detailed commands.
- Start with **three** disconnected VPCs.
  - Set up gcloud commands in lesson description.
  - Start with overlapping subnets — observe results.
  - Instance in each VPC
- Create network peering connection between **network-1 and network-2.**
- Create network peering connection between **network-1 and network-3.**
- Notice we **cannot peer** with overlapping subnets.
  - Delete offending subnets as needed.
- View routes
- Attempt to ping internal IP of **network-2** instance from **network-1** instance.
  - Should fail
- Create firewall rule to allow ICMP access to **network-3** from **network-1** and **network-2** subnets.
  - Source filter = subnet IP ranges
- Ping **network-3** instance from **network-1** instance.
  - Should be successful
- Ping **network-3** instance from **network-2** instance.
  - Should fail due to no transitive peering



Growing Your Network

Shared VPC vs. VPC Network Peering

Consideration	Network Peering	Shared VPC
Across organizations?	Yes	No
Administrative management	Decentralized	Centralized
Single project?	Yes	No
Existing resources?	Can keep — peer existing networks together	Must recreate existing resources in new VPC
Transitive connectivity?	No	Yes — service projects can communicate with each other

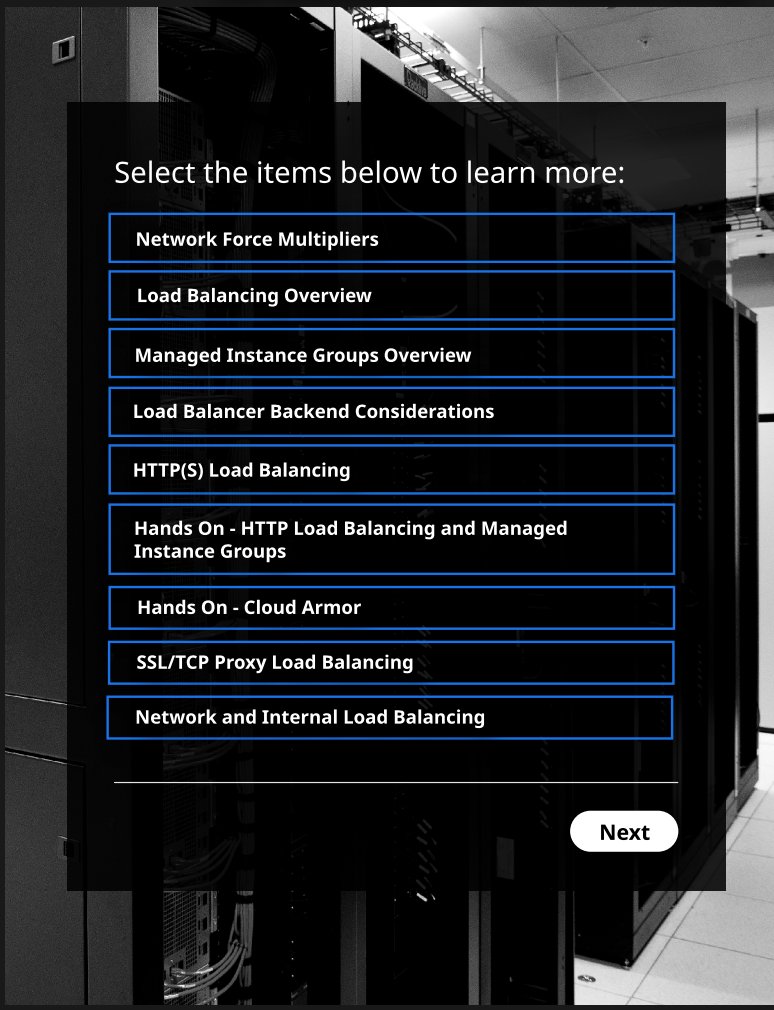
Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing



Select the items below to learn more:

- Network Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Managed Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

What Are "Force Multipliers"?

Scaling and Automation

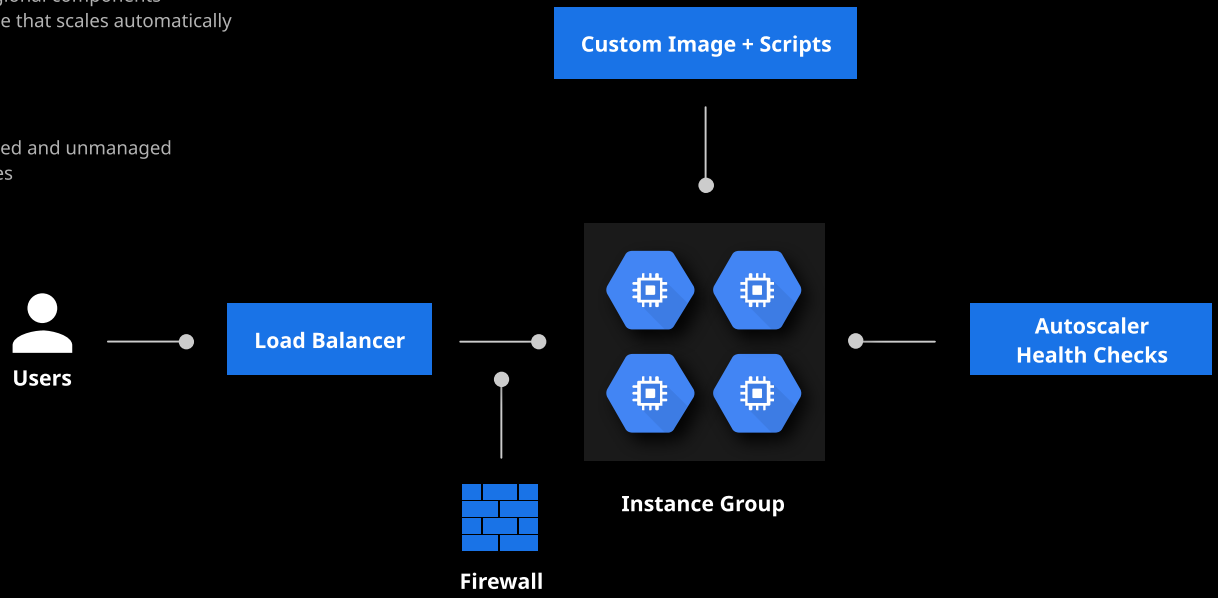
- Instead of working with a single instance, working with **many** instances at once
- Automatically scaling capacity up and down to meet need
  - No more overloaded servers!
- Global scope for your application

Putting It All Together

- Combine VPC network fundamentals (firewalls, subnets, etc.)
  - Add scalable/multi-regional components
  - Result: Global presence that scales automatically

Topics Covered

- Instance groups — managed and unmanaged
- Load balancers — five types
- Lots of hands-on demos



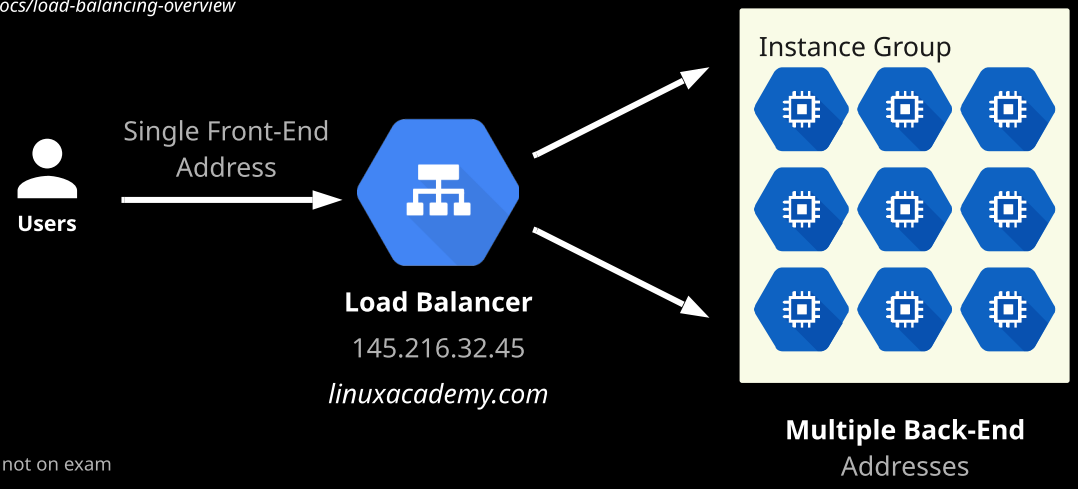
Course Navigation

Growing Your Network
Section 3

- Shared VPC
  - Shared VPC Overview
  - Shared VPC Hans On
  - Shared VPC Scenarios
- VPC Network Peering
  - VPC Network Peering Overview
  - Hands On - VPC Network Peering
  - Shared VPC vs. VPC Network Peering
- Load Balancing and Managed Instance Goups
  - Networking Force Multipliers
  - Load Balancing Overview
  - Managed Instance Groups Overview
  - Load Balancer Backend Considerations
  - HTTP(S) Load Balancing
  - Hands On - HTTP Load Balancing and Manged Instance Groups
  - Hands On - Cloud Armor
  - SSL/TCP Proxy Load Balancing
  - Network and Internal Load Balancing

What is a Load Balancer?

- Distributes (balances) user network requests among a back-end pool of instances
- Single front-end point of access — multiple back-end targets to serve traffic
- Software defined — not physical
- Necessary for distributed applications
- Global or regional in scope — depends on load balancer type
- Back-end traffic subject to firewall rules
- Big picture overview:
  - *https://cloud.google.com/load-balancing/docs/load-balancing-overview*



Load Balancer Types

- **HTTP(S) — Layer 7 Load Balancer**
- **SSL Proxy**
- **TCP Proxy**
- **Network Load Balancer**
- **Internal Load Balancer**
- (In Beta) **HTTP(S) Internal Load Balancer** — not on exam

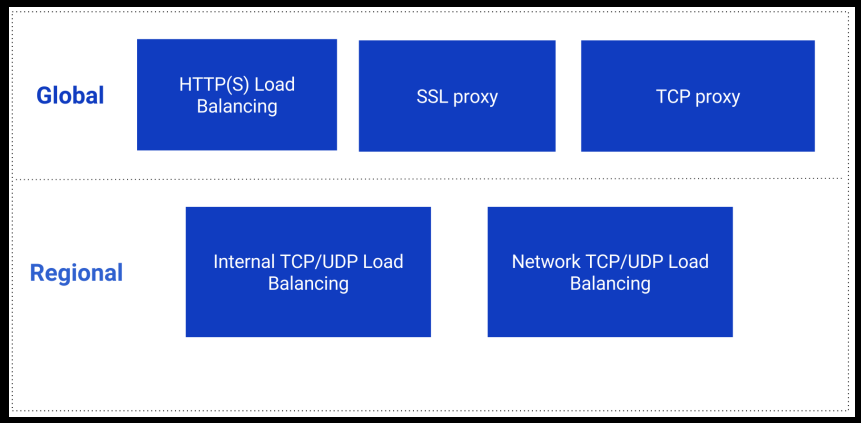
Load Balancer Differences

- Global (multi-regional) vs. regional
- External vs. internal
- HTTP vs. TCP/UDP
  - Layer 7 vs. Layer 4 traffic
- Proxy vs. non-proxy (pass-through) traffic
- SSL?
- IPv6?

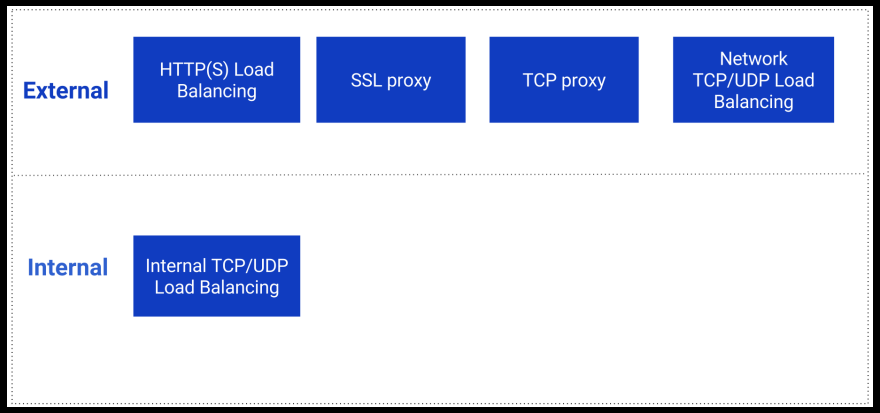
Next

Back to Main
Next Topics

Global vs. Regional



External vs. Internal



Back Next

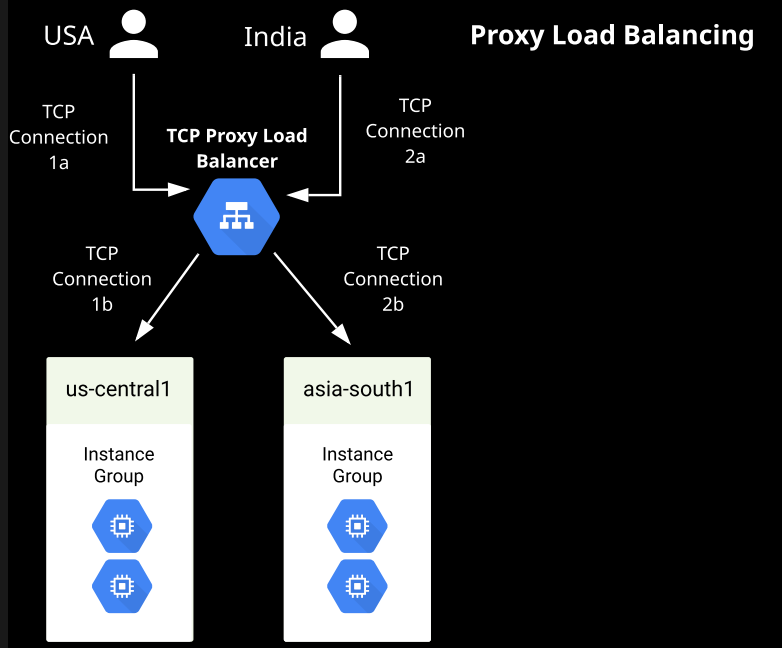
Proxied vs. Non-Proxied Traffic

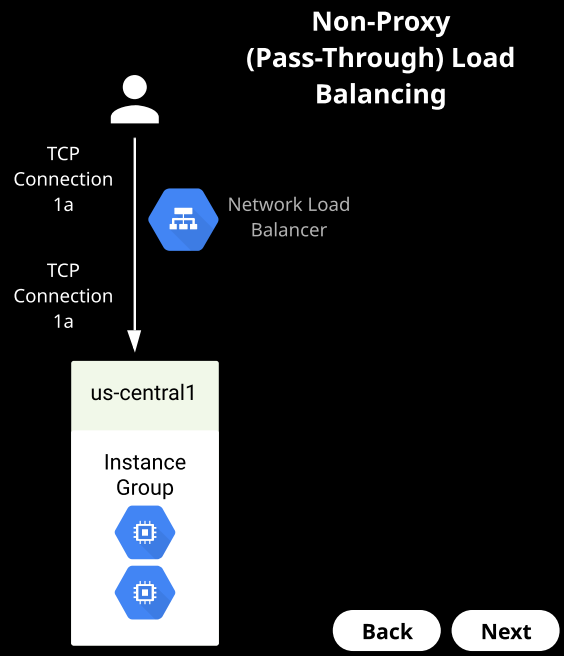
Proxy Traffic

- Proxied = terminate incoming traffic at load balancer layer, then creates new connection to VM
- On all global load balancers (HTTP(S), SSL/TCP proxy)
- Supports IPv6

Non-Proxy (or Pass-Through)

- Non-proxied = traffic goes through load balancer (pass-through) and forwarded on to the VM
- All regional-only load balancers (Network, Internal)
- Does not support IPv6
- Preserves client IP





Back Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Session Affinity

- What if multiple requests from a user need to be directed to the same individual instance?
- **Session affinity** makes it possible
- Most common is **Client IP** = all traffic from a client's address goes to the same instance

Traffic Type Scenarios — UDP

- Some load balancers support HTTP, SSL, TCP, UDP
  - Some global, some not
- What about VOIP traffic?
- VOIP = streaming traffic = UDP
- Regional scope = **Network Load Balancer**
- (Non-exam) Global scope? = **HTTP Load Balancer** ... Huh?
  - Supports QUIC (Quick UDP Internet Connections) protocol
  - Think of it as "UDP over HTTP"
  - That being said, a Network Load Balancer is the better choice if regional, and it is the more likely exam choice

Back
Next

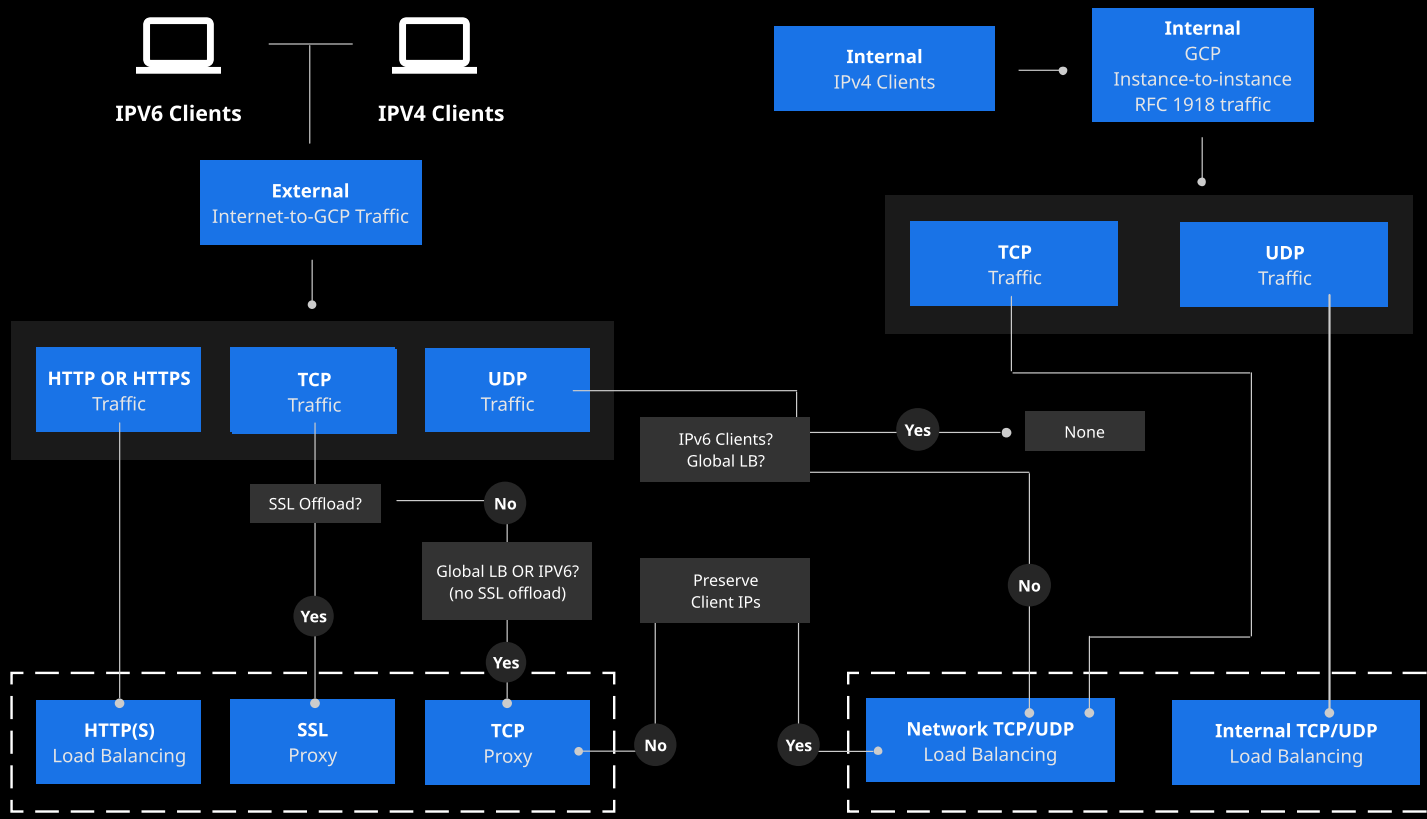
Back to Main
Next Topics

Course Navigation

Growing Your Network
Section 3

- Shared VPC
  - Shared VPC Overview
  - Shared VPC Hans On
  - Shared VPC Scenarios
- VPC Network Peering
  - VPC Network Peering Overview
  - Hands On - VPC Network Peering
  - Shared VPC vs. VPC Network Peering
- Load Balancing and Managed Instance Goups
  - Networking Force Multipliers
  - Load Balancing Overview
  - Managed Instance Groups Overview
  - Load Balancer Backend Considerations
  - HTTP(S) Load Balancing
  - Hands On - HTTP Load Balancing and Manged Instance Groups
  - Hands On - Cloud Armor
  - SSL/TCP Proxy Load Balancing
  - Network and Internal Load Balancing

Flowchart Time! Memorize Every Scenario



Back Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Breakdown in Table Format

Load Balancer	Traffic Type	Global/Regional	External/Internal	External Ports
HTTP(S)	HTTP/S	Global IPv4 and IPv6	External	HTTP:80/8080 HTTPS:443
SSL Proxy	TCP with SSL offload			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
TCP Proxy	- TCP with no SSL offload - Both TCP/SSL Proxies (does not preserve) client IP address (by default)			
Network TCP/UDP	- TCP with no SSL offload - Passes through (preserves) client IP address	Regional IPv4 only		Any
Internal TCP/UDP	TCP/UDP		Internal	Any

Back

What Are Instance Groups?

- Groups of instances
- **Back-end targets** for load balancers
  - Cloud Storage buckets are another back-end type
- Manage as a group, not one machine at a time
- **Managed** and **unmanaged** varieties
  - **Unmanaged instance groups** — ideal for migrating existing configurations for load balancing tasks with minimal modification
  - Serve load balancer traffic to separately managed, dissimilar machines
  - Managed preferred, and what we'll cover

Key Concept: Individual/Group Server Management

Pets vs. Livestock

- **Pets:** Individual, high-value servers — lovingly hand-crafted
  - Indispensable, can never have downtime
  - Examples: domain controller, mail server, database server
  - If single server goes down, it's a very bad thing
  - Server is individually backed up — backups are very important (snapshots)
  - Limited scalability (vertical scaling)
- **Livestock:** Group (or "herd") of servers, built en masse using automated tools
  - Individual servers not as important — disposable
  - Designed with failure in mind — can be replaced
  - Individual backups not important — don't use snapshots on instance group VMs
  - Much greater scalability — horizontal scalability
    - Highly available (multiple regions) and scalable (increase as load increases)
  - Examples: managed instance groups, Kubernetes, App Engine, Hadoop/MapReduce
  - Stateless workloads — don't need to save state on machine

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Features of Managed Instance Groups

- Resizable
- Automatically scale (autoscaler)
- Often paired with load balancers
- Health checks — auto-healing groups
  - Remove "unhealthy" instance and replace with new one
- Deploy new versions of app with no downtime

Back
Next

Back to Main
Next Topics

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Creating a Managed Instance Group

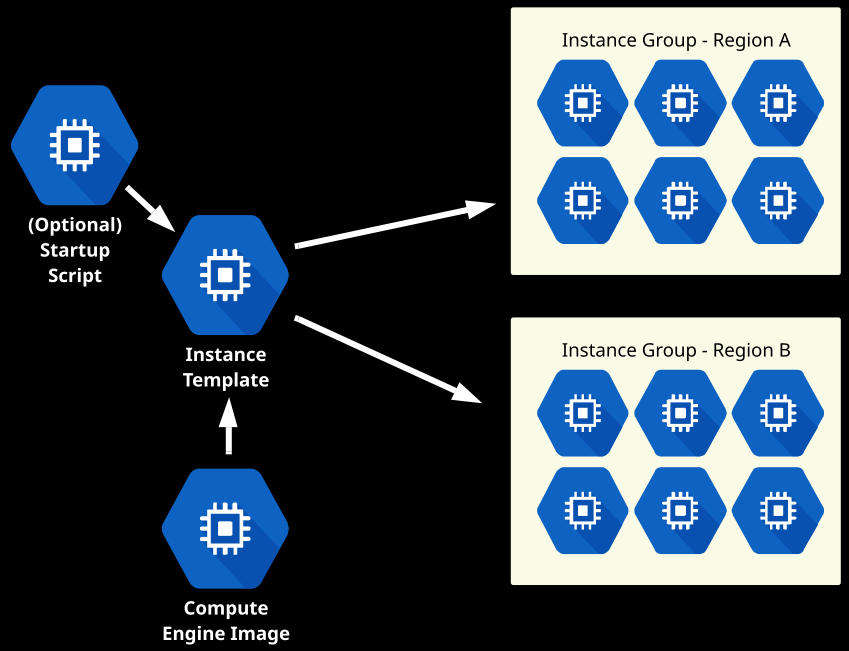
1. Create **instance template**
2. Create **instance group** from **instance template**

Instance Template

- Defines group configuration
- Machine type, zone, image, scripts
- Reusable for multiple group configurations
- Instance template = **Global** — not region bound
- Can specify zonal resources (e.g., read-only disk), which effectively binds it

From template — create managed **instance group**

- Instance group = **Regional** — can use more than one zone in single group



Back
Next

Back to Main
Next Topics

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

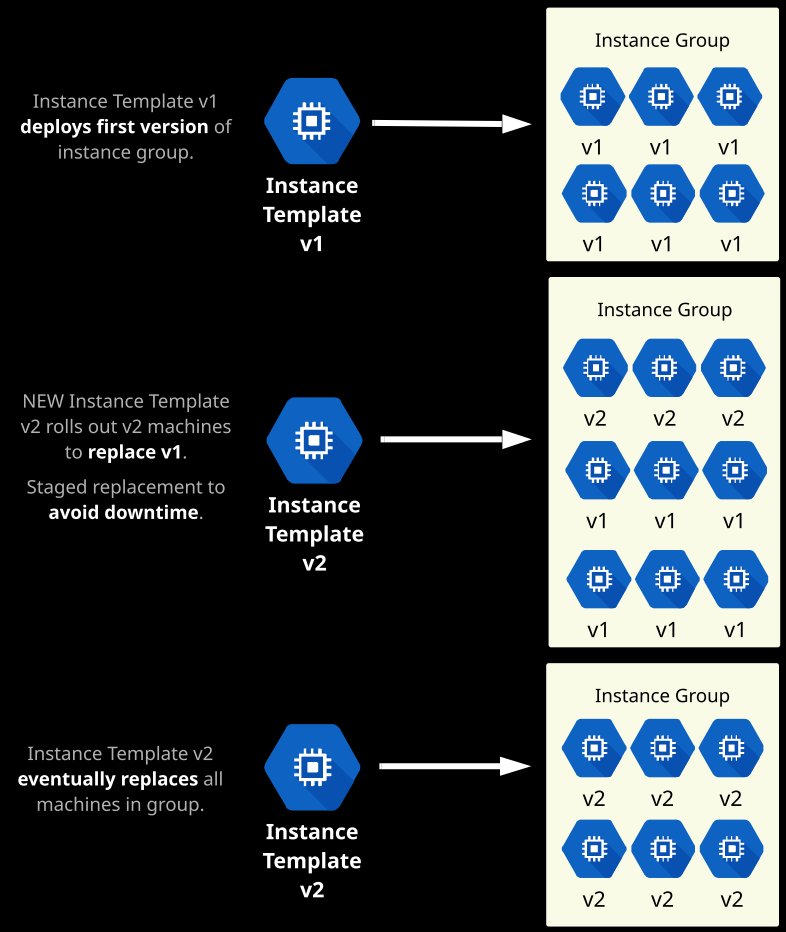
Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Updating Managed Instance Groups

- Short version: Replace one machine with a different machine
- **Managed Instance Group Updater**
  - Update entire group — not just individual machines
  - Deploy new versions of software
  - Control pace of update rollout
    - Rollout happens automatically
    - Can do partial rollouts for **canary** testing
  - Deploy inside existing managed instance group

Autoscaling

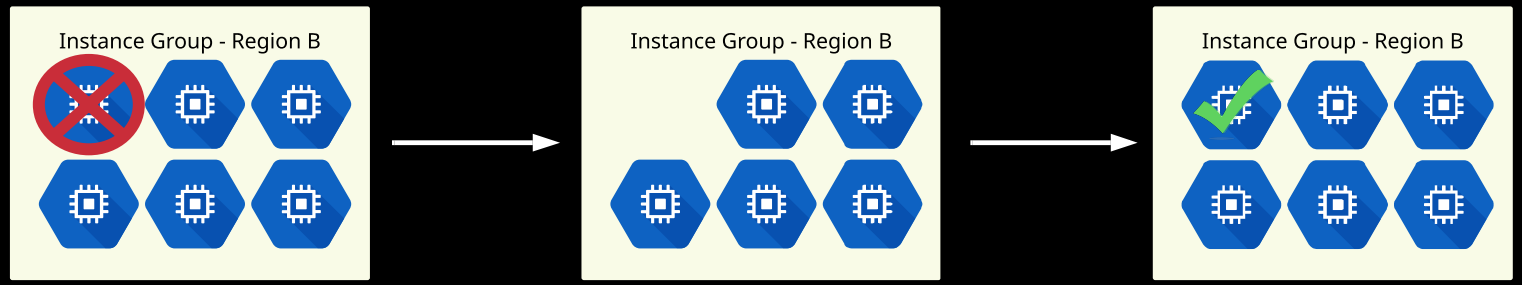
- Add/remove instances as load increases/decreases
- Set autoscaling policy
  - CPU utilization
  - Load balancing capacity
  - Stackdriver Monitoring metrics
  - Queue workload (batch computing)



Back Next

Health Checks

- Autohealing for managed instance groups
- If an instance (or service) fails — delete and recreate failed instance
- Health check **probe** checks instances at set interval (e.g., every fuve seconds)
  - If instance unresponsive for (x) continuous attempts, instance is "unhealthy," deleted, and replaced



Health Check/LB Firewall Rules

- Instance group must allow health check traffic for check to pass
  - Same IP range for load balancers as well
  - Must allow ingress traffic from below IP ranges
  - Possible troubleshooting step

Load Balancer	Probe IP ranges
• Internal • TCP Proxy • SSL Proxy • HTTP(S)	• 35.191.0.0/16 • 130.211.0.0/22
• Network	• 35.191.0.0/16 • 209.85.152.0/22 • 209.85.204.0/22

Back

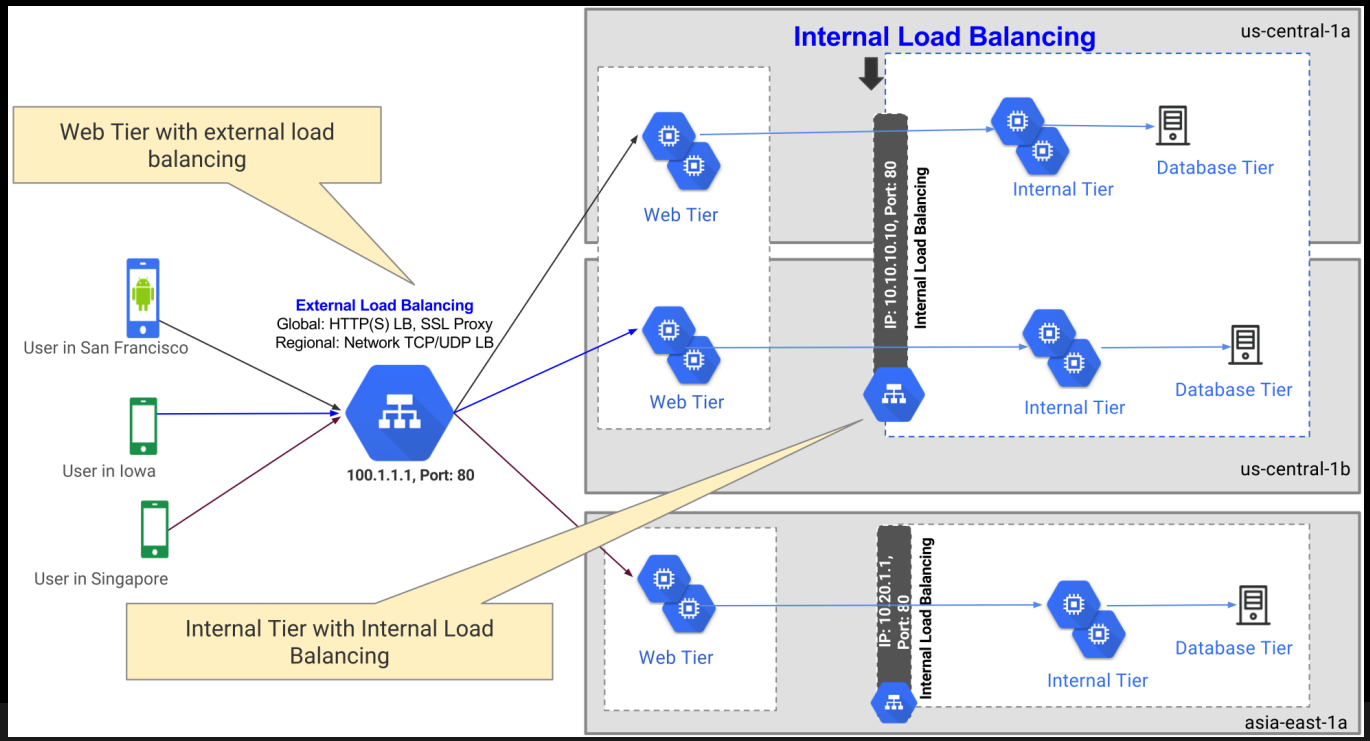
Course Navigation

Growing Your Network
Section 3

- Shared VPC
  - Shared VPC Overview
  - Shared VPC Hans On
  - Shared VPC Scenarios
- VPC Network Peering
  - VPC Network Peering Overview
  - Hands On - VPC Network Peering
  - Shared VPC vs. VPC Network Peering
- Load Balancing and Managed Instance Goups
  - Networking Force Multipliers
  - Load Balancing Overview
  - Managed Instance Groups Overview
  - Load Balancer Backend Considerations
  - HTTP(S) Load Balancing
  - Hands On - HTTP Load Balancing and Manged Instance Groups
  - Hands On - Cloud Armor
  - SSL/TCP Proxy Load Balancing
  - Network and Internal Load Balancing

Load Balancing and Managed Instance Groups - A Perfect Pairing

- **Managed Instance Groups** often paired with **load balancers** - distribute traffic across instances in group (or even multiple groups)
  - Load balancers must be assigned to a backend – target pool or managed instance group
  - Load balancer contains one or backend service
  - **Backend service** links to one or more **backends**
  - Backend links to one instance group
  - Backend service knows which backends to use – directs traffic
- Subject to firewall rules for allowed traffic
  - Rules applied to instances, not load balancer
  - Backends must allow load balancer addresses - same as health check addresses in previous lesson



Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Primary Concepts

- **Layer 7** load balancing
- Operates over HTTP(S) protocols
- **HTTP** = TCP:80, 8080
- **HTTPS** = TCP:443
- Global/multi-regional scope
  - Distribute traffic to backends in multiple regions
- Supports IPv4 and IPv6 — IPv6 terminated at LB then proxy by IPv4 to backend
- QUIC protocol support
- Native support for **WebSocket** protocol
- Proxied traffic
  - Client connection terminated at Edge POP (as are all global LBs)

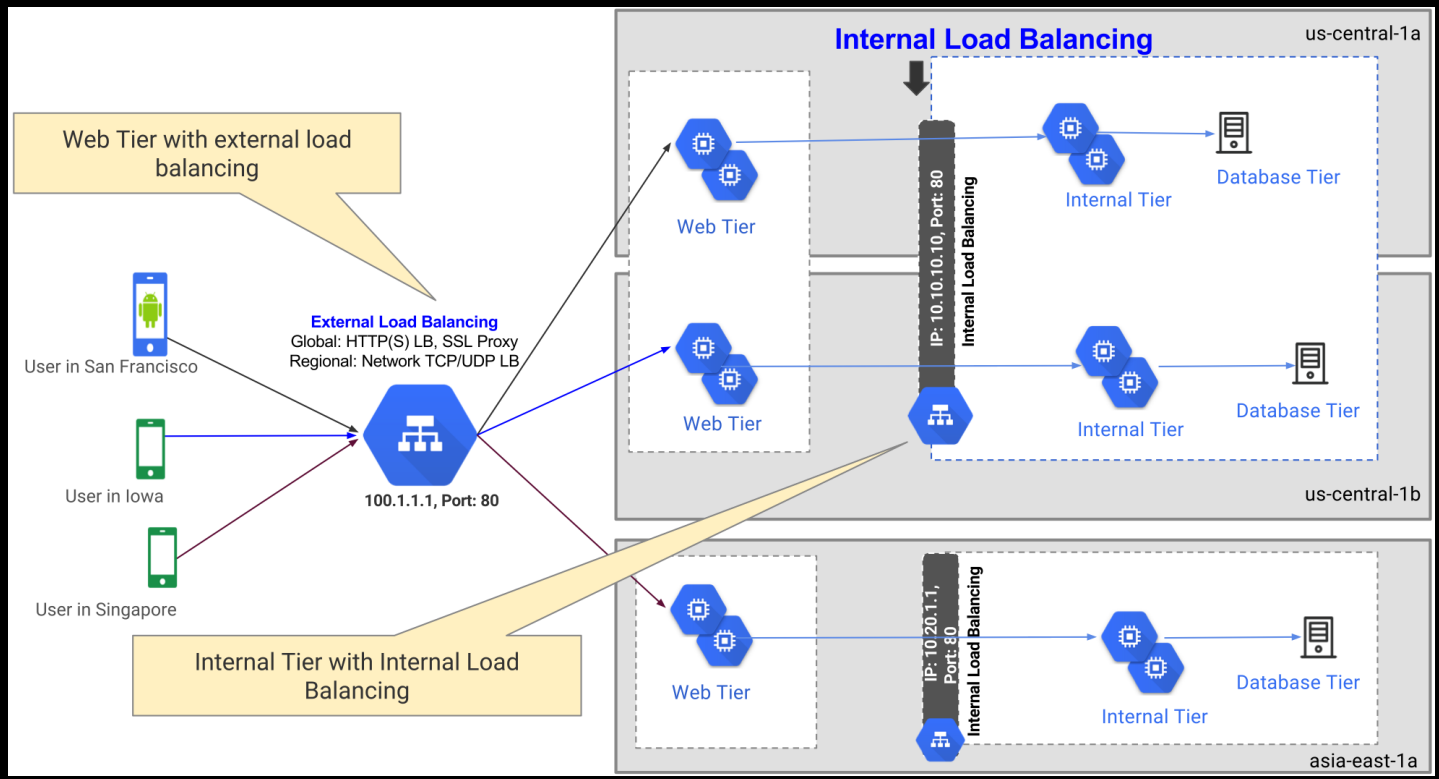
Traffic Distribution — Forwarding Rules

- **Forwarding rule:** Forwards traffic to backend by matched criteria
  - **Location** and **URL map** (content type)
- **Location** = Closest regional backend to user request — which Edge POP received request
- **URL map** = Use URL address to map which backend to send traffic
  - e.g., *linuxacademy.com/video* to "video" backend

Next

Back to Main
Next Topics

Location-Based Load Balancing



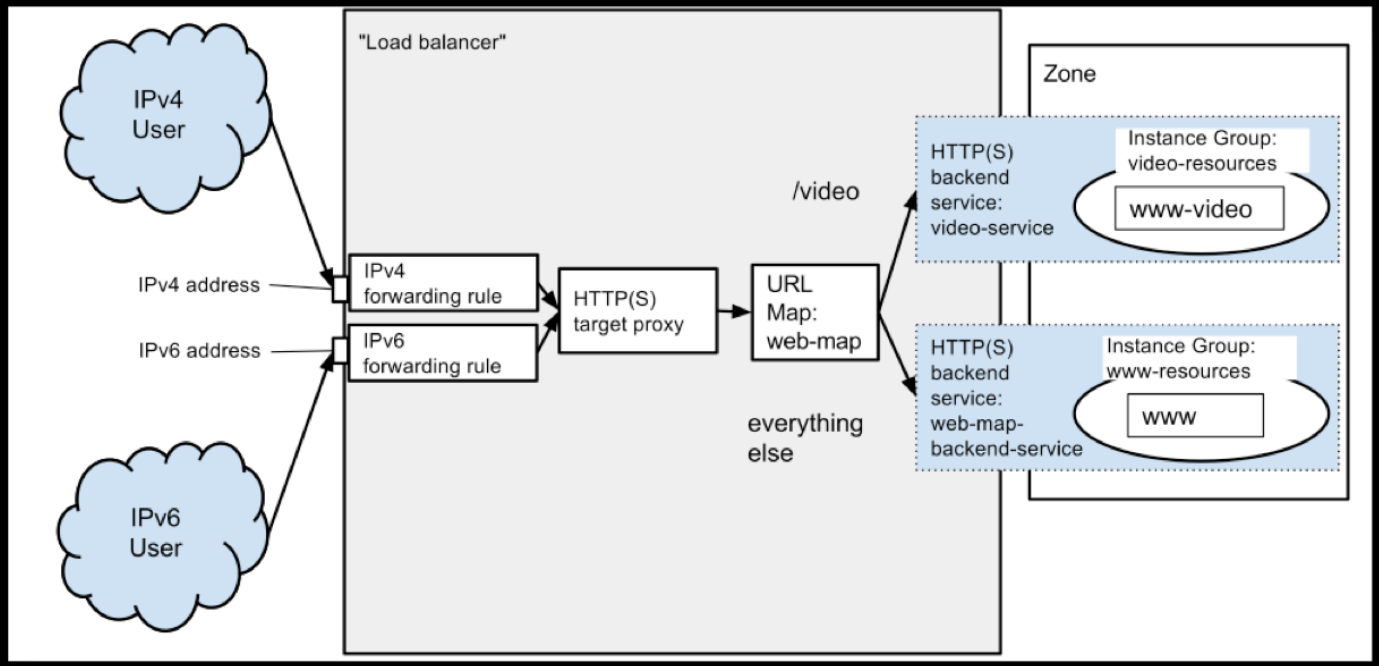
Back Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

Content-Based Load Balancing



Back

What Are We Doing?

- Create firewall rules to allow HTTP and load balancer/health check access to instance group
- Create instance template using startup script
- Create two instance groups
  - Set up autoscaling and health checks
- Create HTTP load balancer
  - Add instance groups as backends
- Update instance group with "version 2" using new instance template
- Force autoscaling via stress test

Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Instance Template, Group, and Health Check

- Create firewall rules to allow HTTP and load balancer/health check access to tagged instance group
- Create instance template, using startup script to create custom web page
- Create instance group from instance template
- View health check options
- Update instance group with newer version using rolling updates

Back
Next

Back to Main
Next Topics

HTTP Load Balancer

- Adjust `us-central1` group for autoscaling
- Create second instance group in `europe-north1` region
- Create HTTP load balancer
  - Observe settings, especially host/path rules
- Observe results
  - Send light traffic to load balancer

Note on path rules:

- Wildcards (*) can be used at end of path, not beginning/middle
- Valid:
  - /video/*
  - /video/hd/*
- Not valid:
  - /*/video

Load Balancer Metrics and Stress Testing

- Create "stress tester" instance with Apache Bench
- Send steady trickle of traffic to application
  - `ab -n 1000000 -c 3 http://<your-frontend-ip>/`
- Check session affinity settings
  - HTTP = Client IP and generated cookie only
  - Other load balancers can include protocol and port
- View load balancing backend metrics
- Overload application, and view how load balancer compensates
  - `ab -n 1000000 -c 1000 http://<your-frontend-ip>/`

Back

What Are We Doing?

- Create firewall rules to allow HTTP and load balancer/health check access to instance group
- Create instance template using startup script
- Create two instance groups
  - Set up autoscaling and health checks
- Create HTTP load balancer
  - Add instance groups as backends
- Update instance group with "version 2" using new instance template
- Force autoscaling via stress test

Next

Course Navigation

Growing Your Network
Section 3

Shared VPC
Shared VPC Overview
Shared VPC Hans On
Shared VPC Scenarios
VPC Network Peering
VPC Network Peering Overview
Hands On - VPC Network Peering
Shared VPC vs. VPC Network Peering
Load Balancing and Managed Instance Goups
Networking Force Multipliers
Load Balancing Overview
Managed Instance Groups Overview
Load Balancer Backend Considerations
HTTP(S) Load Balancing
Hands On - HTTP Load Balancing and Manged Instance Groups
Hands On - Cloud Armor
SSL/TCP Proxy Load Balancing
Network and Internal Load Balancing

SSL Proxy Load Balancer

- Load balancing for encrypted non-HTTP traffic
  - HTTP traffic should use HTTP load balancer instead
- Global/multi-regional in scope
- Terminates client traffic at load balancing layer
- Manages SSL certificates for you
  - Reduce certificate management overhead
- Automatic patching to fix vulnerabilities
- Traffic between load balancer and backends is SSL (recommended) or TCP

TCP Proxy Load Balancer

- Like SSL proxy, but for unencrypted, non-HTTP traffic
- Has other benefits, including intelligent routing, security patching
- Both load balancer types support many, but not all, TCP protocols
- Both load balancers types do not preserver client IP **by default**, but can enable PROXY protocol to preserve

Load Balancer	Traffic Type	Global/ Regional	External/ Internal	External Ports
HTTP(S)	HTTP/S	Global IPv4 and IPv6	External	HTTP:80/8080 HTTPS:443
SSL Proxy	TCP with SSL offload			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
TCP Proxy	- TCP with no SSL offload - Both TCP/SSL proxies (does not preserve) client IP address by default			

Course Navigation

Growing Your Network
Section 3

Shared VPC
- Shared VPC Overview
- Shared VPC Hans On
- Shared VPC Scenarios

VPC Network Peering
- VPC Network Peering Overview
- Hands On - VPC Network Peering
- Shared VPC vs. VPC Network Peering

Load Balancing and Managed Instance Goups
- Networking Force Multipliers
- Load Balancing Overview
- Managed Instance Groups Overview
- Load Balancer Backend Considerations
- HTTP(S) Load Balancing
- Hands On - HTTP Load Balancing and Manged Instance Groups
- Hands On - Cloud Armor
- SSL/TCP Proxy Load Balancing
- Network and Internal Load Balancing

Network vs. Internal Load Balancer

- Network = external traffic
- Internal = private VPC traffic only
  - Example: second tier of two tier web app

Regional, non-proxy traffic

- Only serves single region
- Forwards source client traffic directly to instance
- Supports **all** TCP/UDP ports

Backend Options

- Instance group or individual instances **(target pool)**
- No GCS buckets

Target Pools

- Not the same as instance group
  - Collection of individual instances
- Uses fowarding rules - load balancers picks instance from pool, and forwards client traffic to it
- **Regional load balancers only**

- Health checks are HTTP only

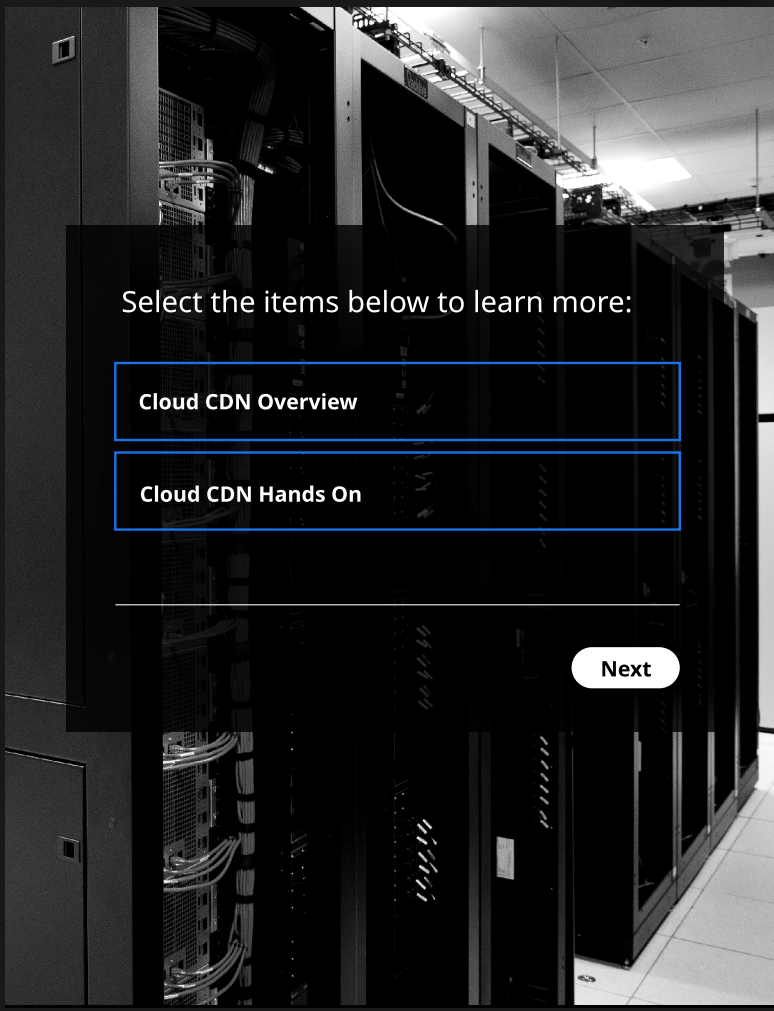
Course Navigation

Growing Your Network
Section 3 (Continued)

- **Cloud CDN**
  - Cloud CDN Overview
  - Cloud CDN Hands On
- **Google Kubernetes Engine**
  - GKE Networking Concepts
  - Planning for Growth of your GKE Cluster
  - Hands On - GKE IP Load Balancing (with Ingress Objects)
  - Private GKE Cluster and Network Policies
  - Hands On - Private GKE Clusters
- Cloud DNS
  - Cloud DNS Overview
  - DNSSEC Overview
  - Cloud DNS Hands On

Hybrid Networking
Section 4

Pulling It All Together
Section 5



Select the items below to learn more:

Cloud CDN Overview

Cloud CDN Hands On

Next

Course Navigation

Growing Your Network
Section 3 (Continued)

Cloud CDN
- Cloud CDN Overview
- Cloud CDN Hands On

Google Kubernetes Engine
- GKE Networking Concepts
- Planning for Growth of your GKE Cluster
- Hands On - GKE IP Load Balancing (with Ingress Objects)
- Private GKE Cluster and Network Policies
- Hands On - Private GKE Clusters

Cloud DNS
- Cloud DNS Overview
- DNSSEC Overview
- Cloud DNS Hands On

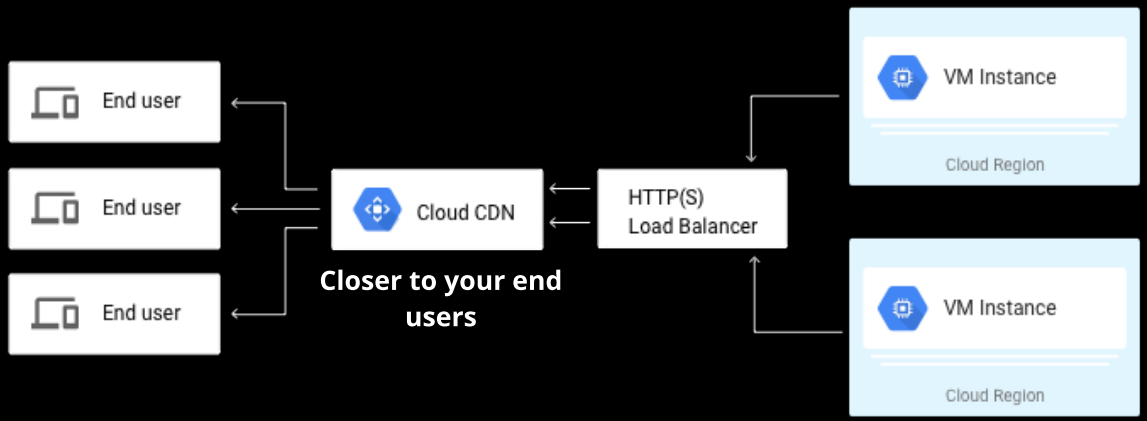
Hybrid Networking
Section 4

Pulling It All Together
Section 5

What is Cloud CDN?

- Short for **Content Delivery Network**
- Cloud CDN caches website and application content closer to your users
  - Cached at **Edge Point of Presence (POP)**

Why is this important?
- The closer your data is to the end user, the better the performance and user experience



Next

Course Navigation

Growing Your Network
Section 3 (Continued)

Cloud CDN
Cloud CDN Overview
Cloud CDN Hands On

Google Kubernetes Engine
GKE Networking Concepts
Planning for Growth of your GKE Cluster
Hands On - GKE IP Load Balancing (with Ingress Objects)
Private GKE Cluster and Network Policies
Hands On - Private GKE Clusters

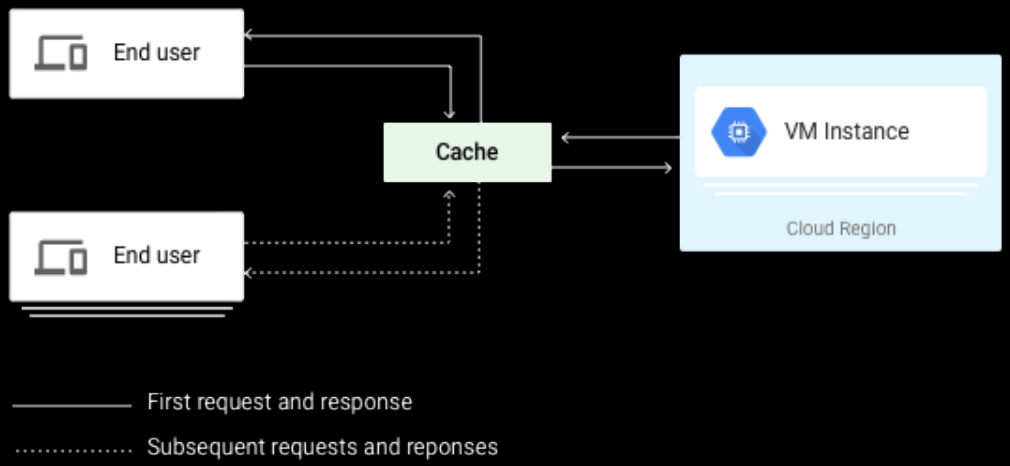
Cloud DNS
Cloud DNS Overview
DNSSEC Overview
Cloud DNS Hands On

Hybrid Networking
Section 4

Pulling It All Together
Section 5

How it works

- **HTTP/S load balancers only**
- Caches contents from **backends** (instance groups, Cloud Storage object)
- The first time backend content is requested in general location, it is pulled from backend and cached at a CDN location (edge location)
  - Usually MUCH closer than the datacenter where backend resides
  - Later requests from location will serve from CDN location, not backend



Terminology

- **Cache miss** - First time content request not at cache location
- **Cache fill** - after requested for first time, cache loaded by either backend or another nearby cache
  - Subsequent requests will pull from cache location
- **Cache key** - Identifier for cached content
  - By default, in form of complete URL (https://linuxacademy.com/images/logo.jpg)
  - Must have exact match

Back
Next

Back to Main
Previous Topic

Course Navigation

Growing Your Network
Section 3 (Continued)

Cloud CDN
Cloud CDN Overview
Cloud CDN Hands On

Google Kubernetes Engine
GKE Networking Concepts
Planning for Growth of your GKE Cluster
Hands On - GKE IP Load Balancing (with Ingress Objects)
Private GKE Cluster and Network Policies
Hands On - Private GKE Clusters

Cloud DNS
Cloud DNS Overview
DNSSEC Overview
Cloud DNS Hands On

Hybrid Networking
Section 4

Pulling It All Together
Section 5

Configuring Cache Scope - Exam Perspectives

Improving Cache Key Hit Ratio

- By default, uses entire URL
- Optimize cache hit ratio for better performance/scalability
  - Example- same company logo on different domains
- How to customize
  - Remove unneeded URL aspects
    - Company domain
    - Protocol (HTTP/HTTPS)
    - Part of query string

Managing what is and is not cached

- Both instance group and GCS objects utilize headers for cache control
- Whitelist/blacklist query strings to limit caching
  - **Whitelist** = No query except whitelisted path will be cached
  - **Blacklist** = All queries except blacklisted path will be cached
- Instance group (web server)
  - Add headers to web server config to allow caching
    - `Cache-Control=public, max-age=(seconds)`
- GCS objects - per-object controls
  - Objects must be public (or signed URL) to be cached
  - Edit object header via metadata to restrict per-object caching
    - `Cache-Control:no-cache,max-age=0`

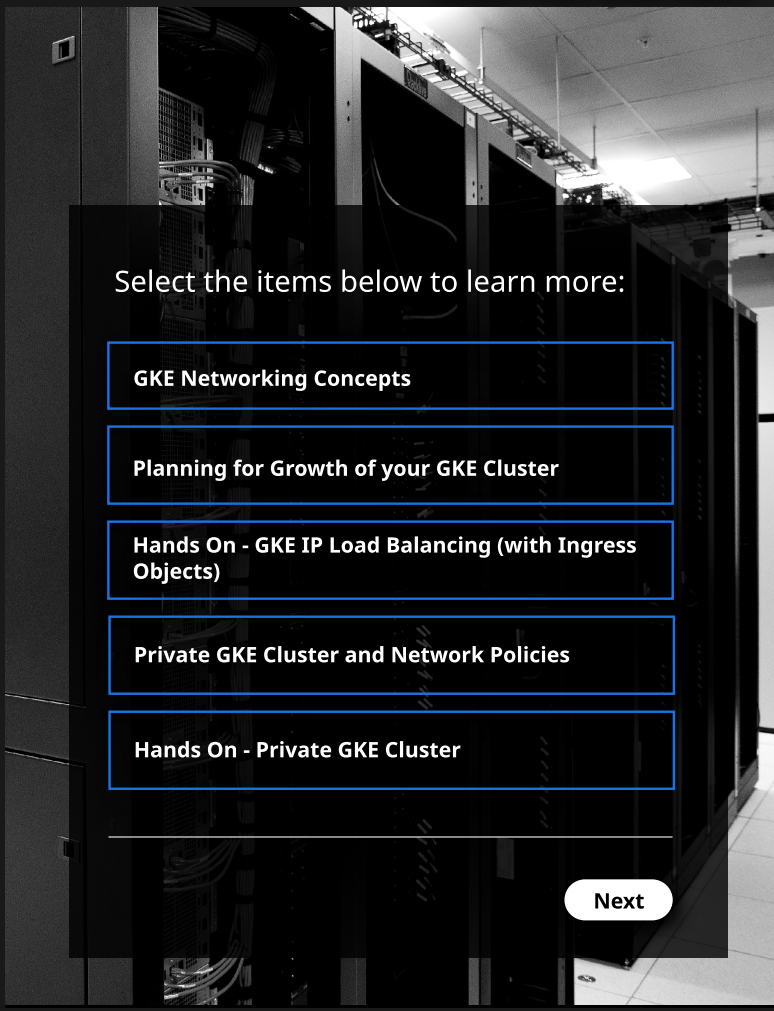
Additional Considerations

- **Compressed content**
  - Cloud CDN does not compress/decompress content, but can serve compressed content from backend server
    - Troubleshoot compression = troubleshoot backend server
- **Expiration for time-sensitive content**
  - Some content needs to be "fresher" than others
  - Set expiration time in Cache-Control header (`max-age=(seconds)`)

Back

What we are doing

- Using pre-built HTTP load balanced application (script in lesson description)
  - Single backend instance group in Australia, accessing from US (very far away)
- Measure site responsiveness without CDN (hint: it will not be good)
- Enable Cloud CDN
- View configuration options
- Measure performance on cached site and view logs



Select the items below to learn more:

- **GKE Networking Concepts**
- **Planning for Growth of your GKE Cluster**
- **Hands On - GKE IP Load Balancing (with Ingress Objects)**
- **Private GKE Cluster and Network Policies**
- **Hands On - Private GKE Cluster**

Next

Expectations for this section

- Not a beginner's introduction to Kubernetes, GKE, or containers
  - Will provide brief "crash course" for context
- Laser focus on networking components of GKE
  - Development concepts not a focus

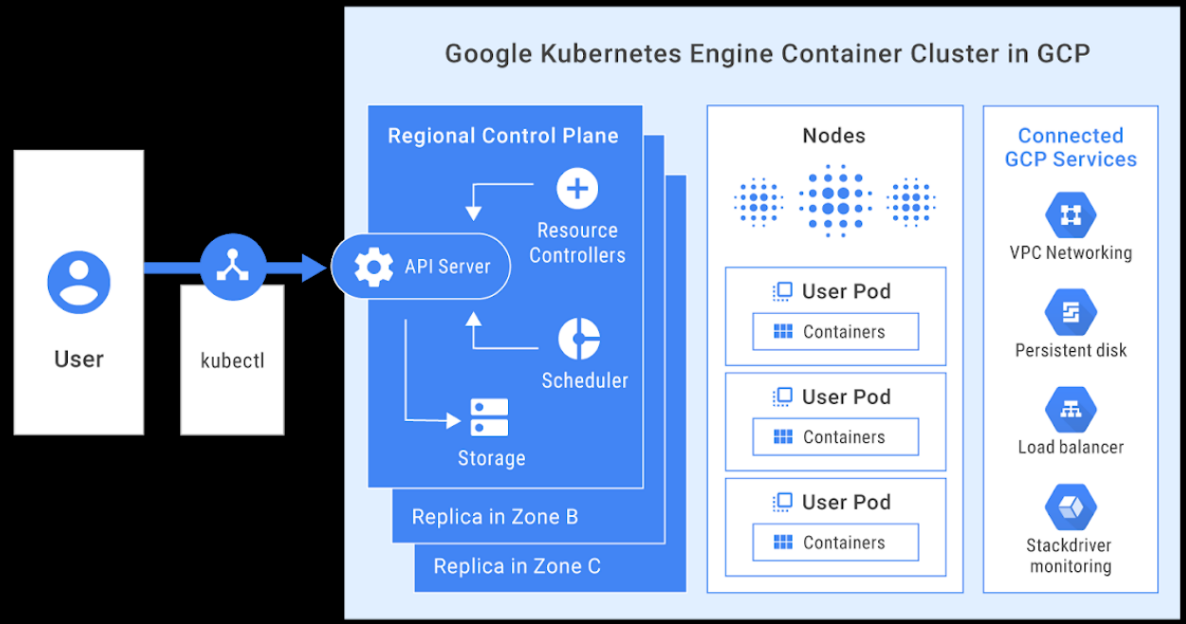
Topics we will cover:

- "Big picture" view of GKE networking
  - **Nodes** vs. **pods** vs. **services**
- How GKE routes traffic (**VPC Native** vs. **route based**)
- GKE integration with other GCP network services
- Breakdown of node/pod/service IP allocation
  - Planning for cluster growth
  - Subnetting calculations (CIDR ranges and binary, oh my!)
- **Private GKE clusters**
  - Connecting via **master authorized networks**
- Restricting intra-cluster communications via **Network Policy**

Next

GKE Quick Crash Course

- Instead of managing VMs, manage **containers**
  - **Containers** = lightweight, portable, self-contained packages that can be run virtually anywhere
- **Kubernetes** = Google-invented, open-source platform for managing your containers
  - Handles container deployment, scaling, updates, load balancing, and more
- **Kubernetes Engine (GKE)** = Managed Kubernetes service for container workloads
  - Google handles: OS management, master node, scaling, health checks, replication controller, services
  - You focus on your containers, Google handles the rest



Key Benefits

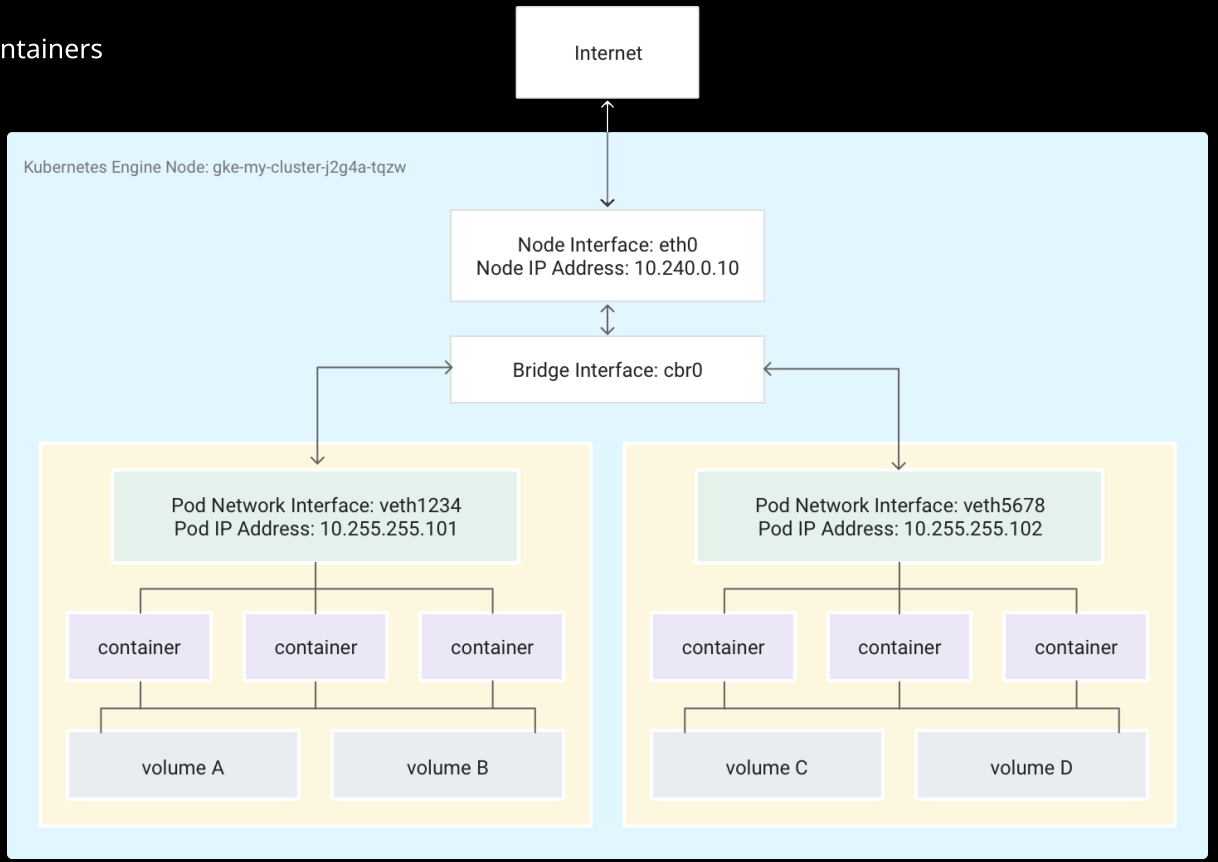
- Rich, powerful UI
- SRE monitoring
- Automated repair of apps
- Resource optimized app deployments
- Load balancing & auto-scaling of resources
- Global Virtual Private Cloud
- SLA of 99.95%

Back Next

GKE Components

- **Cluster** = group of worker nodes + managed cluster master
  - **Master** = Central administrative point for node/container management via **`kubectl`** commands (or API calls)
  - Cluster is either zonal or regional (multi-zone in same region) in scope
  - For high availability, pair multiple regional clusters with **HTTP Load Balancer**
- **Nodes** = Individual Compute Engine instances
  - GKE cluster workers
  - Host one or more **pods** per node
  - Behind the scenes - Nodes are arranged in managed instance groups
- **Pods** = smallest deployable unit
  - Include 1 or more containers
    - Multiple co-dependent containers can be placed in a single pod
  - Deployed to **Nodes**
  - Each pod has a single IP address
    - Communicate with other pods in cluster
    - Which node is on which pods in cluster is largely irrelevant
- **Services** = logical unit of multiple related pods
  - Similar concept as managed instance groups, but at the pod vs. VM level
  - Each service has its own single IP address
    - Provides single IP address for groups of similar pods
    - Think frontend address for backend groups - but again at the pod level
    - Pod/Service grouping determined by **labels**
    - **Expose** a service to pair with load balancer
      - Network (external/internal) and HTTP load balancers
      - HTTP Load Balancer requires Ingress object

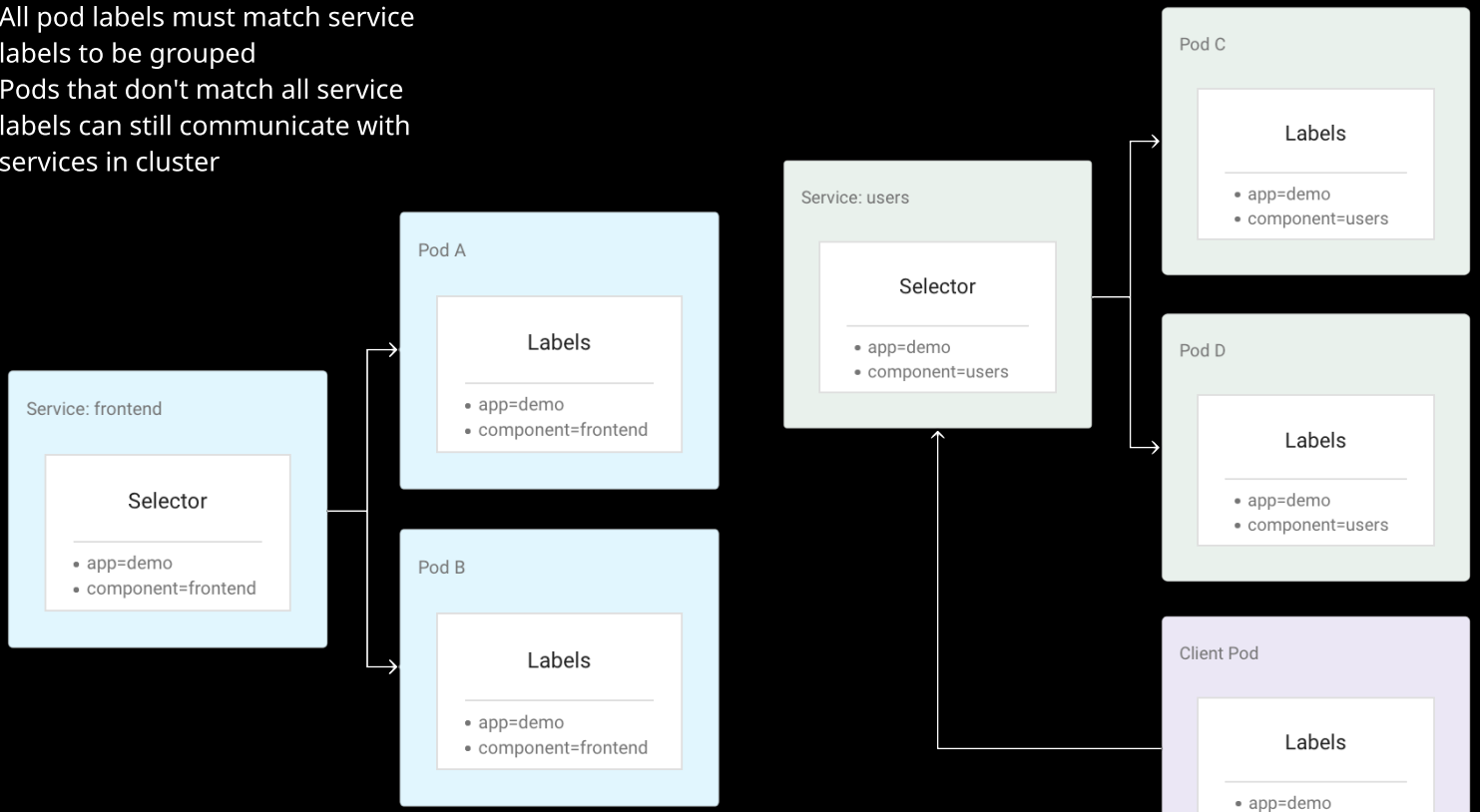
Nodes, Pods, and Containers



Back Next

Services, Pods, and Labels

- All pod labels must match service labels to be grouped
- Pods that don't match all service labels can still communicate with services in cluster



Back Next

Integration with other GCP networking services

- Cloud Armor - pair with Ingress object (same as HTTP Load Balancer)
- Shared VPC
  - Enable GKE API in all host and service projects
  - Grant **service project** service accounts (below) the **Network User role** to **host project**
  - Each project has both GKE and Google API service accounts
    - service-[project-number]@container-engine-robot.iam.gserviceaccount.com
    - [project-number]@cloudservices.gserviceaccount.com
  - Additionally, grant the **Host Service Agent User** role of the **service project GKE service account** to the host project
  - Summary:
    - Service project GKE service account
      - Network User role in host project
      - Host Service Agent User role in host project
    - Service project Google API service account
      - Network User role in host project

Back

IP Allocation

- Nodes, Pods, and Services have their own IP range
- Nodes = primary VPC subnet IP range
- Pods/Services = secondary VPC IP ranges
  - Create your own secondary ranges, or have GKE create them for you
- Pods:
  - Subnet secondary range PLUS per-node allocation
  - **BY DEFAULT:** Per-node IP allocation for pods = /24 CIDR range = 256
    - /24 is max per-node range - can assign smaller per-node range if desired
  - Separate from default per-node IP allocation, limited to 110 pods max per node
    - More IP's than pods for IP re-use, fluctuating pods
    - Any per-node IP range must at least twice as large as desired max pods per node
- Services:
  - IP allocation for entire cluster (no per-node allocation)
- **IMPORTANT:** You must specify pod and service IP range at or before cluster creation, not afterwards

Maximum Pods per Node	CIDR Range per Node
8	/28
9 to 16	/27
17 to 32	/26
33 to 64	/25
65 to 110	/24

Next

Planning for cluster growth

- Short version: Plan your node and pod IP ranges
- Node IP range = primary VPC subnet range
- Pod IP range
  - VPC subnet secondary IP range (for pods)
  - Divide total range by per-node allocation

Examples:

- You have a single node GKE cluster, and plan on growing it to 3 nodes max. Using the default per-node pod IP allocation, what is the smallest subnet/CIDR range you can assign for total pod addresses to account for maximum cluster pod IP's?
  - Default per-node pod range = 256
  - 256 x 3 nodes = 768 pod IP's needed for entire cluster
  - Pod secondary IP range calculation:
    - /24 = 256 - too small
    - /23 = 512 - too small
    - /22 = 1024 - more than total needed = **winner!**
  - Note: if you are at the upper end of max IP's per range, Google recommends going one CIDR range higher
    - Example: 4 nodes max = 1024 pod IP's = use /21 CIDR range
- You are planning on creating a GKE cluster with 100 nodes to start but will scale to a max of 900 nodes. What primary subnet IP range do you need to create to allow for planned growth?
  - /22 = 1024 - 4 reserved = 1020 addresses, which can encompass 900 nodes

Back Next

Back to Main
Previous Topic

Node/Pod IP Ranges
Includes max per-node (/24) allocation

Subnet size for nodes	Maximum nodes	Maximum Pod IP addresses needed	Recommended Pod IP range
/29	4	1,024	/21
/28	12	3,072	/20
/27	28	7,168	/19
/26	60	15,360	/18
/25	124	31,744	/17
/24	252	64,512	/16
/23	508	130,048	/15
/22	1,020	261,120	/14
/21	2,044	523,264	/13
/20	4,092	1,047,552	/12
/19	8,188	2,096,128	/11 (maximum Pod address range)

Back

What we are covering

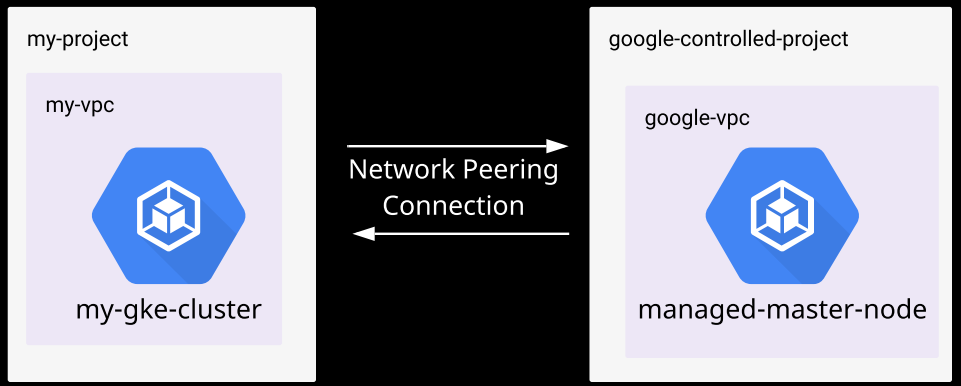
- Start with default VPC/default GKE cluster
  - View network settings
    - VPC-Native enabled
    - Assigning secondary IP range for pods/services
    - Max pods per node
- Create customized addressing cluster
  - Create secondary range in advance
- Pair cluster with HTTP Load Balancer
  - Requires pairing service with **Ingress** object
    - Cloud Armor also requires same Ingress object pairing
  - View URL mapping rules

What are private clusters?

- By default, both nodes and master are publicly accessible
- Private clusters:
  - Remove external IP from nodes
  - Restrict master plane communication with **master authorized networks** or **public endpoints**

Behind the scenes...

- GKE manages the admin overhead of Kubernetes for you
  - Managed "parts" include master node
  - Master node is actually in a Google controlled project, which you interact with to manage your cluster
- By default, managed master node has public access via **`kubectl`** commands
- Private cluster creates **Network Peering** connection to GCP controlled project
  - Network Peering = **PRIVATE** RFC1918 communication with master node



Next

Enable public endpoints/master authorized

- Private cluster cut off from full external access outside of "whitelist" and same subnet as cluster
- Whitelist IP addresses/ranges to access master node
- Two configs: **public endpoint**, **master authorized network**

Public Endpoint
- Whitelisted external (non-RFC 1918) resource

Master Authorized Network
- Whitelisted private (RFC 1918) located resource
  - Example: subnet in same VPC network
- Both connections types authorized/whitelisted with `--master-authorized-networks` command

Back Next

Network Policy

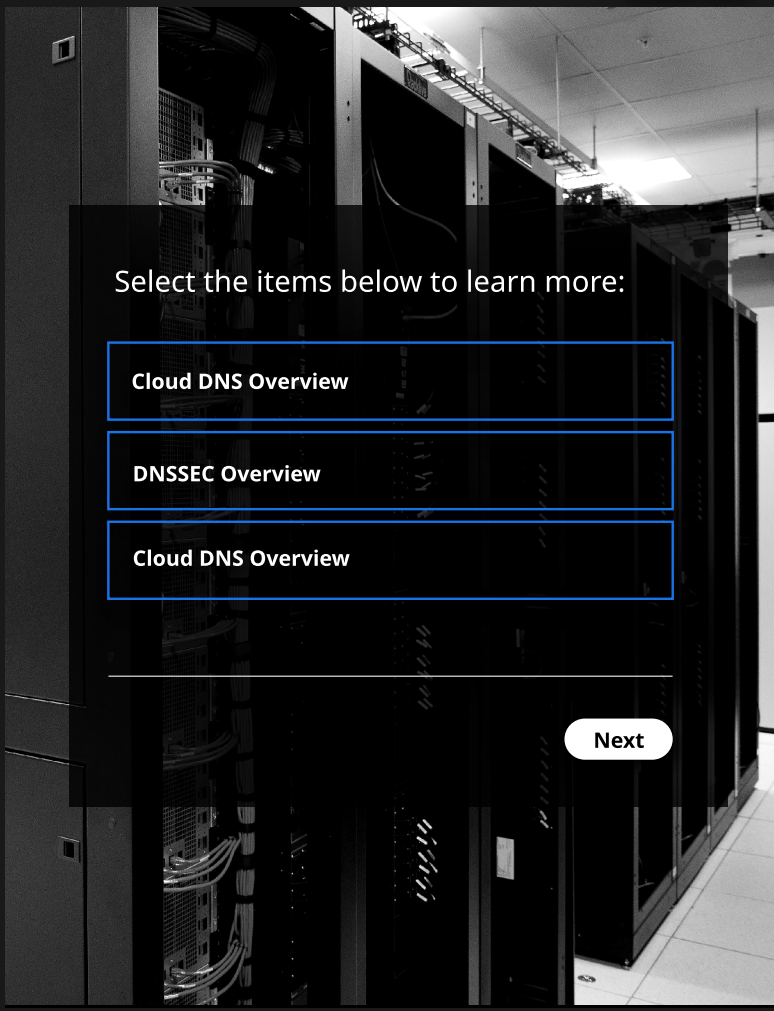
- By default, all cluster pods and services can communicate with each other
- **Network Policy** limits communication by **labels**
- Declared via YAML file via kubectl command
- Network policy must be enabled on cluster config (can be changed after creation)

```yaml
kind: NetworkPolicy
apiVersion:
networking.k8s.io/v1
metadata:
  name:
hello-allow-from-foo
spec:
  policyTypes:
  - Ingress
  podSelector:
    matchLabels:
      app: hello
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: foo
```

Back

What are we doing?

- Create a private cluster with public endpoints and master authorized networks (leave blank for now)
- View subnet and network peering settings
  - Notice Private Google Access automatically enabled
    - Necessary for interacting with GCP services
    - Note on container deployments: must use Container Registry, as external container sources are cut off (e.g. Docker Hub)
- Attempt to use kubectl commands from:
  - Instance in same subnet
  - Instance in another subnet
  - Cloud Shell
- Add public IP of Cloud Shell and subnet IP range to master authorized networks, and try again.



Select the items below to learn more:

Cloud DNS Overview

DNSSEC Overview

Cloud DNS Overview

Next

Course Focus

- Mostly high level, with exam perspectives emphasized
- Assumes basic understanding of DNS, though we will cover the basics for context

What is DNS (Domain Name System)?

- Translating human-readable **domain names** into IP addresses
- Computers only understand IP addresses, but humans don't want to type in IP addresses for every site
- DNS acts as the "phone book" for accessing remote servers
  - Example: google.com = 172.217.6.110

How It Works: DNS Records

- Mapping between DNS resource (IP address) and domain name
- Examples:
  - A record: Address - map host names (google.com) to IPv4 address (172.217.6.110)
    - Includes sub-domains (myaccount.google.com)
  - MX record: Mail exchange - route email to correct mail servers
  - Name server (NS) record: Delegate DNS queries to an authoritative server
    - Map DNS registrar NS records to Cloud DNS

What is Cloud DNS?

- Managed DNS host
  - Both public (external) and private (internal) hosting
- Not a registrar, but an authoritative DNS record host
  - Requires pointing registrar name servers to Cloud DNS
- Only GCP service with 100% SLA
- Managed service = automatic scaling
  - Reliably create millions of DNS records
  - Name servers automatically scale to handle query volume

Cloud DNS Organization

- Hosted in a GCP **project**
- Create one or more **managed zones** for DNS records
  - Container for all DNS records for your **domain**
  - Each **domain** has its own managed zone
    - linuxacademy.com, professionalwireless.net, google.com
      - Typically includes subdomains (account.google.com)
  - Optional - **Delegated Subzone**
    - Subdomain requests (account.google.com) forwarded to separate set of name server records

Cloud DNS Managed Zones

Public Zone

- Visible to the public internet
  - Queried over the public internet
- Public websites (A records), email routing (MX records)

Private Zone

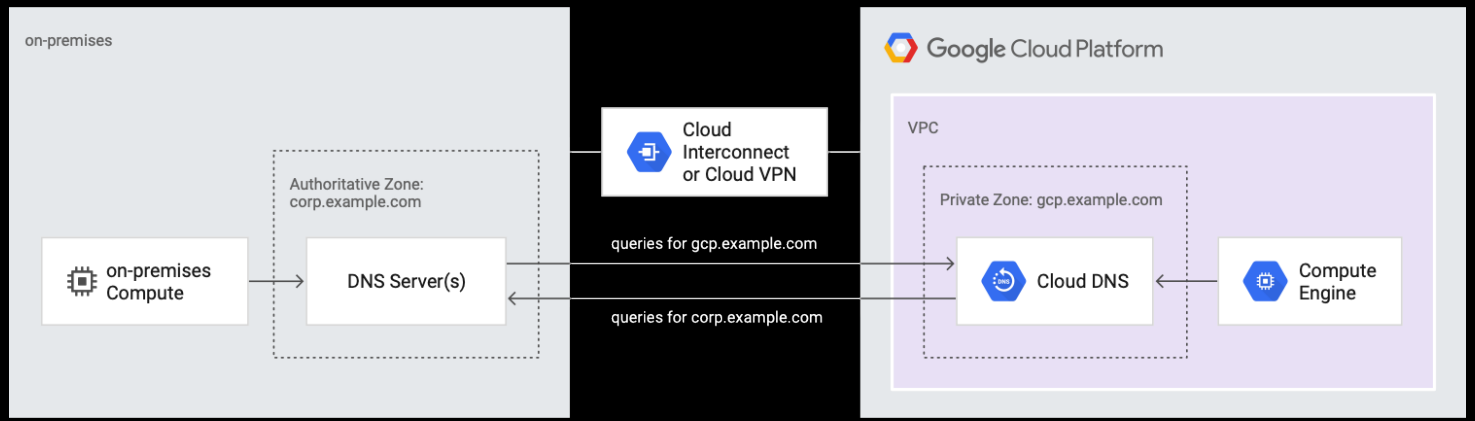
- Not visible or queried over the public internet

Hybrid DNS Scenarios (Private Zones)

- Connecting private networks together over VPN/network peering/etc.
- Need to "merge" DNS resolution between environments
- **GCP to GCP** or **GCP to non-GCP**

DNS Forwarding Zones

- Overrides normal DNS resolution for a zone
  - Instead, forwards queries to a different forwarding target
  - All matching queries forwarded to alternate destination DNS servers instead
  - Used for GCP/non-GCP combinations of environments **only**
  - For GCP to GCP, use **DNS peering** instead
- Integration with on-premises environments
  - Both environments contain their own **authoritative** DNS resolver
  - Cloud DNS handles GCP DNS resolution
  - On-premises DNS server handles on-premises DNS resolution
  - "Other environment" requests forwarded to other DNS resolver via DNS forwarding zone



DNS Peering

- For **GCP to GCP** VPC connections, regardless of connection method
- **Producer** and **consumer** network
  - Only one of the VPC networks is **authoritative** DNS resolver
  - Consumer network forwards requests to producer network

Back

What is DNSSEC (DNS Security)?

- Authenticates responses to DNS lookups
  - Prevents "poisoning" (or spoofing) of responses to requests
  - Provides strong authentication of lookups

How to Enable DNSSEC

- The order is important
  - First, enable DNSSEC in Cloud DNS
  - Then, enable at DNS registrar
    - Add **DS record** to registrar at **top level domain (TLD)**
    - Not all registrars support DNSSEC

Disabling DNSSEC

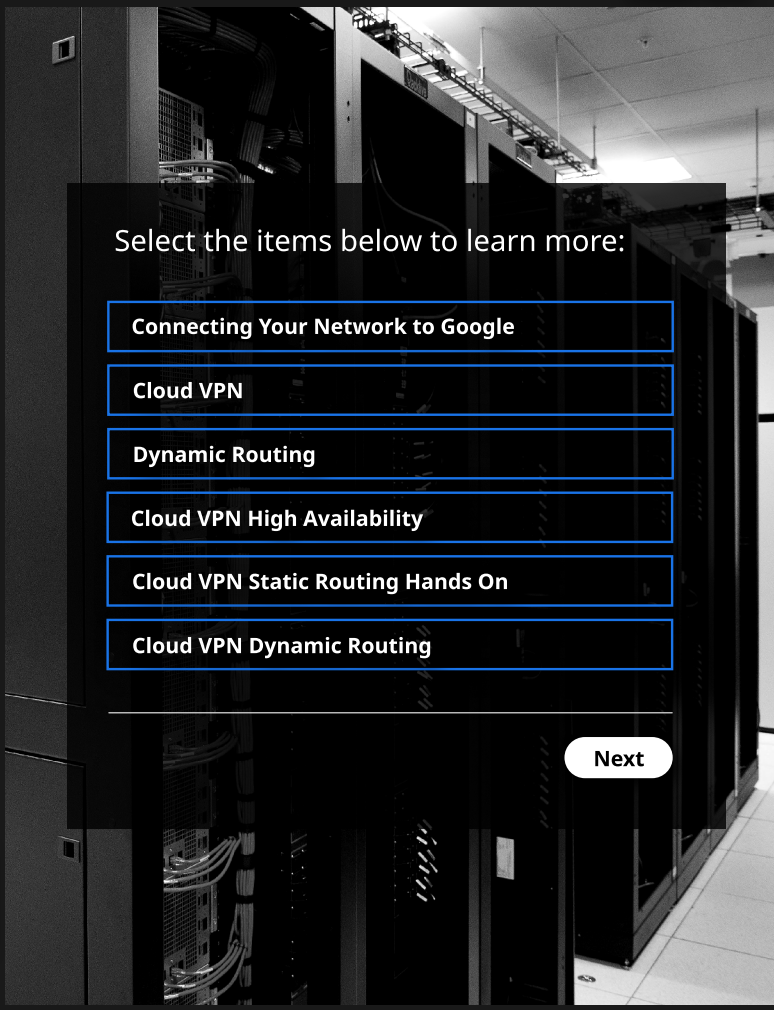
- The order is **critically** important!
- First: Disable DNSSEC at domain registrar
  - Ensures that DNSSEC resolvers can still resolve names in your Cloud DNS zone
- After DS records expire in registrar cache, then turn off DNSSEC in Cloud DNS

Best Practice - TTL settings

- Proper DNSSEC TTL settings prevent resolution errors
- Both too long and too short can cause problems
- Avoid TTLs longer than three days (259200 seconds)

What Are We Doing?

- Create public zone for live domain professionalwireless.net
- Import DNS records from registrar to Cloud DNS zone
  - Note: For BIND format, you must add `--zone-file-format` flag, otherwise it will assume YAML format
  - `gcloud dns record-sets import --zone=(zone name) --zone-file-format (importfile.txt)`
  - For BIND format, must remove name servers and SOA fields to avoid attempted overwrite
- Switch name servers at registrar to Cloud DNS settings
- Enable DNSSEC
  - Enable in Cloud DNS first
  - Enable in domain registrar
  - Add provided settings in Cloud DNS to establish proper chain of trust
- Disable DNSSEC
  - Disable in registrar first, then in Cloud DNS



Select the items below to learn more:

- **Connecting Your Network to Google**
- **Cloud VPN**
- **Dynamic Routing**
- **Cloud VPN High Availability**
- **Cloud VPN Static Routing Hands On**
- **Cloud VPN Dynamic Routing**

Next

What Do We Mean by "Hybrid Networking"?

- Connecting your network/infrastructure (outside of GCP) to GCP over private connection
  - Your network and GCP's network working together
  - Private, internal, RFC 1918 connection from your network (or another cloud network) to GCP
  - As Google puts it: "Get the most out of your cloud"

Three Hybrid Connectivity Products

- **Cloud VPN**
- **Cloud Interconnect**
- **Cloud Peering**

At a Glance:

- **Cloud VPN** = Private, encrypted tunnel to GCP VPC over public internet connection
- **Cloud Interconnect** = dedicated, physical connection to GCP VPC
- **Cloud Peering** = dedicated connection to Google, but not a GCP VPC
  - **Not** the same as **VPC Network Peering**
  - Not an RFC 1918 connection

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On
Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

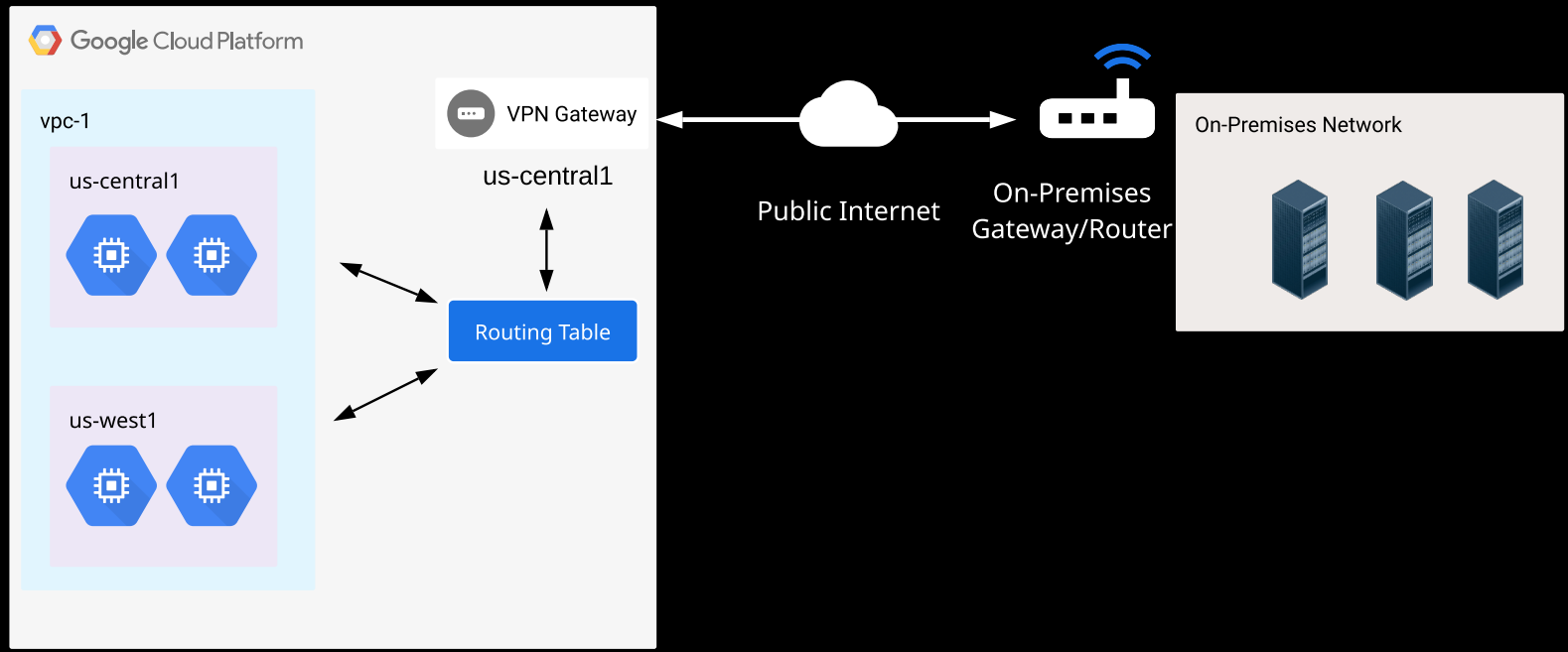
Network Design and Monitoring
Section 5

"Just the Facts"

- The option for "most of us"
  - Low cost
  - Available from any location with internet connection
  - Ideal for low-volume data connections
    - Example: Transfer legacy resources to GCP
- Site-to-site VPN connection over IPSec
- Cloud VPN gateways are a **regional** resource
  - Can serve other regions in VPC
- Connect internal network to GCP over encrypted tunnel over public internet
  - Up to **3 Gbps per tunnel**
  - Best performance over Cloud Peering connection
  - Can use multiple (up to eight) tunnels for increased performance
  - 3 Gbps x 8 = 24 Gbps per gateway combined
- **Static** and **dynamic** routes (using Cloud Router)
- Supports **IKEv1** and **IKEv2** using shared secret
- Site-to-site connection only
  - No site-to-client option available
  - Example: Connecting to GCP VPN via laptop

Next

Back to Main



Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On

Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering

Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

Back to Main

Routing Methods

- Dynamic vs. static routing
- **Dynamic:** Routes are created/updated automatically
  - Preferred option whenever possible
  - Uses Border Gateway Protocol (BGP)
  - Requires using **Cloud Router** service
- **Static:** Manually create/update local and peer routes
  - Only when BGP routing is not available
  - **Route-based** vs. **policy-based** routing options
    - Route-based = specify remote (peer) IP ranges to connect to (right side)
    - Policy-based = specify both local ranges (left side) and remote range (right side)
      - Route-based preferable over policy-based; use policy for peer routers that require it

Back

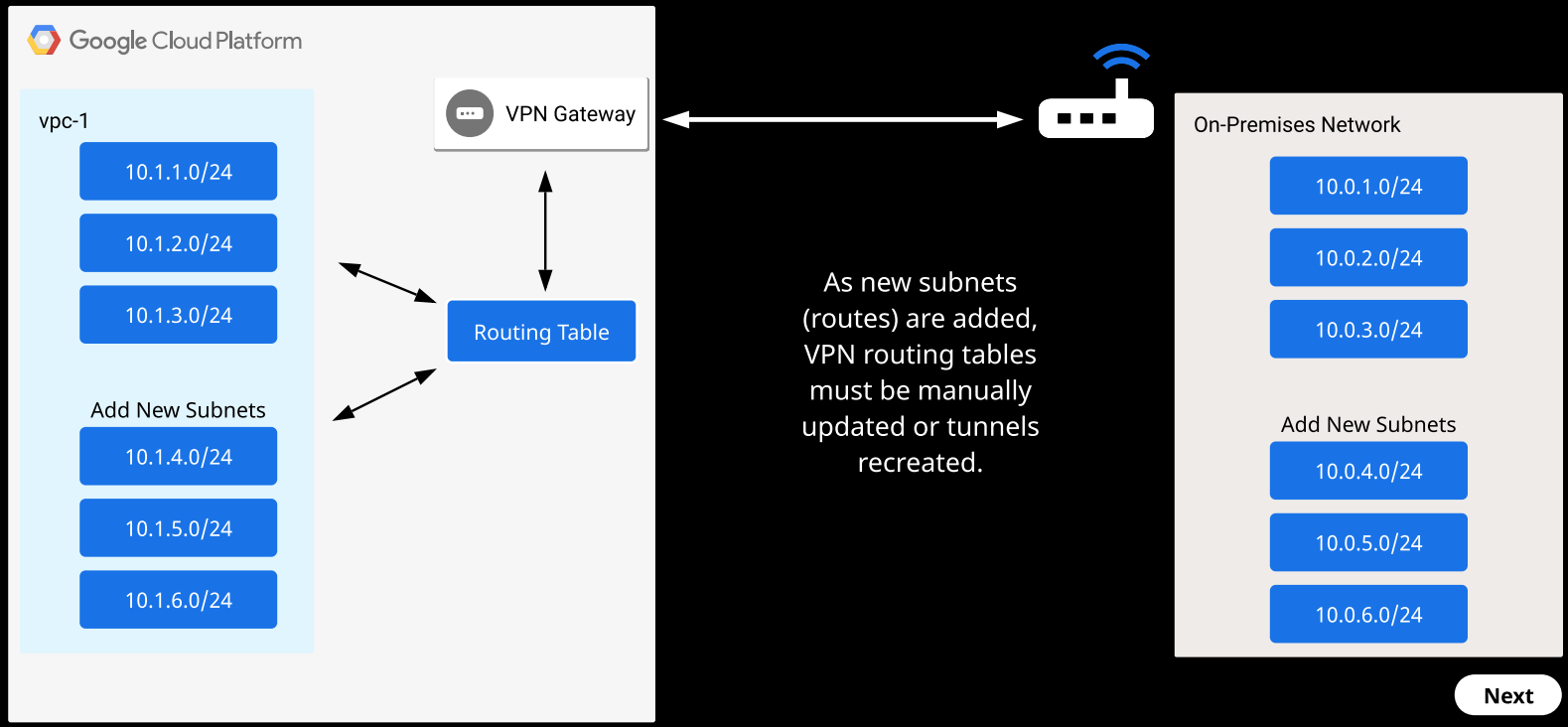
What is Dynamic Routing?

- Not required for Cloud VPN, but makes life much easier

Static vs. dynamic routing

- Static = manually specify every subnet to connect
  - If you later add a new subnet to peer network, must add route in GCP
  - Result: higher administrative overhead

Static Routing and Network Growth

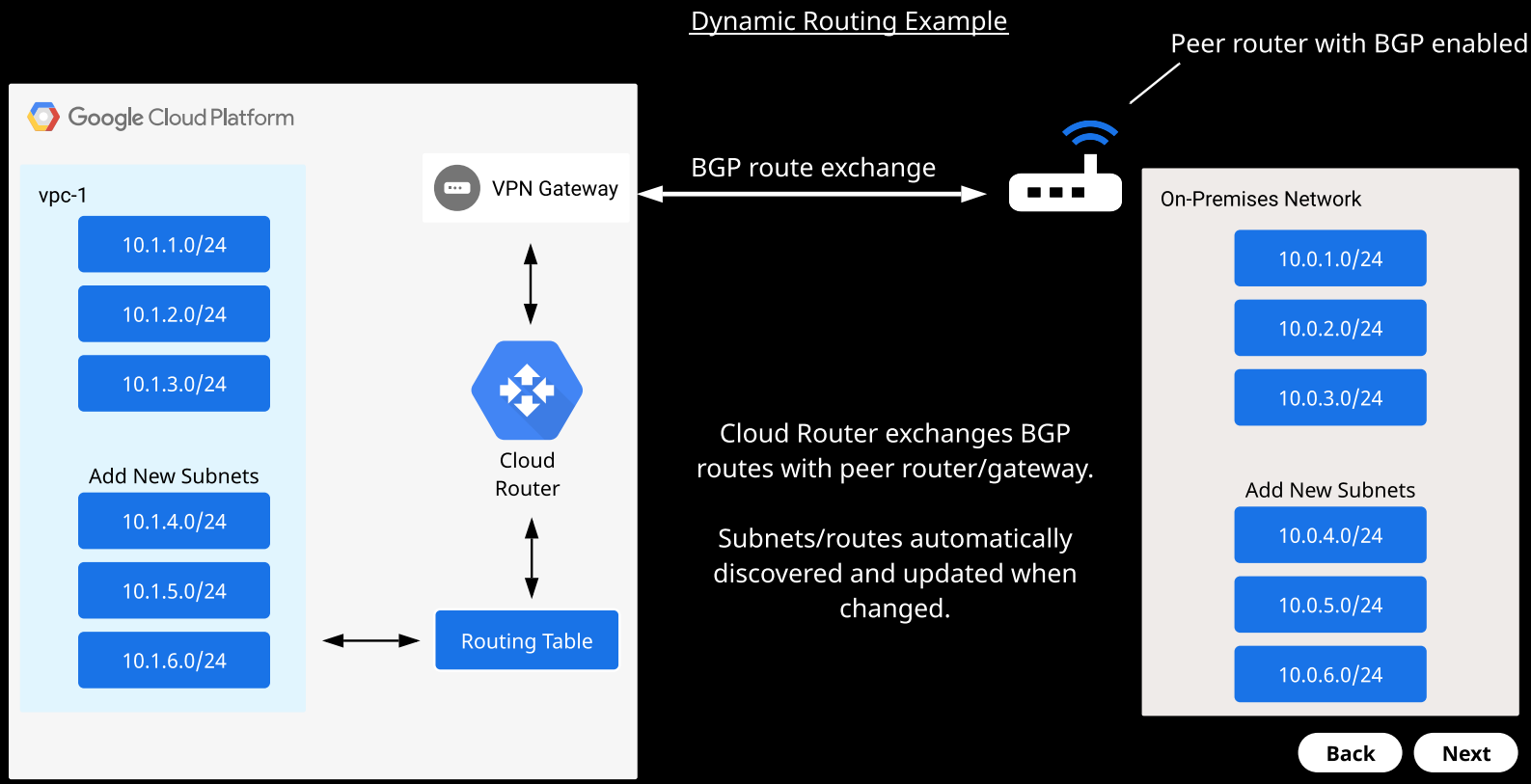


As new subnets (routes) are added, VPN routing tables must be manually updated or tunnels recreated.

Next

Benefits of Dynamic Routing

- Routes automatically discovered and updated as needed
- Requires **Cloud Router** and peer connection that supports BGP



Back Next

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On

Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering

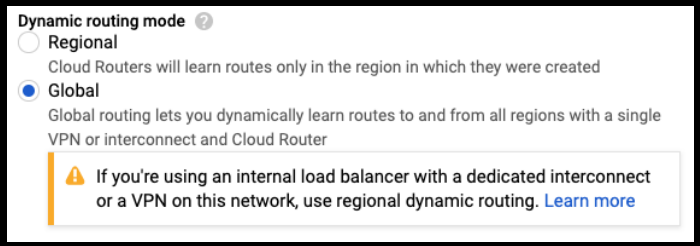
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

Back to Main

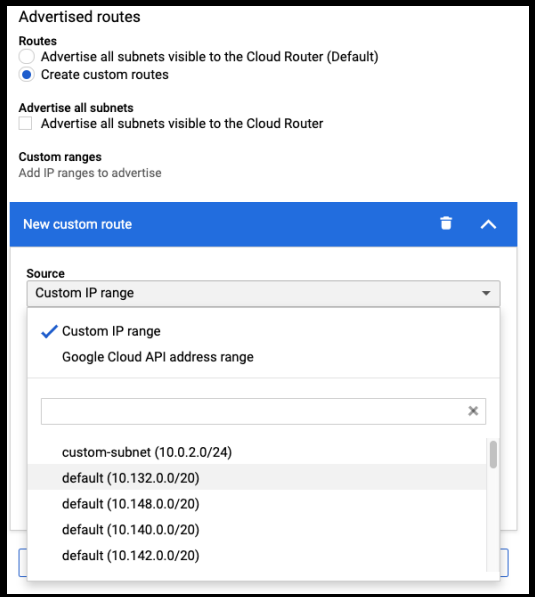
Anatomy of BGP Session

- **Private Autonomous System Number (ASN)** for both Cloud Router and peer gateway
  - Unique identifier
- **Link-local** IP address for Cloud Router and peer gateway
  - Example: 169.254.1.1 and 169.254.1.2
- Misconfiguration of peer session in either direction will cause session to fail



Dynamic Routing Modes

- Specify scope of automatically advertised and learned routes
  - **Regional** (default): Only advertises/learns routes for its own region
  - **Global**: Advertises/learns routes for all regions in VPC
- Applied across entire VPC
- Why use regional dynamic routing?
  - Don't want to advertise all region subnets
  - Internal load balancing over VPN/interconnect



Limit Advertised Routes

- By default, Cloud Router advertises all known VPC routes/subnets (regional or global) to peer network
- Can choose which subnets are advertised
- Must manually update newly needed routes/subnets
- Does not affect learning new peer network routes

Firewall Rules

- BGP operates over TCP port 179
- Most peer gateways enable it automatically; some have to manually configure it
- Do not need to configure on GCP end

Back

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On

Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering

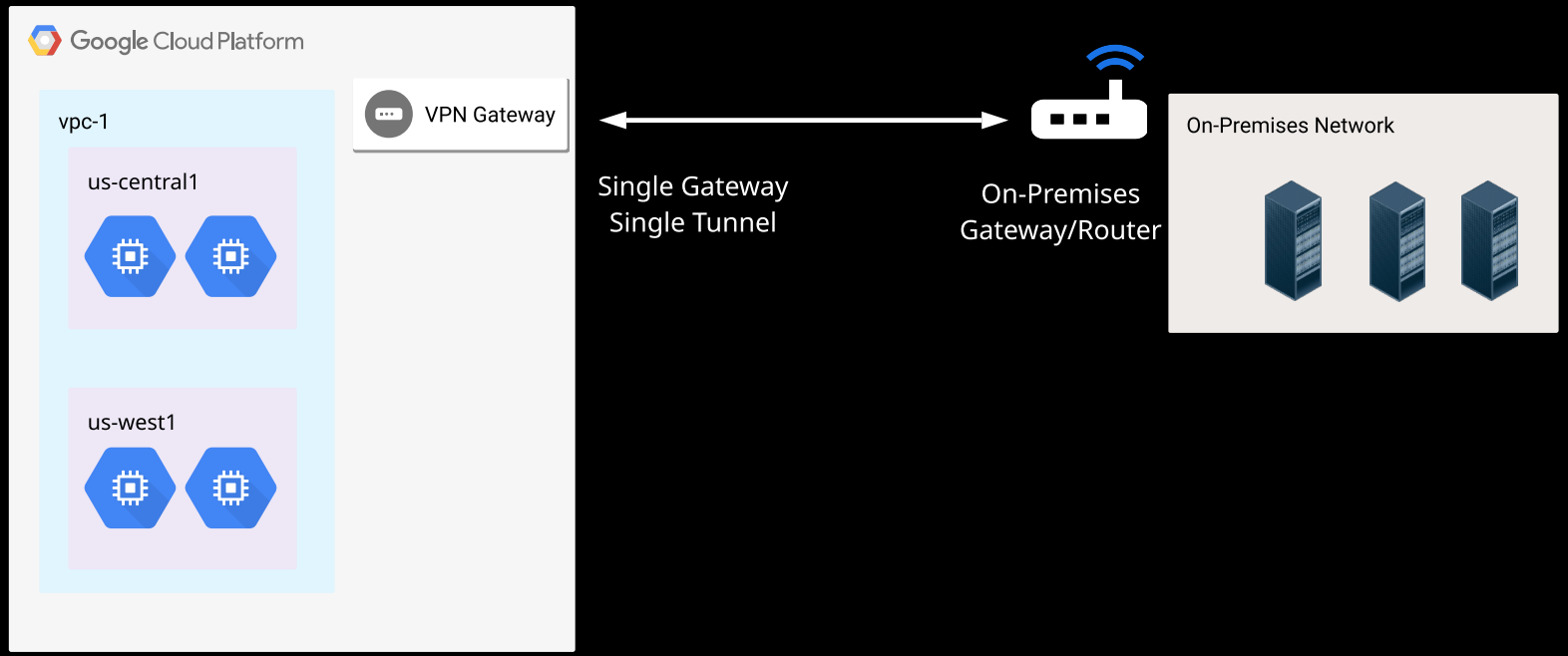
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

What is High Availability?

- Key terms:
  - **Redundancy**, **failover**, and **throughput**
- "Don't put all your eggs in one basket"
- Allows for interruption of service of individual gateways and/or tunnels
  - A second gateway/tunnel will continue to serve traffic in parallel

Non-Redundant Example



Next

Back to Main

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On

Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering

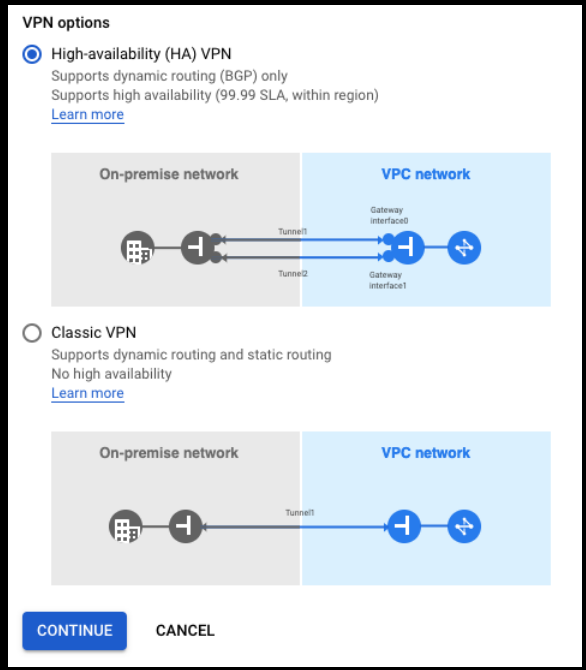
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

Back to Main

Classic vs. High-Availability VPN (New Service)

- May 2019: High-availability VPN made available
  - More easily manage redundant gateways
  - Requires Cloud Router
- "Regular" VPN now referred to as "classic" VPN
- Exam should test on general topologies, not on difference between services

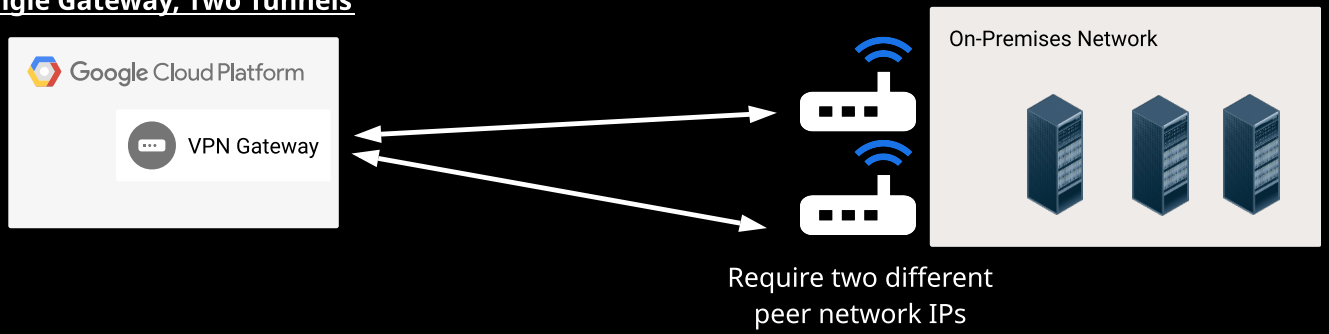


Back
Next

High-Availability Topologies

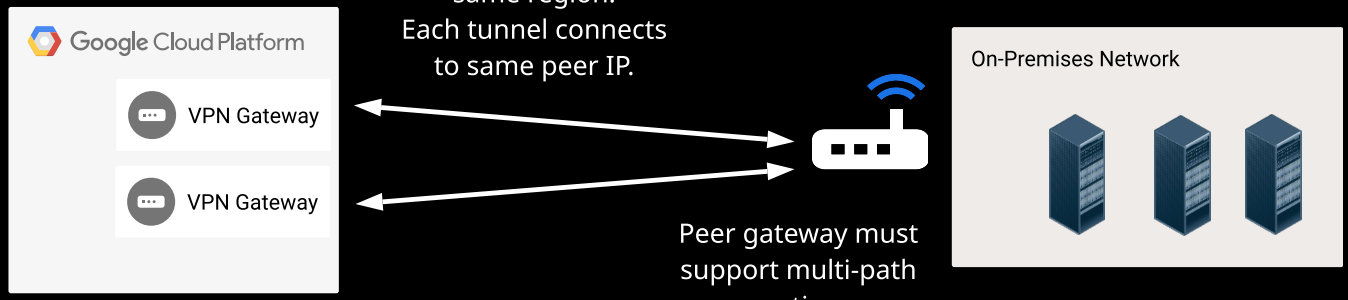
- Three types
  - **Single gateway/two tunnels**
  - **Two gateways/one tunnel each**
  - **Two gateways/two tunnels each**
- Different setups have different peer gateway requirements
  - Multiple peer IP addresses
  - Peer supports multi-path routing
- Route failover/load balancing
  - Failover: Assign one route/tunnel higher route priority. If it fails, VPN service will failover to secondary route.
  - Load balance: Assign each route/tunnel same route priority. Load is balanced between each tunnel.
    - Also useful for increasing throughput

Single Gateway, Two Tunnels

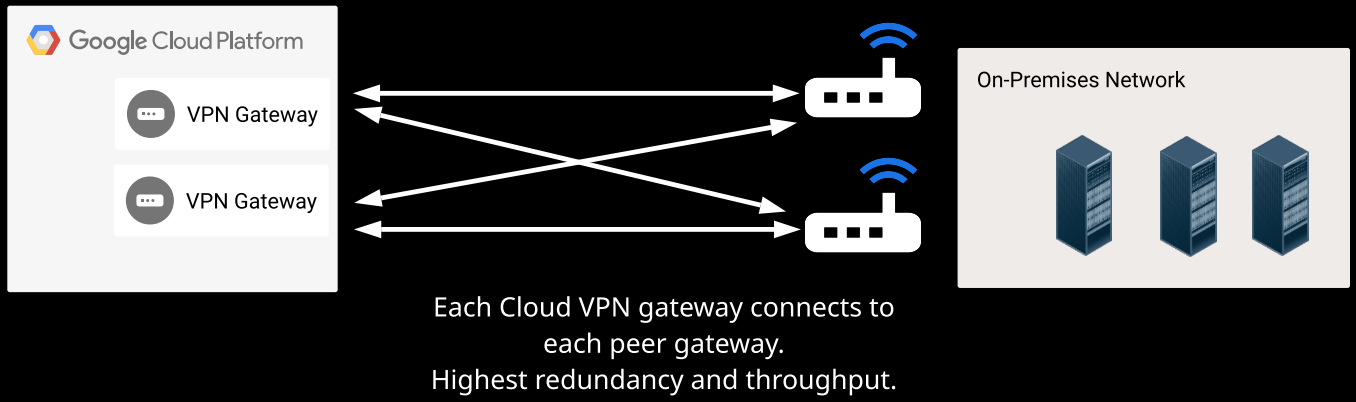


Require two different peer network IPs

Two Gateways, One Tunnel Each



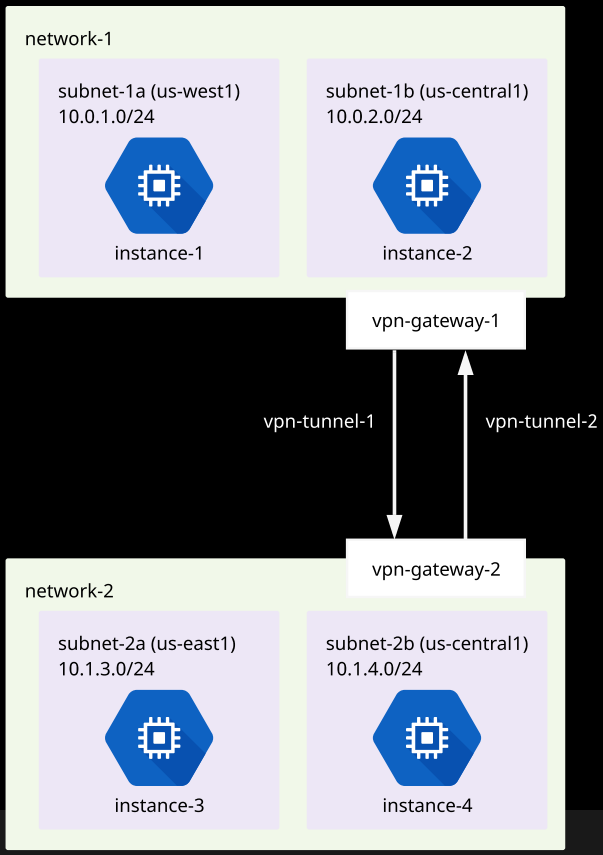
Two Gateways, Two Tunnels Each



Back

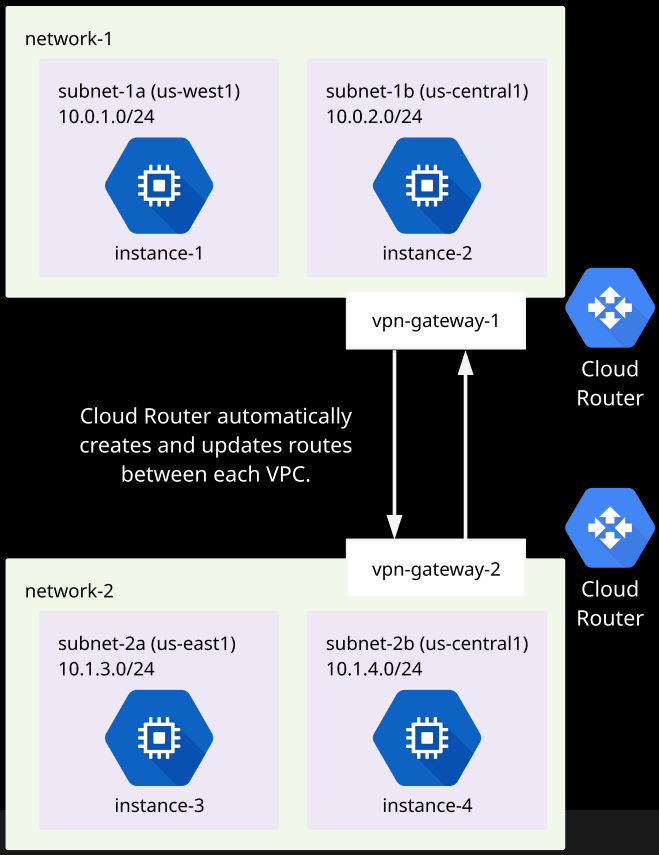
What Are We Doing?

- Start with two custom VPCs with populated resources plus reserved static IP addresses
- Thoroughly explore Cloud VPN service and options
- Create static route between two VPCs



What Are We Doing?

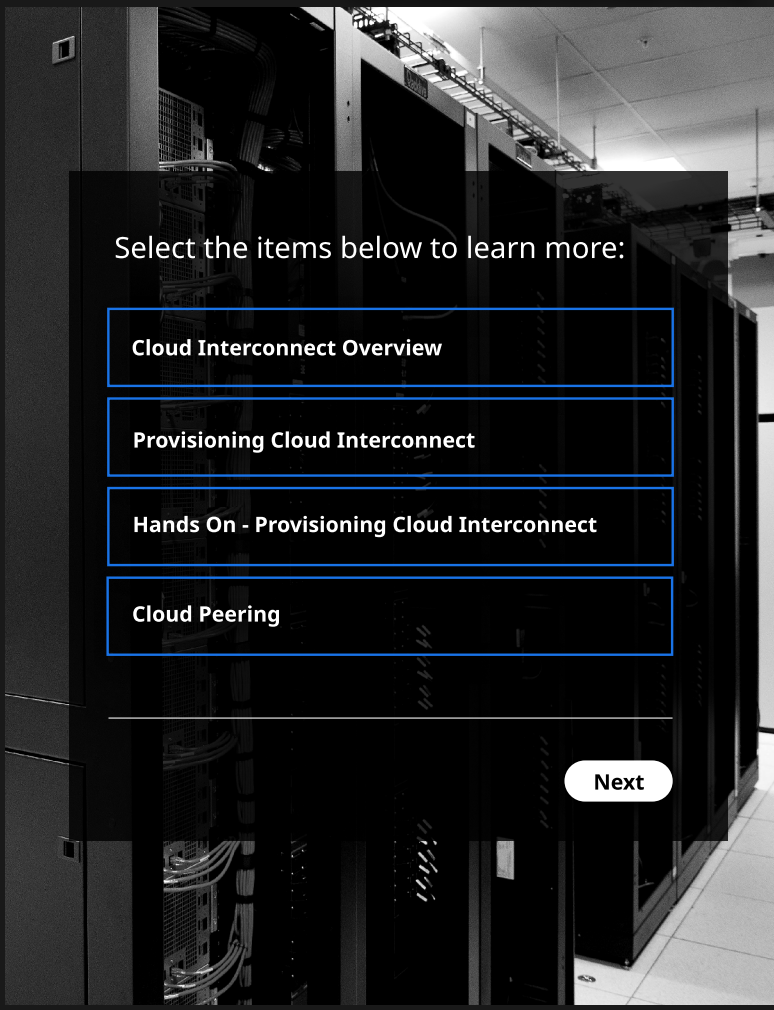
- Start with two custom VPCs with populated resources
- Create Cloud Router for each VPC, and explore options
- Create dynamic route between two VPCs



Back to Main

Hybrid Networking

Cloud Interconnect and Cloud Peering



Select the items below to learn more:

- Cloud Interconnect Overview
- Provisioning Cloud Interconnect
- Hands On - Provisioning Cloud Interconnect
- Cloud Peering

Next

What is Cloud Interconnect?

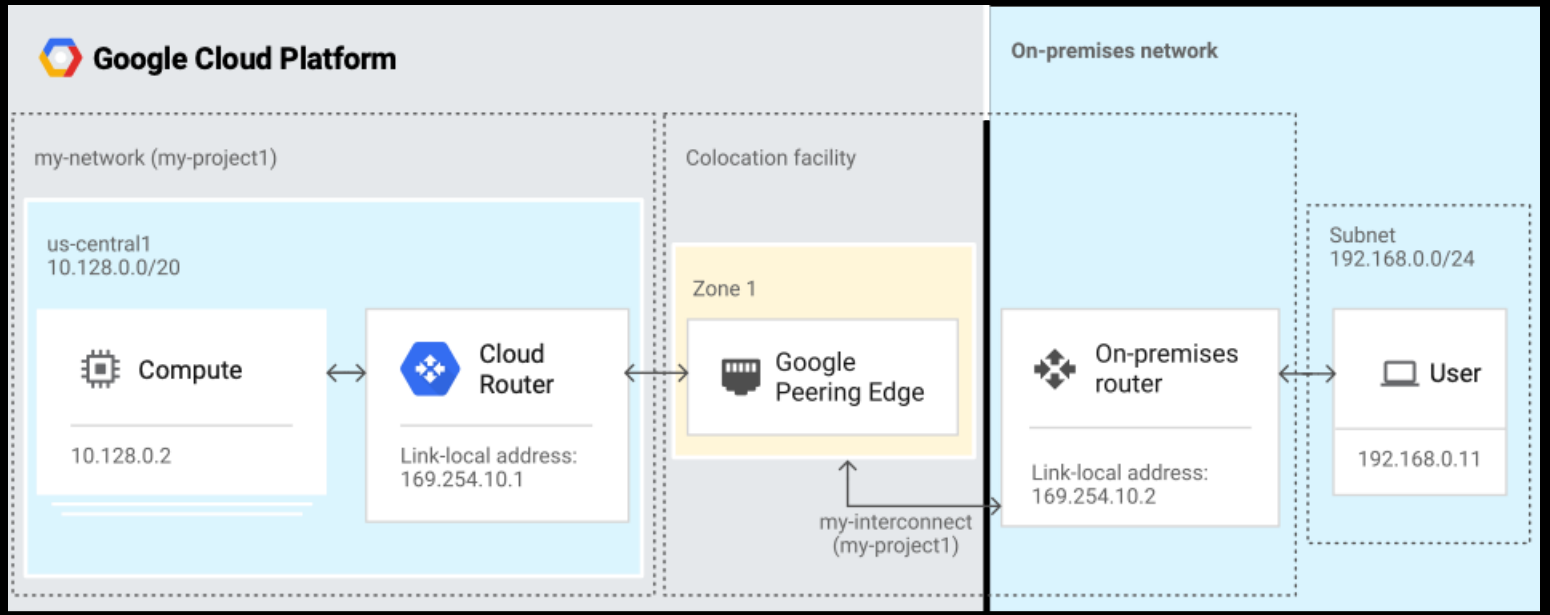
- Extend your **on-premises** network to a **Google Cloud VPC** over a **direct physical connection**
  - Not over public internet
  - Dedicated or partner connection
  - Starting from 50 Mbps to 200 Gbps
  - Connection not encrypted
    - Use application-level encryption or your own VPN (not Cloud VPN)
  - Overall more expensive than Cloud VPN
    - Recommended to start with Cloud VPN, and determine if your needs require an interconnect option

Interconnect Types

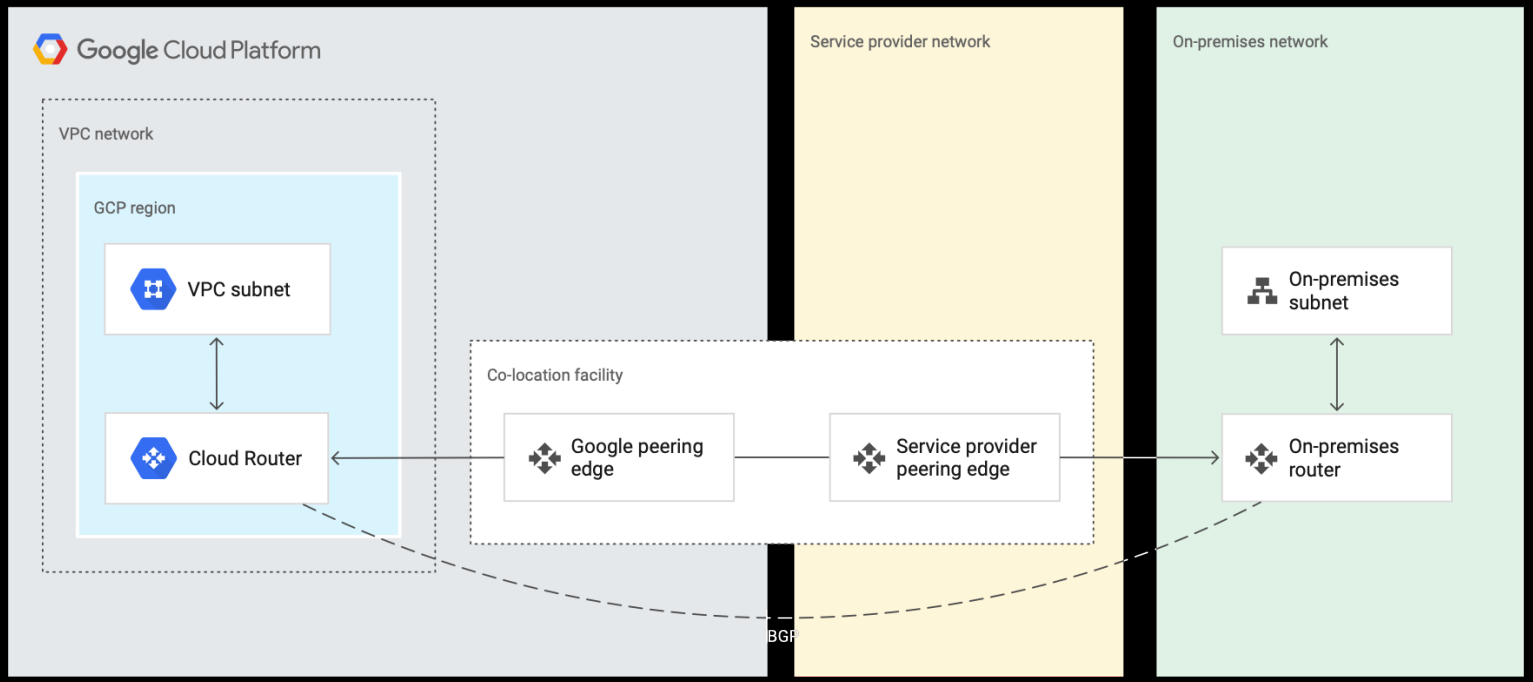
Dedicated and Partner

- **Dedicated Interconnect**
  - Dedicated connection to GCP
  - Requires on-premises router in GCP **colocation facility**
    - Currently 62 locations worldwide (subject to change)
  - Minimum of **10 Gbps**, up to **200 Gbps** (can create multiple circuits)
    - Available in 10 Gbps increments to 80 Gbps, or 100/200 Gbps circuit
  - **Layer 2** connection
- **Partner Interconnect**
  - Connect to GCP VPC via service provider connection
    - More widely available
  - Minimum of **50 Mbps**, up to **10 Gbps** per circuit (can create multiple circuits)
  - **Layer 2** and **Layer 3** connection varieties

Dedicated Interconnect



Partner Interconnect



Back Next

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On
Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

Back to Main

Comparing Each Interconnect Type

	Dedicated Interconnect	Partner Interconnect
Speed	10 or 100 Gbps circuits	Flexible options with a minimum of 50 Mbps up to 10 Gbps per circuit. Traffic between networks flows through a service provider, not through the public internet.
Availability	Requires routing equipment in colocation facility that supports the regions you want to connect to	Use any supported service provider to connect to Google
BGP configuration	Requires Cloud Router and BGP configuration for on-premises router	• Layer 2 connections: Must configure BGP on your on-premises routers and Cloud Routers • Layer 3 connections: Configuration of your Cloud Routers and their peers are fully automated; service provider handles BGP configuration
SLA	End-to-end SLA	Between Google's end and service provider
Pricing	• $1,700 a month for each 10 Gbps circuit or $13,000 a month for each 100 Gbps circuit • $0.10 per hour for each interconnect attachment (VLAN) • Discounted egress charges from VPC network to your on-premises network	• Service provider sets rates for speed levels • $0.10 an hour for each interconnect attachment • Discounted egress charges from VPC network to your on-premises network

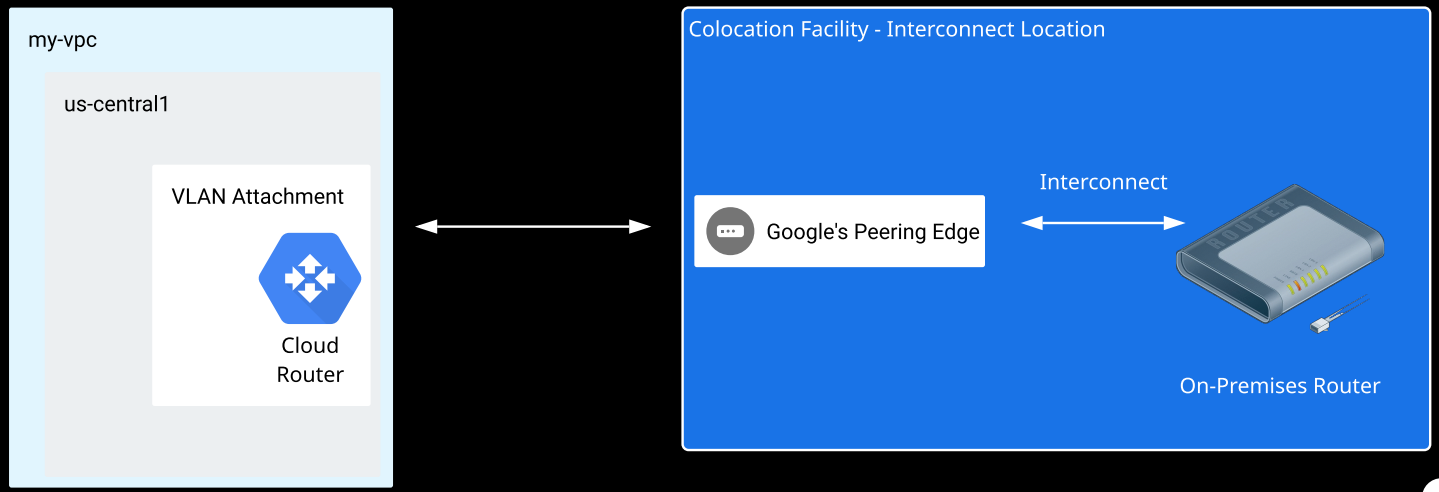
Back

Anatomy of a Cloud Interconnect Connection

- Some differences between direct and partner, but many similarities

Key Terms

- **Interconnect:** Physical connection between Google and on-premises network/service provider
  - In a **colocation** facility at Google's peering edge
    - Peering edge = hop-off points from public internet to Google's private network
- **Cloud Router:** Same Cloud Router in Cloud VPN discussion
  - Required to exchange BGP routes between VPC network and your network/partner network
- **VLAN Attachment:** Virtual point-to-point tunnel between interconnect and single region in VPC network
  - Also known as **Interconnect Attachment**
    - Where the Interconnect attaches to your VPC
  - Can create multiple VLAN attachments to different VPC regions
    - Or multiple attachments from single Interconnect to multiple VPCs



Next

Direct Interconnect

- **Layer 2** connection
- Increments of 10 Gbps circuits
- Must meet at Google colocation center (62 worldwide — subject to change)
- Requires BGP capabilities in on-premises router

Provisioning Dedicated Interconnect — Four Steps

1. Order the Dedicated Interconnect — submit details to Google
2. Send emailed LOA-CFA to your vendor
    - Letter of Authorization and Connecting Facility Assignment
3. Test the interconnect
    - Sent configurations are lightly tested
    - Before attaching to VPC
4. Create VLAN attachments in VPC and set up BGP session via Cloud Router

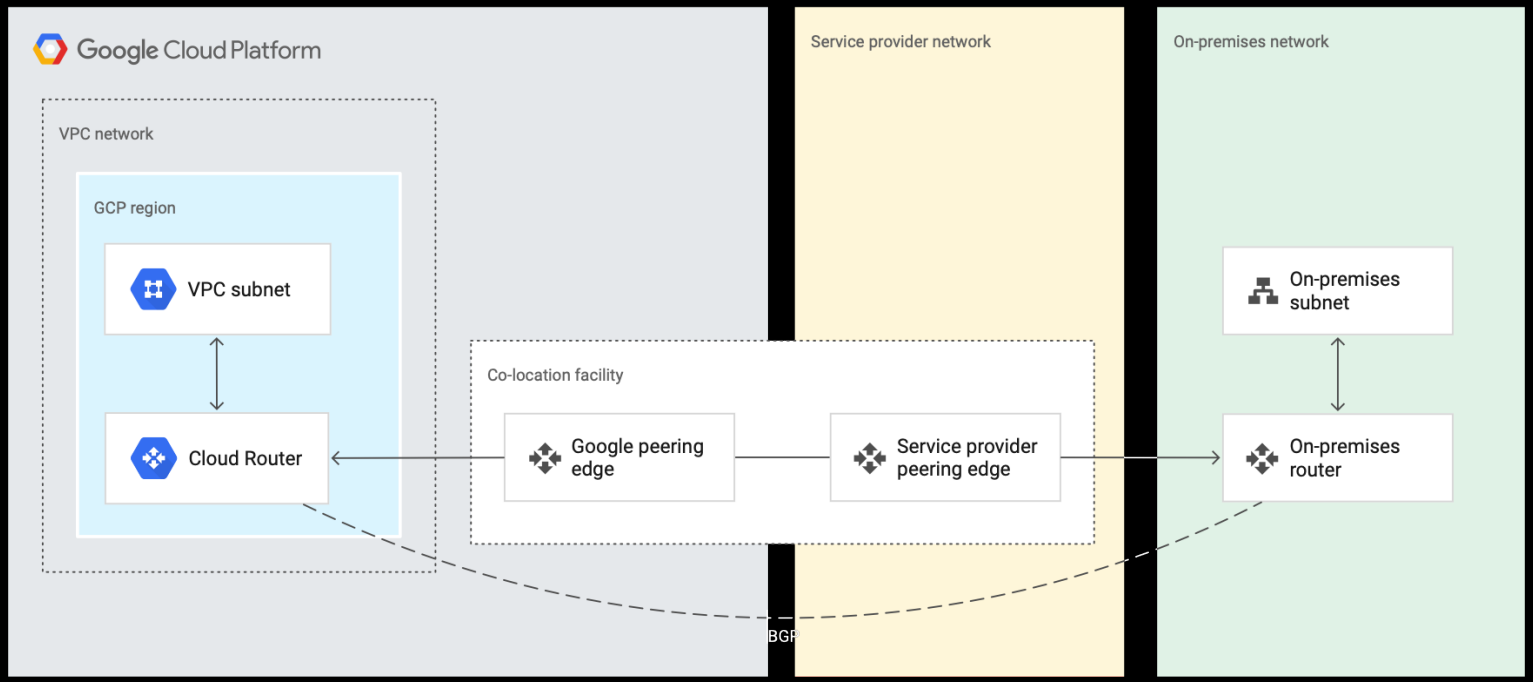
Partner Interconnect

- **Layer 2 or 3** connection
    - Layer 2 requires on-premises BGP negotiation; Layer 3 does not
- Increments between 50 Mbps and 10 Gbps

Provisioning Partner Interconnect

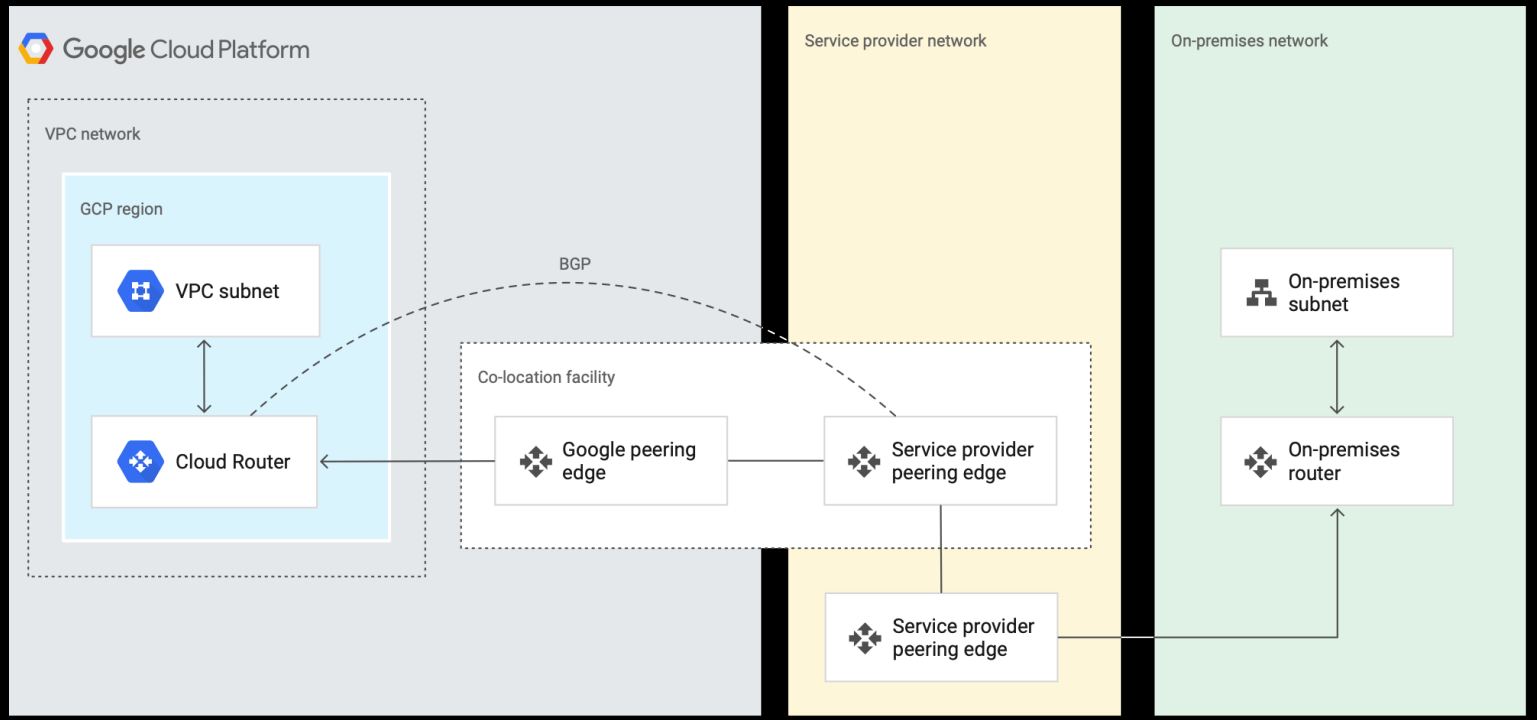
1. Create VLAN attachment
    - Creates unique pairing key to send to service provider
2. Request connection from service provider
3. Activate connection within VLAN attachments (above)
4. Configure BGP
    - Requires Cloud Router
    - Layer 2 connection: Establish BGP session with on-premises router
    - Layer 3 connection: Service provider establishes BGP session with Cloud Router
    - Google automatically adds service provider BGP info to Cloud Router

Partner Interconnect Layer 2 Connection



Back Next

Partner Interconnect Layer 3 Connection



Back Next

Course Navigation

Hybrid Networking
Section 4

Cloud VPN
Connecting Your Network to Google
Cloud VPN
Dynamic Routing
Cloud VPN High Availability
Cloud VPN Static Routing Hands On
Cloud VPN Dynamic Routing Hands On
Cloud Interconnect and Cloud Peering
Cloud Interconnect Overview
Provisioning Cloud Interconnect
Hands On - Provisioning Cloud Interconnect
Cloud Peering
Private Networking
Private Access on Google Cloud Platform
Cloud NAT Overview
Cloud NAT Hands On

Network Design and Monitoring
Section 5

Back to Main

Other Notes

- **Shared VPCs**
  - VLAN attachments go in **Host Project**
- **IAM Roles**
  - VLAN attachments require **Network Admin** role or higher to create/edit/delete

Back

What Are We Doing?

- Limited demonstration of creating interconnect
  - Full demo requires ordering an interconnect
- Creating VLAN attachments
- Cloud Router configurations

What is Cloud Peering?

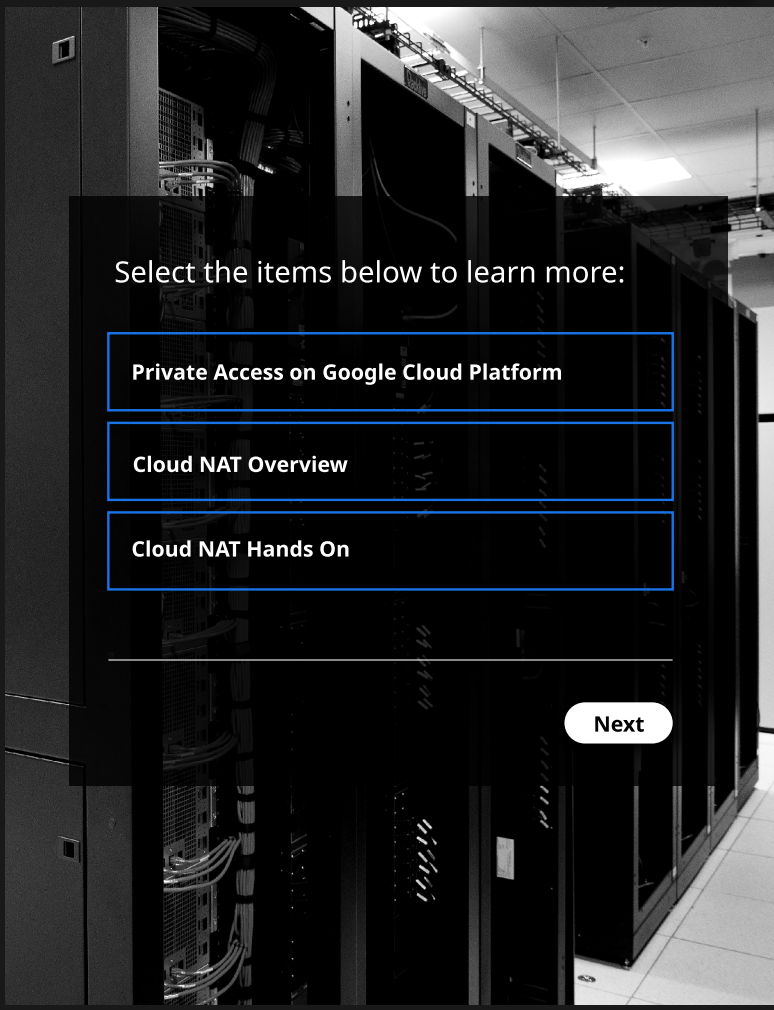
- Direct connection from your business to Google's edge network
  - Not a GCP product, but need to know the differences
    - **Not** tied to a Google Cloud VPC
    - Can use GCP resources with Cloud Peering, but not required
  - Direct access to YouTube, G Suite, other Google services (including GCP)
- **Layer 3** connection
- **Dedicated** and **Carrier** options
  - Similar to Dedicated vs. Partner Interconnect
- **Dedicated** = directly through Google
  - 10 Gbps per link
  - Meet at over 100 locations worldwide
- **Carrier** = work with third-party partner
  - Speed depends on carrier

Use Case (Why Not Just Use the Public Internet?)

- Scenario: Require limited public internet exposure for on-premises network
  - Org needs G Suite access, which operates over the public internet
- Solutions:
  a. Create an **on-premises DMZ** (perimeter network) that is exposed to the internet
  b. Connect on-premises network directly to Google's network, allowing access to Google services without exposing to the rest of the internet = **Cloud Peering**
- In other words: Cloud Peering connects you to Google services and APIs without requiring a public internet connection

Other Cloud Peering Benefits

- Shorter route to Google services (less hops)
- Reduced egress pricing for GCP resources
- Better Cloud VPN tunnel performance



Select the items below to learn more:

- **Private Access on Google Cloud Platform**
- **Cloud NAT Overview**
- **Cloud NAT Hands On**

Next

Why Do We Care About Private Access?

- Most GCP-managed services communicate over public IP
- Problem: How do you connect to GCP-managed services without a public IP address?
- Solution: Enable private access to managed GCP services

Private Access Methods

- **Private Google Access**
- **Private Google Access for on-premises hosts**
- **Private services access**

What We Need to Know

- Use case for each private access solution
- How to enable each one

Two Types of Managed Services on GCP

Hosted on Google Infrastructure

- Accessed via API or internal Google Proxy
- Not in form of instances to interact with
- GCS, Spanner, BigQuery
- **Solution: Private Google Access**

Hosted on Google Cloud Infrastructure

- Similar infrastructure to VPC-hosted resources
- "Managed instances"
- Cloud SQL, Private GKE Cluster
- **Solution: Private services access**

Next

Back to Main

Private Access Options

Private Google Access

- Connect to public Google APIs and services without external IP address
- For GCP VPC resources
- Setup: Click private Google access checkbox for subnet
  - Join internal instance to private access-enabled subnet

Private Google Access for on-premises hosts

- Same Private Google Access as above, but for non-GCP resources
  - Connected over VPN/Interconnect
- Setup: Add private domain and IP ranges to on-premises DNS CNAME records and firewall, plus routes on GCP
  - **restricted.googleapis.com**
    - **199.36.153.4/30**
  - **private.googleapis.com**
    - **199.36.153.8/30**

Private services access

- Connect to Google or third-party services through VPC Network Peering connection
  - Examples: GKE Private Cluster, Cloud SQL internal access
  - Above managed services negotiate the connection for you
- Key terms: Service producer, service producer network
  - **Service producer:** Third party you are connecting to
  - **Service producer network:** Network created exclusively for your private connection
- Setup: Create private connection in VPC settings
  - Allocate internal IP range
  - Connect to service
  - On-premises access: Export custom routes in VPC peering

Challenges with Private Resources

- Problem: GCP private instance needs to access public resources
  - Update repositories, install software
- Solution: Use **network address translation (NAT)**
  - Managed service: **Cloud NAT**

What is Network Address Translation?

- Enables private resources without a public IP address to connect to the internet
  - NAT gateway (traditionally your router) will "translate" internal IPs into public addresses for external communication
  - Example: Traditional home WiFi network, corporate office network

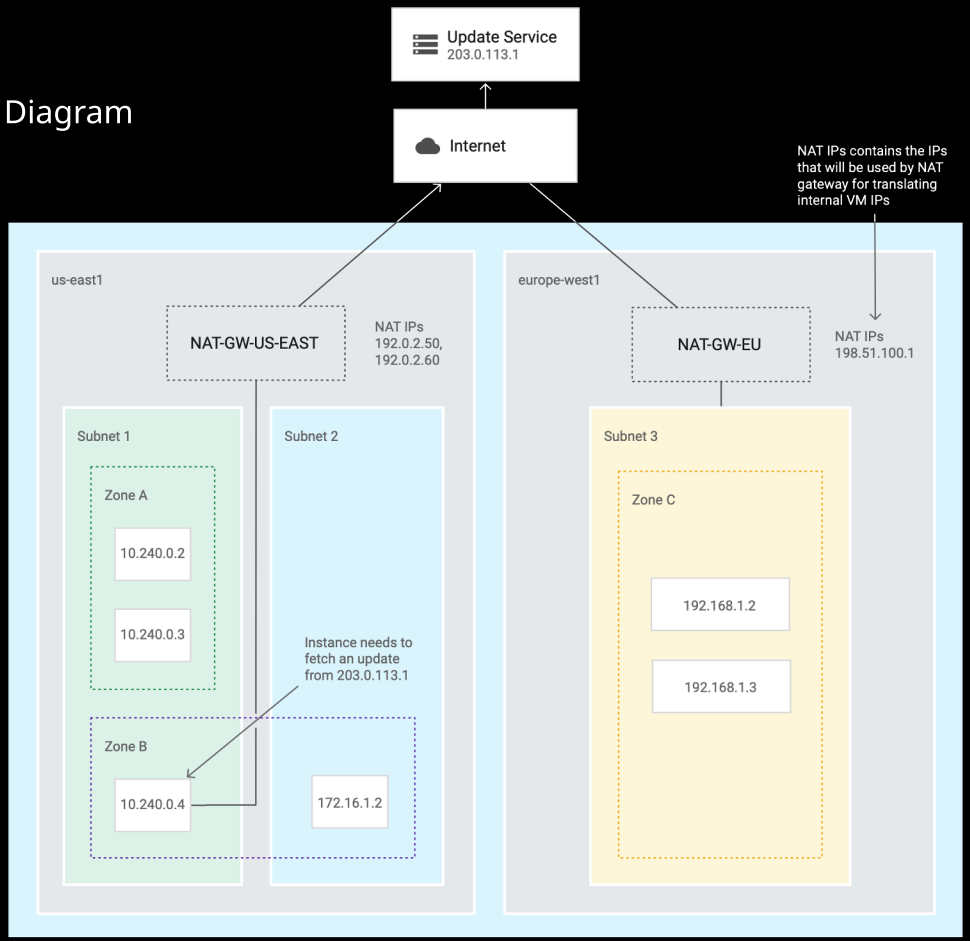
What is Cloud NAT?

- Managed NAT gateway for GCP VPC resources
- Fully managed, software defined service
  - No instance/appliance to manage
  - No NAT proxy instance to configure
  - High scalability
- Requires Network Admin role or higher
- Works in conjunction with Cloud Router to dynamically manage routes

Next

Back to Main

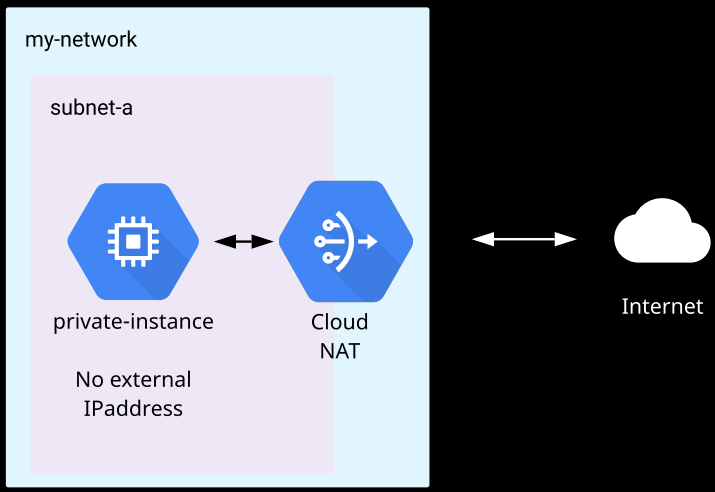
Cloud NAT Diagram

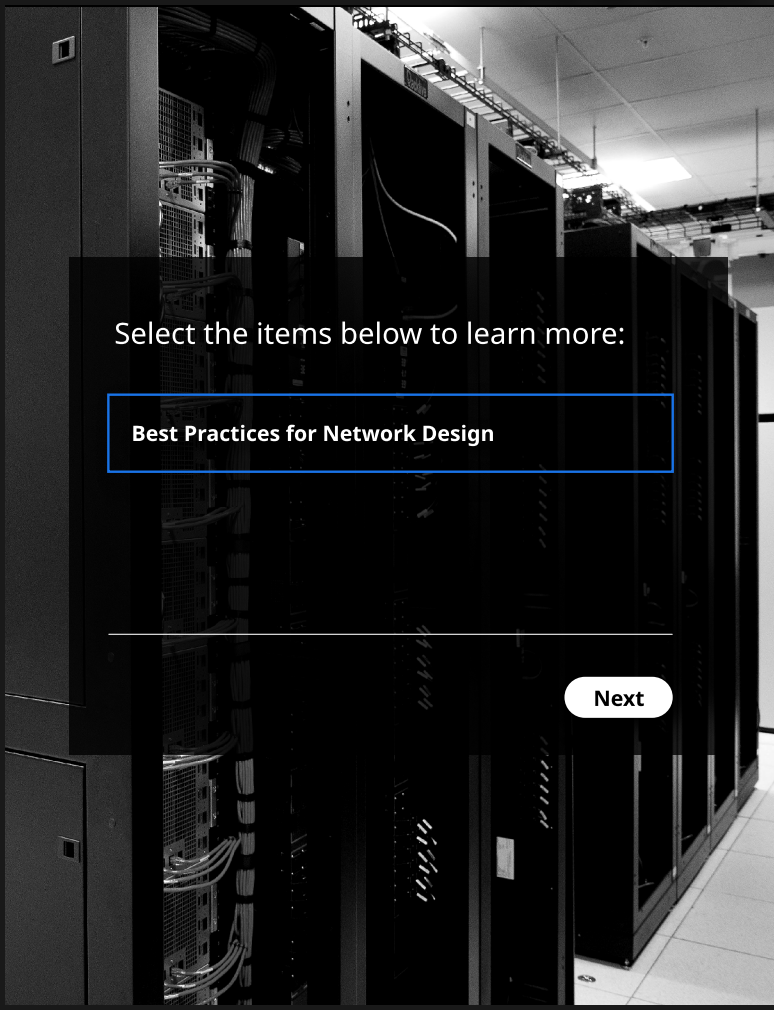


Back

What Are We Doing?

- Enabling public internet access for a private (no external IP) instance
- Start with custom VPC, private instance, and firewall rule to allow port 22
  - Script to create environment in lesson description
- Create a Cloud NAT gateway to enable public internet access





Select the items below to learn more:

Best Practices for Network Design

Next

Detailed Network Design Guide

- https://cloud.google.com/solutions/best-practices-vpc-design

Plan Network Design in Advance

- “Measure twice, cut once”
- Work with stakeholders
  - Application owners, security architects, operations managers
- Who needs access to what
  - Everyone gets same rights in same project?
  - Use shared VPC across projects for different permissions?
- Some decisions cannot be easily changed later
  - Subnet addressing
  - Resource placement - cannot move instance to new subnet, must recreate

Use custom-mode VPC networks

- Default auto-mode VPC is great for quickly testing something, but complex, connected networks need custom subnets
- Connecting networks must avoid conflicting/overlapping subnets
  - Auto-mode networks use identical subnet ranges = conflicts
  - "Conflict" includes different, but encompassing subnets
    - 10.2.0.0/16 overlaps with 10.2.1.0/24 = conflict
    - 10.1.0.0/16 does not overlap with 10.2.1.0/24 = no conflict
- Tailor subnet addressing scheme for other connected networks
- Descriptive subnet names = easier to maintain

Multiple VPCs vs. Multiple Projects?

- What needs to be separated?
- VPCs separate **resources**
  - Single VPC = shared firewall rules, routing, private communications
- Projects separate **member access** to resources
  - IAM roles
  - Most Compute/Network IAM roles are project wide, across all VPC's in project

Shared VPCs

- Additional connections go in host project
  - Multi-NIC instances
  - VLAN attachments
  - Cloud VPN gateways

High availability and redundancy

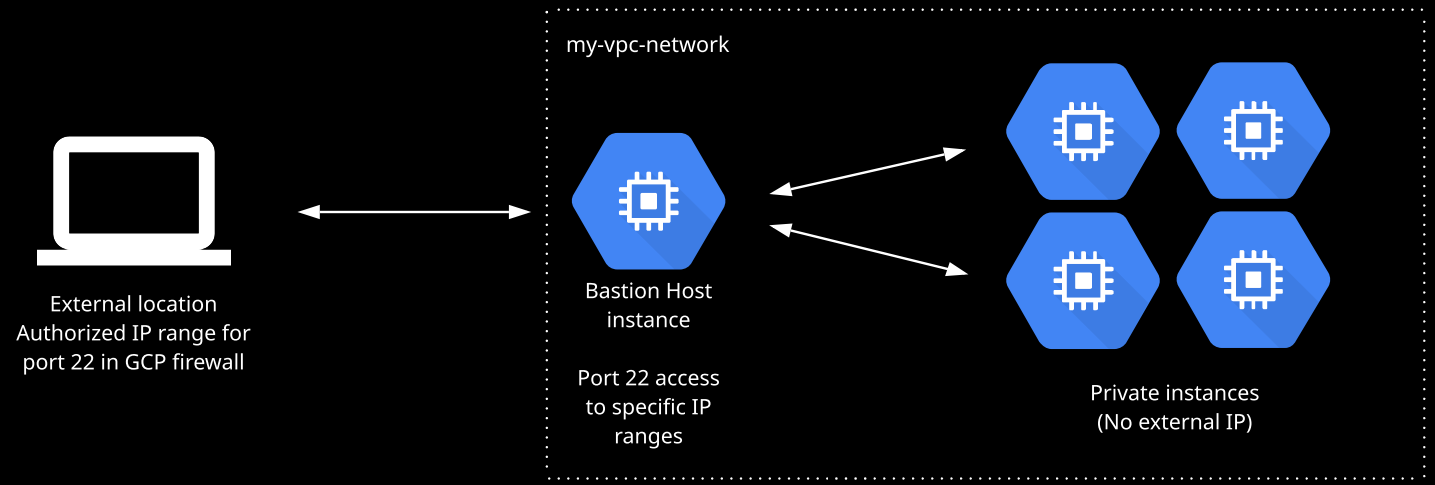
- Plan for failure in mind
  - If a zone or region goes down, can fail over to another one
- Deploy resources across zones and/or regions when possible
- Instance groups, interconnects/VPN tunnels, and more
- What can be deployed across **zones** vs. **regions**?

Multi-zone, regional services

- Individual Managed Instance Groups
- GKE cluster
- Pair with global load balancer for multi-region availability

Access private resources with a Bastion Host

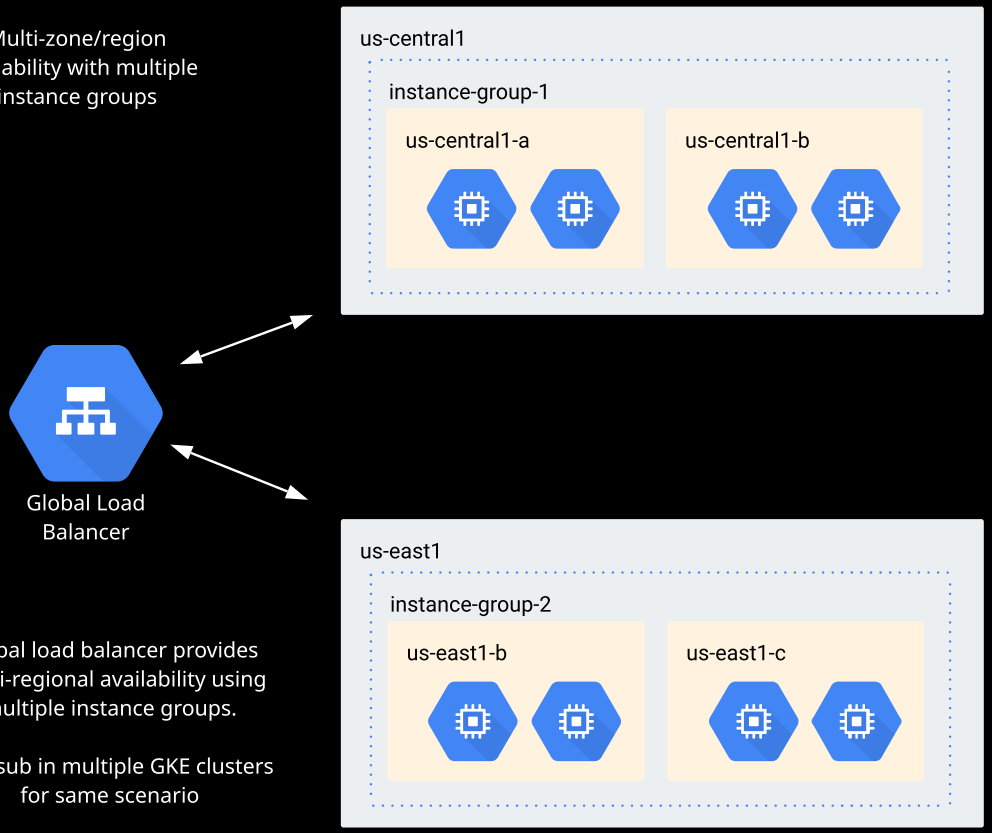
- How do we access resources without an external IP address?
  - VPN/Interconnect - limited to specific locations, full access from those locations
  - Bastion Host - secured 'jump off point' from public to internal resource
    - External facing point of entry to connect to private network instances
    - Hardened instance with limited access scope



Network Design and Monitoring

Best Practices for Network Design

Multi-zone/region availability with multiple instance groups



Single instance group provides multi-zone availability

Global load balancer provides multi-regional availability using multiple instance groups.

Can sub in multiple GKE clusters for same scenario

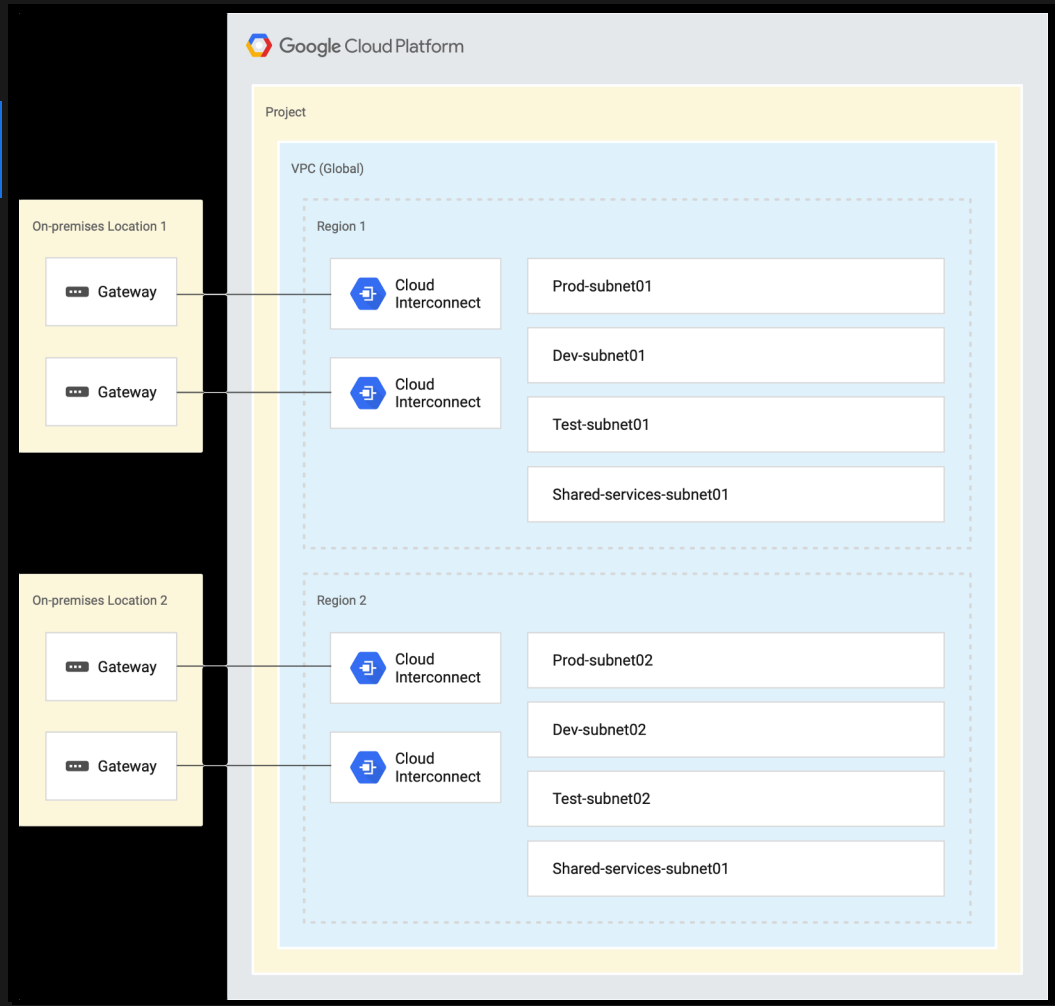
Back Next

Back to Main

Linux Academy

Network Design and Monitoring

Best Practices for Network Design



Interconnect/Cloud VPN High Availability

- Cloud VPN Gateway/VLAN Attachments are regional resource.
- Multiple connections, multiple regions
- If one gateway/connection or region goes down, connectivity is not affected

Back Next

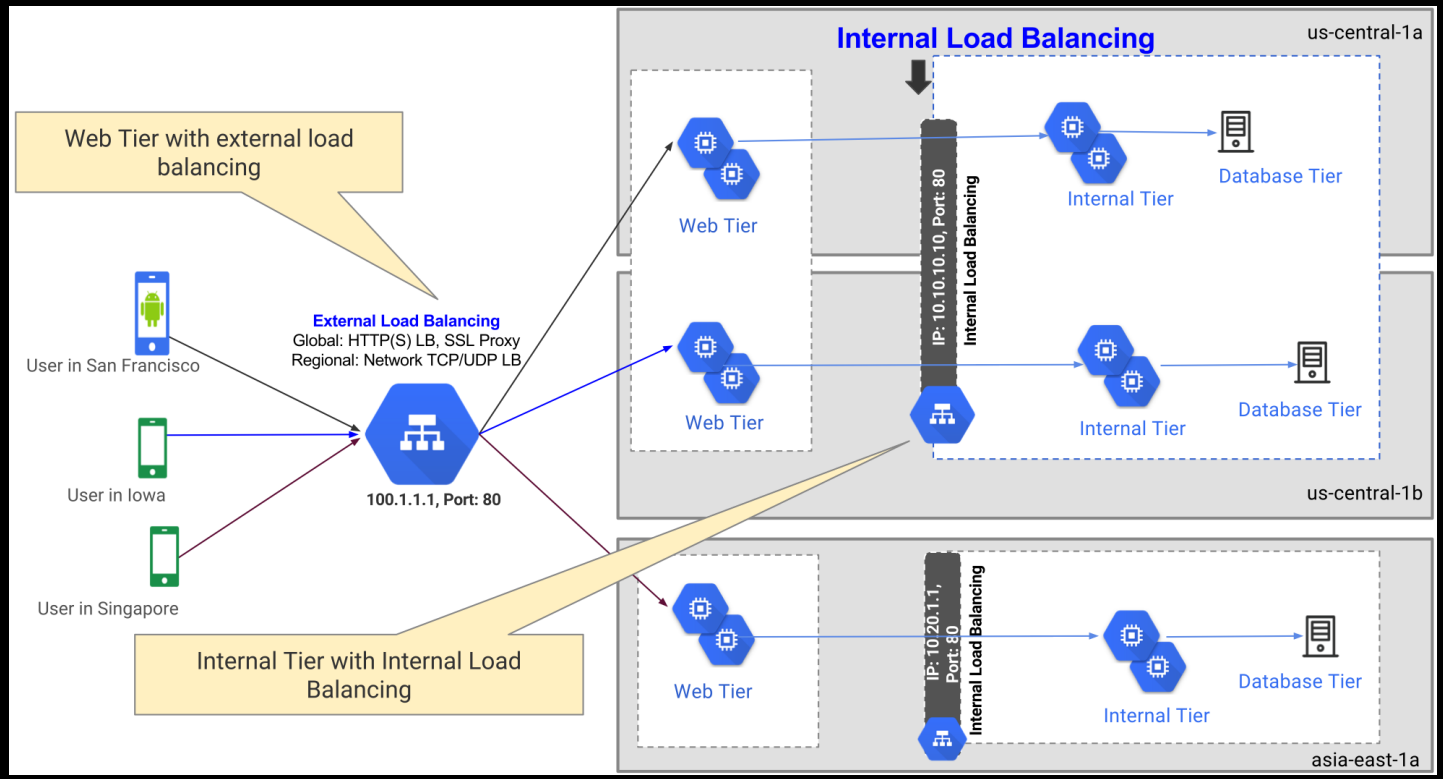
Back to Main

Linux Academy

Network Design and Monitoring

Best Practices for Network Design

Multi-tier application with multiple load balancers, public and private resources

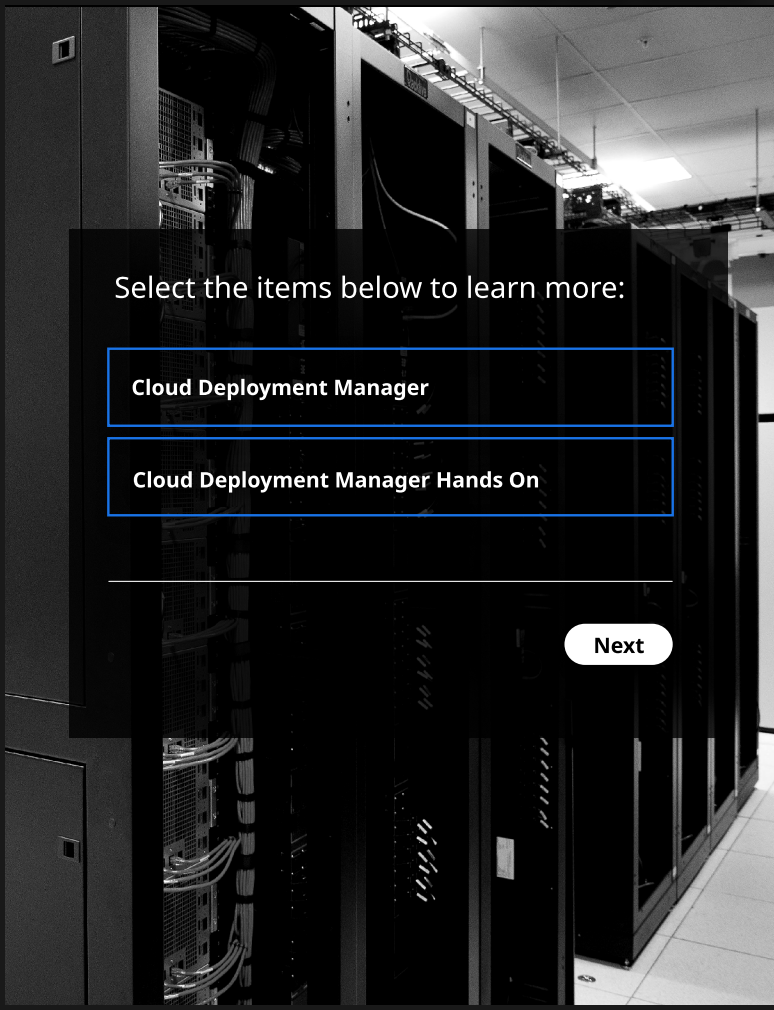


Back

Back to Main

Network Design and Monitoring

Deploying Networks



Select the items below to learn more:

Cloud Deployment Manager

Cloud Deployment Manager Hands On

Next

Network Design and Monitoring

Cloud Deployment Manager

What is Cloud Deployment Manager?

- Infrastructure deployment service
  - Also known as **Infrastructure As Code**
  - Similar to Terraform, Ansible, etc
- Automates creation/management of GCP resources
- Create and manage resources with configuration files and templates

Why is it important?

- As infrastructure grows in size and complexity, so does the chance of human error
- Standardized and repeatable
  - Create resources over and over with repeatable results
  - Highly structured templates and configuration
  - Document infrastructure in easy to understand format
- Used by GCP Marketplace to create easy, one-click deployments

How it works

- Deploy with command line only
- Calls on API resources
- Configuration file – YAML format
  - Lists each resource to create and its properties
  - Contains resources section followed by list of resources
  - Resource components
    - **Name** – user-defined string to identify (my-deployment-project)
    - **Type** – type of resource to deploy (compute.v1.instance, compute.v1.disk)
    - **Properties** – resource parameters (zone: us-central1, boot: true)

Next

Back to Main

Example configuration - single instance

```
resources:
- type: compute.v1.instance
  name: my-instance
  properties:
    zone: us-central1-f
    machineType: https://www.googleapis.com/compute/v1/projects/my-project-id/zones/us-central1-f/machineTypes/f1-micro
    disks:
    - deviceName: boot
      type: PERSISTENT
      boot: true
      autoDelete: true
      initializeParams:
        sourceImage: https://www.googleapis.com/compute/v1/projects/debian-cloud/global/images/family/debian
    networkInterfaces:
    - network: https://www.googleapis.com/compute/v1/projects/my-project-id/global/networks/default
      accessConfigs:
      - name: External NAT
        type: ONE_TO_ONE_NAT
```

Templates

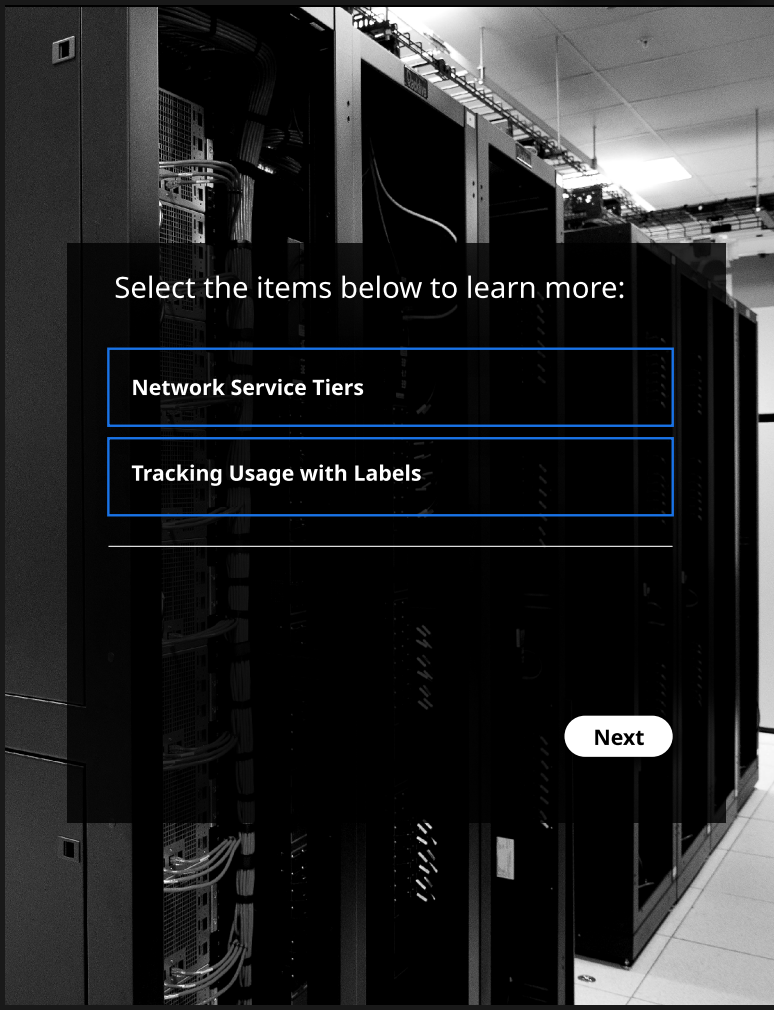
- Configuration file can link to templates
- Separate configurations into modular components
  - Update and re-use
- Python or Jinja2 format
- Advantages:
  - Easier to manage and maintain
  - Reusable
  - Keep consistent definitions in one place

Exam perspective

- Know when to use deployment manager for resource deployment
  - Repeatable infrastructure as code process for automated deployments
- Network configurations
  - Auto mode vs. custom mode VPC
    - Custom mode = declaring subnet is required, optional in auto mode

What are we doing?

- Review configuration file
  - View network portion of configuration
- Create a deployment from single configuration file
  - `gcloud deployment-manager deployments create (deployment_name) --config (config_file.yaml)`
- Delete deployment (web console or command line)
  - `gcloud deployment-manager deployments delete (deployment_name)`
- View configuration for complex multi-VPC deployment, with dependencies
- View and deploy template for multi VPC/instance deployment



Select the items below to learn more:

Network Service Tiers

Tracking Usage with Labels

Next

What are Network Service Tiers?

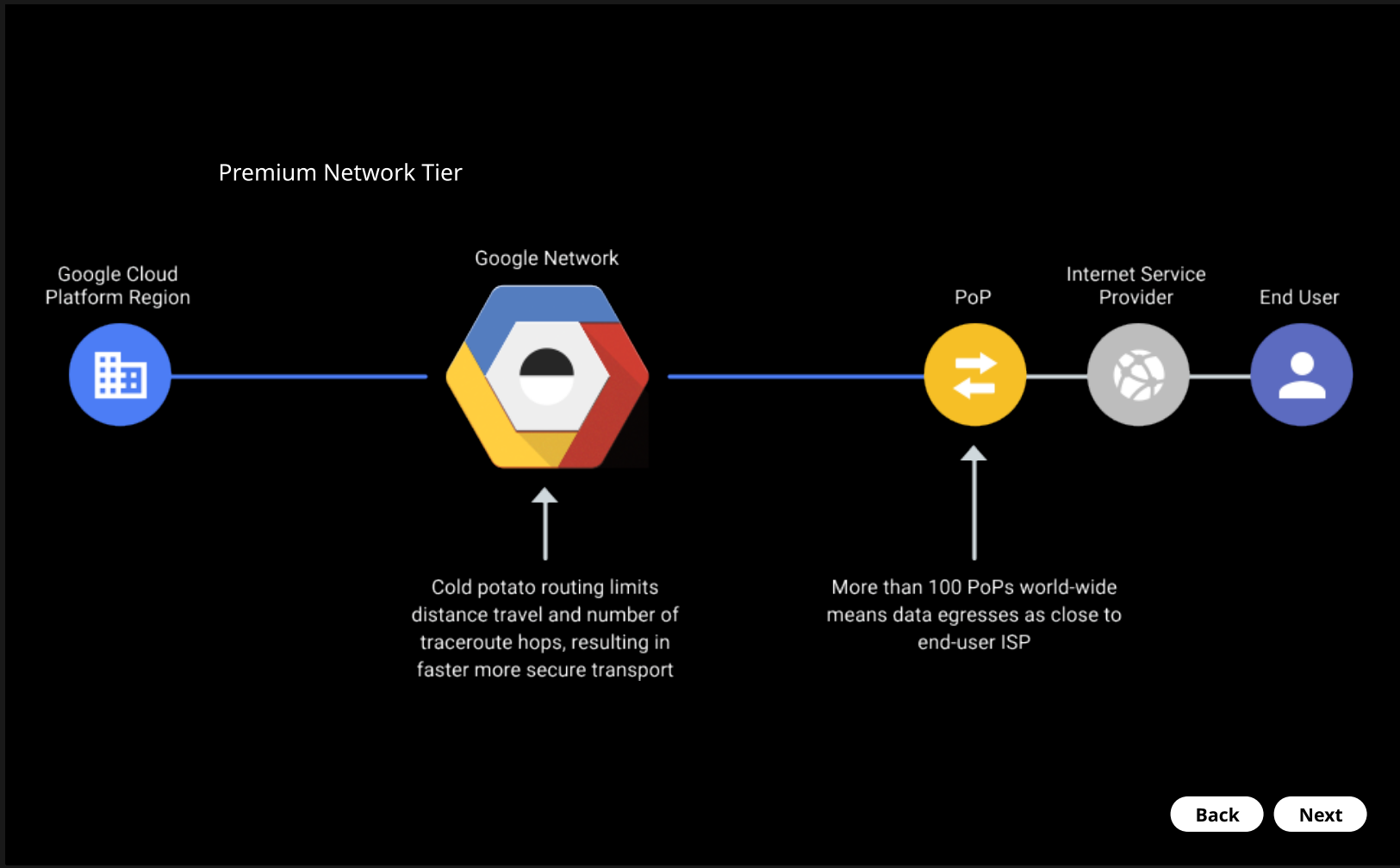
- Determines path and performance of network communications with GCP resources
- You are charged on amount of **egress** (outgoing) traffic from VPC resources. **Ingress** (incoming) traffic is always free.
- Choose between optimal network performance (**Premium tier)** or cost savings (**Standard tier**)
- Premium tier has been the only standard until recently, when the lower price Standard tier was introduced

What is Premium/Standard Tier?

- **Premium tier** = performance unique to GCP
  - Traffic kept in Google's network as much as possible
  - Better performance than any other cloud vendor
- **Standard tier** = performance equivalent to other cloud providers
  - More hops over general Internet = longer path to VPC resources
  - Lower egress costs
  - Ideal for regional-hosted services

Network Design and Monitoring

Network Service Tiers



Back

Next

Back to Main

Linux Academy

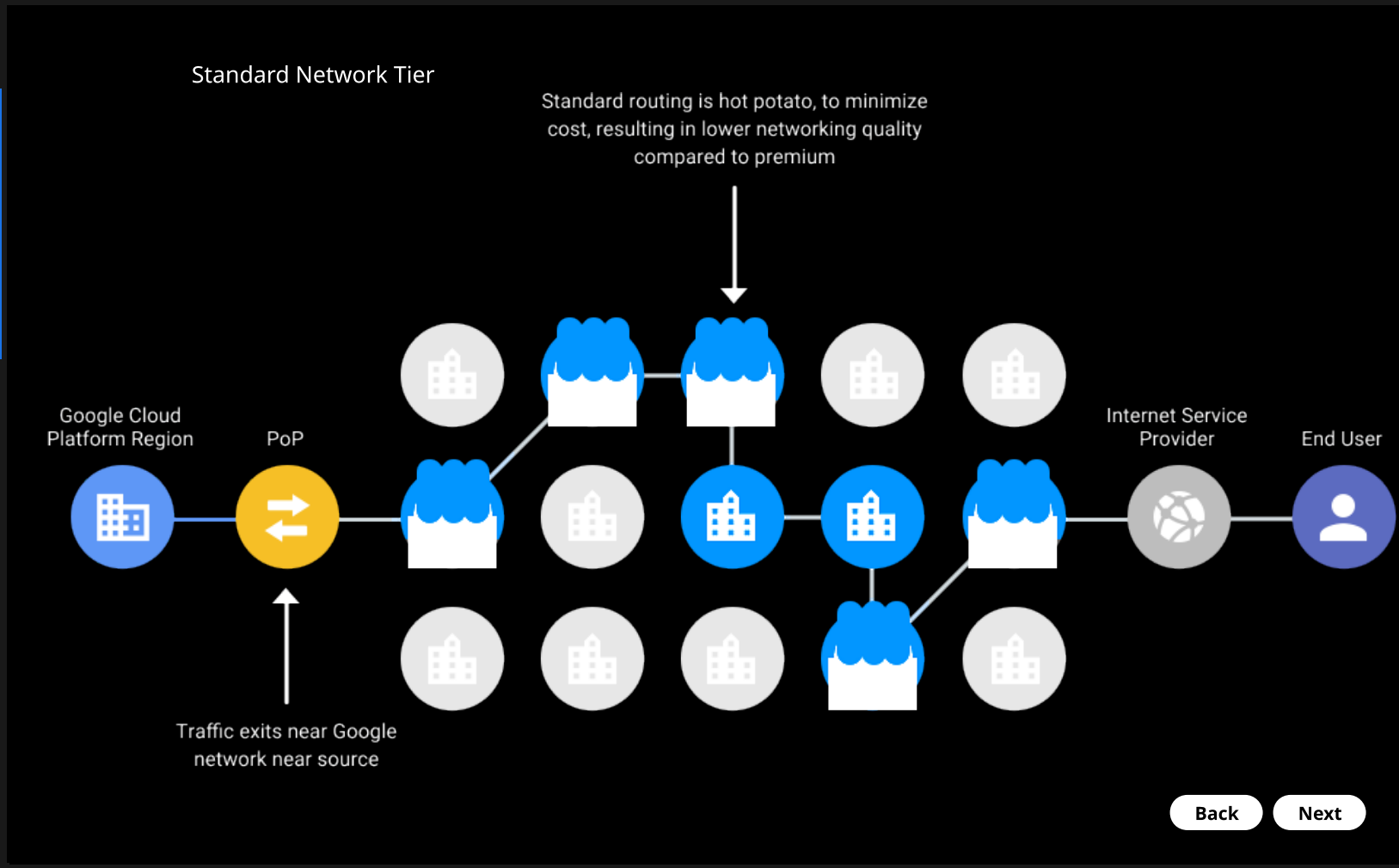
Network Design and Monitoring

Network Service Tiers

Course Navigation

Network Design and Monitoring
Section 5

- **Designing Your Network**
  - **Best Practices for Network Design**
- **Deploying Networks**
  - **Cloud Deployment Manager**
  - **Cloud Deployment Manager Hands On**
- **Managing Costs**
  - **Network Service Tiers**
- **Monitoring and Logging**
  - VPC Flow Logs
  - VPC Flow Logs Hands On
  - Firewall Logs
  - Firewall Logs Hands On
- **Cloud Storage**
  - Optimize Cloud Storage Performance



Back
Next

Back to Main

Linux Academy

Enabling Network Service Tiers

- Enable per-VM or instance template
- Can also set as project-wide default

Restrictions

- Most Multi-regional services require premium tier
  - Global load balancers with multi-region backends
  - Cloud CDN
- Other services also require premium tier
  - Cloud VPN
  - Cloud NAT

When to choose Premium or Standard Tier?

- **Premium:** application requires high performance, high availability, low latency
- **Standard:** cost sensitive, do not require high performance
  - Working with VM's in single region

Network Design and Monitoring

Network Service Tiers

All about choice

Large enterprise

"Our workloads and services need to be up and running across the globe with low latency and high performance."

Premium Tier

Cloud native service provider

" Downtime means we lose customers and money."

Premium Tier

Give me options customer

"For my mission critical workloads, I want high levels of availability and performance across the globe. For the other workloads, I care more about optimizing for cost."

Premium Tier

Standard Tier

Cost-sensitive customer

"My services are deployed only in a single cloud region. I am on a tight budget and willing to trade-off some of the performance and availability for lowered cost."

Standard Tier

Back Next

Back to Main

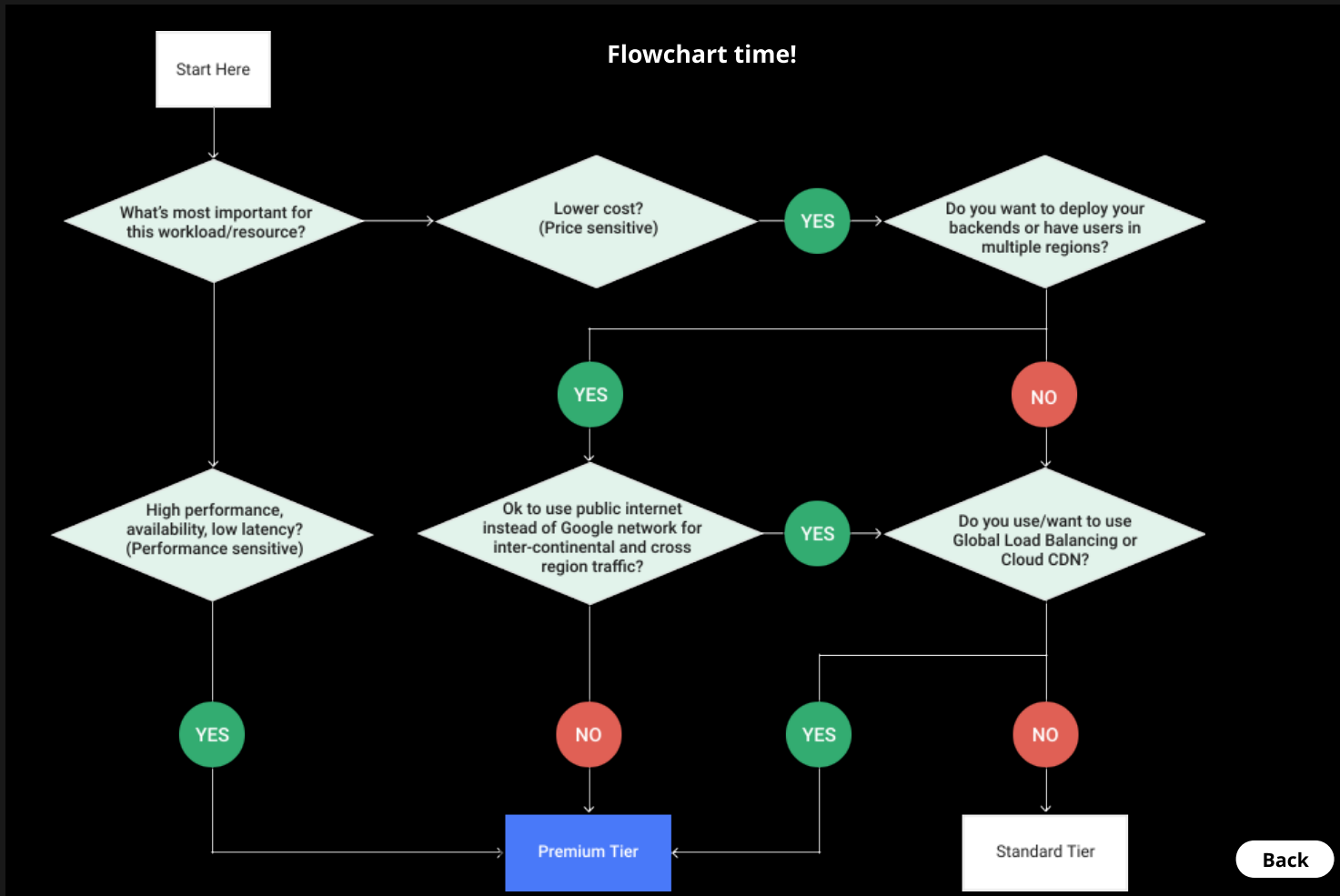
Network Design and Monitoring

Network Service Tiers

Course Navigation

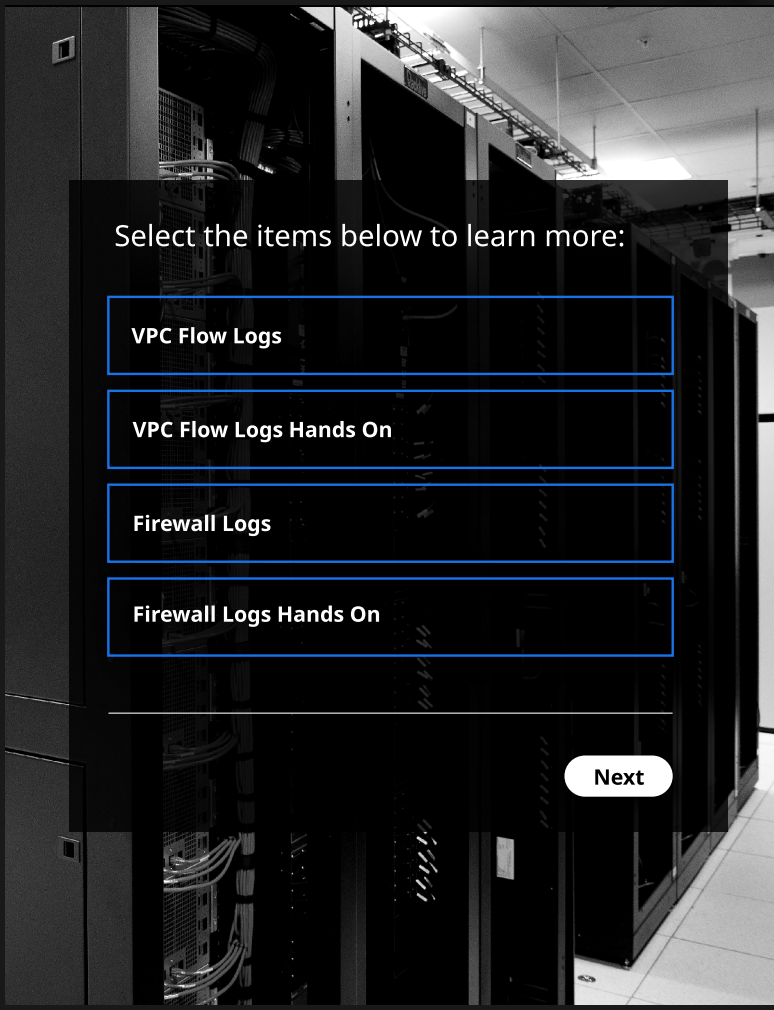
Network Design and Monitoring
Section 5

- **Designing Your Network**
- Best Practices for Network Design
- **Deploying Networks**
- Cloud Deployment Manager
- Cloud Deployment Manager Hands On
- **Managing Costs**
- Network Service Tiers
- **Monitoring and Logging**
- VPC Flow Logs
- VPC Flow Logs Hands On
- Firewall Logs
- Firewall Logs Hands On
- **Cloud Storage**
- Optimize Cloud Storage Performance



Back to Main

Linux Academy



Select the items below to learn more:

- VPC Flow Logs
- VPC Flow Logs Hands On
- Firewall Logs
- Firewall Logs Hands On

Next

Network Design and Monitoring

VPC Flow Logs

What are VPC Flow Logs?

- Record sample of network flows sent/received by VPC resources
  - Near real-time recording
- Used for monitoring, forensics, security analysis
- View in **Stackdriver Logging** - can export to other sources (e.g. BigQuery)
- No network performance penalty
- Enabled at VPC subnet level

Use Cases

- **Network Monitoring** - Understand traffic growth for capacity forecasting
  - Plan network egress costs
- **Forensics** - who are your instances talking to?
- **Real-time security analysis**
  - Integrate with Pub/Sub for streaming transport
  - Integrate with other security products

Considerations/Troubleshooting

- Generate large amount of potentially chargeable log files
  - Can adjust aggregation/sampling amount to balance costs
- Does not capture 100% of traffic
  - Samples approximately 1 out of 10 packets. This cannot be adjusted.
  - Packets in low volume flows might be missed
  - TCP/UDP only
- **Shared VPC** - all VPC flow logs in host project

What are we doing?

- Create custom VPC with single subnet
- Enable VPC Flow Logs in custom subnet
- Generate flow logs for web server in subnet
- View in Stackdriver Logging
- Export flow logs to BigQuery - generate more logs
- View logs and run queries in BigQuery

What are Firewall Logs?

- Logs of firewall rule effects
- Useful for auditing, verifying, and analyzing effect of rules
- Applied per firewall rule, across entire VPC
- Creates **connection record** each time rule allows/denies traffic
- Can be exported to BigQuery or Pub/Sub for analysis

Considerations and Restrictions

- Logs every firewall connection attempt - best effort basis
  - Larger machine types will log more firewall connections per interval
- TCP/UDP protocols only
- Default 'deny all' ingress and 'allow all' egress rules are NOT logged

How to view denied ingress/allowed egress connections?

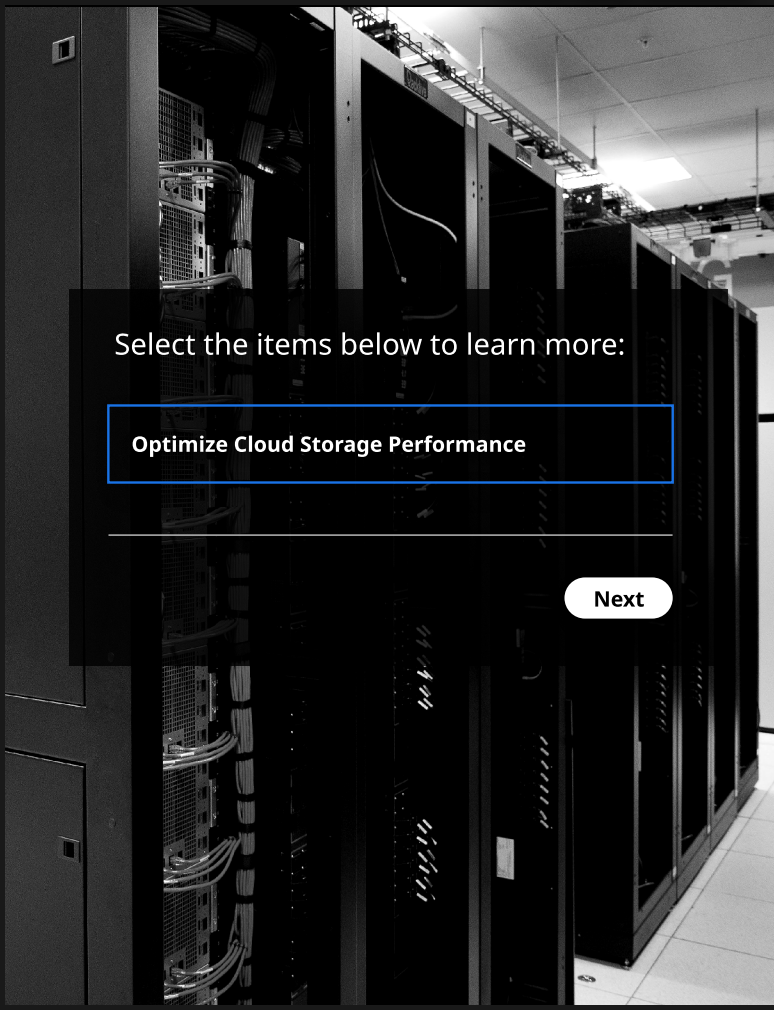
- Create explicit firewall rule for denied/allowed traffic you want to view
- Example: View all SSH attempts from outside of allowed location
  - Create rule to deny all TCP:22 access from all locations
  - Assign second-lowest priority figure of 65534
  - Assign higher priority 'ssh-allow' rule for allowed location in source filter
  - Result: all allowed SSH connections will be logged, and all denied connections will also be logged
- WARNING: This may generate large amounts of log files, use as temporary troubleshooting tool

What are we doing?

- Create custom VPC with single subnet
- Create web-server instance and http-allow firewall rule
- Generate firewall logs for web server by accessing website
- View in Stackdriver Logging - requires advanced filter
- Create second 'deny-all' rule, denying ALL TCP/UDP traffic
- Export deny-all logs to BigQuery
- View logs and run queries in BigQuery

Network Design and Monitoring

Third-Party Solutions



Select the items below to learn more:

Optimize Cloud Storage Performance

Next

Network Design and Monitoring

Optimize Cloud Storage Performance

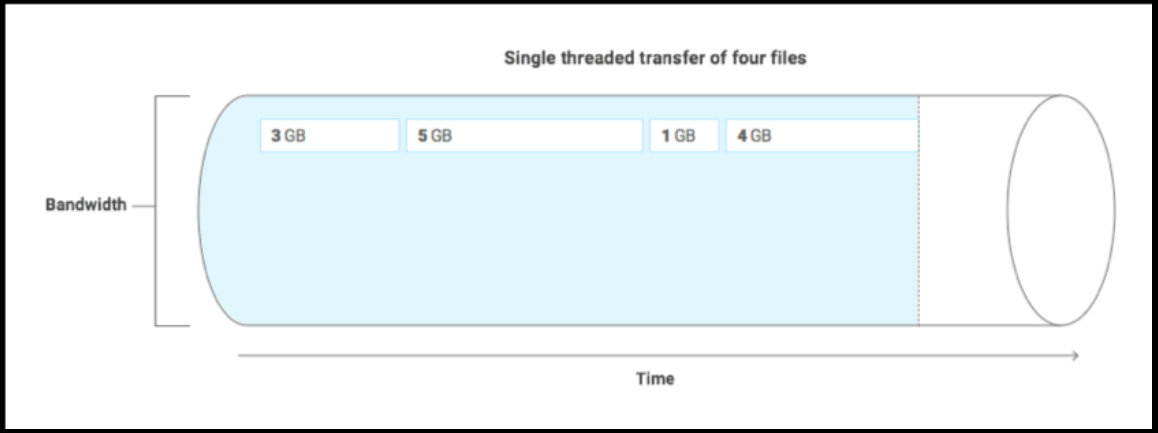
Why is this important?

- Not related to a VPC, however transferring files to/from GCS may be part of network engineer role
- Need to know best practices and diagnostic tools to optimize transfer speeds
- Focus on **gsutil** commands

Best Practice: Transferring many small files

- **Problem:** By default, gsutil transfers one file at a time, each transfer has overhead which can affect performance
  - The smaller the file, the more overhead affects total transfer time
- **Solution:** Transfer multiple files at a time with multi-threading
- Use **-m** in your transfer command
- `gsutil -m cp -r /my-local-directory gs://my-bucket`

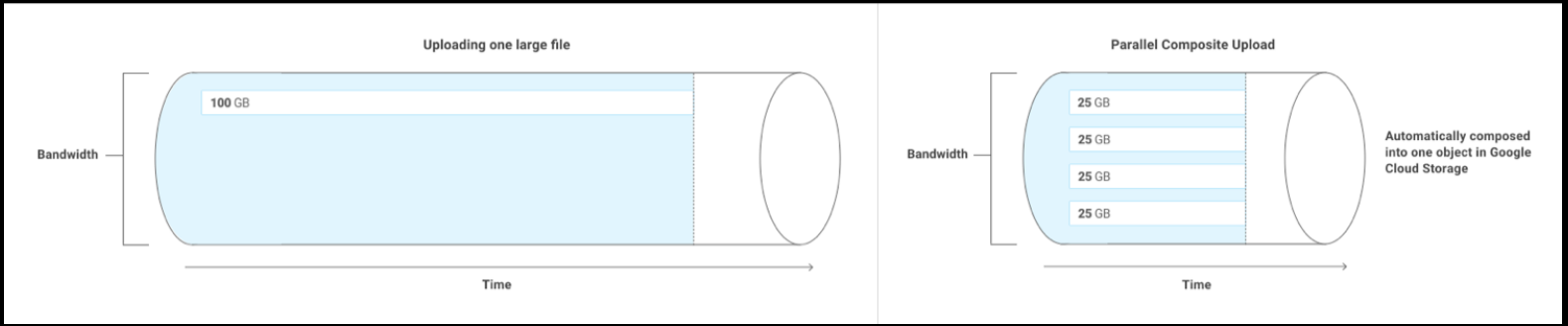
Multi-threaded transfers



Back
Next
Back to Main

Best Practice: "Slice" large files into smaller pieces for transfer

- **Problem:** Large files take a long time to transfer, even with sufficient bandwidth
- **Solution:** Parallel Composite Transfer
- 'Slice' large files into smaller pieces (called **composite objects**), then transfer multiple pieces over multiple threads
  - GCP will re-assemble the pieces when transfer is complete
- `gsutil -o GSUtil:parallel_composite_upload_threshold=150M cp ./my-large-file gs://my-bucket`
- Note: do not use with Nearline/Coldline storage class - will result in 'modification' fee



Diagnosing and Troubleshooting Transfer Performance

- **Problem:** need to measure performance for standard transfers and optimized transfers
- **Solution:** use 'perfdiag' option in gsutil - **performance diagnostics**
- Diagnostic tool only, does not facilitate optimized transfers
- Test transferring, reading, and deleting multiple test files of different sizes to your GCS bucket
  - Can adjust testing parameters for file sizes, multi-threading, number of tests, etc to view different results
- `gsutil perfdiag gs://my-bucket`

Back