GCP Professional Cloud Architect Crash Course

Get Fully Prepared to Crush the Exam

Victor Dantas
Author, Certified Cloud Architect

Pearson

Segment 1: Intro and overview of GCP exam

Objectives

- Introduction to the course
- Overview of GCP exam and what to expect

Day 1 Schedule

- **Segment 1: Intro and overview of GCP exam, what to expect**
- **Segment 2: Designing network, storage, and compute**
  - Integration with on-premises
  - Multicloud environments
  - Designing VPC networks
  - Choosing appropriate storage types
  - Choosing data processing technologies
  - Choosing compute resources
  - Demo: designing network, storage, and compute
  - Q&A (10min)
  - Break (10min)
- **Segment 3: Configuring network, storage, and compute**
  - Securing networks
  - Setting up hybrid and multicloud networking
  - Provisioning data storage
  - Configuring data retention and lifecycle management
  - Configuring Cloud SQL for high availability
  - Provisioning compute resources
  - Configuring Kubernetes
  - Demo: configuring network, storage, and compute
  - Q&A (10min)
  - Break (10min)

Day 1 Schedule

- **Segment 4: Designing for observability, security and compliance**
  - Identity and Access Management (IAM)
  - Separation of duties
  - Resource hierarchy
  - Service Control Policies
  - Securing data at rest and in transit
  - Secrets and certificate management
  - Compliance considerations
  - Cloud Monitoring and Logging
  - Application performance monitoring
  - Demo: designing for observability, security, and compliance
- **Q&A and Day 1 wrap-up**

Audience Poll Question

What is your experience level with GCP? (Single response)

- Basic knowledge of GCP but experienced with public cloud (AWS/Azure)
- Basic knowledge of GCP but experienced with private cloud / IT
- Foundational knowledge of GCP (< 6 months), no prior experience
- 6-months to 1-year working with GCP
- 1-year to 3-years working with GCP
- More than 3 years working with GCP

Exam Overview

- **Length**: 2 hours
- **Registration fee**: $200 (plus tax where applicable)
- **Languages**: English, Japanese
- **Exam format**: Multiple choice and multiple select, taken remotely or in person at a test center.
- **Prerequisites**: None
- **Recommended experience**: 3+ years of industry experience including 1+ years designing and managing solutions using Google Cloud
- **Certification Validity**: Two years from the date of certification. Candidates must recertify in order to maintain their certification status.

Exam Overview

A Google Cloud Certified Professional Cloud Architect:

- **Designs**, **develops**, and **manages** solutions that drive business objectives
- Is proficient in all aspects of enterprise **cloud strategy** and **architectural best practices**
- Is experienced in **software development methodologies**
- Is experienced with solutions that include **distributed applications** which span **multicloud** or **hybrid** environments

Source: https://cloud.google.com/certification/guides/professional-cloud-architect

Exam Overview

Four Case Studies:

- **EHR Healthcare**
- **Helicopter Racing League**
- **TeramEarth**
- **Mountkirk Games**

Each Case Study:

- Company overview
- Solution concept
- Existing technical environment
- Business requirements
- Technical requirements
- Executive statement

https://cloud.google.com/certification/guides/professional-cloud-architect

Segment 2: Designing network, storage, and compute

Objectives

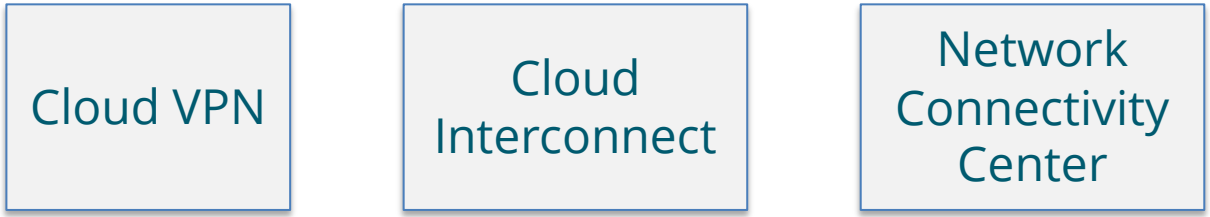
- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

Segment 2: Designing network, storage, and compute

Objectives

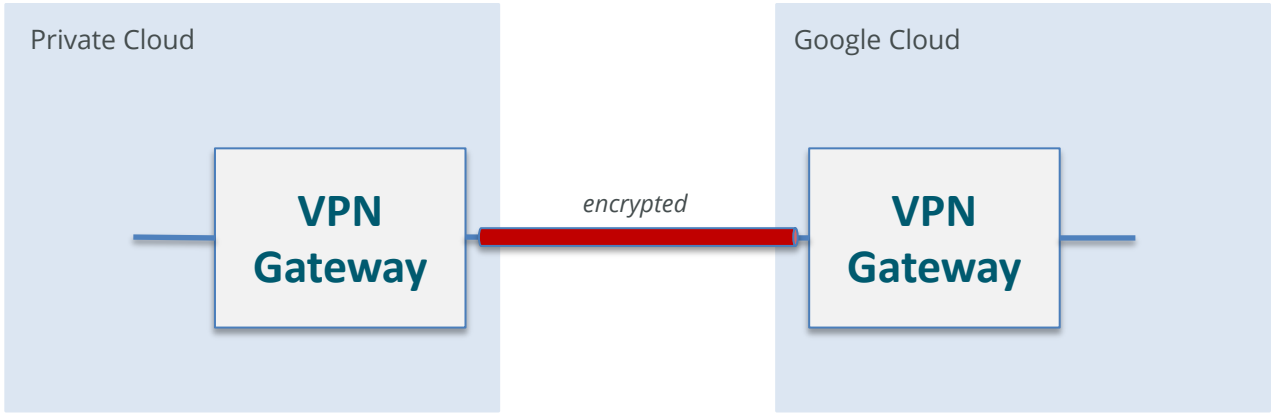
- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

Hybrid Networking Services



Hybrid Networking Services: Cloud VPN

- IPSec VPN tunnel
- Traffic travels over the public internet, encrypted by one VPN gateway, then decrypted by another VPN gateway



Integration with On-Premises: Cloud VPN

Two types of Cloud VPN:

- HA VPN: High availability VPN (99.99% SLA) solution
- Classic VPN: 99.9% availability SLA (being deprecated!)

Integration with On-Premises: Cloud VPN

Two types of Cloud VPN:

- HA VPN: High availability VPN (99.99% SLA) solution
- Classic VPN: 99.9% availability SLA (being deprecated!)

Recommended configuration

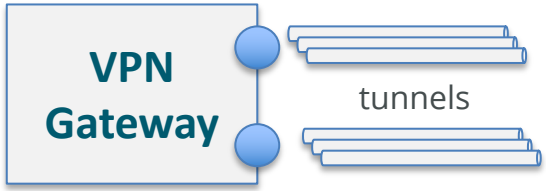
Two interfaces, two external IPV4 addresses

Integration with On-Premises: Cloud VPN

Two types of Cloud VPN:

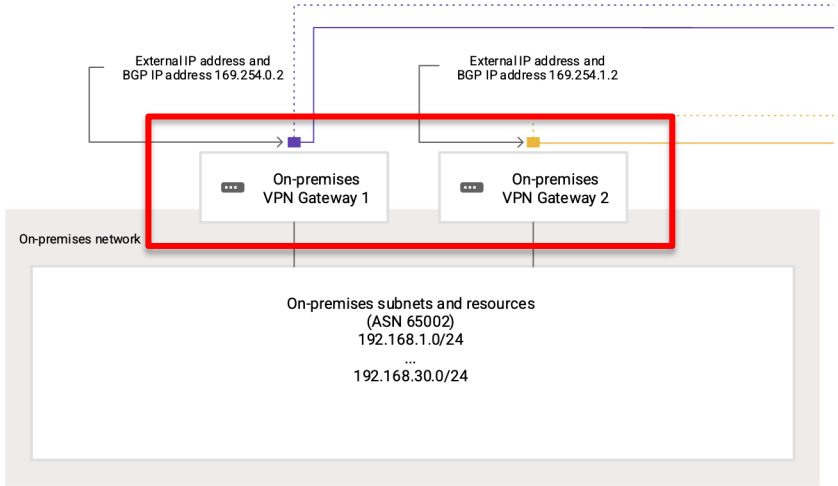
- HA VPN: High availability VPN (99.99% SLA) solution
- Classic VPN: 99.9% availability SLA (being deprecated!)

HA VPN: Each interface supports multiple tunnels



High Availability VPN Requirements

- 99.99% SLA is guaranteed on Google Cloud side only
- For end-to-end 99.99% availability:



VPN device configured with adequate redundancy

Configure two tunnels

Cloud Interconnect

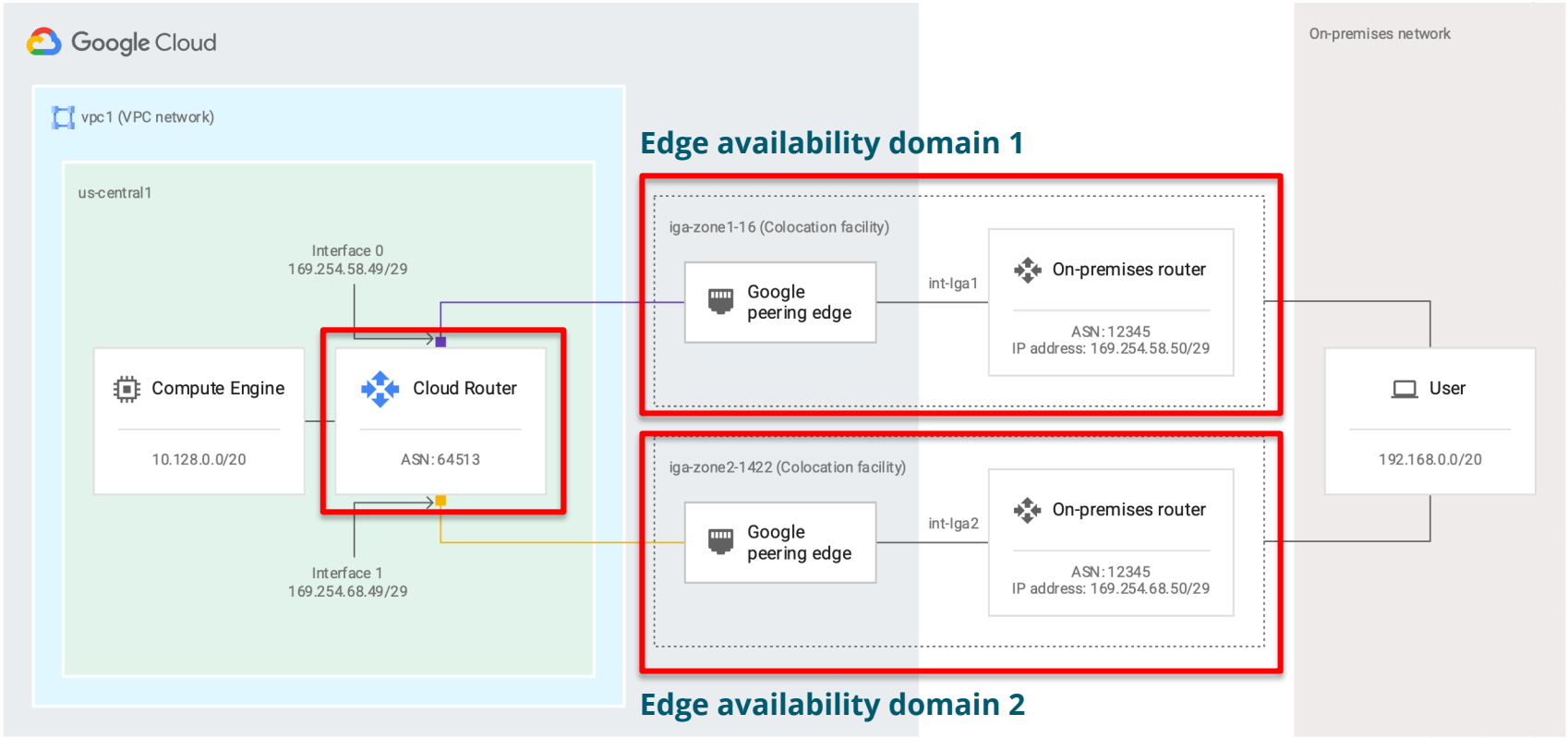
Dedicated Interconnect

- Direct connection to Google's network with end-to-end SLA
- Must be able to physically meet Google's network
- **10-Gbps** or **100-Gbps** circuits with flexible VLAN attachment capacities from 50 Mbps to 50 Gbps.
- Maximum of 8x10Gbps (**80Gbps** aggregate bandwidth) or 2x100Gbps (**200Gbps** aggregate bandwidth) circuits

Partner Interconnect

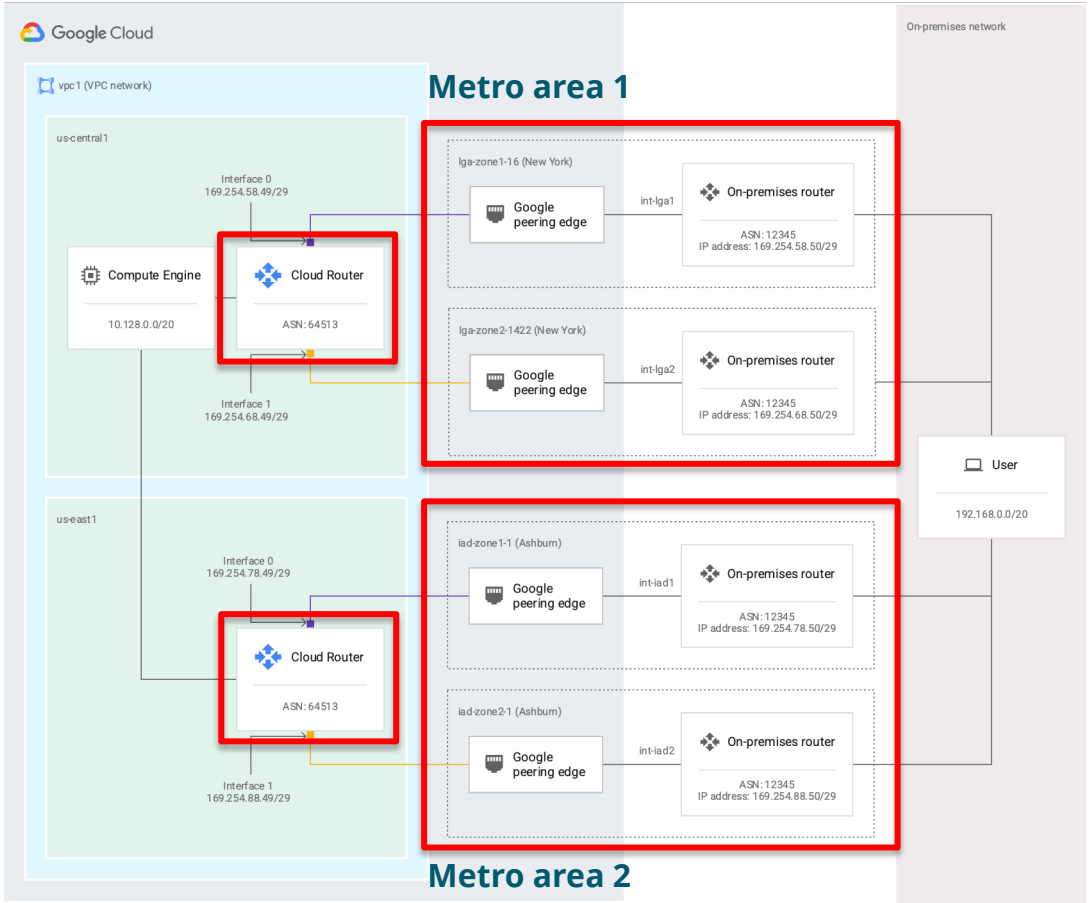
- Traffic passes through service provider's network, but **not** the public internet
- More points of connectivity through one of the supported service providers
- Flexible VLAN attachment capacities from **50Mbps** to **50Gbps**
- Google provides SLA for Google-Partner connection

Dedicated Interconnect Topology: 99.9% SLA



Pearson

Dedicated Interconnect Topology: 99.99% SLA



Pearson

Integration with On-Premises: Interconnect Options

Dedicated vs. Partner Interconnect: What to choose

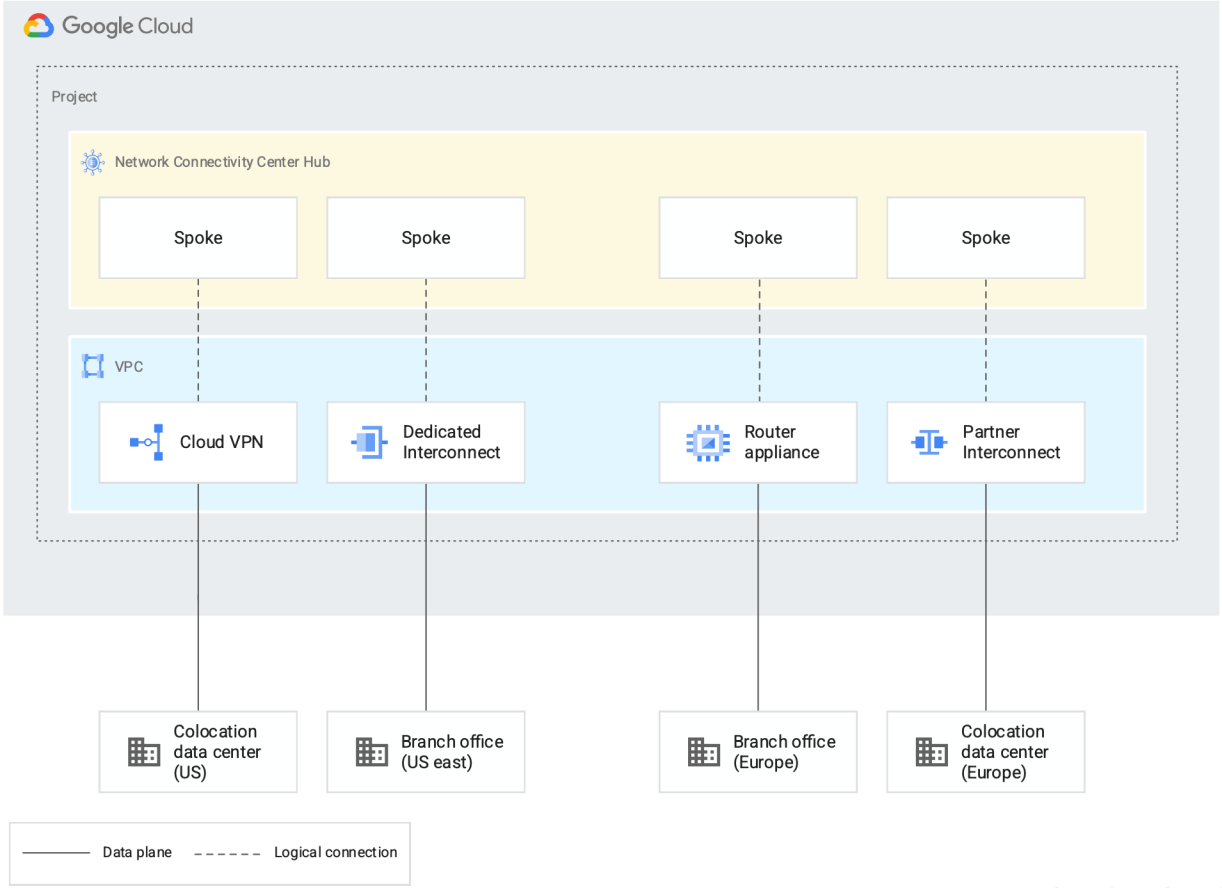
Dedicated Interconnect	Partner Interconnect
High bandwidth needs (10s of Gbps)	Bandwidth needs are in the 100s of Mbps or low Gbps
Can reach Google's network directly at a colocation facility	Not able to reach Google's network directly
Don't want traffic to pass through a service provider network	Don't want to setup and/or maintain routing equipment at colocation facility

Network Connectivity Center

- Supports connecting different enterprise sites by using Google's network as a wide area network (WAN).
- On-premises networks can consist of data centers and branch or remote offices.
- **Hub and spoke model**: on-premises networks (spokes) connect to Network Connectivity Center (hub)

Integration with On-Premises: Network Connectivity Center

Network Connectivity Center



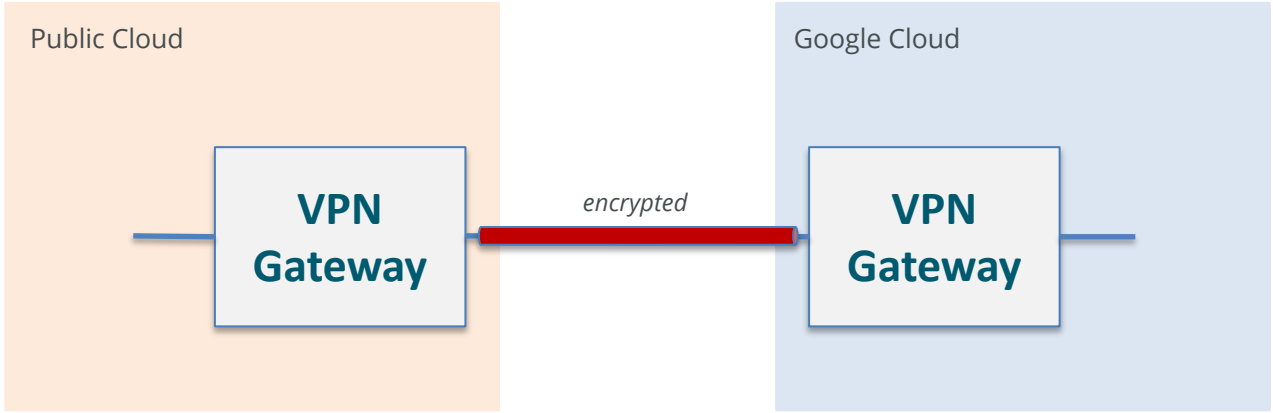
Segment 2: Designing network, storage, and compute

Objectives

- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

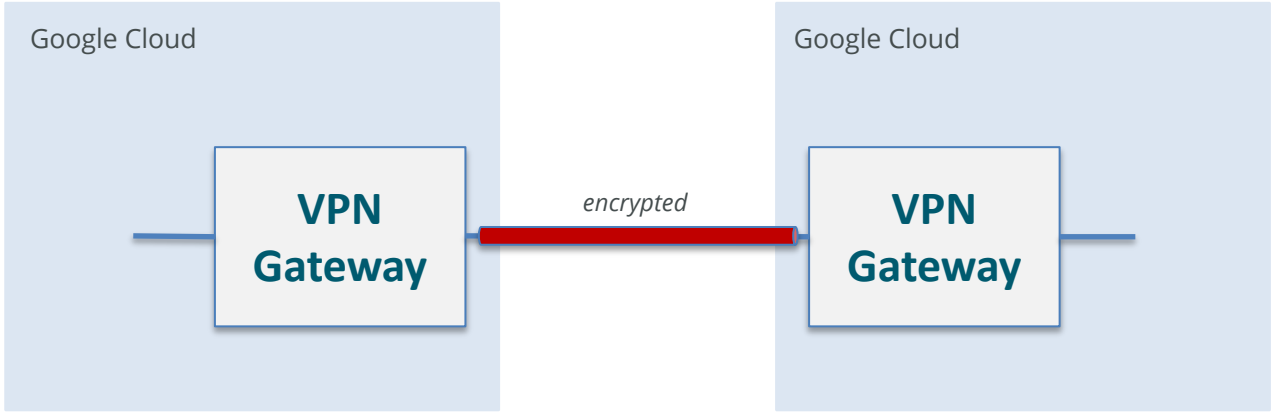
Multicloud Networking

- You can connect a Google Cloud VPC network to another cloud provider's network (AWS, Azure, etc.)



Multicloud Networking

- You can connect a Google Cloud VPC network to another cloud provider's network (AWS, Azure, etc.)
- You can also connect two Google Cloud VPC networks, whether they belong to the same organization or not

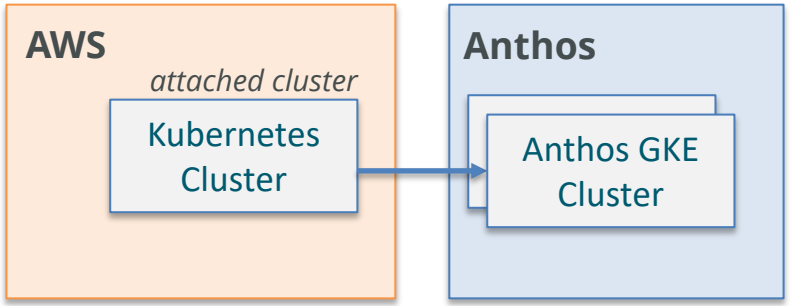


Multicloud Solutions

Multicloud Solutions: Anthos

Anthos enables you to manage **GKE clusters** and workloads running on **virtual machines** across environments.

Consistent managed Kubernetes experience with upgrades validated by Google.

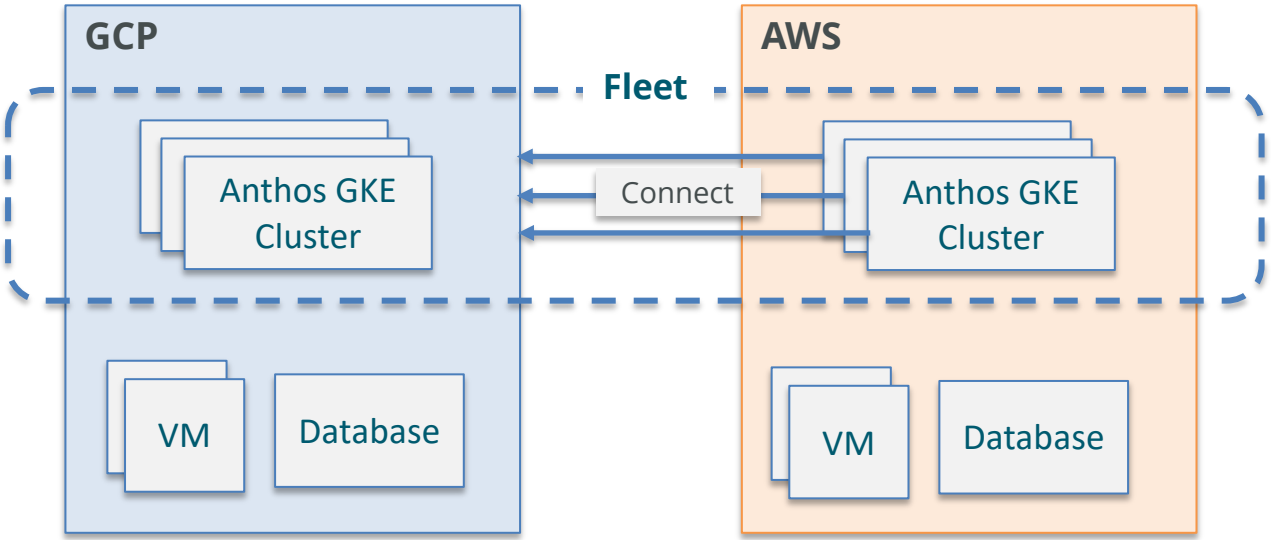


Anthos Fleets and Connect

- A **fleet** is a logical grouping of Kubernetes cluster and other resources that can be managed together
- When you register a cluster outside of GCP, Anthos uses a Kubernetes Deployment called the **Connect Agent**
- **Connect** establishes a long-lived, encrypted connection between the cluster's Kubernetes API server and Google Cloud

Anthos Fleets and Connect

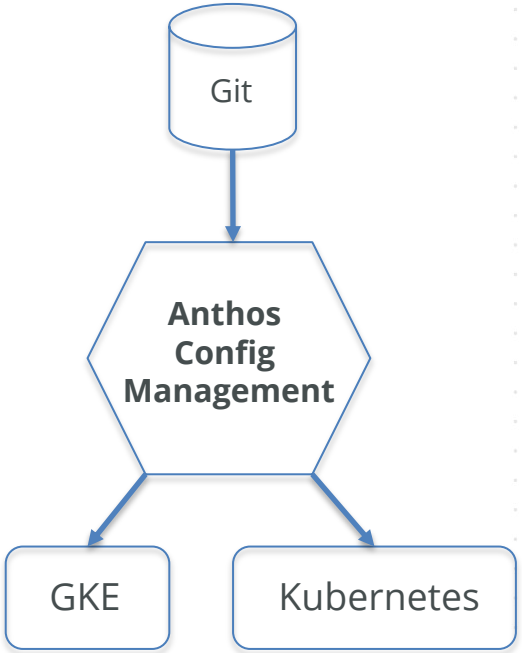
Example



Unified management (control plane) and user interface for all your clusters

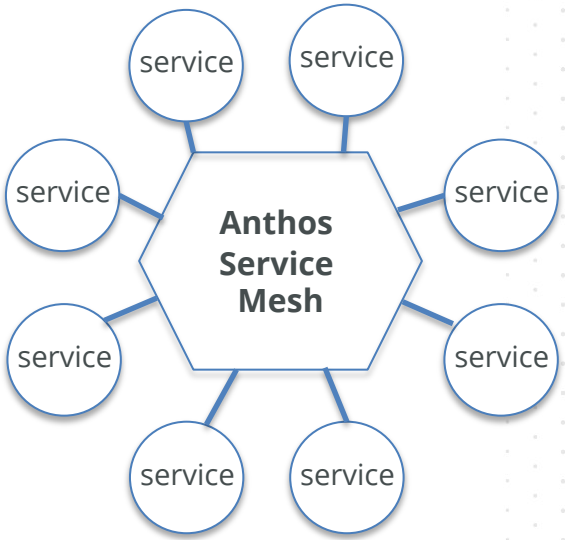
Anthos Config Management

- You create a common configuration across all your infrastructure
- Once you declare a new desired state, it continuously checks for changes that go against state
- Changes are rolled out to all clusters to reflect the desired state



Anthos Service Mesh

- Fully managed service mesh based on **Istio**
- Out-of-the-box telemetry with all traffic monitored through a proxy
- Enforce policies across VMs and containers
- Can be used for traffic management (canary, location-based routing)



Anthos Migrate to Containers

- Convert VM-based workloads into containers
- VM disk imported as a persistent volume
- Container packaged as a StatefulSet

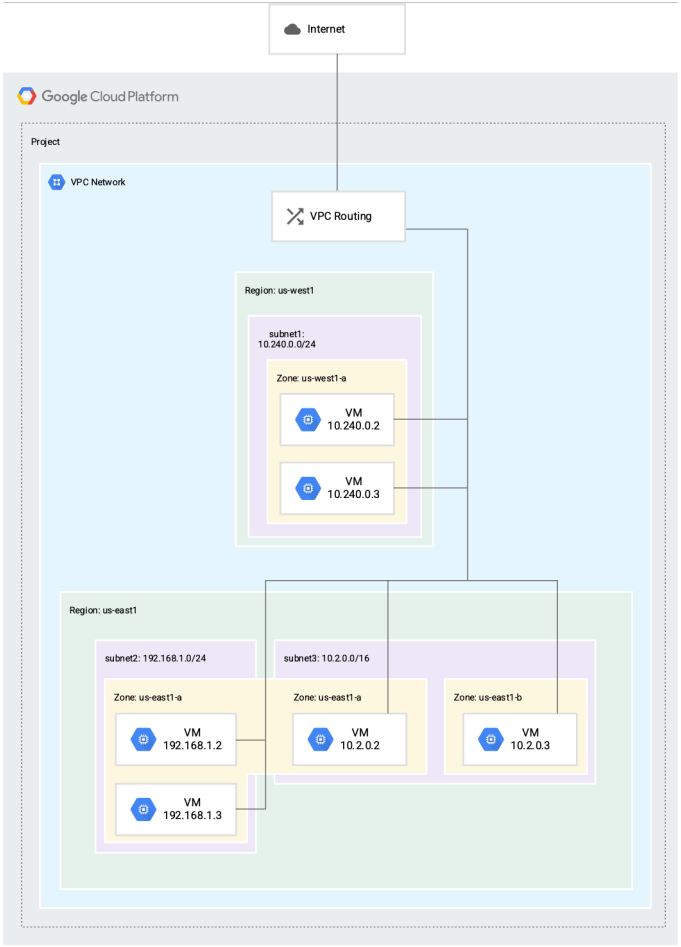
Segment 2: Designing network, storage, and compute

Objectives

- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

Cloud-native Networking: VPC

- Global resource, logically isolated
- Consisting of a list of regional subnetworks (subnets)

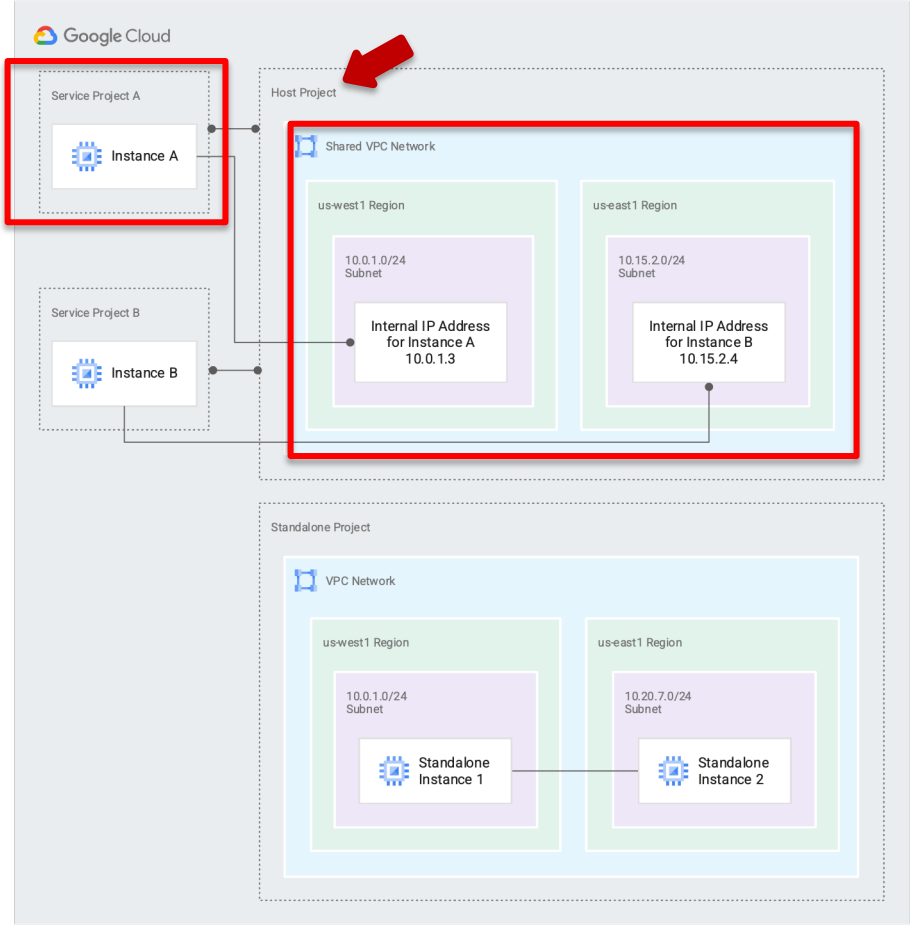


Cloud-native Networking: Shared VPC

- You can share a VPC network from one project (**host** project) to other projects (**service** projects)
- Benefits:
  - Separation of concerns
  - Enforce consistent security policies for multiple (service) projects
  - Help separate budgeting and internal cost allocation

Cloud-native Networking: Shared VPC

Shared VPC

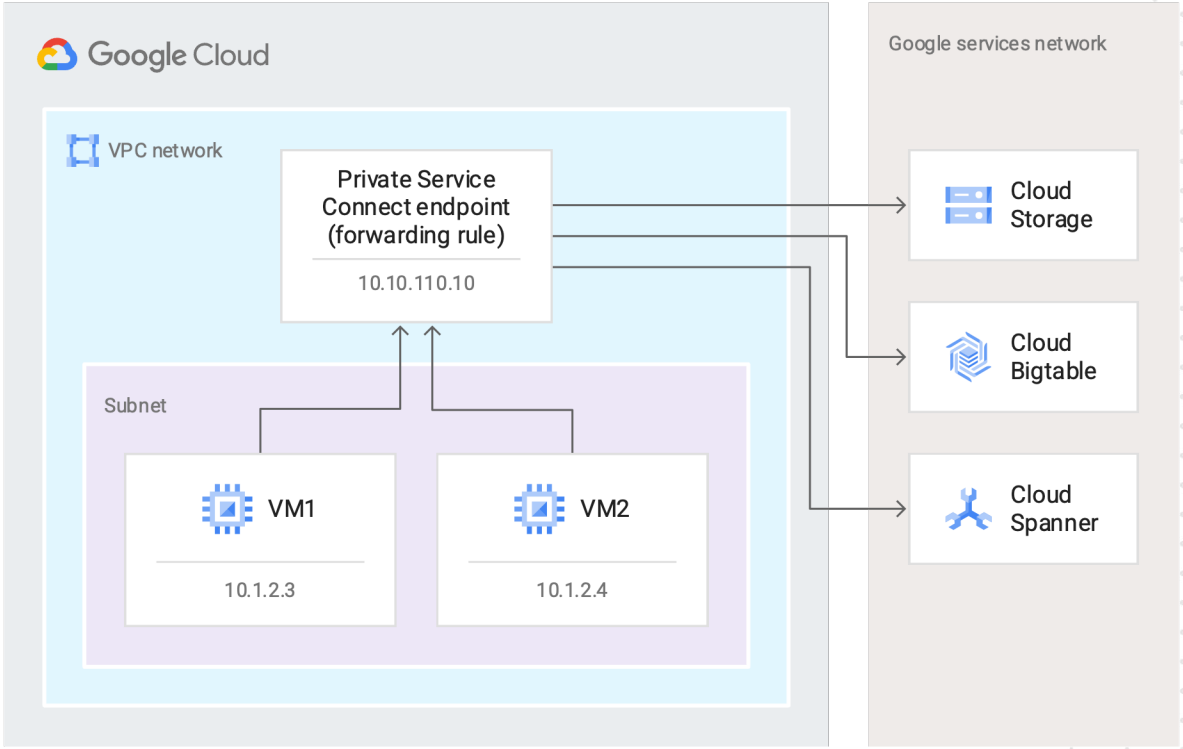


Cloud-native Networking: Private Service Connect

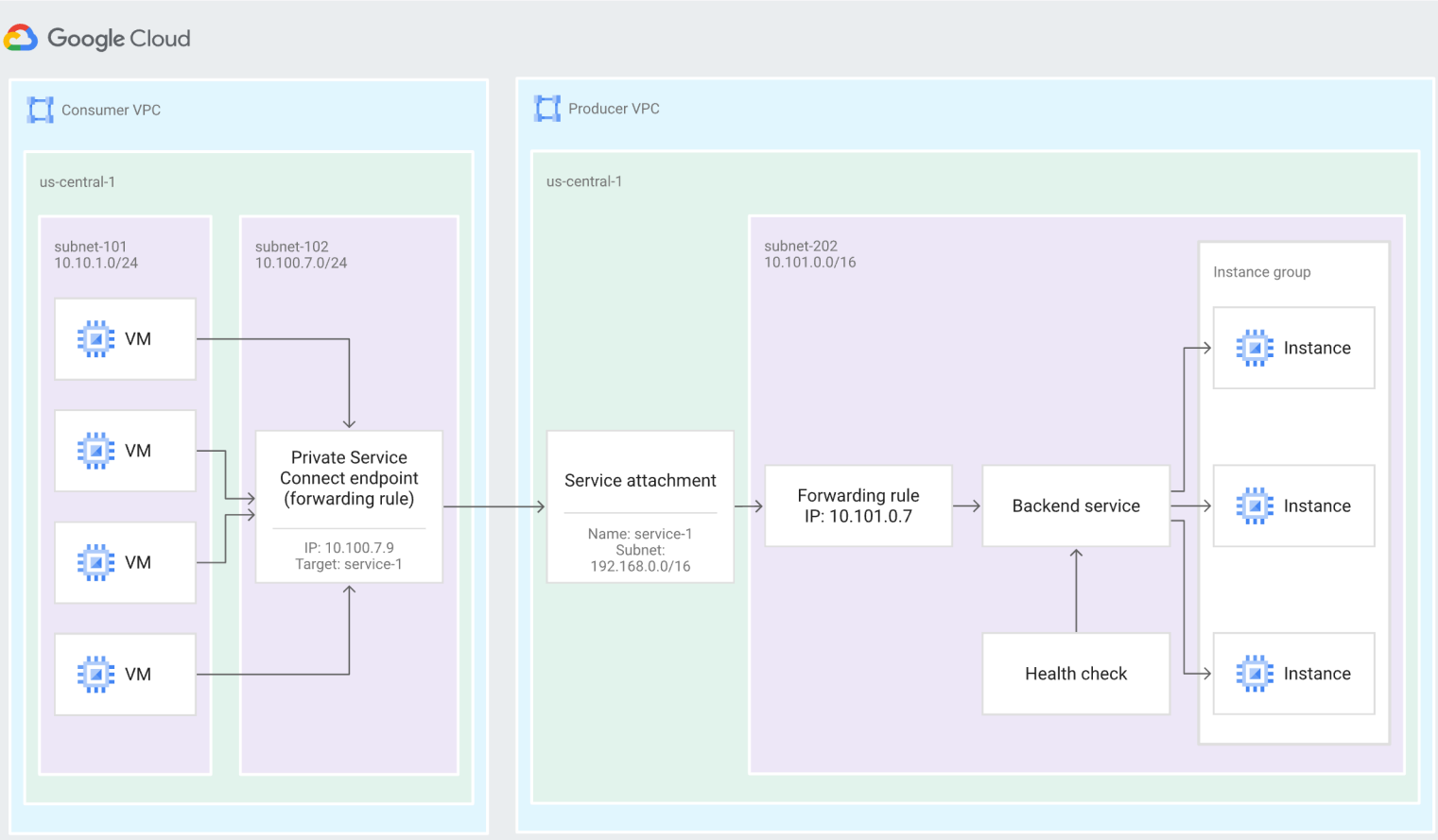
- Allows private consumption of services across VPC networks
- Can be used to access supported Google APIs and services or non-GCP managed services in another VPC network

Cloud-native Networking: Private Service Connect

Private Service Connect



Cloud-native Networking: Private Service Connect



Pearson

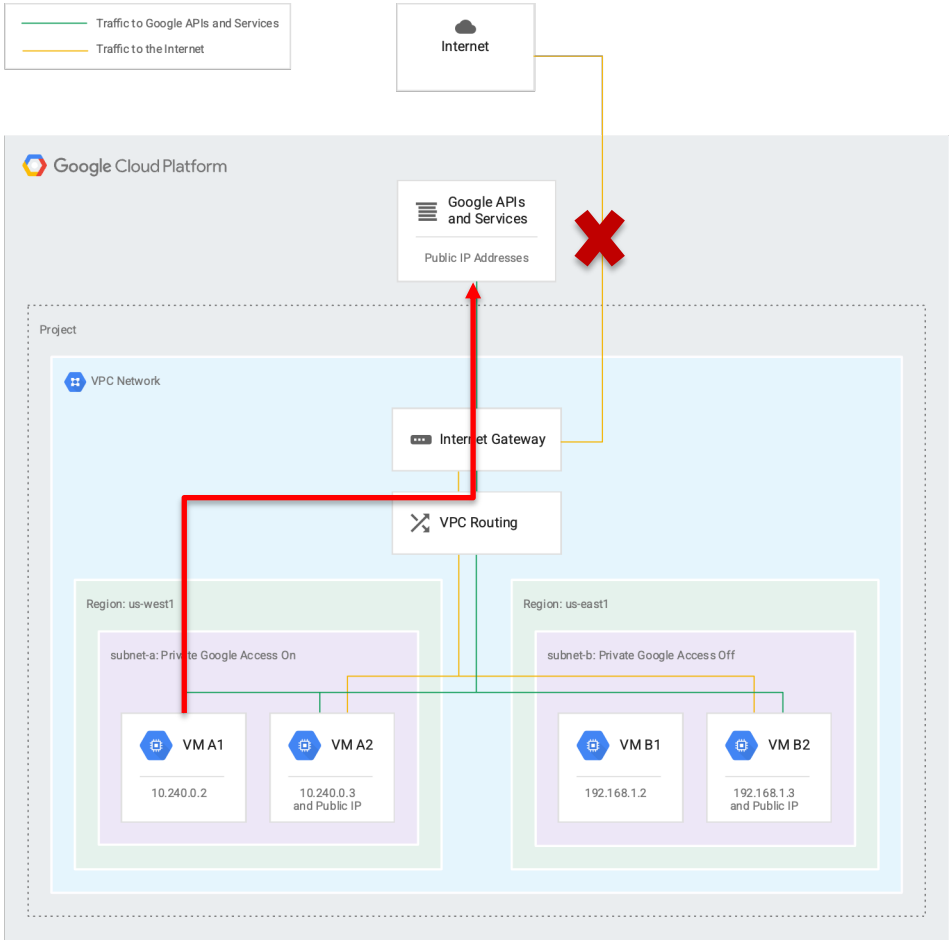
Cloud-native Networking: Private Google Access

- Allows VMs without public IP address to reach Google APIs and services
- Enabled on a subnet
- **Private Google Access for on-premises hosts** allows on-premises hosts to reach Google APIs and services through Cloud VPN or Cloud Interconnect

Cloud-native Networking: Private Google Access

Private Google Access

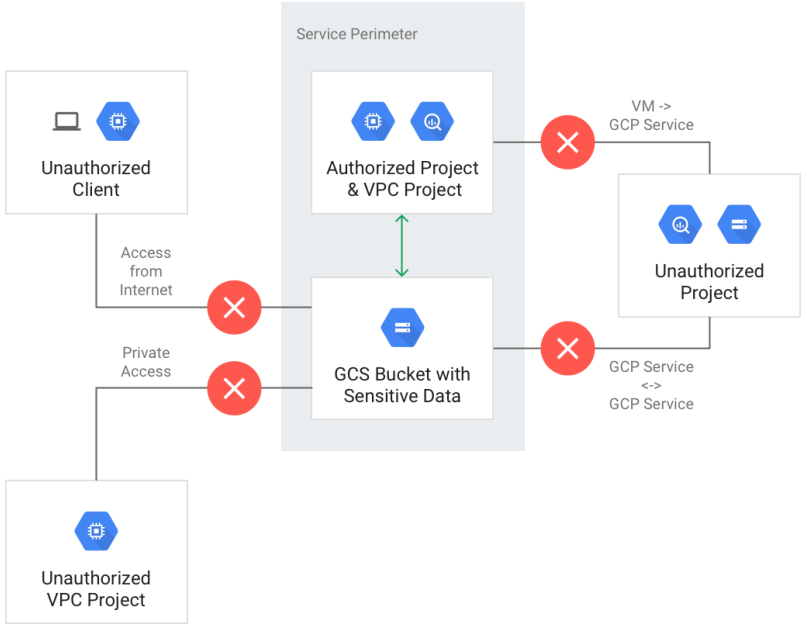
Private Google Access has **no effect** on instances that have external IP addresses



Pearson

Cloud-native Networking: VPC Service Controls

- Create a **perimeter** that protects resources and data
- Clients within perimeter do not have access to (unauthorized) resources outside the perimeter
- Unauthorized clients outside the perimeter don't have access to resources inside the perimeter



Pearson

Cloud-native Networking: Load Balancing

Fully distributed, software-defined load balancing

Load balancer	Scope	Type	Protocol
Global External HTTP(S) Load Balancer	Global, external	Proxy	HTTP(S)
SSL Proxy Load Balancer	Global, external	Proxy	Non-HTTP(S) SSL
TCP Proxy Load Balancer	Global, external	Proxy	TCP (Layer 4)
External TCP/UDP Network Load Balancer	Regional, external	Pass-through	TCP, UDP
Internal TCP/UDP Load Balancer	Regional, internal	Pass-through	TCP, UDP
Internal HTTP(S) Load Balancer	Regional, internal	Proxy	HTTP(S)

Load balancer decision tree

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

Cloud-native Networking: Other Services

Network Tiers

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

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

Pearson

Segment 2: Designing network, storage, and compute

Objectives

- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

Storing Data

- Relational vs. non-relational
- Transactional vs. non-transactional
- Structured vs. unstructured (vs. semi-structured)

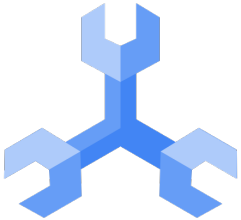
Relational and Structured Storage

Cloud SQL

MySQL, PostgreSQL, and SQL Server

Availability SLA of **99.95%**

Cloud Spanner



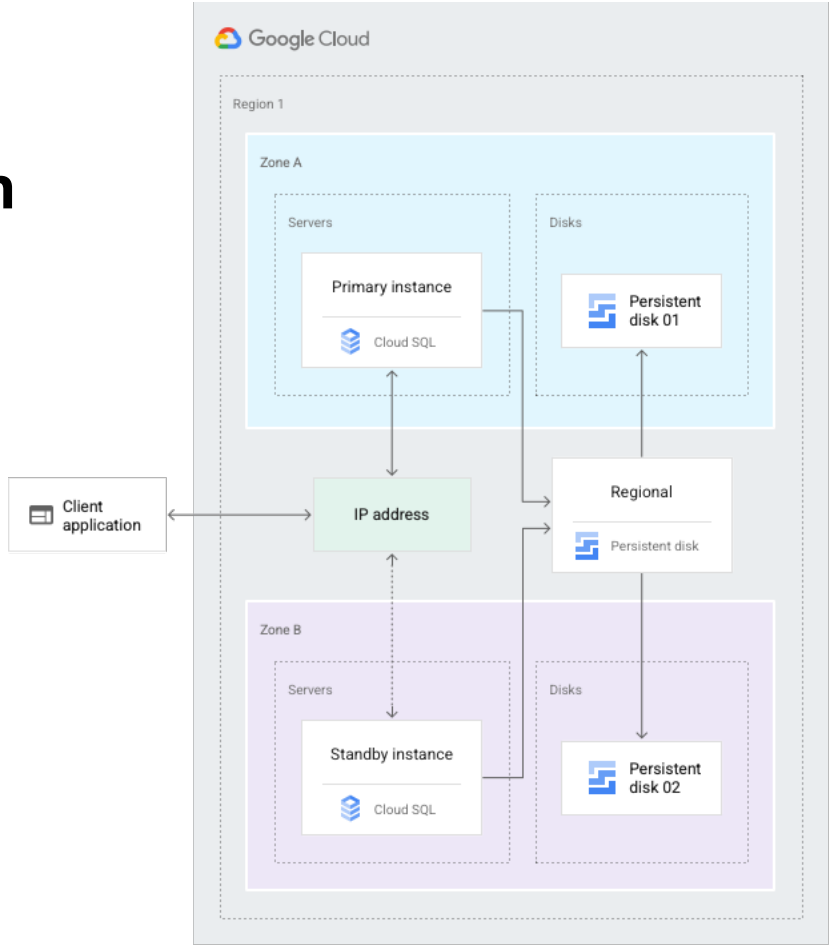
Google Standard SQL and PostgreSQL

Automatic, synchronous replication

Availability SLA of **99.99%** (regional instance)
or **99.999%** (multi-regional instance)

Cloud SQL High Availability (HA) configuration

Cloud SQL HA Configuration



Pearson

Relational and Structured (Non-transactional)

BigQuery

- Serverless petabyte-scale data warehouse
- Storage separate from compute
- For analytics and business intelligence (BI) workloads
- Availability SLA of **99.99%**

Non-Relational and (Semi-)structured Storage

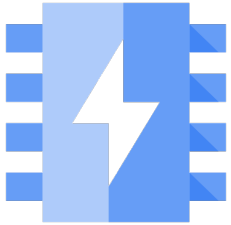
Cloud Firestore

Automatic scaling and availability SLA of **99.999%** (multi-regional)

Real-time updates, direct database connectivity for mobile, web, and IoT apps

Strong consistency, ACID support

Cloud Memorystore



Fully-managed Redis and Memcached

Up to 5 Read replicas (Redis)

Horizontally scale for reads and writes (Memcached)

Vertically scale up to 300GB (Redis) and up to 256GB per node (Memcached)

Availability SLA of **99.9%**

Non-Relational and (Semi-)structured Storage

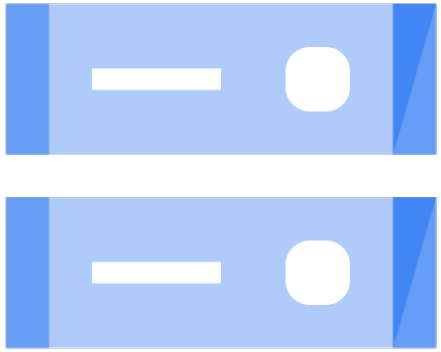
Cloud Bigtable

- Fully managed, NoSQL database for large analytical and operational workloads
- Consistent sub-10ms latency
- Handles millions of requests per second
- Storage scales seamlessly with demand
- Easily connect to Apache ecosystem or BigQuery
- **Up to 99.999%** availability

Unstructured Storage (“Object” Storage)

Cloud Storage

- Object storage for any amount of data
- 99.999999999% annual storage durability
- Different storage classes for cost saving opportunities
- Object Lifecycle Management (OLM) features
- Archival storage
- Availability SLA of **99.9%** (regional), **99.95%** (dual-, multi-region)



File Storage

Cloud Filestore

- Fully managed service for file migration and storage
- Mount file shares on Compute Engine VMs
- Can automatically scale up or down based on demand
- Regional availability SLA of **99.99%**

Block Storage

Persistent Disks

- Durable, high-performance block storage for virtual machines
- Performance scales with the size of the disk and with the number of vCPUs on the VM
- Data stored redundantly

Local SSD

- High-performance block storage for virtual machines
- Physically attached to the server
- Higher throughput and lower latency than persistent disks
- **Data persists only until instance is stopped or deleted**
- Each local SSD is 375GB

Choosing Appropriate Storage Types

Use cases	Service to think of
Data lakes Videos, images, and web assets Backups Media archives	**Cloud Storage**

Choosing Appropriate Storage Types

Use cases	Service to think of
Disks for VMs Storage for databases Sharing read-only data across multiple VMs VM disk backups	**Persistent Disk**

Choosing Appropriate Storage Types

Use cases	Service to think of
Flash-optimized databases Hot caching layer for analytics Application scratch disk	**Local SSD**

Choosing Appropriate Storage Types

Use cases	Service to think of
Rendering and media processing Filesystem migrations Web content management	**Filestore**

Choosing Appropriate Storage Types

Use cases	Service to think of
Mobile apps User-generated content Robust uploads over mobile networks	**Cloud Storage for Firebase**

Choosing Appropriate Storage Types

Use cases	Service to think of
Time-series data Big data IoT Adtech Personalization	**Bigtable**

Choosing Appropriate Storage Types

Use cases	Service to think of
Big data analytics Business intelligence Data warehousing Machine learning	**BigQuery**

Choosing Appropriate Storage Types

Use cases	Service to think of
User profiles User session management Real-time capabilities Cross-device data synchronization Collaborative multi-user mobile apps, Gaming	**Cloud Firestore**

Choosing Appropriate Storage Types

Use cases	Service to think of
Application caching Gaming Stream processing Very low-latency data access	**Cloud Memorystore**

Data Growth Planning: Cost Considerations

Cost-effective for small (and large) data

- Cloud Storage
- BigQuery
- Cloud Firestore
- Cloud SQL

Cost-effective only for large data

- Bigtable ***(>1TB)***
- Cloud Spanner ***(>2TB)***

Data Growth Planning: Unit Size Limits

	Cloud Firestore	Cloud SQL	Cloud Storage	Bigtable	Cloud Spanner	BigQuery
Capacity	Terabytes+	Terabytes	Petabytes+	Petabytes+	Petabytes	Petabytes+
Unit sizes	1MB/entity	Depends on engine	5TB/object	~10MB/cell ~100MB/row	10,240MiB/ row	10MB/row

Data Growth Planning: Going Global

- Consider using Cloud CDN for static web assets
- Consider Spanner instead of Cloud SQL
- Leverage multi-region locations for Cloud Storage buckets
- Leverage multi-region locations for Cloud Firestore
- Leverage Bigtable cross-region replication

Segment 2: Designing network, storage, and compute

Objectives

- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

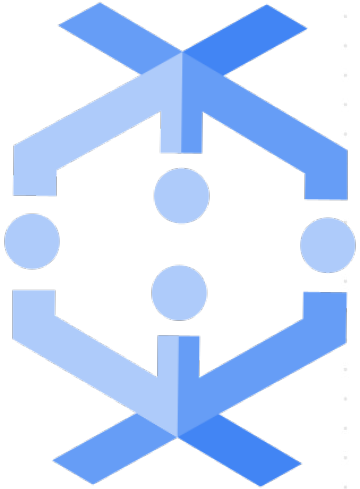
Data Processing Design Considerations

- Streaming vs. batch
- Real-time analytics vs. storing for later use
- Structured vs. unstructured data

Design Data Pipelines: Cloud Dataflow

Cloud Dataflow

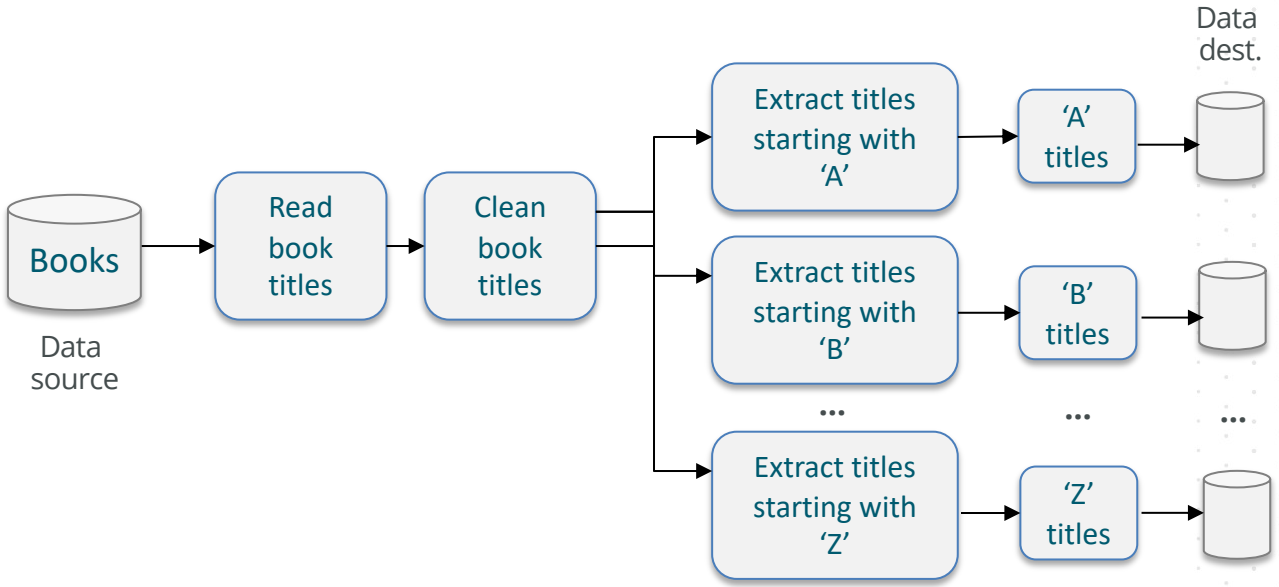
- Unified stream and batch
- Processing resources are provisioned automatically
- Data-aware, horizontal resource autoscaling
- Apache Beam SDK
- Consistent exactly-once processing



Apache Beam Runner

Design Data Pipelines: Cloud Dataflow

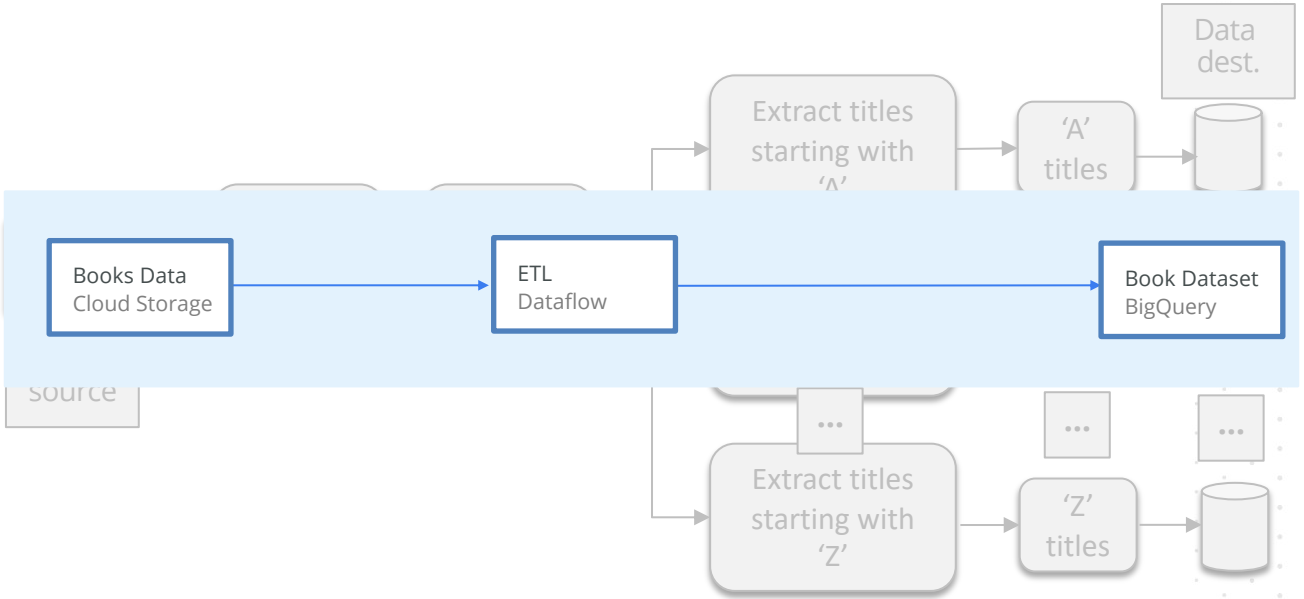
Example ETL Scenario



Pearson

Design Data Pipelines: Cloud Dataflow

Example ETL Scenario

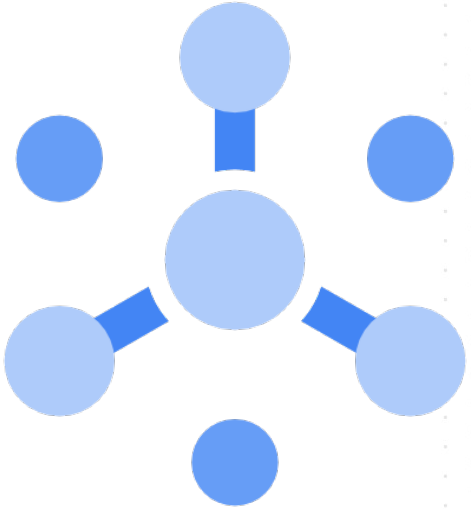


Pearson

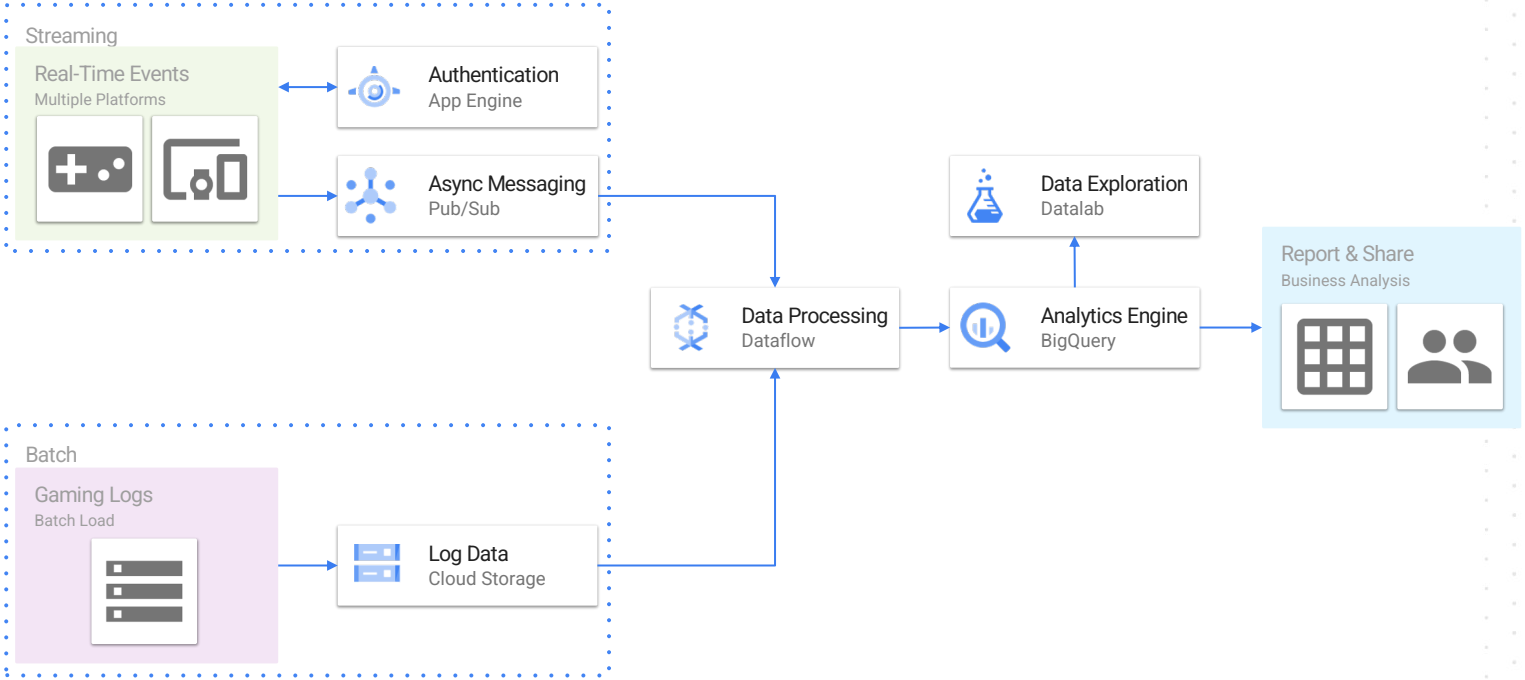
Design Data Pipelines: Cloud Pub/Sub

Cloud Pub/Sub

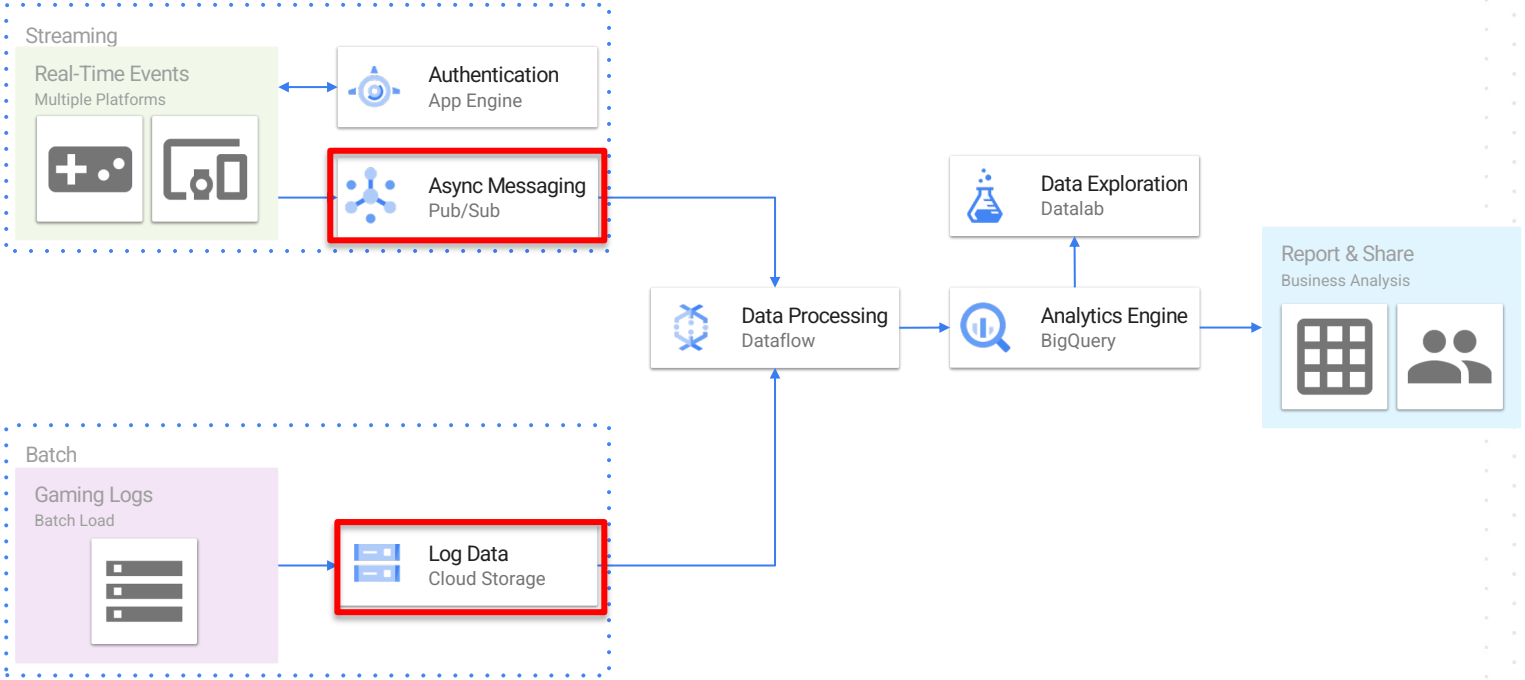
- No-ops streaming ingestion
- Scalable messaging or queueing system
- At-least-once message delivery
- In-order or any-order
- Push and Pull modes



Streaming and Batch Data Pipeline Example

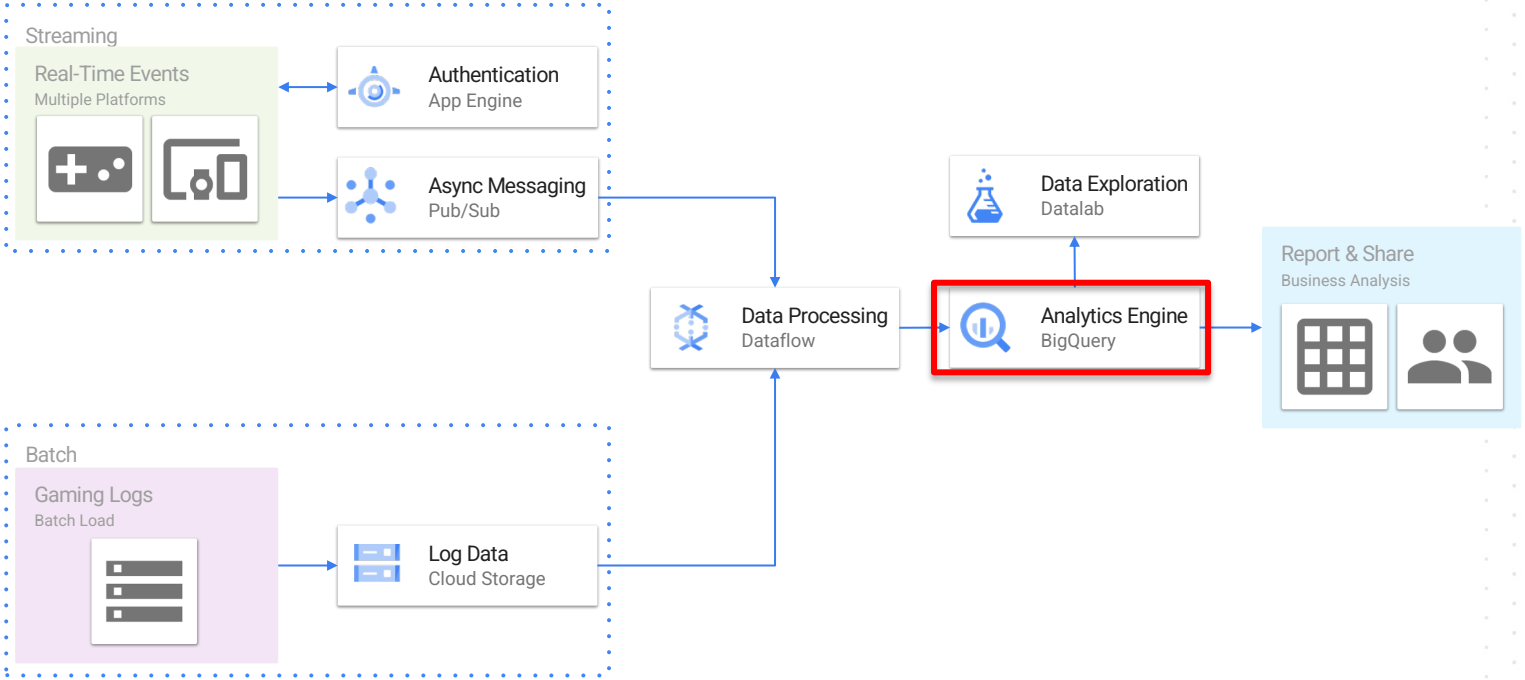


Streaming and Batch Data Pipeline Example



Streaming and Batch Data Pipeline Example

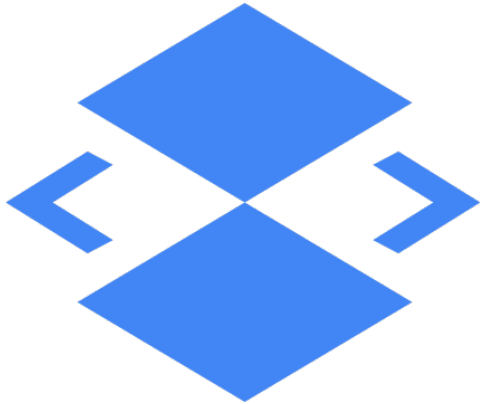
Streaming and Batch Data Pipeline Example



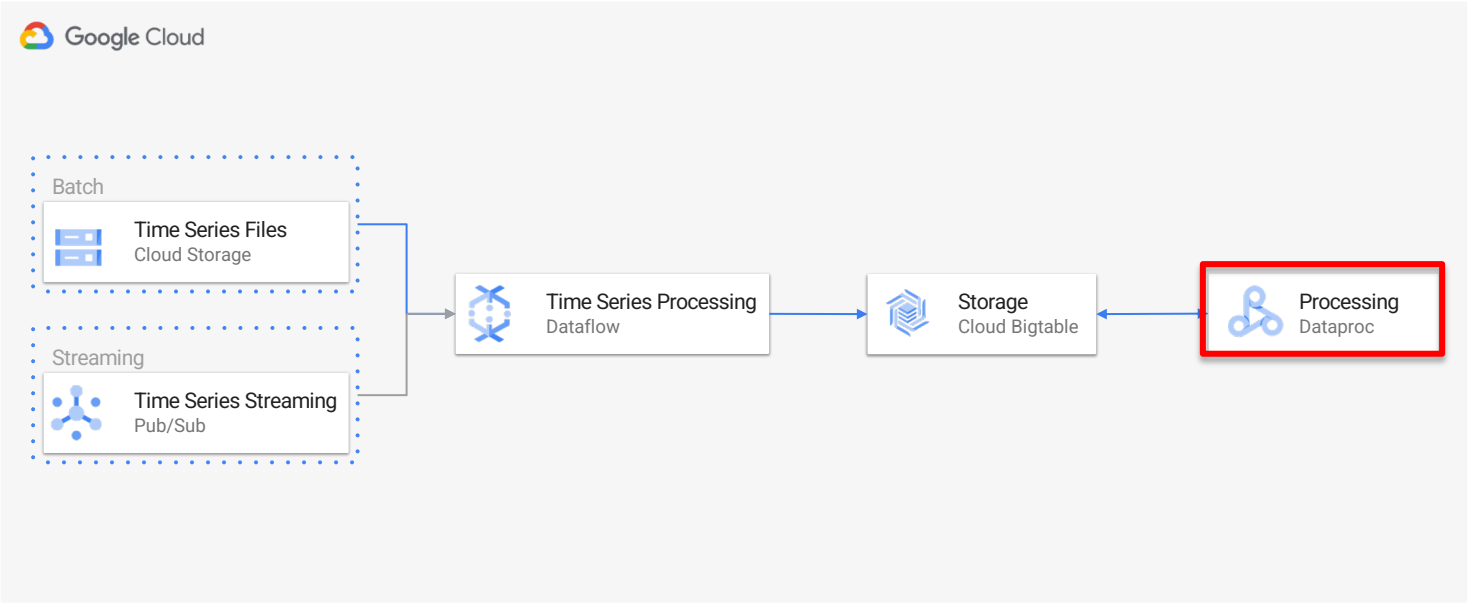
Big Data Processing: Cloud Dataproc

Cloud Dataproc

- Fully managed **Apache Spark** and **Hadoop** service
- Apache Flink, Presto and 30+ other OSS frameworks
- For large-scale batch processing, querying, streaming, and machine learning
- Can run serverless, or on Kubernetes/VMs
- Built-in integration with Cloud Storage, BigQuery and Bigtable



Bid Data Processing: Time Series Analysis



Other Data Processing Services

Cloud Dataprep

- Integrated partner service operated by Trifacta
- Visual data wrangling tool to explore, combine, and transform data
- A set of transformation steps is called a “recipe”

Cloud Data Fusion

- Powered by the open source project CDAP (with pipeline portability)
- Visual data wrangling and data pipelining tool (code-free ETL)

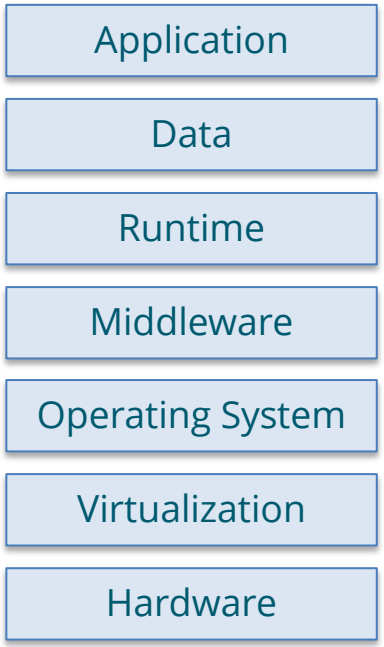
Segment 2: Designing network, storage, and compute

Objectives

- Integrations with on-premises
- Multicloud environments
- Designing VPC networks
- Choosing appropriate storage types
- Choosing data processing technologies
- Choosing compute resources

Compute Strategies

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)

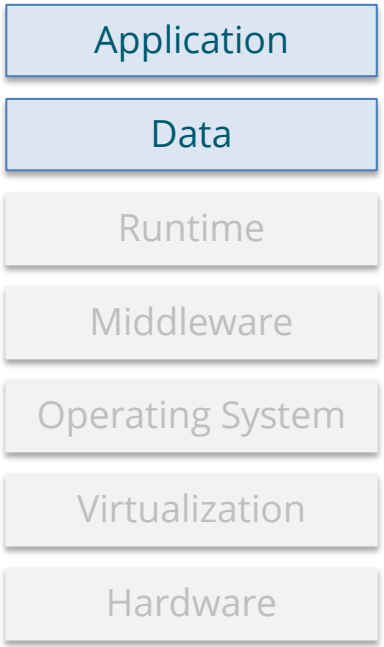


Compute Strategies

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)

Compute Strategies

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)



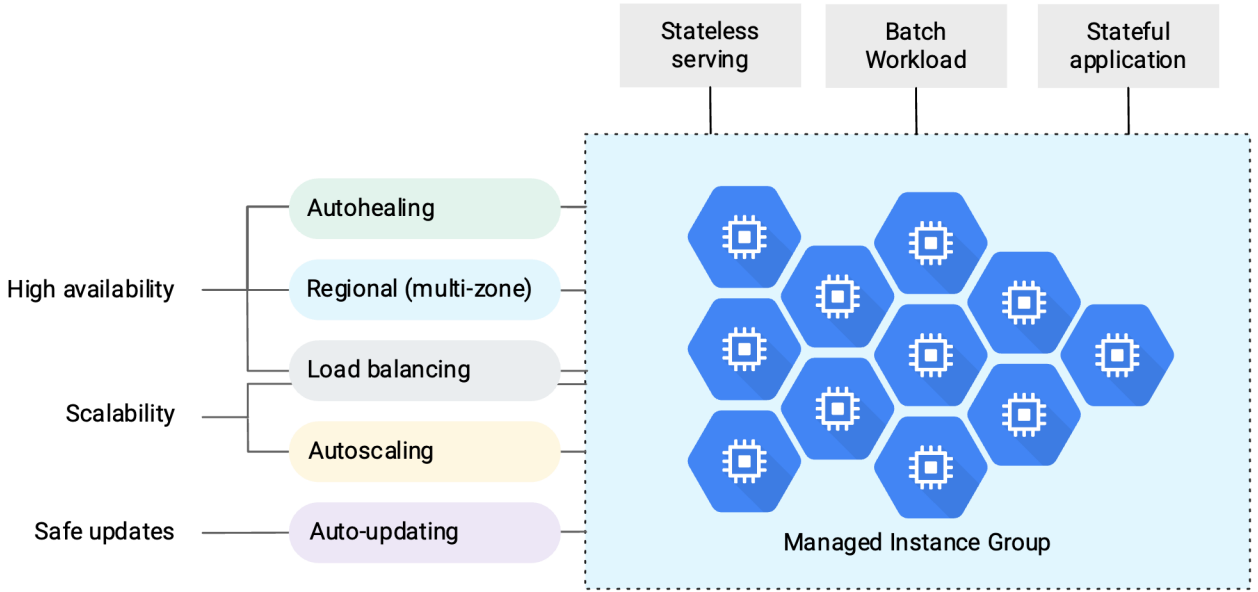
IaaS: Compute Engine instances

When to choose Compute Engine instances?

- Full access to OS settings and/or underlying filesystem is required
- Speed up and/or derisk a data center migration (**lift-and-shift**)
- Legacy applications without a suitable platform product

Managed Instance Groups (MIGs)

Horizontal scaling solution for instances



IaaS-PaaS: Google Kubernetes Engine (GKE)

Google Kubernetes Engine (GKE)

- Managed Kubernetes platform
- High-availability control plane
- Cluster autoscaling
- Two modes of operations: **Standard** and **Autopilot**

IaaS-PaaS: Google Kubernetes Engine (GKE)

You're responsible for:

- Creating and maintaing container images
- Installing and maintaining application libraries and runtime
- Configuring some aspects of networking, storage, and observability
- Architecting for fault-tolerance and scalability (easier to do)

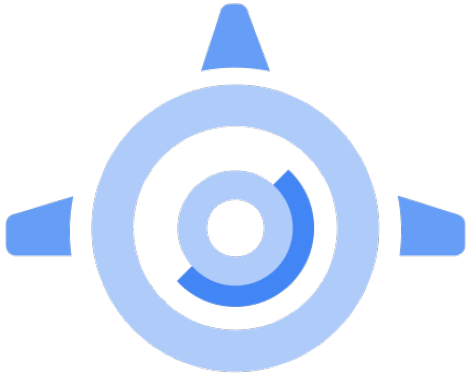
When to Choose GKE

- Containerized workloads of sufficient complexity
- Stateful microservices
- Hybrid- and multi-cloud workloads
- Application modernization

PaaS products: App Engine

App Engine

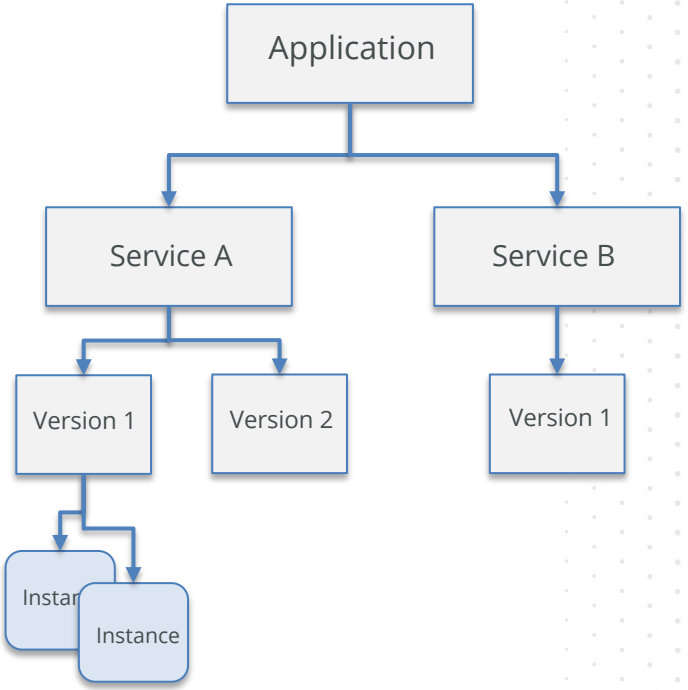
- Fully-managed, serverless platform for developing and hosting web applications
- Can choose from several languages, libraries, and frameworks
- Scales automatically
- Availability SLA of 99.95%
- Two environments: Standard and Flexible



PaaS products: App Engine

App Engine

- Fully-managed, serverless platform for developing and hosting web applications
- Can choose from several languages, libraries, and frameworks
- Scales automatically
- Availability SLA of 99.95%
- Two environments: Standard and Flexible



PaaS products: App Engine

App Engine (Standard)

- Containers are preconfigured with one of several available runtimes.
- The instance class determines the amount of memory and CPU available
- Can scale to zero if no traffic
- Cannot SSH or use custom libraries
- Supported languages: Python, Java, Node.js, PHP, Ruby, and Go
- Supported runtimes: https://cloud.google.com/appengine/docs/standard/runtimes

App Engine (Flexible)

- Uses Compute Engine instances (managed by App Engine)
- Greater CPU and memory instance types
- You can take advantage of custom libraries, use SSH for debugging
- You can deploy your own Docker containers
- Can access resources in the same network
- Does **not** scale to zero

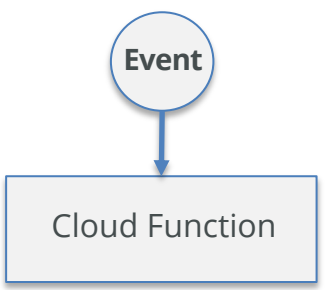
PaaS products: Cloud Functions

Cloud Functions

- Serverless lightweight compute service
- For single-purpose, standalone functions that respond to events
- Can be written using JavaScript, Python 3, Go, or Java runtimes

Example events that can trigger functions:

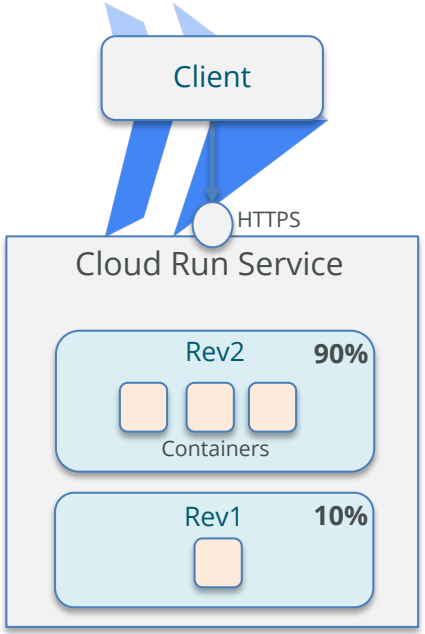
- Changes to data in database
- Files added to a storage system
- New VM created



PaaS products: Cloud Run

Cloud Run

- Serverless container platform
- For highly scalable containerized applications
- No need to build your own container if using **Go**, **Node.js**, **Python**, **Java**, **.NET Core**, or **Ruby**.
- Request-based auto scaling and scale to zero
- Built-in traffic management



Mapping Compute Needs to Platform Products

Scenarios / needs	Compute platform
ETL when a file is created, changed, or removed in Cloud Storage Webhooks for events from 3rd party systems Lightweight APIs Mobile backend that listens and responds to events IoT ETL	**Cloud Function**

Mapping Compute Needs to Platform Products

Scenarios / needs	Compute platform
Web server Django app Web and mobile backend Microservices	**App Engine**

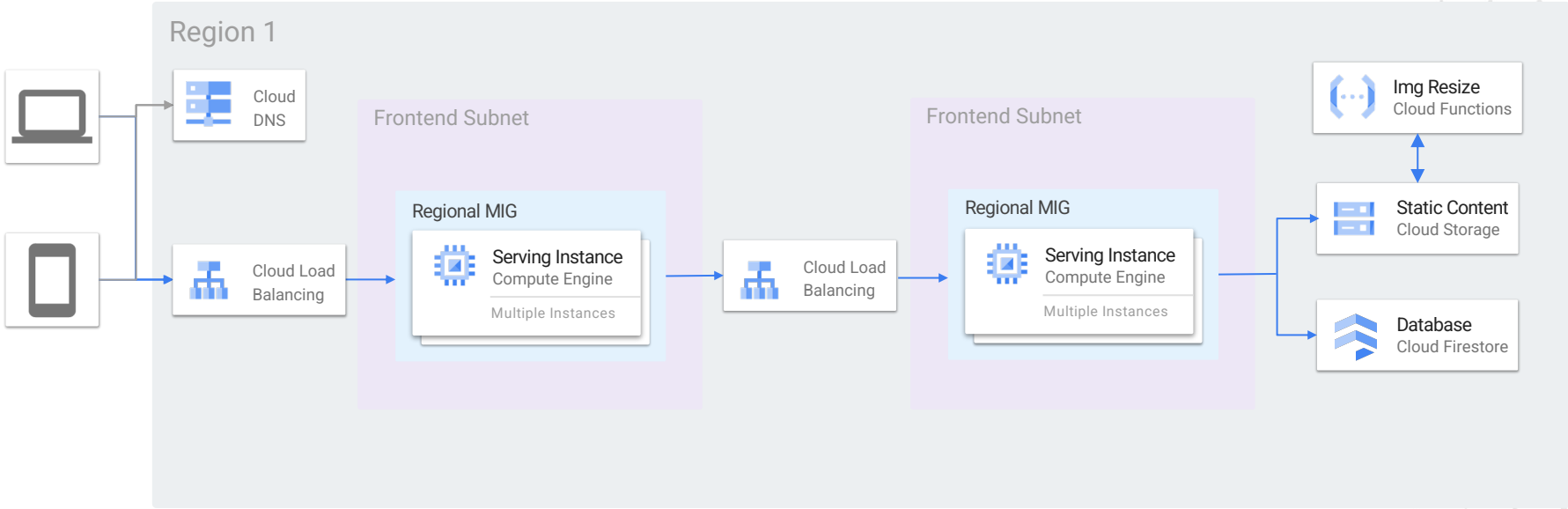
Mapping Compute Needs to Platform Products

Scenarios / needs	Compute platform
Websites and web applications APIs and microservices Processing streaming data from Pub/Sub Custom Runtimes	**Cloud Run**

Demo: Designing Network, Storage, and Compute

Requirements

- Regional 3-tier web app
- IaaS Compute with Debian Linux
- A document database
- Object storage for images
- Event-driven processing for new images (image resizing)


Region 1
Cloud DNS
Cloud Load Balancing
Frontend Subnet
Regional MIG
Serving Instance
Compute Engine
Multiple Instances
Cloud Load Balancing
Frontend Subnet
Regional MIG
Serving Instance
Compute Engine
Multiple Instances
Img Resize
Cloud Functions
Static Content
Cloud Storage
Database
Cloud Firestore

Questions Breakdown

Following an acquisition of another company, you are tasked with simplifying the management of Kubernetes clusters that are deployed on AWS and consolidate them under the GKE control plane.

What solution in Google Cloud can you leverage to meet this requirement?

A. Anthos with Fleets and Connect
B. Anthos Migrate to Containers
C. Google Kubernetes Engine
D. Kubernetes Multicluster Ingress

Questions Breakdown

Following an acquisition of another company, you are tasked with simplifying the management of Kubernetes clusters that are deployed on **AWS** and consolidate them **under the GKE control plane**.

What solution in Google Cloud can you leverage to meet this requirement?

A. Anthos with Fleets and Connect
B. Anthos Migrate to Containers
C. Google Kubernetes Engine
D. Kubernetes Multicluster Ingress

Questions Breakdown

You've been tasked with deploying a new workload on Google Cloud, on a new, isolated VPC. The workload's security requirements dictate that Google APIs and services can only be accessed through a private network using non-public IP addresses.

How should you design the network for this new workload to meet the security requirements?

A. Set up a Dedicated Interconnect connection to connect the VPC with a Google data center.
B. Deploy an Internal HTTP(S) Load Balancer that points to the required Google APIs.
C. Deploy a Cloud NAT instance and configure the VPC's default route to point to the NAT gateway.
D. Configure Private Service Connect on the VPC.

Questions Breakdown

You've been tasked with deploying a new workload on Google Cloud, on a new, isolated VPC. The workload's security requirements dictate that **Google APIs and services** can only be accessed **through a private network** using **non-public IP addresses**.

How should you design the network for this new workload to meet the security requirements?

A. Set up a Dedicated Interconnect connection to connect the VPC with a Google data center.

B. Deploy an Internal HTTP(S) Load Balancer that points to the required Google APIs.

C. Deploy a Cloud NAT instance and configure the VPC's default route to point to the NAT gateway.

D. Configure Private Service Connect on the VPC.

Questions Breakdown

Your company has decided to migrate from an on-premises infrastructure to Google Cloud. The workloads to migrate include a web server on Kubernetes, Apache Hadoop jobs, and a Network-Attached Storage (NAS).

What combination of services would you use on GCP?

A. Migrate Kubernetes application to Google Kubernetes Engine (GKE), Apache Hadoop to Cloud Dataproc, and NAS to Cloud Filestore.

B. Migrate Kubernetes application to Cloud Run, Apache Hadoop to Cloud Dataproc, and NAS to Cloud Storage.

C. Migrate Kubernetes application to Compute Engine VMs, Apache Hadoop to Cloud Dataflow, and NAS to Cloud Filestore.

D. Migrate Kubernetes application to Anthos, Apache Hadoop to Cloud Dataproc, and NAS to Cloud Storage.

Questions Breakdown

Your company has decided to migrate from an on-premises infrastructure to Google Cloud. The workloads to migrate include a web server on **Kubernetes**, **Apache Hadoop** jobs, and a **Network-Attached Storage** (NAS).

What combination of services would you use on GCP?

A. Migrate Kubernetes application to Google Kubernetes Engine (GKE), Apache Hadoop to Cloud Dataproc, and NAS to Cloud Filestore.

B. Migrate Kubernetes application to Cloud Run, Apache Hadoop to Cloud Dataproc, and NAS to Cloud Storage.

C. Migrate Kubernetes application to Compute Engine VMs, Apache Hadoop to Cloud Dataflow, and NAS to Cloud Filestore.

D. Migrate Kubernetes application to Anthos, Apache Hadoop to Cloud Dataproc, and NAS to Cloud Storage.

Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data retention and lifecycle management
- Configuring Cloud SQL for high availability
- Provisioning compute resources
- Configuring Kubernetes

Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data retention and lifecycle management
- Configuring Cloud SQL for high availability
- Provisioning compute resources
- Configuring Kubernetes

Securing Networks: General Guidance

- Disable default networks and configure VPC Firewall
- Secure hybrid and multicloud connectivity
  - IPSec VPN
  - TLS encryption over Interconnect
  - Configure Private Service Connect / Private Google Access
- Configure **VPC Service Controls** to secure a perimeter
- Configure **Cloud Armor** to protect against web attacks

Configuring VPC Firewall Rules: Considerations

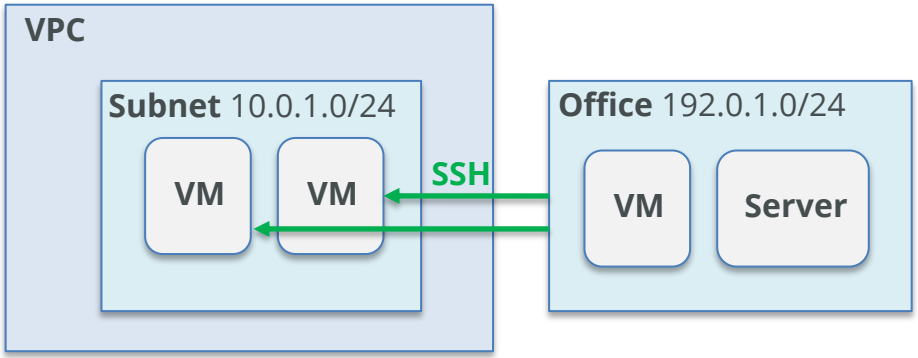
- VPC firewall rules let you allow or deny connections to or from instances in your VPC network:
  - GCE VMs
  - GKE clusters
  - App Engine Flexible Environment
- It does **NOT** let you control connections to/from serverless offerings, such as Cloud Storage, BigQuery, etc.
- It does **NOT** let you control access to fully-qualified domain names (FQDNs), e.g: *www.google.com*.

Every network has two implied firewall rules:
Allow all egress and block all ingress.

VPC Firewall Rules: Common Configurations

- Allow ingress SSH connections to VMs

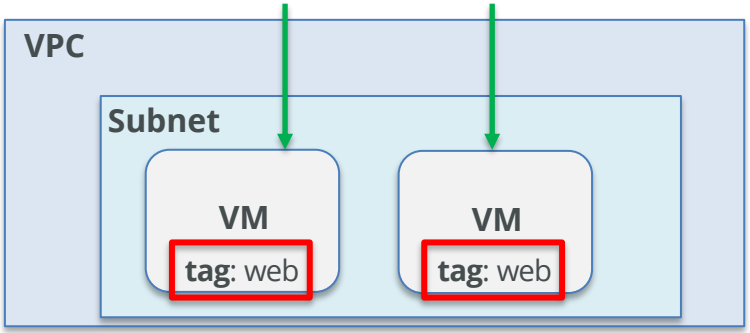
```
gcloud compute firewall-rules create NAME \
    --action=ALLOW \
    --direction=INGRESS \
    --network=NETWORK; default="default" \
    --priority=1000 \
    --rules=tcp:22 \
    --source-ranges=192.0.1.0/24
```



VPC Firewall Rules: Common Configurations

- Allow HTTP traffic only to web servers

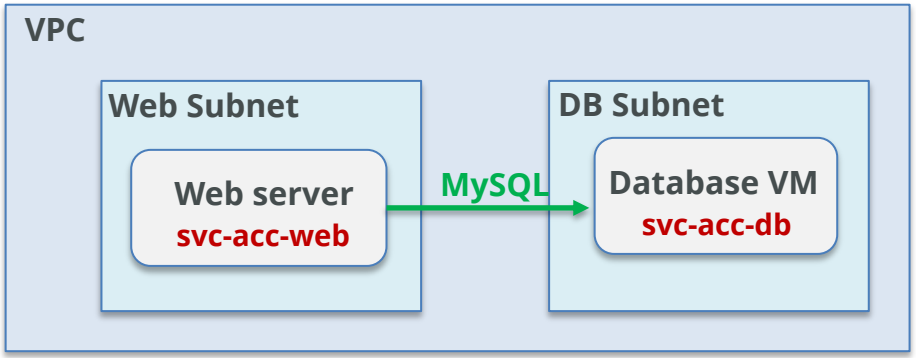
```
gcloud compute firewall-rules create deny-subnet1-
webserver-access \
 --network NETWORK_NAME \
 --action allow \
 --direction ingress \
 --rules tcp:80 \
 --source-ranges 0.0.0.0/0 \
 --priority 1000 \
 --target-tags web
```



VPC Firewall Rules: Common Configurations

- From web server to database using service account

```
gcloud compute firewall-rules create allow-from-web-
to-db \
    --network NETWORK_NAME \
    --allow TCP:3306 \
    --source-service-accounts svc-acc-web@my-
project.iam.gserviceaccount.com \
    --target-service-accounts svc-acc-db@my-
project.iam.gserviceaccount.com
```



Configuring Cloud Armor

Works with:

- Global External HTTP(S) Load Balancer
- TCP Proxy Load Balancer
- SSL Proxy Load Balancer

Configuring Cloud Armor: Security Policies

1) Create a Cloud Armor security policy
2) Add rules to the security policy based on IP address lists, custom expressions, or preconfigured expression sets

   Cloud Armor has a custom rules language used to write advanced match conditions

   Example expression: `inIpRange(origin.ip, '9.9.9.0/24')`

3) Attach the policy to a backend service of the load balancer
4) Optionally: enable Adaptive Protection

Configuring Cloud Armor: Rules and Expressions

Other examples

```
inIpRange(origin.ip, '1.2.3.4/32') && has(request.headers['user-agent'])
&& request.headers['user-agent'].contains('Godzilla')
```

```
has(request.headers['cookie']) &&
request.headers['cookie'].contains('cookie_name=cookie_value')
```

```
origin.region_code == "AU" && !inIpRange(origin.ip, '1.2.3.0/24')
```

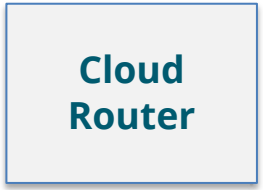
Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data retention and lifecycle management
- Configuring Cloud SQL for high availability
- Provisioning compute resources
- Configuring Kubernetes

Configuring Hybrid and Multi-Cloud Networking: VPN

(Pre-requisite) Create Cloud Router or use existing

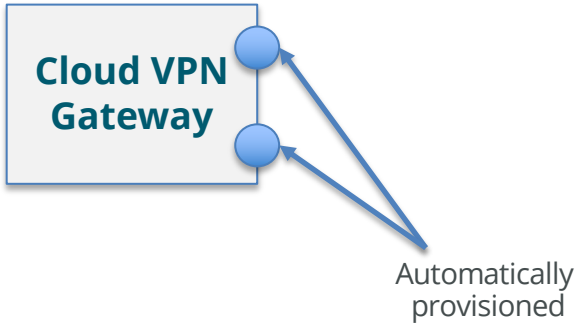


Configuring Hybrid and Multi-Cloud Networking: VPN

Create HA VPN gateway

```
gcloud compute vpn-gateways create GW_NAME \
     --network=NETWORK \
     --region=REGION \
     --stack-type=IP_STACK
```

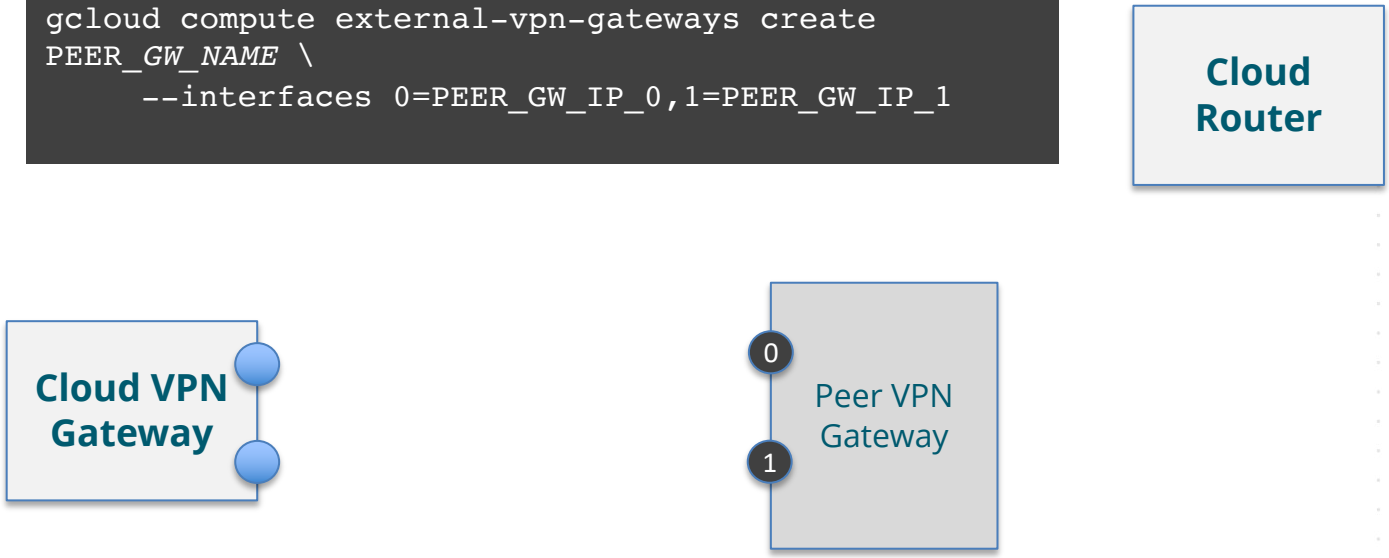
Cloud Router



Configuring Hybrid and Multi-Cloud Networking: VPN

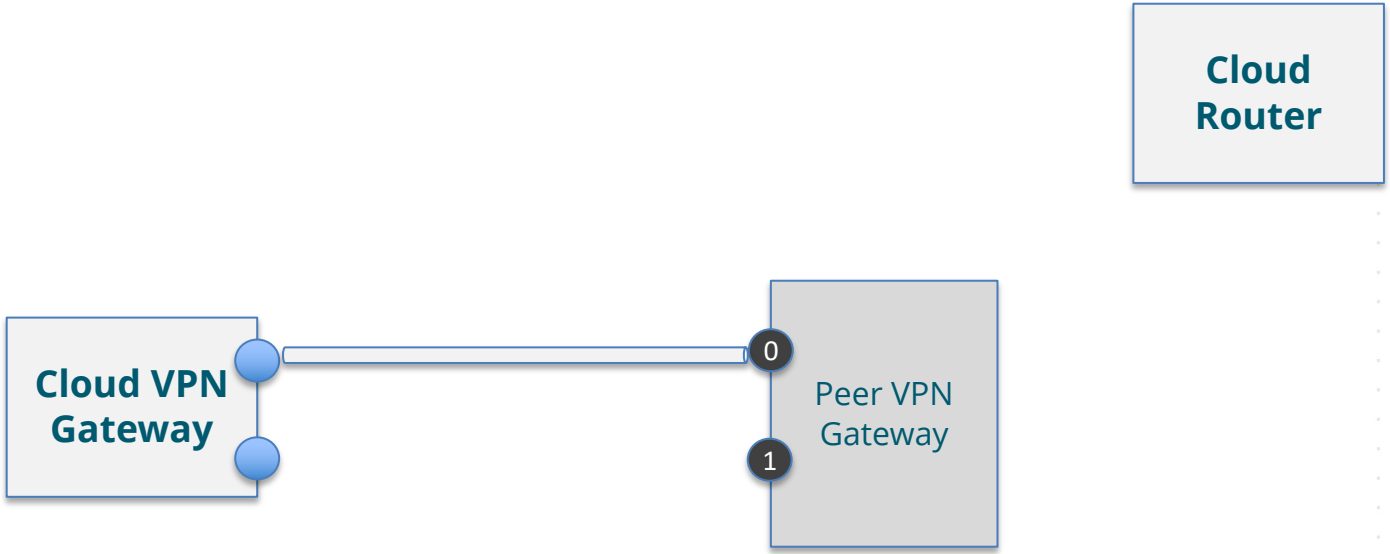
3) Create peer gateway resource

```
gcloud compute external-vpn-gateways create
PEER_GW_NAME \
     --interfaces 0=PEER_GW_IP_0,1=PEER_GW_IP_1
```



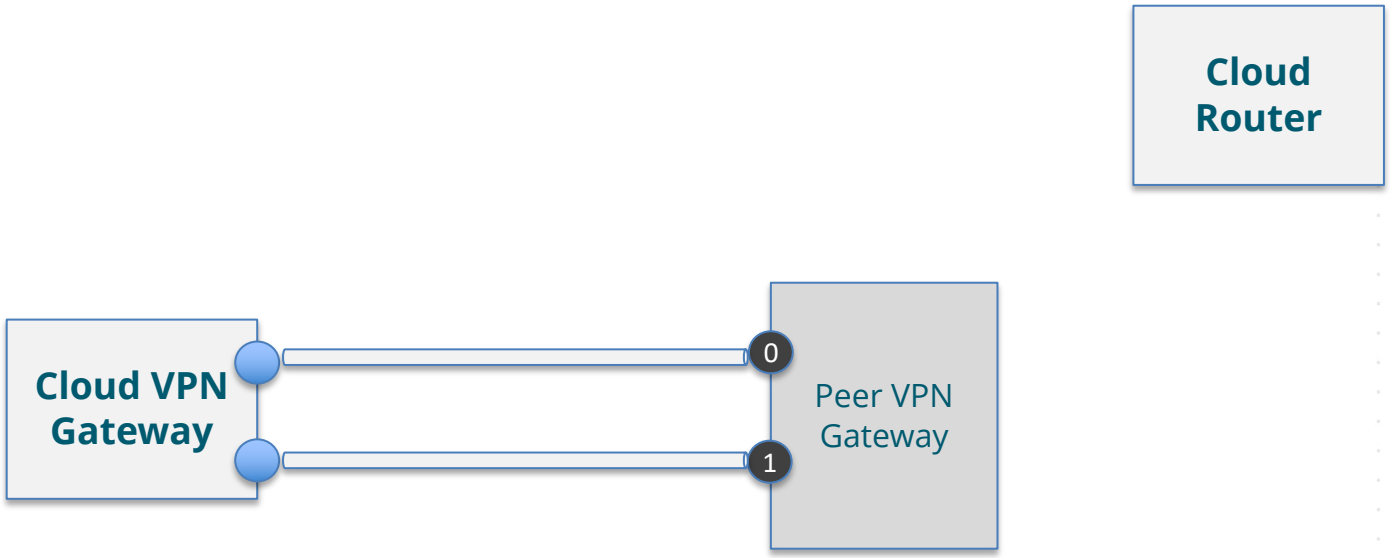
Configuring Hybrid and Multi-Cloud Networking: VPN

4) Create VPN tunnels



Configuring Hybrid and Multi-Cloud Networking: VPN

4) Create VPN tunnels

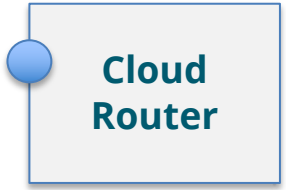


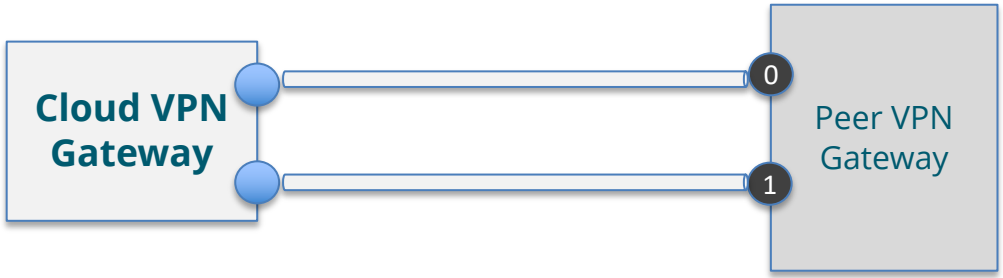
(One tunnel per interface)

Configuring Hybrid and Multi-Cloud Networking: VPN

5) Create BGP sessions

```
gcloud compute routers add-interface ROUTER_NAME \
  gcloud compute routers add-bgp-peer ROUTER_NAME \
     --peer-name=PEER_NAME \
     --peer-asn=PEER_ASN \
     --interface=ROUTER_INTERFACE_NAME_0 \
     --region=REGION
```



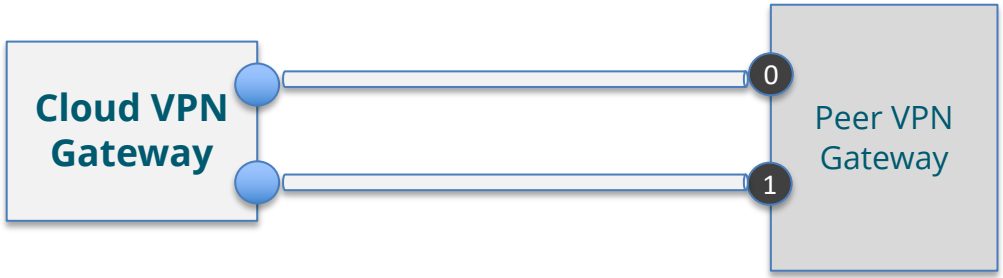


+ repeat both commands for second interface

Configuring Hybrid and Multi-Cloud Networking: VPN

6) Complete the configuration

- Apply corresponding configurations on peer gateway*
- Configure firewall rules (Google Cloud and peer network as required)



*GCP offers specific configuration guidance for certain peer VPN devices

Configuring Dedicated Interconnect

1) Order an Interconnect connection

```
gcloud compute interconnects create INTERCONNECT_NAME \
   --customer-name=NAME \
   --interconnect-type=DEDICATED \
   --link-type=LINK_TYPE \
   --location=LOCATION_NAME \
   --requested-link-count=NUMBER_OF_LINKS \
  [--noc-contact-email=EMAIL_ADDRESS] \
  [--description=STRING]
```

For redundancy, create a duplicate Interconnect connection in a different edge availability domain

Configuring Dedicated Interconnect

2) Google emails you a confirmation and allocates ports
3) Google generates a Letter of Authorization and Connecting Facility Assignment (**LOA-CFA**)
4) Retrieve LOA-CFAs and send to your vendor so that they can install your connections
5) Create VLAN attachments to determine which VPC networks can reach your on-premises network

Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data security and access management
- Configuring data retention and lifecycle management
- Provisioning compute resources
- Configuring Kubernetes

Data Storage and Database Services

Configuring Storage Allocation: SQL Databases

Cloud SQL

- Data storage is limited by instance type
- When creating an instance, you choose:
  - Storage type (SSD or HDD)
  - Storage capacity
  - Whether to enable automatic storage increase
- After instance creation, it is possible to increase storage (but NOT decrease)

Configuring Storage Allocation: SQL Databases

Cloud Spanner

- Data storage is defined by "compute capacity" on instance creation
  - Defined as either number of *processing units* or number of *nodes*
  - 1000 *processing units* = 1 *node*
- Instance can be edited and capacity can be increased or decreased

Configuring Storage Allocation: SQL Databases

Cloud Spanner

- Precise data allocation is done by Spanner:
  - For instances smaller than 1 node: 409.6GB for every 100 processing units in the database
  - For instances larger than 1 node (inclusive): 4TB for each node

Note: Cloud Spanner bills for the storage that instances actually utilize, and not their total storage allotment.

Configuring Storage Allocation: NoSQL Databases

Firestore

- No data allocation. Data capacity scales with usage
- Pay for the amount of data consumed

Configuring Storage Allocation: NoSQL Databases

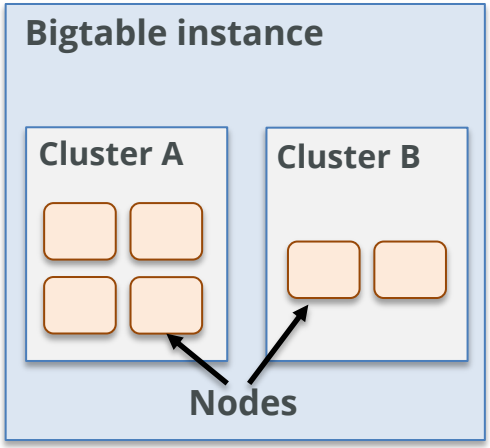
Bigtable

- Data storage capacity is determined by instance capacity (number of nodes)
  - SSD clusters: 5TB per node
  - HDD clusters: 16TB per node
- As a best practice: add enough nodes so you are only using 70% of storage capacity
- If autoscaling enabled, you determine:
  - Min. and max. number of nodes
  - Storage utilization target

Compute Provisioning: NoSQL Databases

Relational (SQL)		NoSQL		Object	Warehouse	In-memory
Cloud SQL	Cloud Spanner	Firestore	Bigtable	Cloud Storage	BigQuery	Memorystore

Bigtable



Pearson

Configuring Storage Allocation: Object Storage

Cloud Storage

- No data allocation. “Unlimited” storage capacity
- Max object size: 5TB

Configuring Storage Allocation: Data Warehouse

BigQuery

- No data allocation. “Unlimited” storage capacity
- Pay for the amount of storage consumed

Configuring Storage Allocation: In-memory Databases

Memorystore

- Data allocated during instance creation
- When creating an instance, you select a size between 1-300GB
- Instance's capacity tier and network performance are determined by size you choose

Data Storage Considerations

- Bigtable and Spanner are not suitable for smaller amounts of data
- Firestore is less expensive per GB, but you also pay for reads and writes
- BigQuery storage is relatively cheap, but you have to pay for running queries

Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data security and access management
- Configuring data retention and lifecycle management
- Provisioning compute resources
- Configuring Kubernetes

Configuring Data Access Control: SQL Databases

Cloud SQL

- Access can be controlled through IAM and "authorized networks"
- If you configure a Cloud SQL instance to have a public IPv4 address, you can specify public IP addresses to accept connections from
- You can update the authorized network list with IP address ranges

Configuring Data Access Control: SQL Databases

Cloud Spanner

- Access can be controlled through IAM

Configuring Data Access Control: NoSQL Databases

Firestore

- Access can be controlled with IAM
- For mobile and web client libraries, use Firebase Authentication and Cloud Firestore Security Rules to handle serverless authentication and authorization.

Configuring Data Access Control: NoSQL Databases

Bigtable

- Access can be controlled through IAM

Configuring Data Access Control: Object Storage

Cloud Storage

- Access can be controlled through IAM

Configuring Data Access Control: Data Warehouse

BigQuery

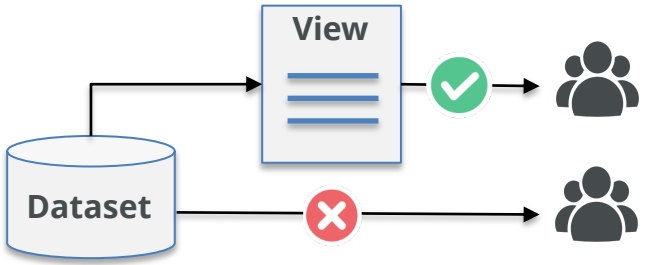
- Granular IAM-based access control
  - Dataset level
  - Table or view level

Configuring Data Access Control: Data Warehouse

Relational (SQL)		NoSQL		Object	Warehouse	In-memory
Cloud SQL	Cloud Spanner	Firestore	Bigtable	Cloud Storage	BigQuery	Memorystore

BigQuery

- Access control by authorization
  - Authorized views



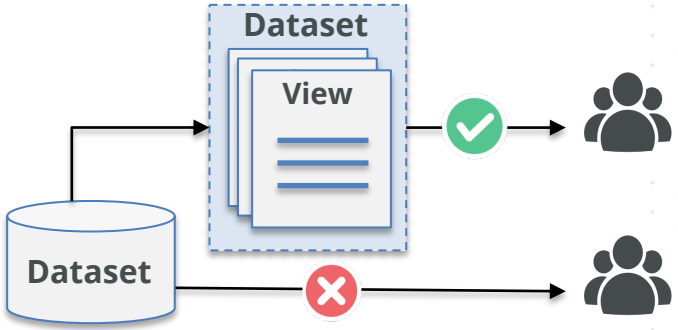
Pearson

Configuring Data Access Control: Data Warehouse

Relational (SQL)		NoSQL		Object	Warehouse	In-memory
Cloud SQL	Cloud Spanner	Firestore	Bigtable	Cloud Storage	BigQuery	Memorystore

BigQuery

- Access control by authorization
  - Authorized views
  - Authorized datasets

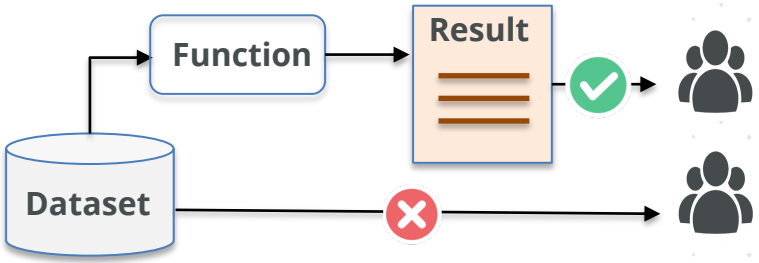


Configuring Data Access Control: Data Warehouse

Relational (SQL)		NoSQL		Object	Warehouse	In-memory
Cloud SQL	Cloud Spanner	Firestore	Bigtable	Cloud Storage	BigQuery	Memorystore

BigQuery

- Access control by authorization
  - Authorized views
  - Authorized datasets
  - Authorized functions



Configuring Data Access Control: In-memory Databases

Memorystore

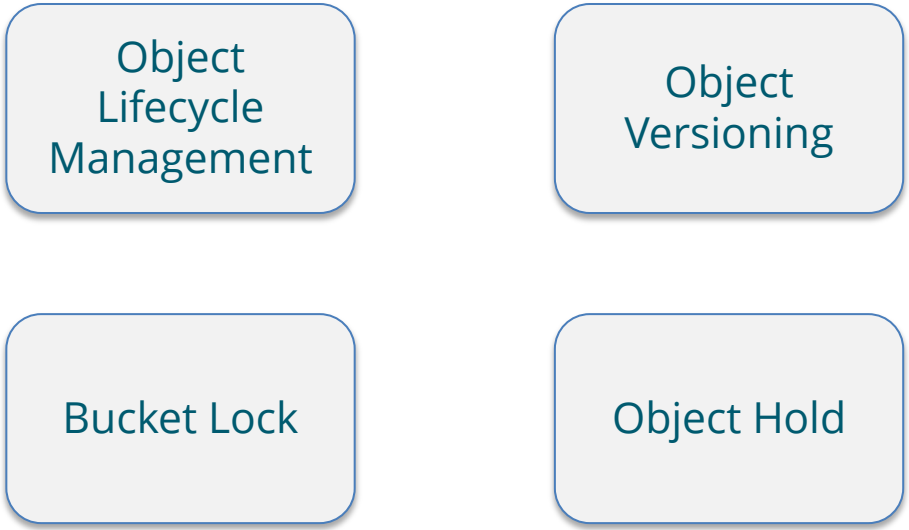
- Access control can be applied with IAM

Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data security and access management
- Configuring data retention and lifecycle management
- Provisioning compute resources
- Configuring Kubernetes

Cloud Storage Data Lifecycle Controls



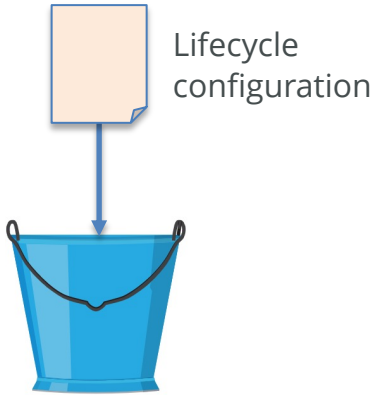
Object Lifecycle Management

- Use cases:
  - Setting a time-to-live (TTL) on objects
  - Retain noncurrent versions of objects
  - Automate “downgrade” of objects’ storage classes
- You set a lifecycle configuration on a bucket

  Downgrade the storage class of objects older than 365 days to Coldline storage.

  Delete objects created before January 1, 2013.

  Keep only the 3 most recent versions of each object in a bucket with versioning enabled.



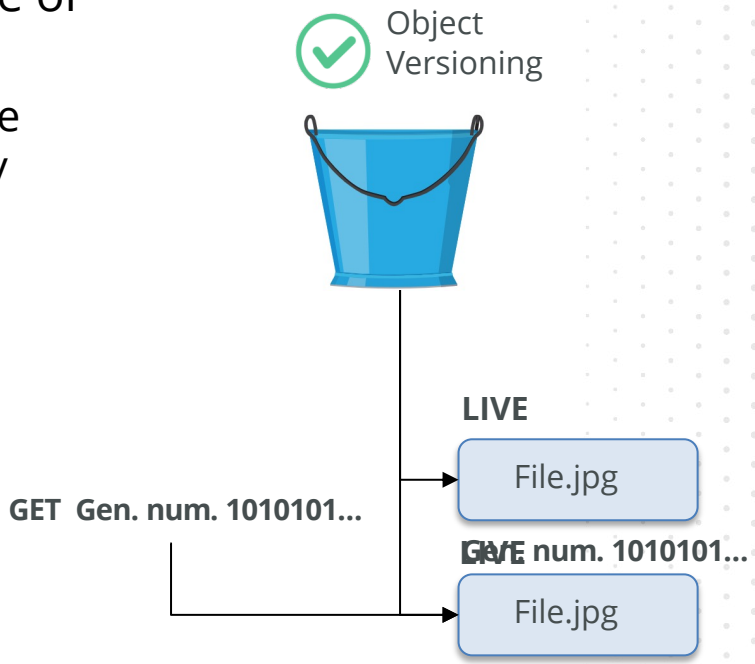
Object Versioning

- Use cases:
  - Support retrieval of objects that are deleted
  - Retrieval of objects that are replaced

- You enable Object Versioning for a bucket

Object Versioning: How it Works

- Cloud Storage retains a noncurrent object version each time you replace or delete a live object version
  - Noncurrent versions retain the same name, but are uniquely identified by generation number
  - Noncurrent versions only appear in requests that explicitly specify the generation number



NOTE: Cannot be enabled on a bucket that has a retention policy

Bucket Lock

- Allows you to configure a data retention policy for a Cloud Storage bucket
- Can help with regulatory and compliance requirements

Bucket Lock: How it Works

- You add a retention policy to a bucket to specify a retention period
  - Objects in the bucket can only be deleted or replaced once their age is greater than the retention period
  - Retention policy retroactively applies to existing objects
- You can lock a retention policy to permanently set it on the bucket
  - You cannot remove the policy or reduce the retention period it has

Object Hold

- Holds are object metadata flags
- While an object has a hold, it cannot be deleted or replaced
- Two types of hold:

Event-based

- Resets the object's time when hold is removed

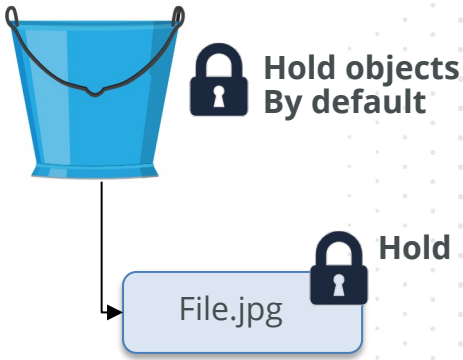
Temporary

- Does not affect the object’s time for the purpose of retention period

Without a retention policy, both hold types behave the same

Object Hold: How it Works

- You place a hold on an object
- You can set the default event-based hold property on a bucket
- A hold can be released any time



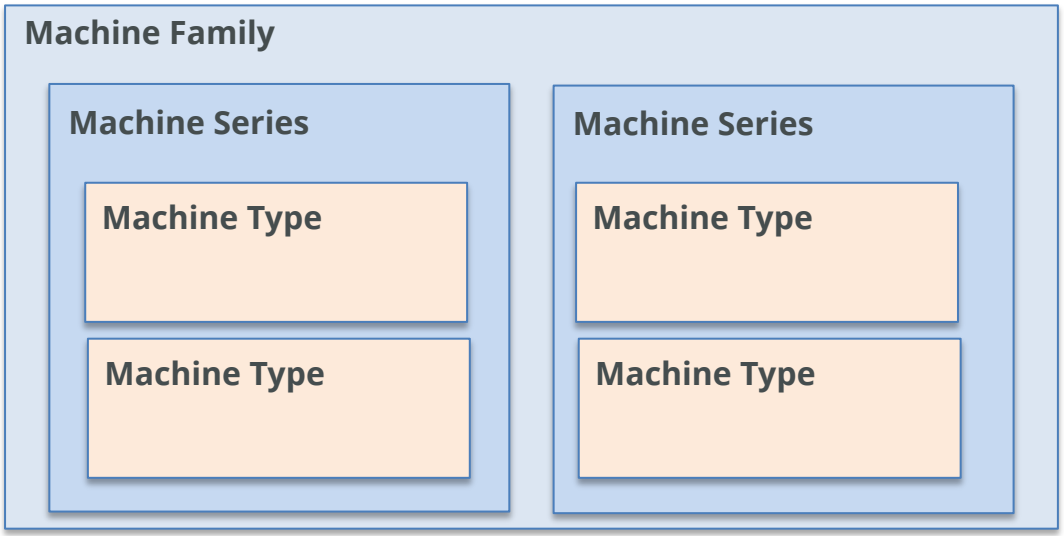
Segment 3: Configuring network, storage, and compute

Objectives

- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data security and access management
- Configuring data retention and lifecycle management
- Provisioning compute resources
- Configuring Kubernetes

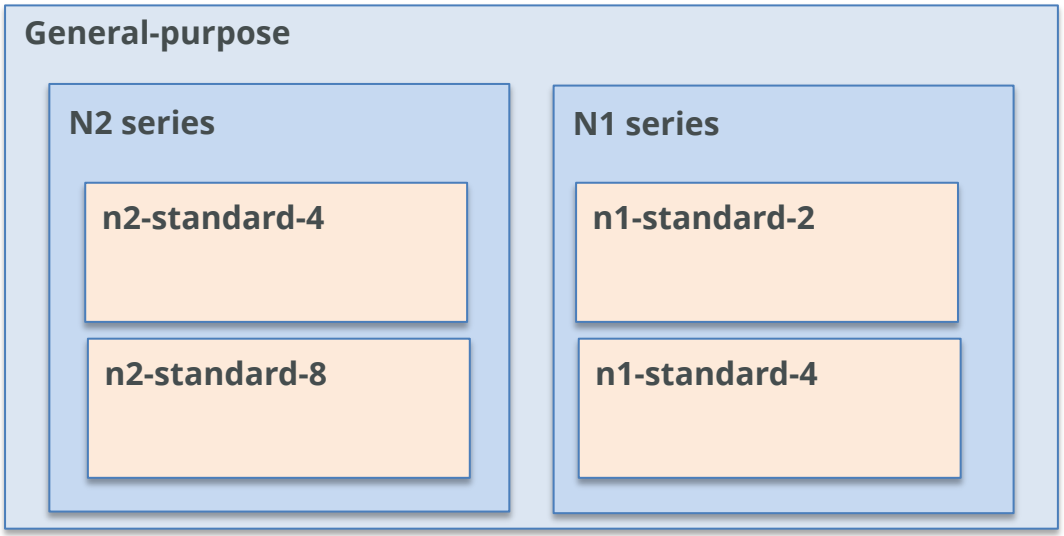
Google Compute Engine

- Customizable virtual machines
- Pre-defined machine types organized into machine families and machine series



Google Compute Engine

- Customizable virtual machines
- Pre-defined machine types organized into machine families and machine series



Compute Engine: Machine Families

- General-purpose
- Compute-optimized
- Memory-optimized
- Accelerator-optimized

Compute Engine: Machine Series

Series	Workload Type	Use cases
E2	General-purpose, cost-optimized	Web/app serving, back-office apps, small-medium databases, dev environments
N2, N2D, N1	General-purpose, balanced	Web/app serving, back-office apps, medium-large databases, cache, media/streaming
Tau T2D	General-purpose, Scale-out optimized	Web serving, containerized microservices, media transcoding, large-scale Java apps

Compute Engine: Machine Series

Series	Workload Type	Use cases
M2, M1	Memory-optimized	SAP HANA, in-memory databases, SQL Server
C2, C2D	Compute-optimized	High-performance computing (HPC), gaming, ad serving, AI/ML, media transcoding
A2	Accelerator-optimized	CUDA-enabled ML training and inference, HPC, massive parallelized computation

VM Instance Templates

- Used to create VM instances and managed instance groups (MIGs)
- Global resource, not tied to a region or zone
- Can be based on existing instance

- ✓ Machine type
- ✓ Boot disk image
- ✓ Labels
- ✓ Startup script

Provisioning Managed Instance Groups

- To enable autoscaling, use *set-autoscaling* and set the autoscaling signals

Example

```
gcloud compute instance-grou
managed-instance-group \
  --max-num-replicas 20 \
  --target-cpu-utilization
  --cool-down-period 90
```

Target utilization metrics:

- Average CPU utilization.
- HTTP load balancing serving capacity, which can be based on either utilization or requests per second.
- Cloud Monitoring metrics.

Provisioning Managed Instance Groups

- To enable autoscaling, use *set-autoscaling* and set the autoscaling signals

Example

```
gcloud compute instance-groups managed set-autoscaling example-
managed-instance-group \
  --max-num-replicas 20 \
  --target-cpu-utilization 0.60 \
  --cool-down-period 90
```

Preemptible Instances

- Instances offered at a discount (60-91%) in periods of excess Compute Engine capacity
- Compute Engine might stop (*preempt*) these instances if it needs to reclaim the capacity
- For fault-tolerant apps
- **Run for a maximum of 24 hours**
- **No SLA**

Spot Instances

- Latest version of preemptible instances
- Similar to preemptible VMs, **except no 24-hour limit**

Handle Preemption

- You can use a shutdown script to perform cleanup actions before each VM is preempted
- Add script to VM using metadata tag *shutdown-script*

```
gcloud compute instances create example-instance \
    --metadata-from-file shutdown-
script=examples/scripts/install.sh
```

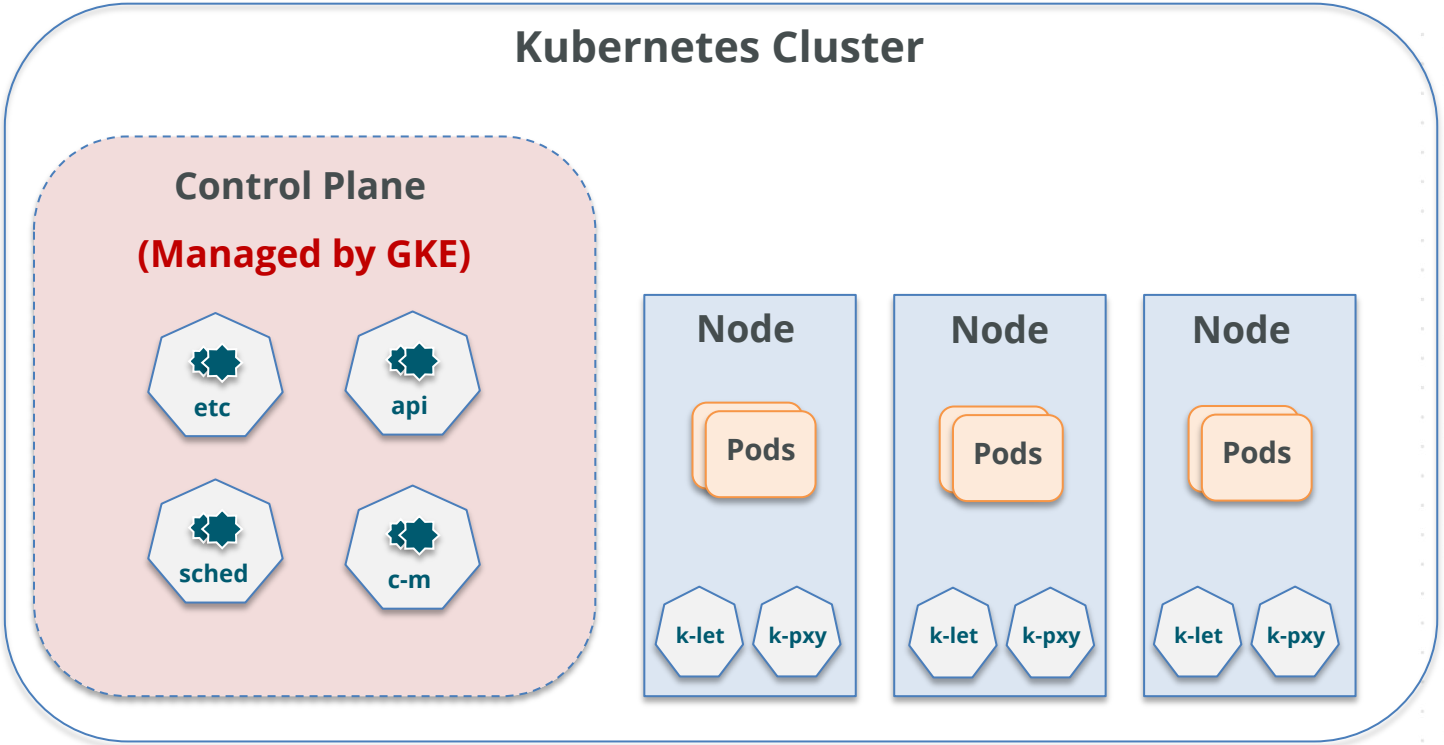
- Example script action: perform a parallel upload of a checkpoint file to a Cloud Storage bucket.

Segment 3: Configuring network, storage, and compute

Objectives

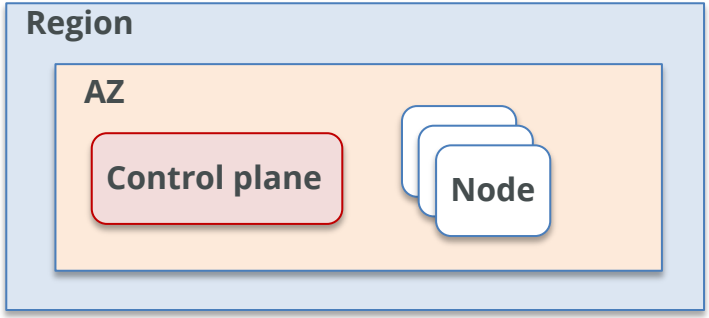
- Securing networks
- Setting up hybrid and multicloud networking
- Provisioning data storage
- Configuring data security and access management
- Configuring data retention and lifecycle management
- Provisioning compute resources
- Configuring Kubernetes

Kubernetes Cluster Components

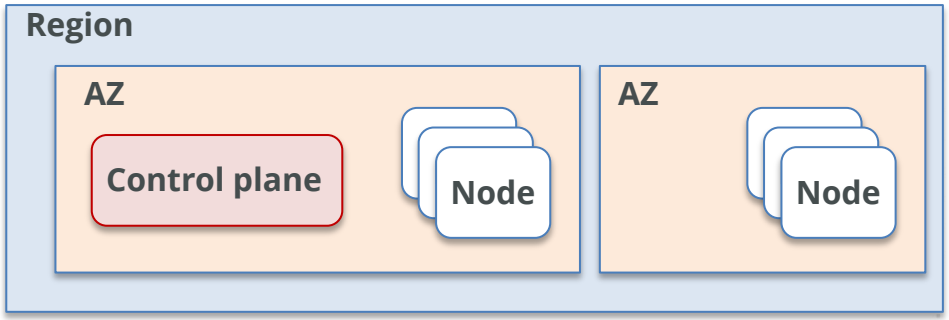


Standard GKE Cluster: Deployment Types

Single-zone cluster

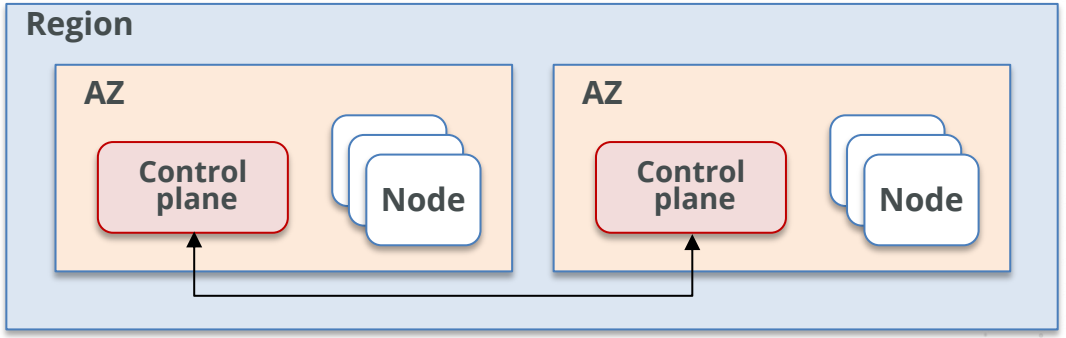


Multi-zonal cluster



Standard GKE Cluster: Deployment Types

Regional cluster



Creating a Standard Zonal Cluster

- Specifying number of nodes:

```
gcloud container clusters create example-cluster \
--zone us-central1-a \
--node-locations us-central1-a,us-central1-b,us-central1-c \
--num-nodes 5
```

Note: when the –num-nodes flag is omitted, the default number of nodes is used which is **three nodes per zone**

Configuring Cluster Autoscaling

- GKE's cluster autoscaler automatically resizes **the number of nodes in a given node pool**

```
gcloud container clusters create CLUSTER_NAME \
    --enable-autoscaling \
    --num-nodes NUM_NODES \
    --min-nodes MIN_NODES \
    --max-nodes MAX_NODES \
    --region=COMPUTE_REGION
```

Configuring Horizontal Pod Autoscaling

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Configuring Horizontal Pod Autoscaling

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Configuring Horizontal Pod Autoscaling

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Configuring Horizontal Pod Autoscaling

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Configuring Vertical Pod Autoscaling

```
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: my-vpa
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind:       Deployment
    name:       my-auto-deployment
  updatePolicy:
    updateMode: "Auto"
```

Configuring Vertical Pod Autoscaling

```
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: my-vpa
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind:       Deployment
    name:       my-auto-deployment
  updatePolicy:
    updateMode: "Auto"
```

Configuring Vertical Pod Autoscaling

Get output resource

```
kubectl get vpa r
```

Sample Output

```
...
  recommendation:
    containerRecommendations:
    - containerName: my-container
      lowerBound:
        cpu: 536m
        memory: 262144k
      target:
        cpu: 587m
        memory: 262144k
      upperBound:
        cpu: 27854m
        memory: "545693548"
```

Deploying a Stateless App

1) Define a Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      run: my-app
  template:
    metadata:
      labels:
        run: my-app
    spec:
      containers:
      - name: hello-app
        image: us-docker.pkg.dev/google-samples/containers/gke/hello-app:1.0
```

Deploying a Stateless App

2) Create a Deployment

```
kubectl apply -f DEPLOYMENT_FILE
```

3) Inspect Deployment

```
kubectl describe deployment DEPLOYMENT_NAME
```

e tool to run commands against Kubernetes clusters

4) Update/Rollback Deployment

```
kubectl apply -f DEPLOYMENT_FILE
```

```
kubectl rollout undo deployment my-deployment
```

Expose Apps as Services: Service Types

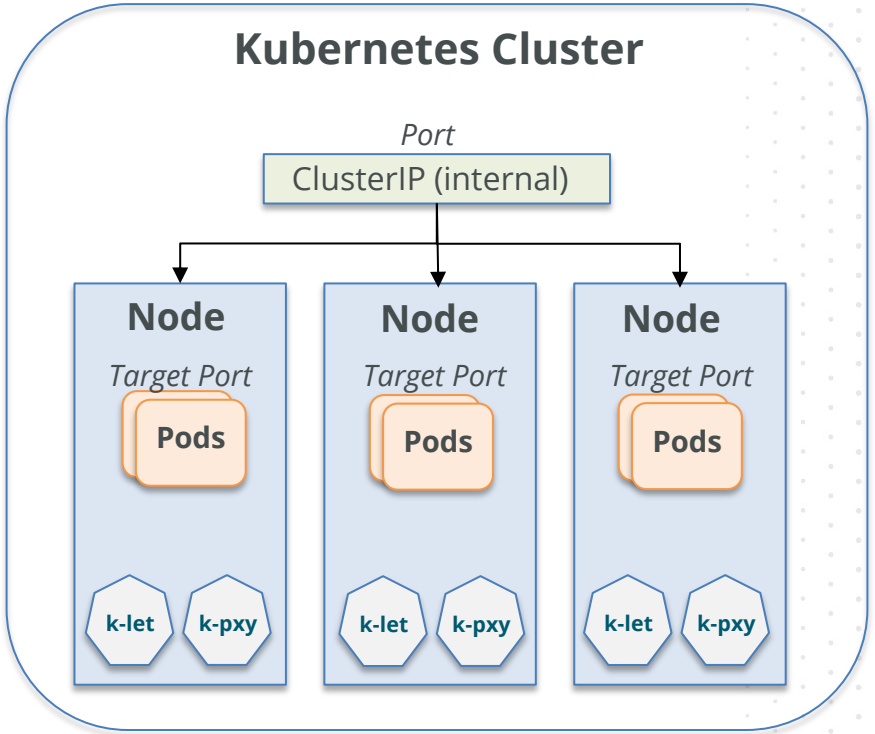
- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

Expose Apps as Services: Service Types

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

Exposes the Service on a cluster-internal IP

Service is only reachable from within the cluster



Example Service Manifest: ClusterIP

```
apiVersion: v1
kind: Service
metadata:
  name: my-cip-service
spec:
  type: ClusterIP
  # Uncomment the below line to create a Headless Service
  # clusterIP: None
  selector:
    app: my-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
```

Example Service Manifest: ClusterIP

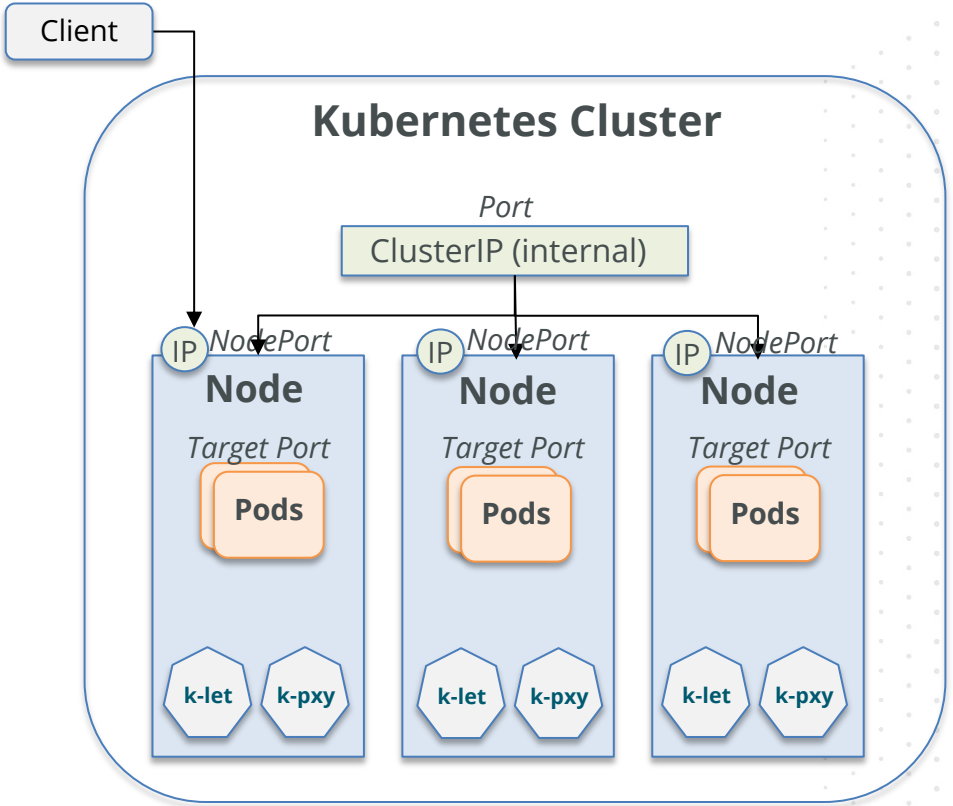
```
apiVersion: v1
kind: Service
metadata:
  name: my-cip-service
spec:
  type: ClusterIP
  # Uncomment the below line to create a Headless Service
  # clusterIP: None
  selector:
    app: my-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
```

Expose Apps as Services: Service Types

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

Extension of ClusterIP

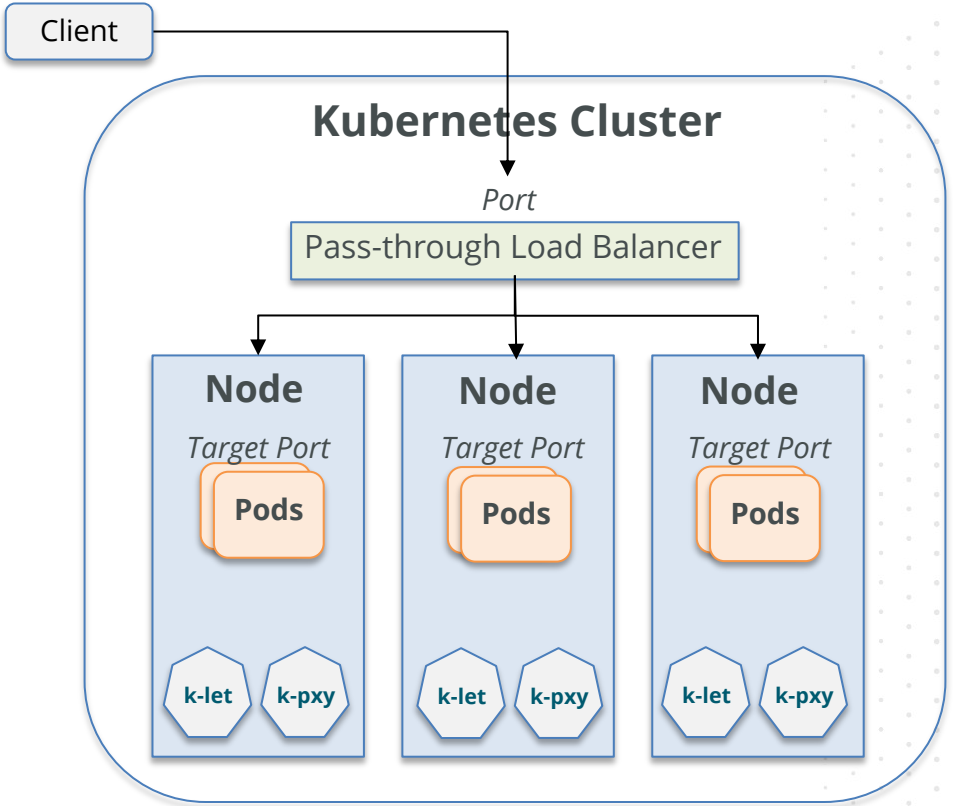
Kubernetes creates a *NodePort* value. The Service is reachable by using the IP address of any node along with the *NodePort* value.



Expose Apps as Services: Service Types

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

GKE configures an L4, pass-through load balancer (can be internal or external)



Example Service Manifest: LoadBalancer

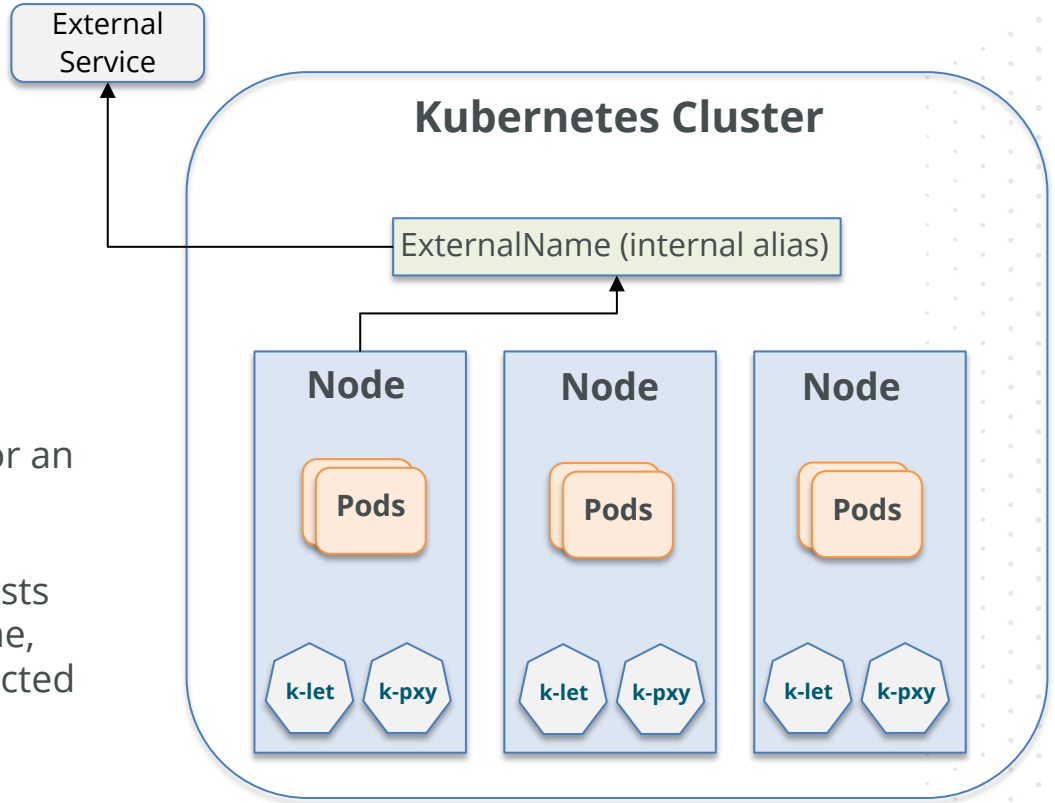
```
apiVersion: v1
kind: Service
metadata:
  name: ilb-service
  annotations:
    networking.gke.io/load-balancer-type: "Internal"
  labels:
    app: hello
spec:
  type: LoadBalancer
  selector:
    app: hello
  ports:
  - port: 80
    targetPort: 8080
    protocol: TCP
```

Expose Apps as Services: Service Types

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

Provides an internal alias for an external DNS name

Internal clients make requests using the internal DNS name, and the requests are redirected to the external name



Example Service Manifest: ExternalName

```
apiVersion: v1
kind: Service
metadata:
  name: my-xn-service
spec:
  type: ExternalName
  externalName: example.com
```

Load Balancing Traffic with Ingress

- An Ingress object defines rules for routing HTTP(S) traffic to applications running in a cluster
- Backed by the HTTP(S) Load Balancer (internal or external)

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: basic-ingress
spec:
  defaultBackend:
    service:
      name: web
      port:
        number: 8080
---
```

Load Balancing Traffic with Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: fanout-ingress
spec:
  rules:
  - http:
      paths:
      - path: /*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web
            port:
              number: 8080
      - path: /v2/*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web2
            port:
              number: 8080
---
```

Load Balancing Traffic with Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: fanout-ingress
spec:
  rules:
  - http:
      paths:
      - path: /*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web
            port:
              number: 8080
      - path: /v2/*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web2
            port:
              number: 8080
---
```

Load Balancing Traffic with Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: fanout-ingress
spec:
  rules:
  - http:
      paths:
      - path: /*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web
            port:
              number: 8080
      - path: /v2/*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web2
            port:
              number: 8080
---
```

Load Balancing Traffic with Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: fanout-ingress
spec:
  rules:
  - http:
      paths:
      - path: /*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web
            port:
              number: 8080
      - path: /v2/*
        pathType: ImplementationSpecific
        backend:
          service:
            name: web2
            port:
              number: 8080
---
```

Choosing How To Expose Apps

Component	Type	Scenarios
Service	ClusterIP	Internal (intra-cluster) access only
Service	NodePort	Need to access the service from outside the cluster For small number of nodes, no load-balancing needed
Service	LoadBalancer	Need to access the service from outside the cluster and to balance the load
Service	ExternalName	Need an internal DNS alias for an external (public) DNS name
Ingress	Internal	Access only on private IP address Need a proxy load balancer and HTTP(S) routing capabilities
Ingress	External	Publicly accessible Need a proxy load balancer and HTTP(S) routing capabilities

Isolate Clusters and Workloads

1) Create a private cluster

```
gcloud container clusters create private-cluster-3 \
    --create-subnetwork name=my-subnet-3 \
    --no-enable-master-authorized-networks \
    --enable-ip-alias \
    --enable-private-nodes \
    --master-ipv4-cidr 172.16.0.32/28
```

Isolate Clusters and Workloads

1) Create a private cluster

```
gcloud container clusters create private-cluster-3 \
    --create-subnetwork name=my-subnet-3 \
    --no-enable-master-authorized-networks \
    --enable-ip-alias \
    --enable-private-nodes \
    --master-ipv4-cidr 172.16.0.32/28
```

Disables authorized networks for the cluster, so any IP address can access the control plane

Isolate Clusters and Workloads

2) Restrict control plane access with *authorized networks*

```
gcloud container clusters create private-cluster-1 \
    --create-subnetwork name=my-subnet-1 \
    --enable-master-authorized-networks \
    --enable-ip-alias \
    --enable-private-nodes \
    --master-ipv4-cidr 172.16.0.0/28
```

At this point, these are the only IP addresses that have access to the cluster control plane:

- The primary range of my-subnet-1.
- The secondary range used for Pods.

Isolate Clusters and Workloads

2) Restrict control plane access with *authorized networks*

Specify authorized networks:

```
gcloud container clusters update private-cluster-1 \
    --enable-master-authorized-networks \
    --master-authorized-networks 203.0.113.0/29
```

Deploying Autopilot GKE Clusters

Create an Autopilot cluster

```
gcloud container clusters create-auto CLUSTER_NAME \
    --region REGION \
    --project=PROJECT_ID
```

- GKE provisions and manages the nodes and node pools
- Clusters are pre-configured with an optimized configuration for production workloads
- Comes with an SLA that covers both the control plane and your Pods

Demo: Configuring Network, Storage, and Compute

1. Create custom VPC with two subnets (**frontend-subnet** and **backend-subnet**)
2. Enable **Private Google Access** on backend-subnet
3. Create a frontend **Compute Engine instance** with an **external IP address** and tagged "**web**"
4. Create a backend **Compute Engine instance without external IP address** and tagged "**app**"
5. Create a regional Cloud Storage bucket

Demo: Configuring Network, Storage, and Compute

https://github.com/vmehmeri/gcp-professional-cloud-architect

Questions Breakdown

Your company wants to adopt a hybrid strategy with some workloads developed on-premises, and other workloads on GCP. Because of the inter-dependence of these workloads, you're responsible for configuring an IPSec VPN tunnel between the on-premises network and a newly created VPC network. What should you do?

A. Configure subnets in the VPC network with the same ip range as you use on-premises. Configure Cloud VPN with static routing.
B. Configure subnets in the VPC network with the same ip range as you use on-premises. Configure Cloud VPN with dynamic routing.
C. Configure subnets in the VPC network with ip ranges that don't overlap with the ones you use on-premises. Configure Cloud VPN with static routing.
D. Configure subnets in the VPC network with ip ranges that don't overlap with the ones you use on-premises. Configure Cloud VPN with dynamic routing.

Questions Breakdown

Your company wants to adopt a **hybrid** strategy with some workloads developed on-premises, and other workloads on GCP. Because of the inter-dependence of these workloads, you're responsible for configuring an **IPSec VPN tunnel** between the on-premises network and a **newly created VPC network**. What should you do?

A. Configure subnets in the VPC network with the same IP range as you use on-premises. Configure Cloud VPN with static routing.

B. Configure subnets in the VPC network with the same IP range as you use on-premises. Configure Cloud VPN with dynamic routing.

C. Configure subnets in the VPC network with IP ranges that don't overlap with the ones you use on-premises. Configure Cloud VPN with static routing.

D. Configure subnets in the VPC network with IP ranges that don't overlap with the ones you use on-premises. Configure Cloud VPN with dynamic routing.

Questions Breakdown

You set up an HTTP(S) load balancer and health check with a managed instance group (MIG) configured as a backend service. You noticed that, after deploying the load balancer, MIG instances are being terminated and re-launched every 60 seconds. What should you do?

A. Configure an ingress VPC firewall rule to allow source traffic to reach the HTTP(S) load balancer.

B. Configure an ingress VPC firewall rule to allow load balancer health checks to reach the instances.

C. Configure a public IP address on each instance and an ingress VPC firewall rule to allow source traffic to reach the instances.

D. Ensure the default “allow all” egress rule is enforced in the VPC firewall.

Questions Breakdown

You set up an **HTTP(S) load balancer** and **health check** with a managed instance group (MIG) configured as a backend service. You noticed that, after deploying the load balancer, **MIG instances are being terminated and re-launched every 60 seconds**. What should you do?

A. Configure an ingress VPC firewall rule to allow source traffic to reach the HTTP(S) load balancer.

B. Configure an ingress VPC firewall rule to allow load balancer health checks to reach the instances.

C. Configure a public IP address on each instance and an ingress VPC firewall rule to allow source traffic to reach the instances.

D. Ensure the default “allow all” egress rule is enforced in the VPC firewall.

Questions Breakdown

You are responsible for deploying a container-based secure web application to Google Kubernetes Engine (GKE). You need to ensure the application scales automatically and that traffic can be served to end users over HTTPS. How should you configure the GKE deployment?

A. Enable cluster autoscaling and use the Horizontal Pod Autoscaler. Define a Service resource of type *LoadBalancer*.

B. Enable cluster autoscaling and use the Horizontal Pod Autoscaler. Define an Ingress resource.

C. Configure an autoscaling policy for the underlying Compute Engine instance group. Define an Ingress resource.

D. Configure an autoscaling policy for the underlying Compute Engine instance group. Define a Service resource of type *LoadBalancer*.

Questions Breakdown

You are responsible for deploying a container-based secure web application to **Google Kubernetes Engine (GKE)**. You need to ensure the application **scales automatically** and that traffic can be served to end users **over HTTPS**. How should you configure the GKE deployment?

A. Enable cluster autoscaling and use the Horizontal Pod Autoscaler. Define a Service resource of type *LoadBalancer*.

B. Enable cluster autoscaling and use the Horizontal Pod Autoscaler. Define an Ingress resource.

C. Configure an autoscaling policy for the underlying Compute Engine instance group. Define an Ingress resource.

D. Configure an autoscaling policy for the underlying Compute Engine instance group. Define a Service resource of type *LoadBalancer*.

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

The Principle of Least Privilege

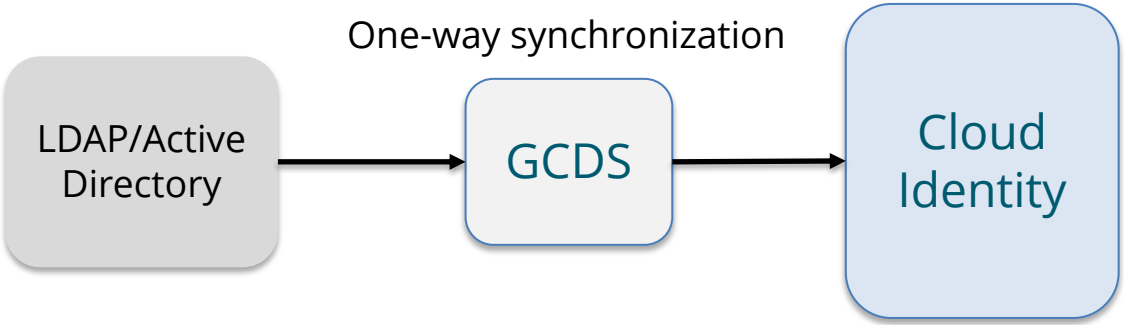
A resource/individual should only have access to the exact resource(s) it needs to function

Authentication on GCP: Cloud Identity

- Identity as a Service (IDaaS) solution
- Single pane of glass to manage users' identities as well as devices and apps
- Single sign-on (SSO), multi-factor authentication (MFA)
- Endpoint management

Cloud Identity: Directory Sync

Cloud Identity supports extending an on-premises directory to the cloud with **Google Cloud Directory Sync** (**GCDS**)



Authorization on GCP: Cloud IAM

- Consistent access control interface for *all* Google Cloud services
- An IAM policy can be broken into three components:

Cloud IAM Roles

- **Basic roles**: Owner, Editor, Viewer
- **Predefined roles**: service-specific roles (example: Pub/Sub Subscriber)
- **Custom roles**: based on user-specified list of permissions (can only be set at the *project* or *organization* level)

Example custom role (YAML)

```
title: "My Company Admin"
description: "My custom role
description."
stage: "ALPHA"
includedPermissions:
- iam.roles.get
- iam.roles.list
```

Role Recommendations

- One of the recommendations from **Recommender** service
- Helps you identify and remove excess permissions using *policy insights*
- It may recommend that you create a new custom role

Service Accounts

- Both a *resource* and an *identity*
  - You can grant a role to a service account (identity)
  - You can grant roles to users to access (impersonate) or manage the service account (resource)
- Associated with a public/private RSA key pair (no password)
- You can prevent the creation of service accounts with an organization policy constraint

Service Accounts: Example

Enforcing Least Privilege: Best Practices

- Do not grant basic roles (“Owner”, ”Editor”, “Viewer”)
- Use role recommendations and grant roles at the smallest scope needed
- Grant roles to Google groups instead of individuals
- Treat each component of your application as a separate trust boundary
- Be cautious when granting the Service Account User role
- Rotate service account keys
- Regularly check Cloud Audit Logs and audit changes to policies

Separation of Duties (SoD)

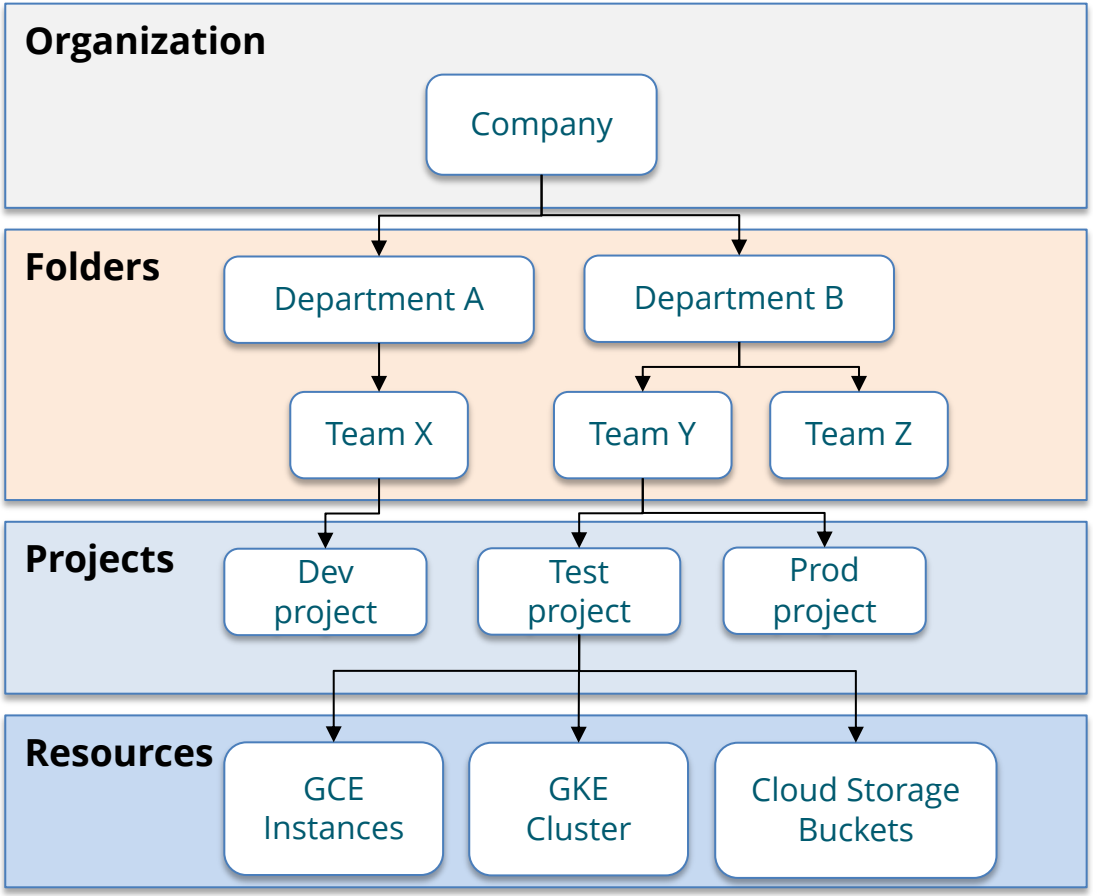
- Ensuring no one individual has all necessary permissions to complete a malicious action
- Highly privileged rights should be spread among multiple people

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

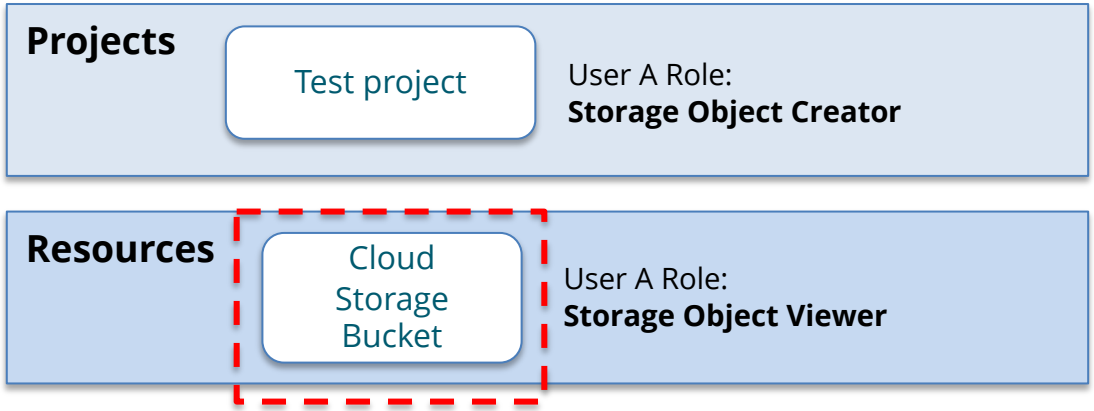
Resource Hierarchy



Using Resource Hierarchy for Access Control

- IAM roles can be set at the organization level, folder level, project level, or (in some cases) the resource level
- IAM roles granted at a level are inherited by all resources under that level
- The effective *allow* policy for a resource is the union of the inherited *allow* policy and the resource *allow* policy
- **There is no way to explicitly remove a permission for a lower-level resource that is granted at a higher level in the resource hierarchy**

Example Policy Inheritance



Allow policy = Parent policy **U** Resource policy

Storage Object Creator **U** Storage Object Viewer

Final Access scope = **Storage Object Creator**

Best Practices

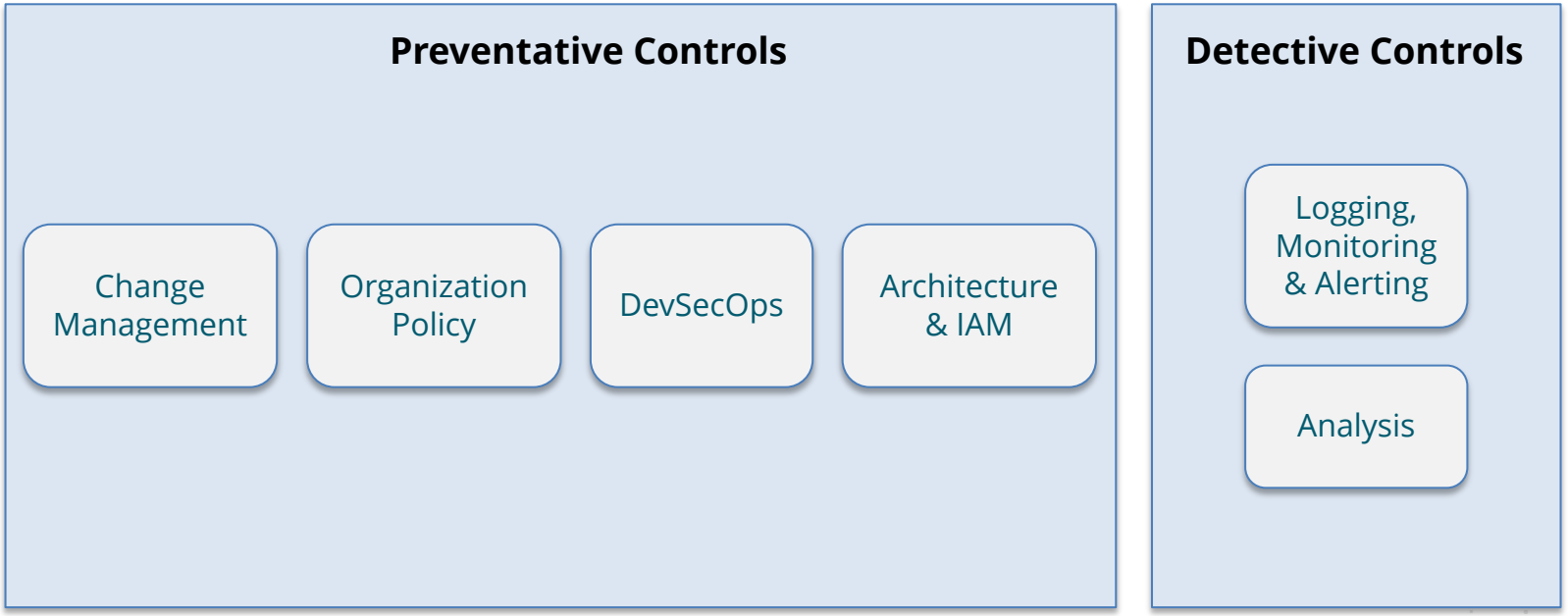
- Mirror resource hierarchy structure to organization structure
- Use projects to group resources that share the same trust boundary
- On every project, ensure that at least two principals have the Owner role (roles/owner)
- Limit project creation by granting the Project Creator role to a single group

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Security Controls



Organization Policy Service

- Centralized and programmatic control over cloud resources
- Enforce constraints across the entire resource hierarchy
- Define and establish guardrails for teams to stay within compliance boundaries

Example Policy Constraints

Constraint	Description	Example
Google Cloud Platform – Resource Location Restriction	Defines the set of locations where location-based GCP resources can be created	*in:us-locations*
Restrict Resource Service Usage	Defines the set of Google Cloud resource services that can be used within an organization, folder, or project	*is:compute.googleapis.com, is:storage.googleapis.com*

Example Policy Constraints

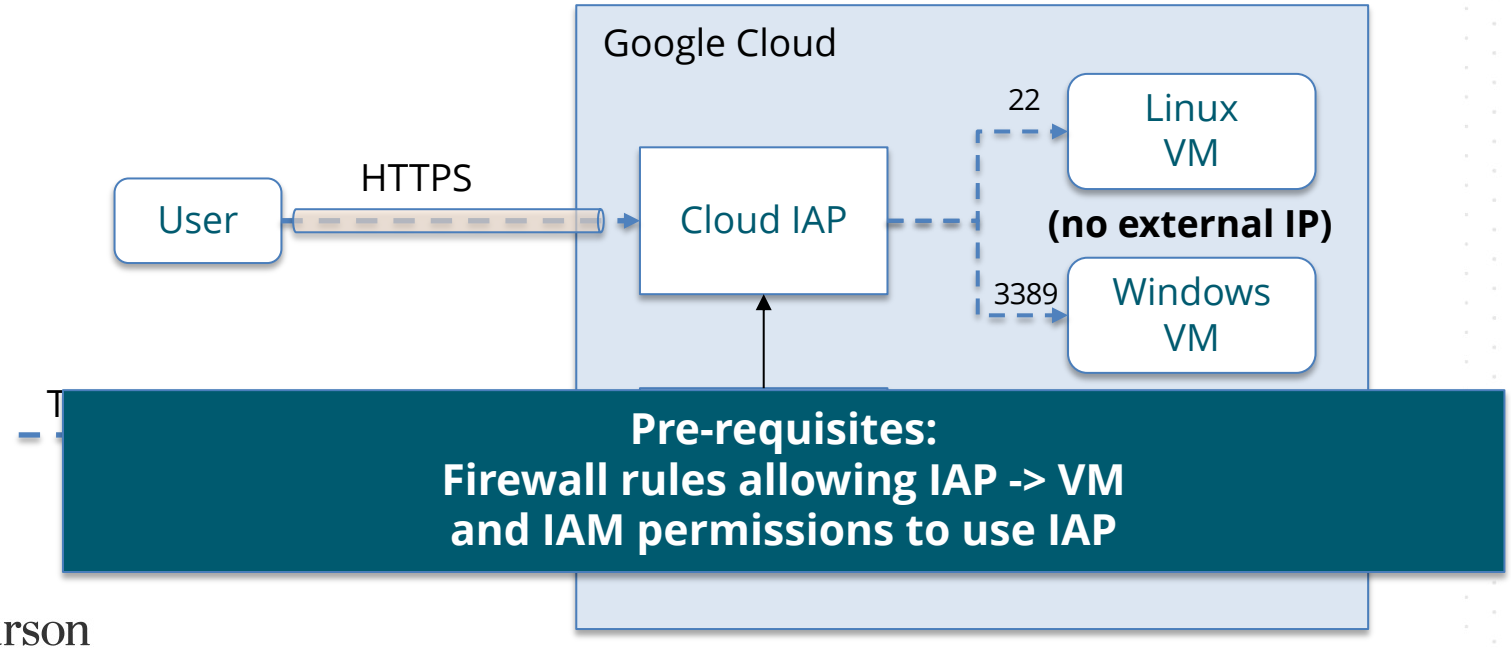
Constraint	Description	Example
Cloud KMS – Restrict which KMS CryptoKey types may be created	Defines the Cloud KMS key types which may be created under a given hierarchy node	*HSM*
Cloud SQL – Restrict Authorized Networks on Cloud SQL instances	Restricts adding Authorized Networks for unproxied database access to Cloud SQL instances	*True*
Compute Engine – Define allowed external IPs for VM instances	Defines the set of Compute Engine VM instances that are allowed to use external IP addresses.	*projects/myproject1234/zones/us-east1-c/instances/INSTANCE1*
Cloud SQL – Restrict Public IP access on Cloud SQL instances	Restricts the configuration of public IPs on Cloud SQL instances	*True*

Identity-Aware Proxy (IAP)

- Manage access to applications in App Engine, Compute Engine, and GKE
- Central authorization layer for applications
- Application-level access control model

Connecting to VMs using IAP TCP Forwarding

For VMs without external IP address, you can enable **Identity-Aware Proxy TCP forwarding**



Detective Controls

Cloud Audit Logs

- Contains the logs that answer the question "*Who did what, where, and when?*"
  - Admin Activity logs (**can't be disabled**)
  - System Event logs (**can't be disabled**)
  - Data Access logs (must be enabled)

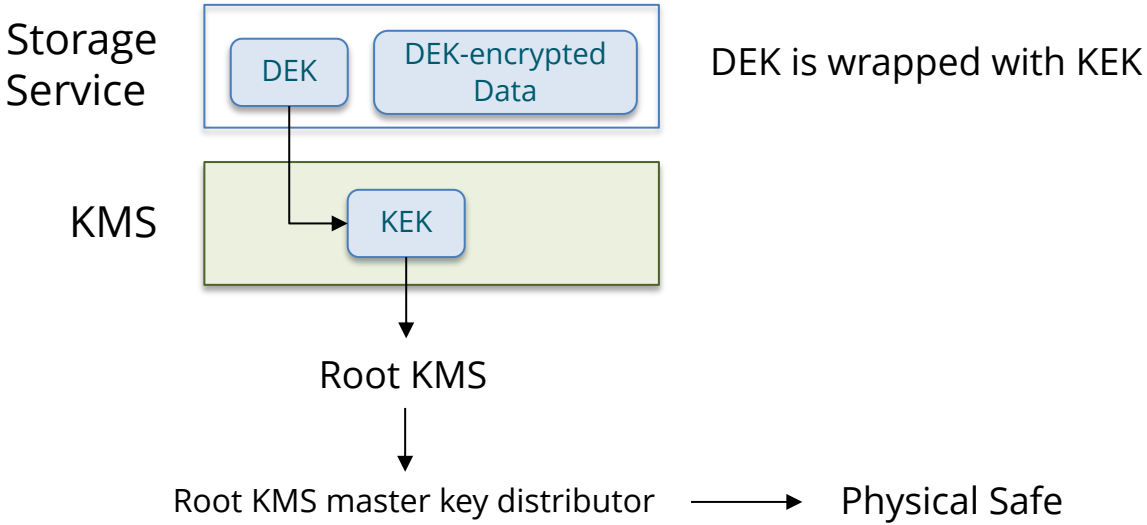
Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Data Encryption on GCP

Google Cloud always encrypts data on the server side (**AES256**), before it is written to disk



Data Encryption on GCP: CMEK

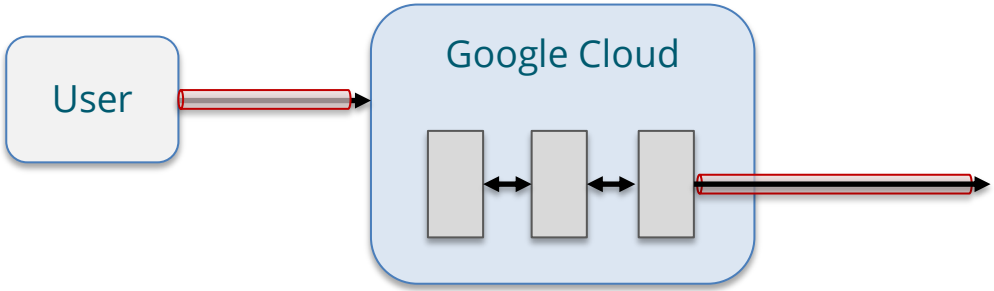
- With customer-managed encryption keys (CMEK), the KEK is under your control
- Using CMEK doesn't necessarily provide more security than Google's default encryption
- What you do get is more control:
  - You can disable the key at any time, preventing decryption
  - You can meet specific locality or residency requirements
  - You can automatically or manually rotate the keys at any frequency you need

Enabling CMEK on Services


```
*roles/cloudkms.cryptoKeyEncrypterDecrypter
```

Data Security in Transit

- Google applies TLS-based encryption in transit by default on all Google Cloud services
- Google encrypts and authenticates data in transit at one or more network layers when data move outside GCP's network boundaries



Data Security in Transit: Additional Options

IPSec VPN Tunnel

SSL Certificates

Anthos Service Mesh

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Certificate Manager

- Provision and manage SSL certificates for global external HTTP(S) load balancers
- Requires **Premium Network Service Tier**

Customer-supplied Encryption Keys (CSEK)

- Only for Cloud Storage
- You provide your own **AES256** key, encoded in standard **Base64**
- You need to supply the key in every request

Client-side Encryption

- Encryption you perform prior to sending data
- You must create and manage your own encryption keys, use your own tools, etc.
- You can use Google's open-source cryptographic SDK, **Tink**

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Legislation Considerations

- Health Insurance Portability and Accountability Act (HIPAA) – US
- General Data Protection Regulation (GDPR) - Europe

- Payment Card Industry (PCI) Data Security Standards (DSS)
- ISO 27001, 27017, 27018
- SOC 2/3
- Federal Risk and Authorization Management Program (FedRAMP)

HIPAA

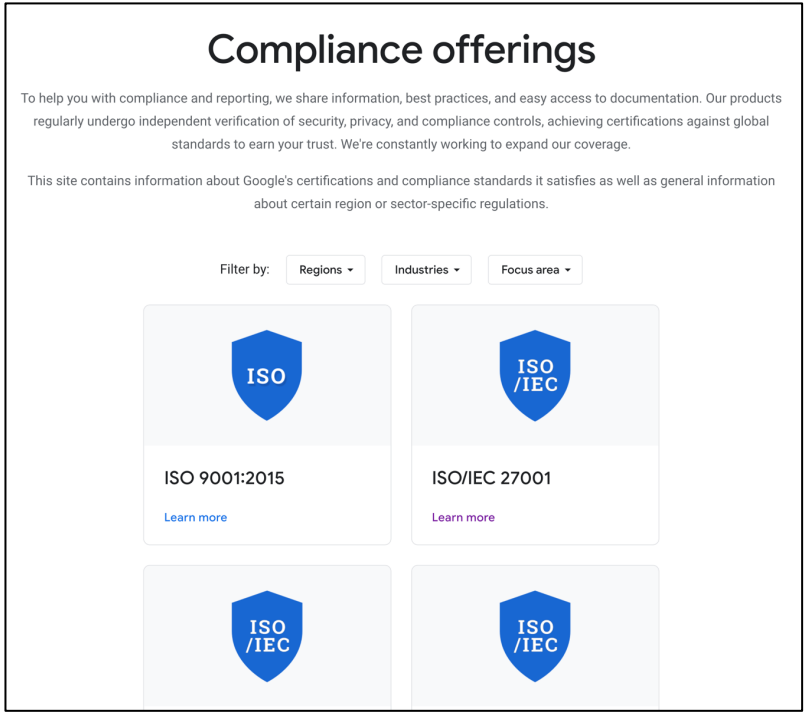
- There is no recognized certification for HIPAA compliance
- Customers that are subject to HIPAA must review and accept Google's **Business Associate Agreement** (**BAA**)
- The BAA covers the majority, but not all GCP services

HIPAA: Customer Responsibilities

- Execute a **Google Cloud BAA**
- Ensure that Google Cloud products not explicitly covered by the BAA aren't used when working with Protected Health Information (PHI)
- Ensure that applications are properly configured and secured according to HIPAA requirements

GCP Industry Certifications

- **Compliance resource center** is the go-to place for compliance resources and offerings
- **Compliance Reports Manager** is the go-to place for downloading reports



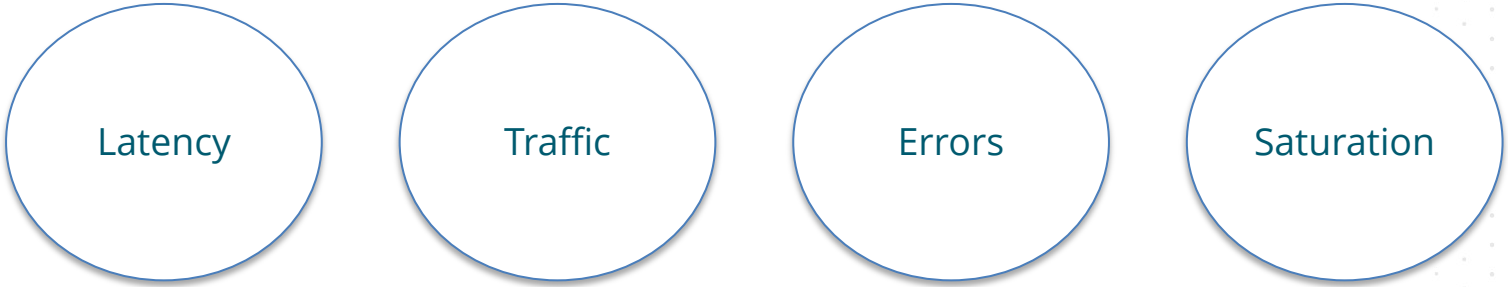
Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Monitoring The Four Golden Signals

The **SRE** book defines the following as the “four golden signals” of monitoring:

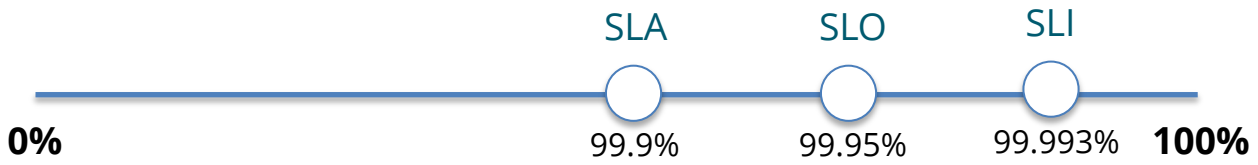


source: https://sre.google/books/

SLIs, SLOs, and SLAs

- **SLI**: Quantifiable measure of service reliability
- **SLO**: Reliability target for an SLI
- **SLA**: Agreed-upon reliability target as a commitment to paying customers

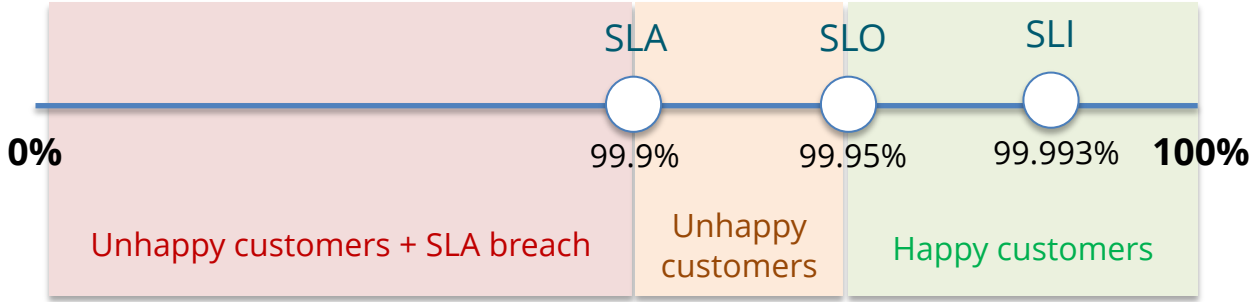
Example: service uptime



SLIs, SLOs, and SLAs

- **SLI**: Quantifiable measure of service reliability
- **SLO**: Reliability target for an SLI
- **SLA**: Agreed-upon reliability target as a commitment to paying customers

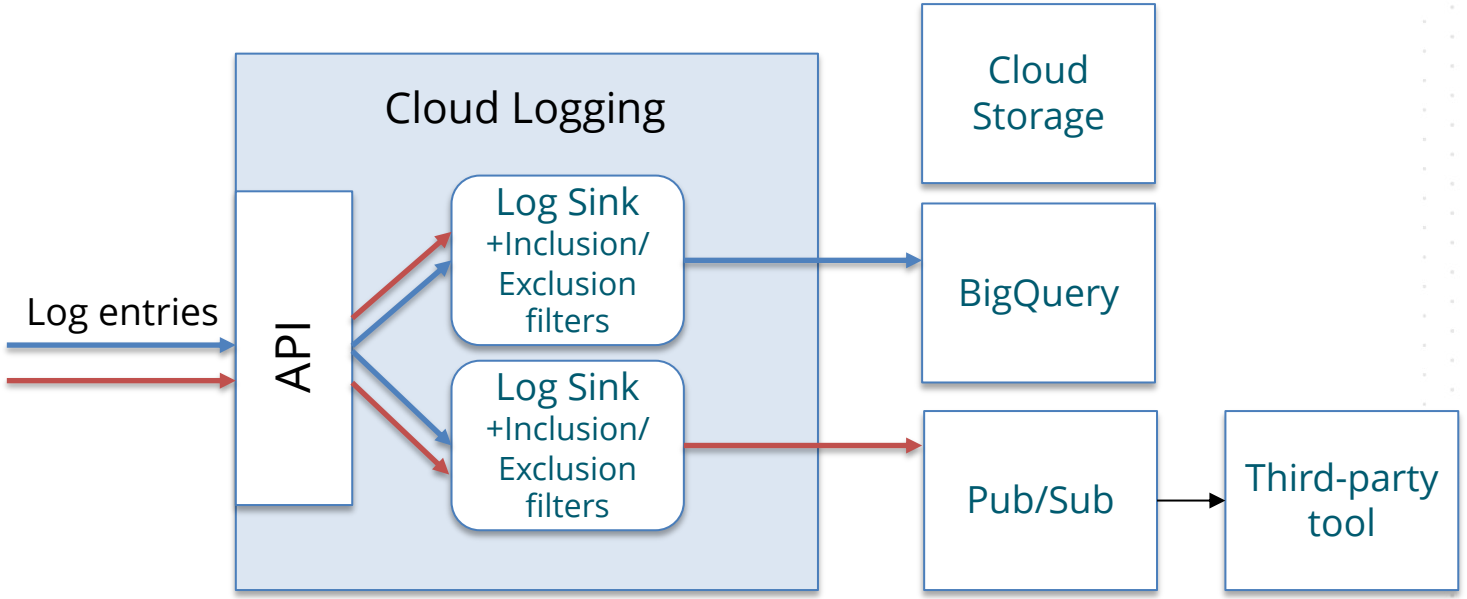
Example: service uptime



Google Cloud Operations Suite

Cloud Logging: Sinks

Routing behavior is controlled with inclusion filters and exclusion filters



Common Use Case: Export for Compliance

1) Create a Cloud Storage bucket
2) (Optional) Configure object lifecycle management
3) Configure sink

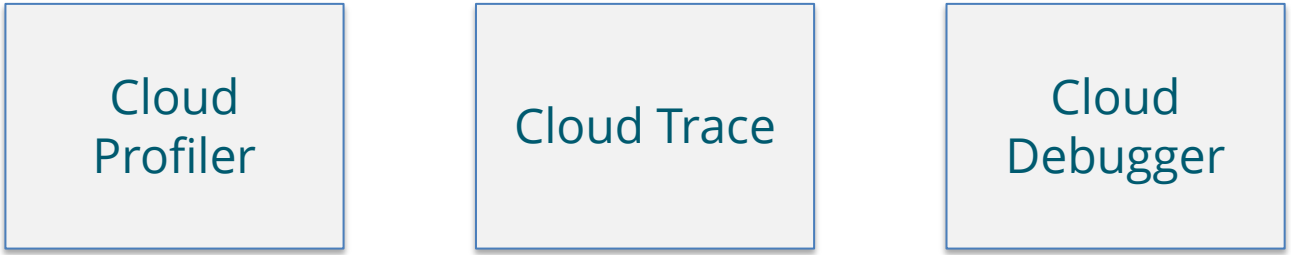
```
gcloud logging sinks create gcp_logging_sink_gcs \
    storage.googleapis.com/gcp-logging-export-000100011000 \
    --log-filter='logName: "/logs/cloudaudit.googleapis.com" \
    --include-children   \
    --organization=324989855333
```

Segment 4: Designing for Observability, Security, and Compliance

Objectives

- Identity and Access Management (IAM)
- Separation of duties
- Resource hierarchy
- Security controls
- Securing data at rest and in transit
- Secrets and certificate management
- Compliance considerations
- Cloud Monitoring and Logging
- Application performance monitoring

Application Performance Monitoring

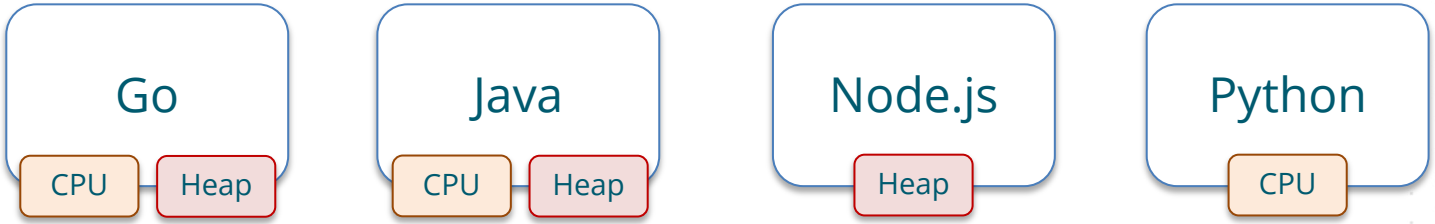


Cloud Profiler

- Low-overhead profile that continuously gathers **CPU usage** and **memory allocation** from applications
- Attributes information to the source code that generated it
- Consists of a profiling agent and a console interface

Cloud Profiler Use Cases

- Identify application performance bottlenecks
- Find ways to make an application faster and more efficient
- Alleviate the need to develop accurate predictive load tests before production



Cloud Trace

- Distributed tracing system that collects **latency** data from applications in near real-time
- Helps answer questions :
  - What is the overall latency of requests?
  - How long does it take to handle a given request?
  - Why is it taking so long to handle a request?
  - Why do some requests take longer than others?
  - Has latency increased or decreased over time?
  - What are my application’s dependencies?

Cloud Trace

- Automatically enabled for **App Engine** (Java 8, Python 2, and PHP 5), **Cloud Functions**, and **Cloud Run**
- Enable it by instrumenting applications with the **Cloud Trace API**
- **OpenCensus** (now part of **OpenTelemetry**) is an open-source library that is simpler to use and **recommended** if available for your programming language

Cloud Debugger

- Inspects the state of an application in real time without impacting performance
- Requires code instrumentation
- Available for: **Java**, **Python**, **Go**, **Node.js**, **Ruby**, **.NET**, and **PHP**

Deprecated and will be shutdown May 31, 2023

Snapshot Debugger

- Google's recommended replacement for Cloud Debugger
- Open source and works in a very similar way

Currently in preview

Demo: Designing for Observability, Security, and Compliance

Requirements:

- Firewall rules with minimal required access
- Least-privilege access permissions
- Restrict use of external IP addresses
- Configure a perimeter around sensitive data

Demo: Designing for Observability, Security, and Compliance

https://github.com/vmehmeri/gcp-professional-cloud-architect

Questions Breakdown

Your company wants to enforce locality requirements by limiting the physical locations where cloud resources can be created. What should you do?

A. Configure an organization policy with a constraint on resources location.

B. Configure an organization policy with a constraint on resource service usage.

C. Configure the organization's resource quotas and set all quotas to zero under the disallowed regions.

D. Ensure all IAM roles that create resources include a condition specifying Google Cloud regions.

Questions Breakdown

Your company wants to enforce locality requirements by **limiting the physical locations where cloud resources can be created**. What should you do?

A. Configure an organization policy with a constraint on resources location.

B. Configure an organization policy with a constraint on resource service usage.

C. Configure the organization's resource quotas and set all quotas to zero under the disallowed regions.

D. Ensure all IAM roles that create resources include a condition specifying Google Cloud regions.

Questions Breakdown

Your company's security requirements dictate that all data must be encrypted at rest with encryption keys that can be manually rotated by the security team when needed. How should you configure the storage services on GCP?

A. Ensure that all storage services are created using customer-supplied encryption keys (CSEKs).
B. Ensure that only storage services with Google-managed encryption keys (GMEKs) are used.
C. Generate an AES-256 key using a third-party tool. Store the key in a Cloud Storage bucket that the security team has access to. Use this key to encrypt all data.
D. Ensure that all storage services are created using customer-managed encryption keys (CMEKs).

Questions Breakdown

Your company's security requirements dictate that **all data must be encrypted at rest** with encryption keys that can be **manually rotated** by the security team when needed. How should you configure the storage services on GCP?

A. Ensure that all storage services are created using customer-supplied encryption keys (CSEKs).

B. Ensure that only storage services with Google-managed encryption keys (GMEKs) are used.

C. Generate an AES-256 key using a third-party tool. Store the key in a Cloud Storage bucket that the security team has access to. Use this key to encrypt all data.

D. Ensure that all storage services are created using customer-managed encryption keys (CMEKs).

Questions Breakdown

You're managing a secured Google Cloud project with several Compute Engine instances. No instances in the project can have a public IP address and there is no hybrid connectivity between Google Cloud and your company's office. You need to connect via SSH into a specific machine for troubleshooting. What should you do?

A. Delete any organization policy constraints in place that prevent the assignment of external IP addresses to Compute Engine instances. Assign an external IP address to the instance during troubleshooting.

B. Configure Cloud NAT on the instance's subnet and a VPC Firewall rule allowing SSH connections from your IP address. Connect via SSH using Cloud NAT's IP address.

C. Configure Identity-Aware Proxy (IAP) with TCP forwarding and a VPC Firewall rule allowing SSH connections from the proxy. Connect to the instance using *gcloud.*

D. Create a bastion host in the same VPC network and a VPC Firewall rule allowing SSH connections from the bastion host. Connect to the instance via the bastion host using an SSH client.

Questions Breakdown

You're managing a secured Google Cloud project with several Compute Engine instances. **No instances in the project can have a public IP address** and there is **no hybrid connectivity** between Google Cloud and your company's office. **You need to connect via SSH** into a specific machine for troubleshooting. What should you do?

A. Delete any organization policy constraints in place that prevent the assignment of external IP addresses to Compute Engine instances. Assign an external IP address to the instance during troubleshooting.

B. Configure Cloud NAT on the instance's subnet and a VPC Firewall rule allowing SSH connections from your IP address. Connect via SSH using Cloud NAT's IP address.

C. Configure Identity-Aware Proxy (IAP) with TCP forwarding and a VPC Firewall rule allowing SSH connections from the proxy. Connect to the instance using *gcloud.*

D. Create a bastion host in the same VPC network and a VPC Firewall rule allowing SSH connections from the bastion host. Connect to the instance via the bastion host using an SSH client.