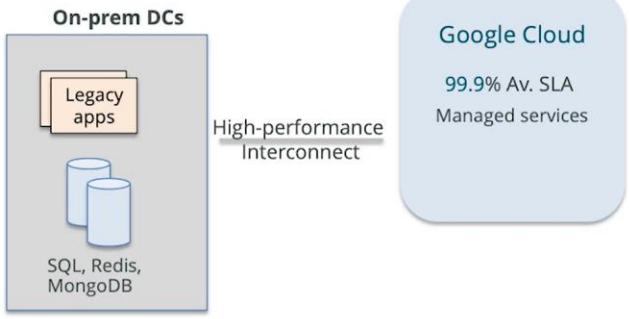
A Google Cloud Certified Professional Cloud Architect:

- **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

Case Study: Medical Software

Case Study Summary

- SaaS Product, web-based, Kubernetes
- MySQL+MS SQL Server, Redis, MongoDB
- Hosted on multiple on-prem locations
- Legacy integrations with partners – remain on-prem
- Requirements: reduce latency, low infrastructure admin cost, centralized monitoring, high-performance interconnect.



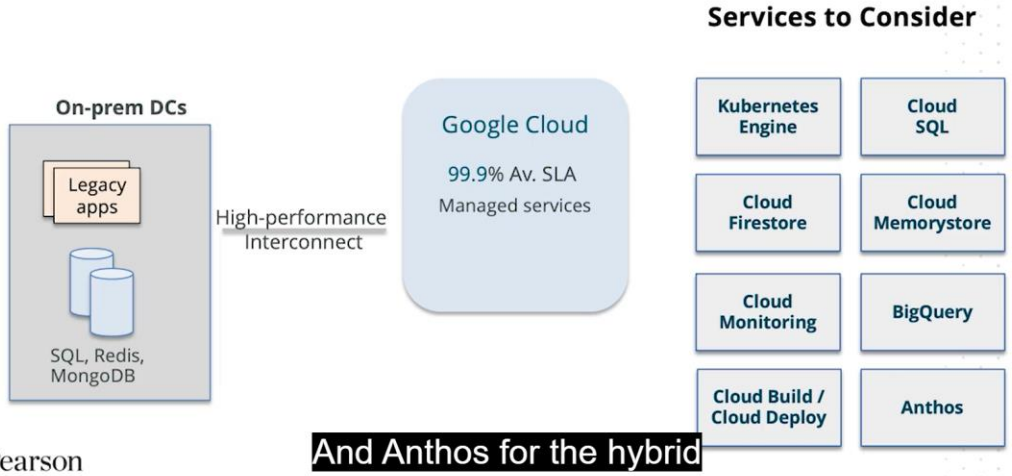
Pearson

where this company has systems

For above scanario , components that can be used in gcp are –
Kubernetes engines , cloud sql – managed db offering , cloud firestore – nosql db , cloud memorystore for redis , cloud build deploy – for cicd , anhos – for hybrid and multi cloud application management .

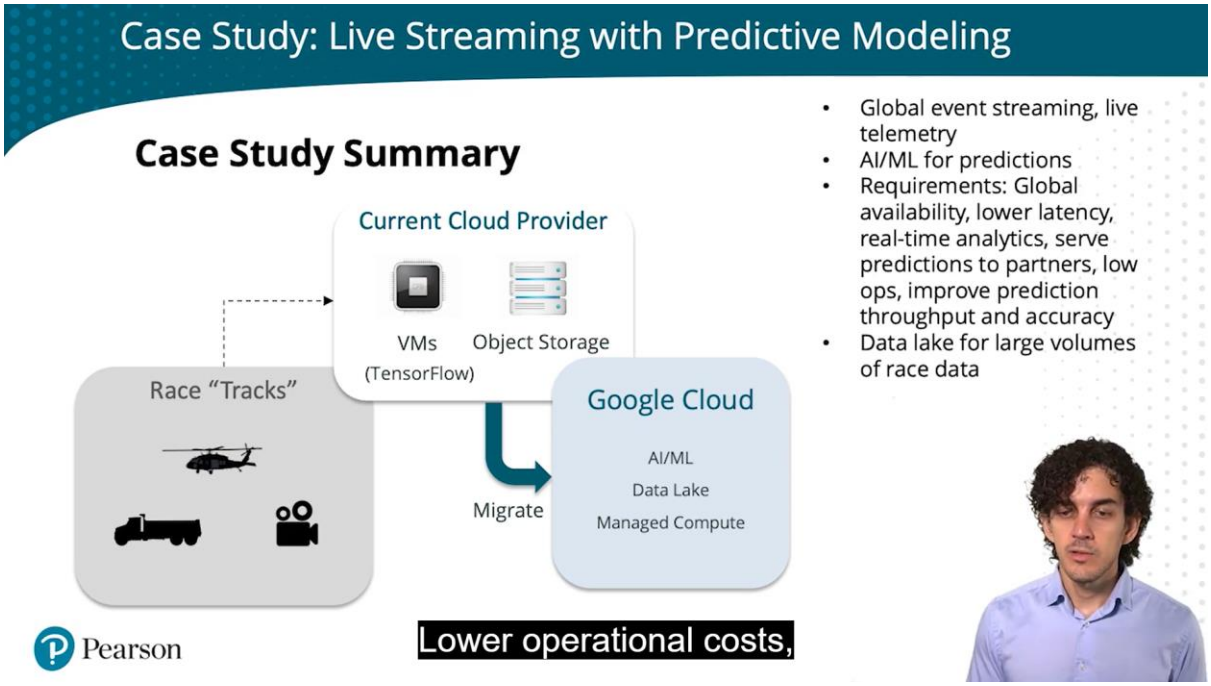
Case Study: Medical Software

Case Study Summary

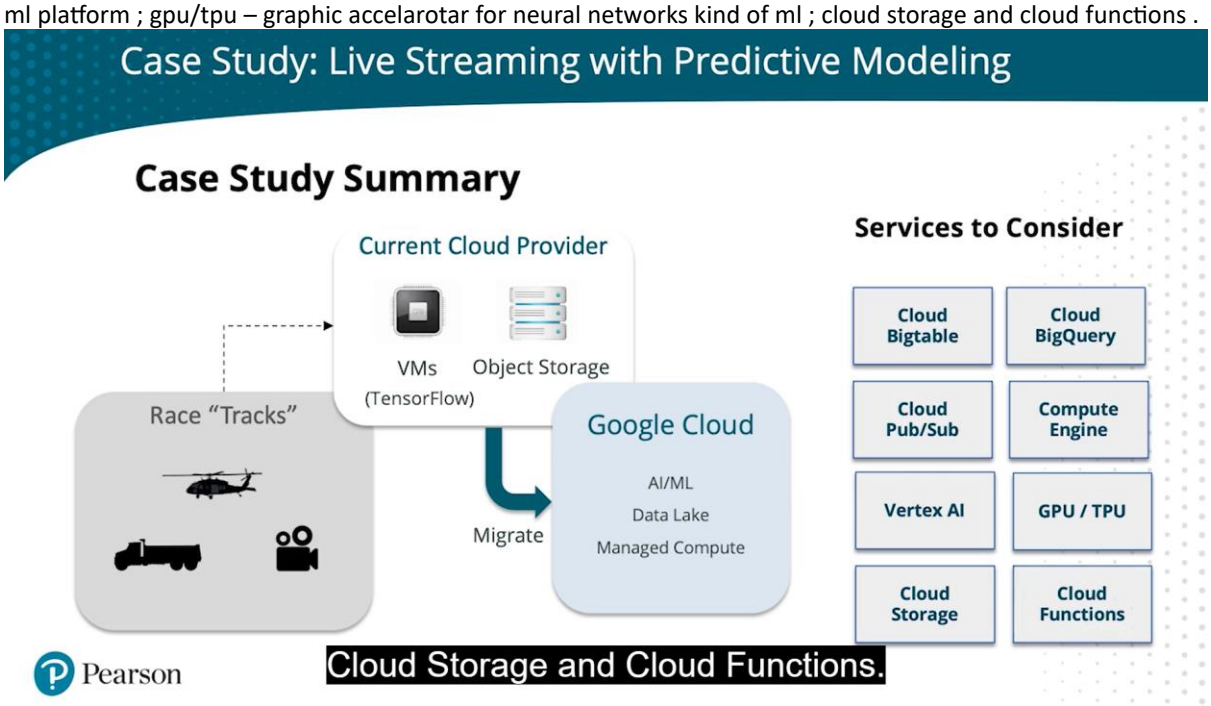


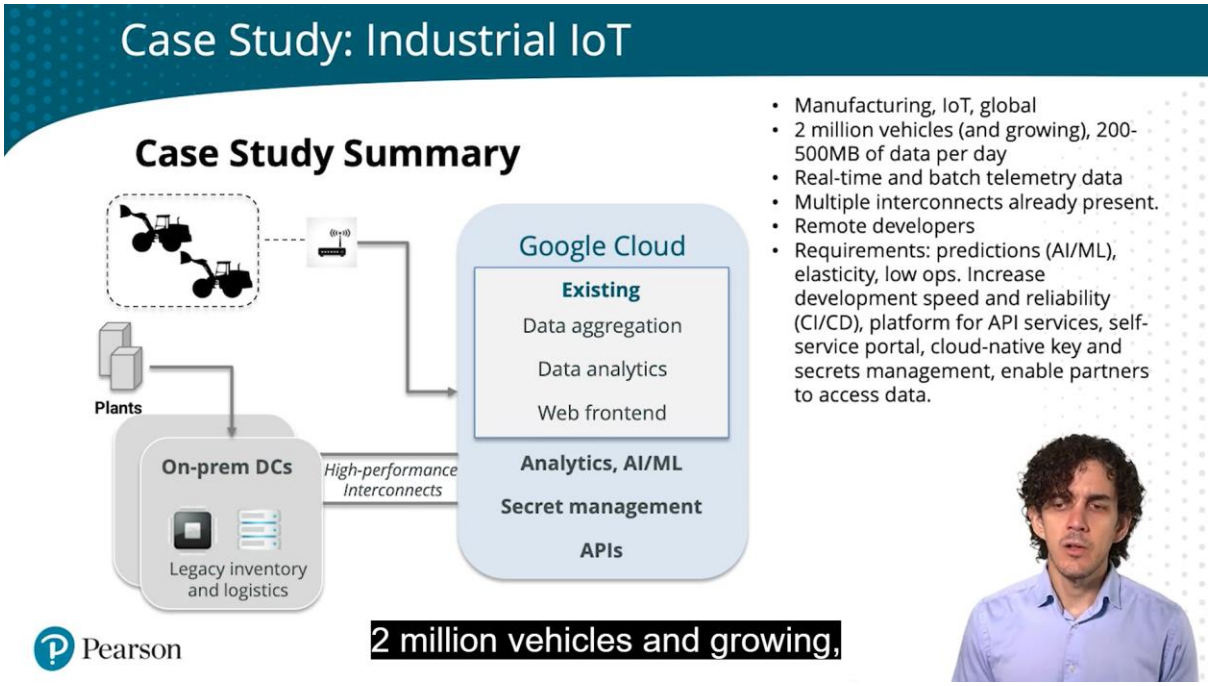
Pearson

And Anthos for the hybrid

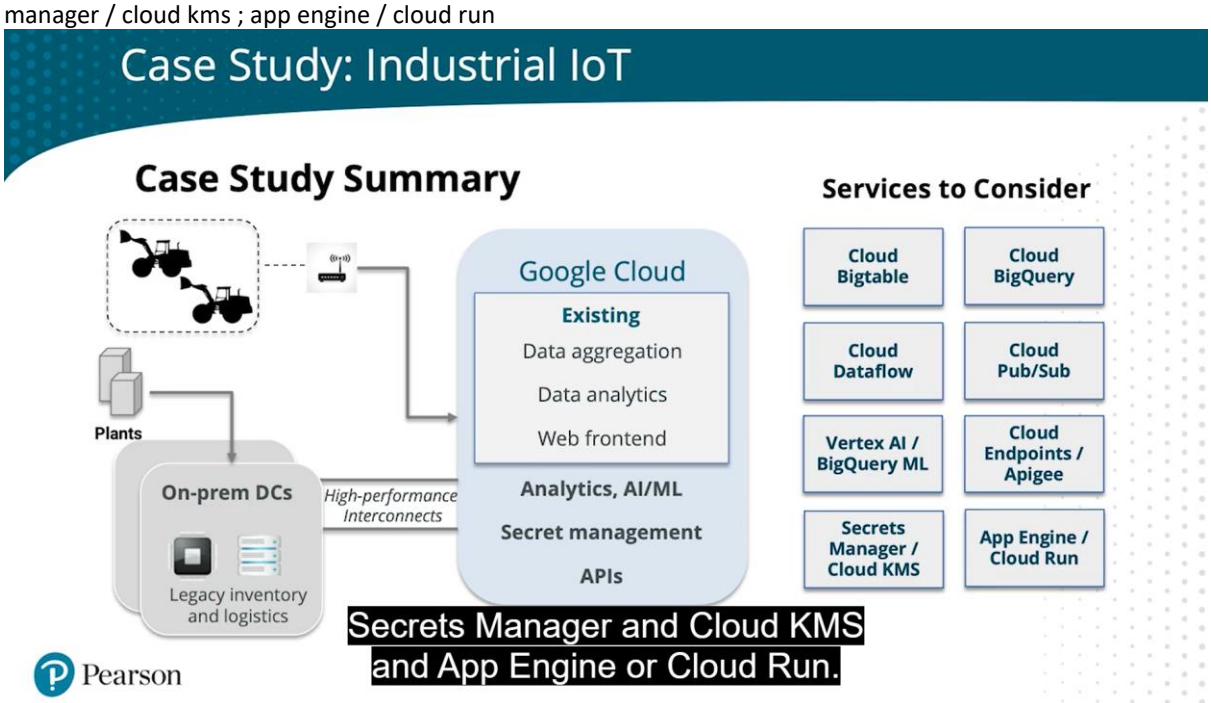


Cloud big table – for time series , high throughput insert ; cloud bigquery for analytics component ; cloud pub/sub – for streaming ingestion of data ; comput engine – for lift and shift migration of vm ; vertex ai – gcp ml platform ; gpu/tpu – graphic accelarotar for neural networks kind of ml ; cloud storage and cloud functions .





Cloud bigtable ; cloud big query ; cloud dataflow – managed apache beam pipeline runner – it works very well for etl and data transformation ; cloud pub/sub ; vertex ai /bigquery ml ; cloud endpoint /apigee ; secret manager / cloud kms ; app engine / cloud run



Case Study: Mobile Gaming Platforms

Case Study Summary

- Online mobile gaming, global, near real-time leaderboard
- Fully on Google Cloud, but still running mostly VMs. Separate dev/test environments
- Plan to use GKE, Global Load Balancer, and multi-region Spanner for leaderboard
- Each game in an isolated project
- Legacy games bundled in single project
- Requirements: Multi-region, low latency (top priority), elastic, rapid development, multiple gaming platforms, managed services and pooled resources, GPU processing, game activity logs in structured files for batch analysis.

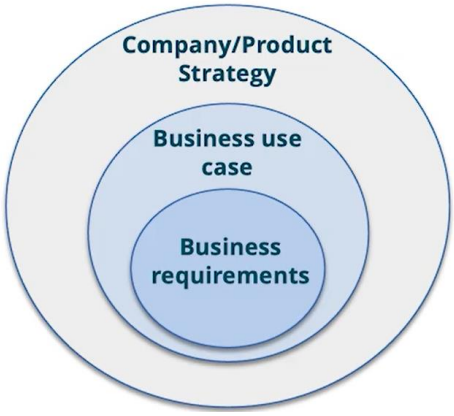
Pearson

that is building mobile gaming platforms.

This is case of application modernization – they need multi region environment with low latency ; elasticity of infrastructure , rapid development , multiple gaming platforms – one for each region ;gpu processing ; game activity logs to be in structural file for batch analysis of logs

GCP services can be used : kubernetes engine ; cloud spanner ; big query for analytics part ; cloud gpu ; cloud storage ; cloud dataflow – for streaming and batch data pipelines ; cloud firestore for nosql db ; cloud pub/sub for streaming data ingestion .

Start from the big picture



Example

- Company Strategy: cloud-first, global reach
- Product Strategy: cloud-native architecture, fast development cycles
- Business use case: global e-commerce app
- Business requirements: low latency to users, high availability

Business Drivers for Cloud Adoption

- Cost savings
- Agility and time to market
- Reduce operations overhead
- Enable expansion into new markets
- Elasticity & performance

Cloud Migration Strategies

- Lift and shift
- Improve and move
- Remove and replace

Lift and shift : move system as is , take vm and disk with their existing state from on prem and move them to cloud – least amount of time and effort required .
Improve and move – first rearchitect your application and move to cloud
Remove and replace – first we decommission from on prem and replace to cloud .

Requirement	What to think of
Reduce infrastructure administration costs	Managed services, serverless, automation, PaaS and SaaS

Requirement	What to think of
Maintain high availability	Multi-AZ or Multi-region deployments. Managed services, serverless. Load balancers.

Requirement	What to think of
Increase development agility	CI/CD, dev/prod environment parity, containers, infrastructure as code, blue/green and canary deployments.

Requirement	What to think of
Increase ability to generate predictions and insights	Modern data warehousing, AI/ML, data pipelines, BI and data visualization tools.

Success Measurements

- **Business measurements of success**
  - Total Cost of Ownership (TCO)
  - Return on Investment (ROI)
  - Development agility (time from code to production)
  - Key Performance Indicators (KPI)
- **Technical measurements of success**
  - Service availability
  - Service response times
  - Error rate
  - Mean time to recovery (MTTR)

on Google Cloud, for example.

Success Measurements

Relevant to users

- Service availability
- Service response time
- Service error rate

Irrelevant to users

- Server availability
- Server or DB latency
- Server errors

Pearson

Measure what matters to users .

(Almost) every decision is a trade-off

- Performance vs. Cost
- Security vs. Flexibility
- Reliability vs. Agility
- Scalability vs. Cost or Reliability
- Etc.

Some decisions are based on a constraint.

- Example: due to regulatory compliance, your organization may require all data to be stored in a single, specific region.

> The solution's design may sacrifice some amount of reliability to meet the constraint

Some decisions are based on a priority.

- Example: Your organization may place very high priority on security. Your organization also wants to drive agility.

> The solution's design may sacrifice some agility in favor of security by e.g., introducing guardrails and policies.

Some decisions are just implicitly better.

- Example: Your organization wants to set up a development environment that is expected to be used only during business days.

> A solution may include always-on VMs. Another solution may include VMs with a shutdown schedule. Both solutions meet the requirements, but one is cheaper.

Integration with external systems :
with private connectivity between a public cloud and a private cloud, or a multi-cloud which would be some private connectivity between different cloud environments with, for example, a VPN and a VPN which could be between any two networks really.

public integrations, we have access to repositories or dependencies over the internet, access to partners' API over the internet, or the other way around, when partners access our APIs to integrate with our systems.

Integration with External Systems

Private integrations

- Hybrid cloud
- Multi-cloud
- IPSec VPN, VPC Peering

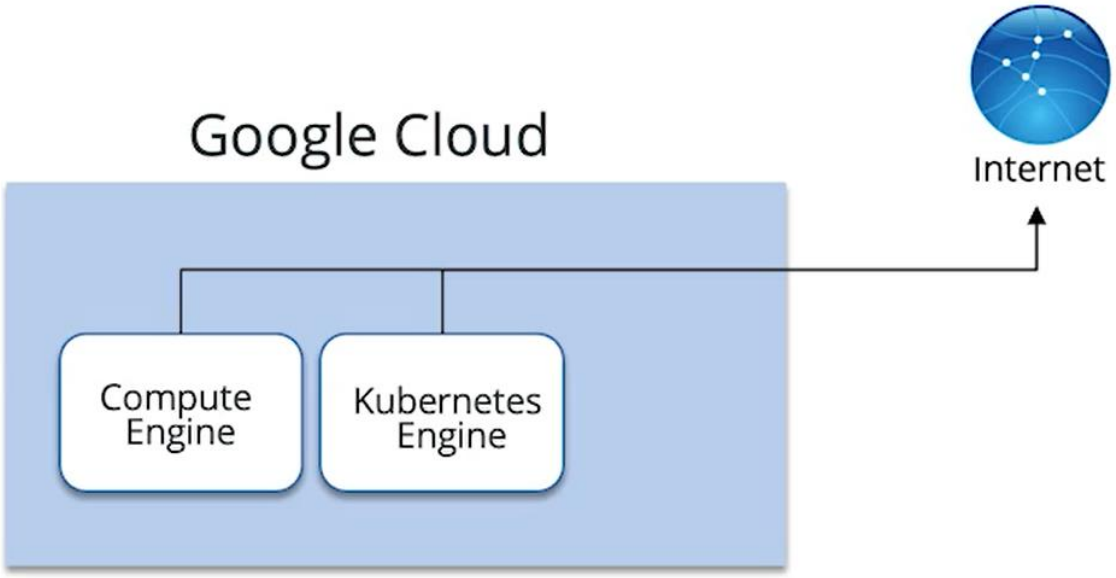
(Will discuss in a later lesson)

Public integrations

- Access to repositories/dependencies over the Internet
- Access to partners' APIs
- Inbound access to APIs from partners

We'll discuss those private integration options later

Pearson

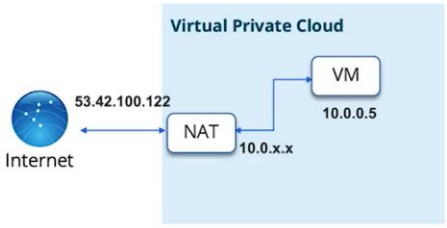


Access gcp from internet .
Cloud NAT :

Allows resources **in a VPC** to create outbound connections to the Interne
without requiring external IP addresses.

- Compute Engine virtual machine (VM) instances without external IP addresses
- Private Google Kubernetes Engine (GKE) clusters
- Cloud Run instances through Serverless VPC Access
- Cloud Functions instances through Serverless VPC Access
- App Engine standard environment instances through Serverless VPC Access

Outbound Internet Access: Cloud NAT



Egress firewall rules still need to allow it

Manual NAT external IP assignment can be leveraged to share static IPs with a 3rd party

! DNAT is only performed for packets that arrive as responses to outbound packets

Pearson

and with inbound net.

Outbound Internet Access: NGFW Appliance

- Typical features include Deep Packet Inspection (DPI), FQDN Filtering, TLS/SSL traffic inspection.
- Many appliances from different vendors can be found in the Google Cloud Marketplace.
- Some appliance vendors also offer customized Compute Engine images for Google Cloud on their website or support pages.
- You can create your own virtualized appliances by using open-source networking software. You also can create your own images.

Example requirement:
All traffic to or from the Internet to secure environments must pass through a centralized virtual appliance.

Pearson

would be when you see that all traffic to

Inbound Integrations: Cloud Endpoints

- API management system
- Access can be controlled with IAM
- Can grant access to API users so they can enable your API in their own Google Cloud project.
- Three endpoint options:
  - Cloud Endpoints for OpenAPI
  - Cloud Endpoints for gRPC
  - Cloud Endpoints Frameworks for App Engine (Python and Java)

Inbound Integrations: Apigee

Inbound Access for Integrations: Apigee

- Enterprise API management system
- Can register app developers who create apps that consume your API
- Can publish APIs to a customizable developer portal
- With Apigee hybrid, APIs can be hosted on-premises, on Google Cloud, or both
- With Apigee Integration, can connect existing data and applications and surface them as accessible APIs (including legacy apps)

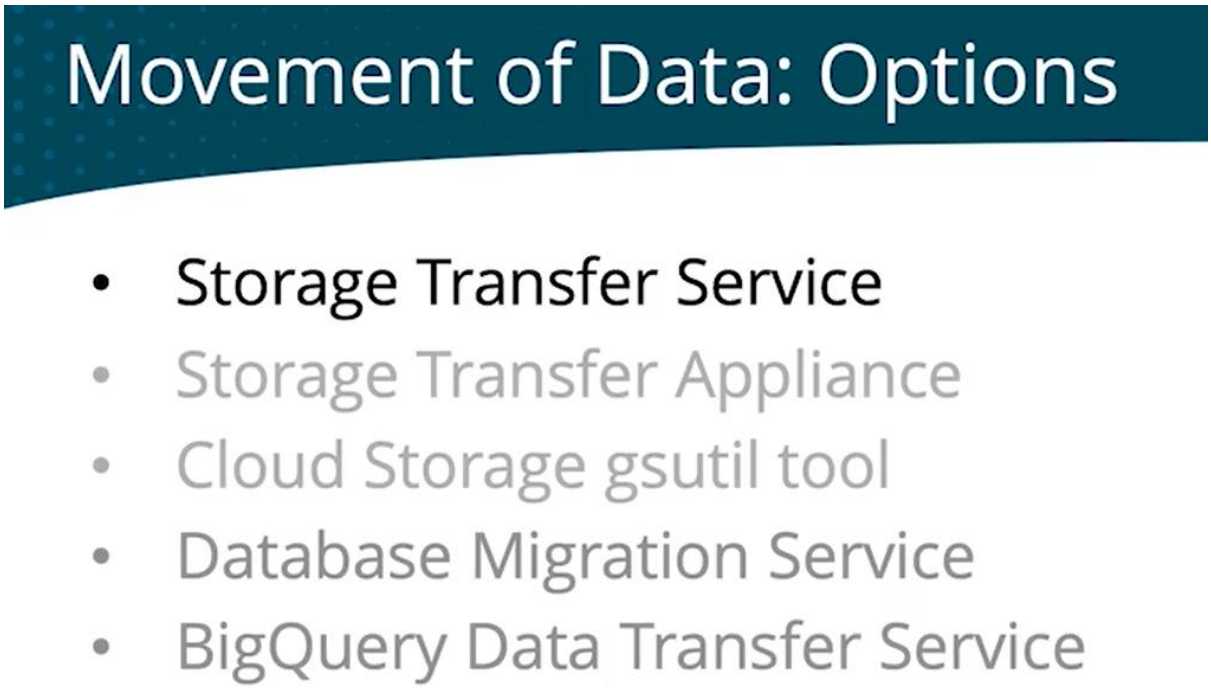
We can divide this into two types, the private integrations and the public integrations. So for the private integrations, we would have hybrid cloud, with private connectivity between a public cloud and a private cloud, or a multi-cloud which would be some private connectivity between different cloud environments with, for example, a VPN and a VPN which could be between any two networks really.

And then when it comes to public integrations, we have access to repositories or dependencies over the internet, access to partners' API over the internet, or the other way around, when partners access our APIs to integrate with our systems.

So let's take a look first into the first two here, which is basically outbound access, or you are accessing external systems over the internet. The more natural solution here, but with perhaps some security implications would be to just access those systems over the internet. But you can also have a Cloud NAT, or a Network Address Translation service, which would give you a little bit more security as you would have in this case, your application's sort of hidden behind this Cloud NAT, and those applications or those virtual machines or whatever you have in a VPC, would not require an external IP address to communicate with those external systems. So Cloud NAT works for compute Engine virtual machines without external IP address, private GKE clusters, and Cloud Run instances through Serverless VPC Access, as well as Cloud Function instances through Serverless VPC Access.

Serverless VPC Access is a feature that basically allows some of the serverless service to integrate with your VPC, and basically communicate with your network through the private

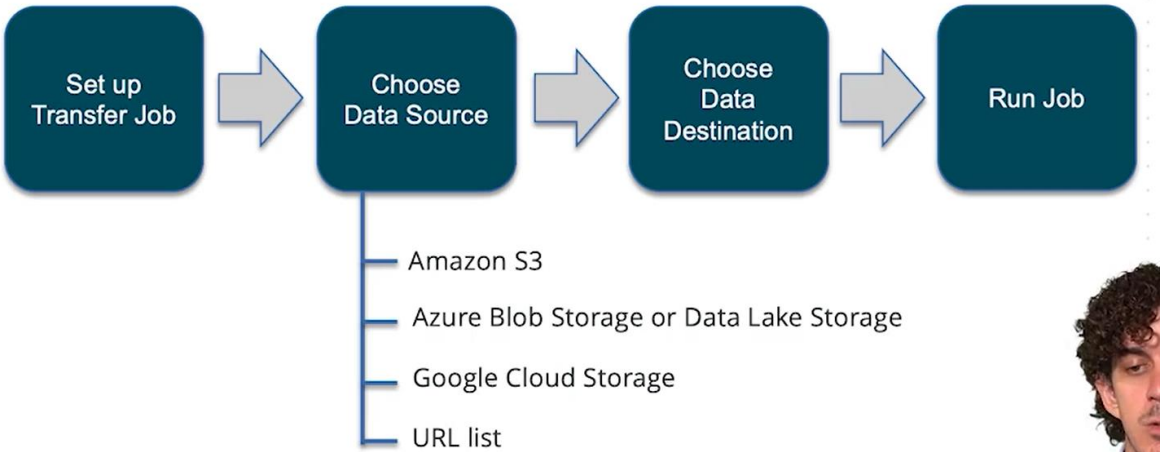
address space of your VPC. So they would work more or less, as if they were running in a VM inside your VPC, even though it's still a serverless managed component. And the same applies to App Engine standard environment,
. So the way Cloud NAT works, is basically you have, for example here, a virtual machine with a private IP address, and you connect to a Cloud NAT instance through the private network, and the Cloud NAT instance will translate that address into a public address for you to communicate with services on the internet. Couple of things to keep in mind here, the Egress firewall rules still need to allow it for the communication to happen. And you can have your Cloud NAT instance, have a manually assigned external IP address, which can be a useful thing to do if you have, for example, third parties and partners that need to whitelist you and your IP addresses so that you can access them. Having statically manually defined IP address is an easy way to basically agree with them. Here's my address, please, a whitelist, this one, this is where my request will be coming from. So you can do that with Cloud NAT. Now one caveat, here's that DNAT which is Destination Network Address Translation or inbound net is only performed for packets that arrive as responses to outbound packets. So you cannot have a scenario where a third party initiates traffic into your systems, and with inbound net. That only happens as a response for traffic you initiate towards external systems. Another option for outbound internet access is to have a next generation firewall appliance. And here a typical feature set of a next generation firewall appliance would include things like, Deep Packet Inspection, FQDN featuring, which is fully qualified domain name featuring for you to for example, restrict which URLs or which, the w.whatever.com addresses you can access, and things like TLS and SSL traffic inspection.'



Movement of Data: Storage Transfer Service

Transfer data securely between **object** and **file** storage across Google Cloud, AWS, Azure, and on-premises.

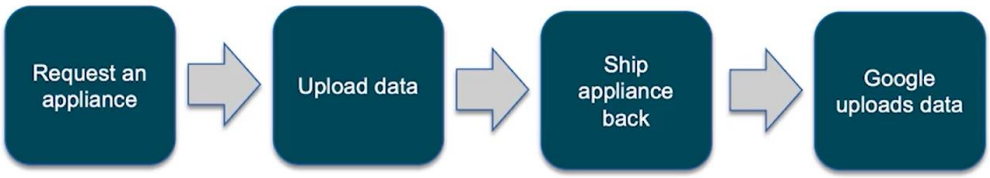
- Encrypts data in transit, supports VPC Service Controls
- Uses checksums to perform data integrity checks
- Does incremental transfer (move files/objects that are new, updated, or deleted since last transfer)
- Preserves object and file metadata during transfer
- Can set up a repeating schedule for transferring data



Transfer Service for on-premises:

- Install a Docker container containing the on-premises agent
- Grant access to resources used by Storage Transfer (by completing the Transfer Service for on-premises set up)
- Start a Transfer Service for on-premises data transfer from Google Cloud

- High-capacity, ruggedized, tamper-resistant storage device
- Ship your data to a Google upload facility
- Data is encrypted with AES 256 encryption
- Data is uploaded to Cloud Storage



Movement of Data: Transferring Large Datasets

- Transfer Service for on-premises can be used to transfer large amounts of data: billions of files and hundreds of TBs of data in a single transfer. Network connections in the tens of Gbps.
- **Data Catalog** can be used to organize data into logical groupings that are moved and used together.
- If not enough bandwidth to meet project deadline: use **Transfer Appliance** for offline transfer.
  - High-capacity, tamper-resistant, storage device
  - Good fit for data > 10TB and if it would take more than one week to transfer over the network

Pearson

If it's either very large amounts of data

- For small transfers (<1TB and enough bandwidth), can use **gsutil** tool
  - For multi-threaded transfers, use **gsutil –m**
  - For a single large file, use **Composite transfers**
    - File is divided into up to 32 chunks and uploaded in parallel, then reassembled.

Movement of Data: Deciding What to Use

Data Source	Scenario	Product
AWS or Azure	Any	Storage Transfer Service
Cloud Storage	Any	Storage Transfer Service
On-premises	Enough bandwidth, less than 1 TB of data	gsutil
On-premises	Enough bandwidth, more than 1 TB of data	Storage Transfer Service for on-premises data
On-premises	Not enough bandwidth	Transfer Appliance

Source: https://cloud.google.com/architecture/migration-to-google-cloud-transferring-your-large-datasets

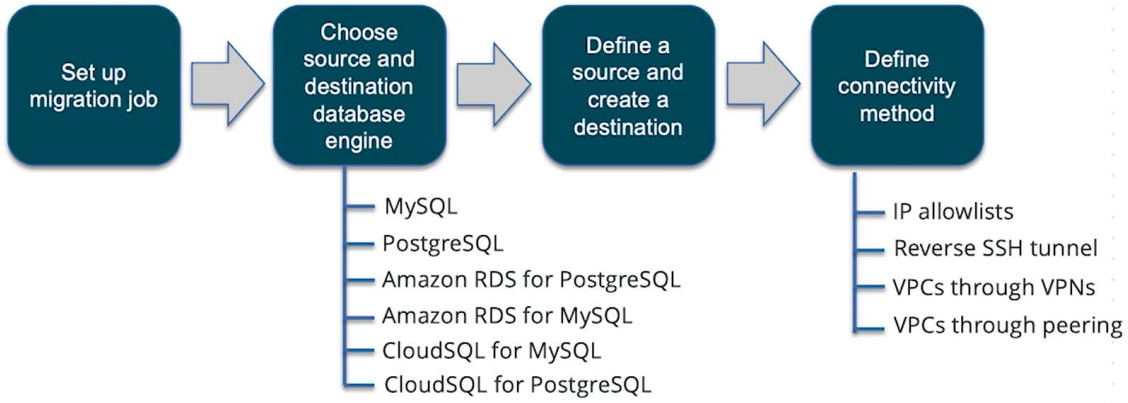
This is about objects and files.

Movement of Data: Database Migration Service

Migrate databases to Cloud SQL

- Fully managed, serverless
- Can migrate from on-premises, GCE, and other clouds
- Can replicate data continuously for minimal downtime migrations
- Available for MySQL and PostgreSQL*

and it's available today *SQL Server and Oracle in preview for MySQL and PostgreSQL,



Automates data movement into BigQuery on a schedule

- Currently can only be used to transfer data into BigQuery
- Supported sources:
  - Cloud Storage
  - Amazon S3
  - Teradata
  - Amazon Redshift
  - Google SaaS apps (Google Ads, Google Play, etc.)
  - Several third-party transfers available in Google Cloud Marketplace
- The service processes and stages data in the same location as the destination dataset

Application Design Considerations

- Fault-tolerance
- Performance
- Security & Compliance
- Portability

- Zones and Regions
  - Deploy over multiple regions
  - Select regions based on geographic proximity
- Security
  - Use Organization Policy Service to enforce guardrails
  - Cloud IAM and least privilege
  - Data encryption at rest and in-transit
- Scalability
  - MIGs to support VM management
  - Pod autoscalers
  - Managed, serverless services

- Leverage billing and cost management tools
  - Organize and structure costs
  - Analyze billing reports
  - Use labels to attribute costs back to departments/teams
  - Build custom dashboards for more granular cost views
  - Use quotas, budgets, and alerts to closely monitor cost trends and forecast costs over time

Cost Optimization Best Practices

- Don't pay for resources you don't use
  - Identify idle VMs (tip: use Recommender service and the Idle Resource Recommender)
  - Schedule VMs to auto start and stop (tip: Google-recommended solution uses Cloud Scheduler, Cloud Pub/Sub, and Cloud Functions to achieve this)
- Rightsize VMs
  - Leverage custom machine types
  - Apply machine type recommendations
- Leverage preemptible VMs

Pearson

And those are just much cheaper VMs,

Cost Optimization Best Practices

Optimize Cloud Storage costs

- Leverage storage classes
- Leverage lifecycle policies

Storage class	Minimum duration	Typical monthly availability
Standard Storage	None	>99.99% in multi-regions and dual-regions 99.99% in regions
Nearline Storage	30 days	99.95% in multi-regions and dual-regions 99.9% in regions
Coldline Storage	90 days	99.95% in multi-regions and dual-regions 99.9% in regions
Archive Storage	365 days	99.95% in multi-regions and dual-regions 99.9% in regions

Retrieval Cost

Storage Cost

Pearson

Cost Optimization Best Practices

- Optimize Cloud Storage costs
  - Leverage storage classes
  - Leverage lifecycle policies
  - Avoid unnecessary object duplication
- Tune your data warehouse
  - Enforce controls to limit query costs
  - Use partitioning and clustering
  - Checking for unnecessary streaming inserts (use batch loading instead, it's free)
  - Use Flex Slots

Pearson

to an amount of BigQuery query capacity.

Cost Optimization Best Practices

- Optimize networking costs
  - Identify "top talkers" and optimize regional and intercontinental network egress
  - Consider standard network tier (as opposed to premium)
  - Filter out logs you don't need in Cloud Logging and enable sampling, if possible, for VPC Flow Logs and Cloud Load Balancing

Pearson

that really creates a lot of logs.

GCP helps support compliance , it's a shared responsibility .
Shared responsibility model – all infrastructure compliance is done by gcp only application compliance need to be taken care of .

Compliance Resource Center

Go-to place for third-party audits and certifications, documentations, and legal commitments.

- Can download reports directly via **Compliance Reports Manager**
  - Report types: Certificate, Audit Report, Statement of Applicability, Vendor Risk Assessment
- Can browse compliance offerings by region (USA, Canada, Latin America, EMEA, Asia Pacific)
- Documentation to aid your own reporting
- Latest industry news and best practices updates

Pearson

and best practices updates
around security and compliance.

Questions Breakdown

Your team has decided to adopt Agile development practices and release new software daily into a production Kubernetes cluster hosting a **public web application**. You want to create baseline metrics to **measure the quality of new software** versions. What should you measure?

A. CPU utilization and memory utilization
B. Pod uptime and Pod-to-Pod latency
C. Client request latency and error rate
D. Service uptime and average CPU utilization

Pearson

client request latency and error rate.

Questions Breakdown

Press Esc to exit full screen

You are responsible for ensuring an application's binary log data is **stored for 1 year**. The file may be **inspected during the first 30 days** but it's very unlikely that it will be accessed again after that. You have decided to store the files in Cloud Storage. What would you do to optimize Storage cost?

A. Choose the Standard storage class for the file. Create a lifecycle rule to move the file to Coldline after 30 days and a second lifecycle rule to delete the file after 365 days.

B. Choose the Standard storage class for the file. Create a lifecycle rule to move the file to Nearline after 30 days and a second lifecycle rule to move the file to Coldline after 365 days.

C. Choose the Coldline storage class for the file. Create a lifecycle rule to delete the file after 30 days.

D. Choose the Coldline storage class for the file. Create a lifecycle rule to move the file to Archive after 30 days.

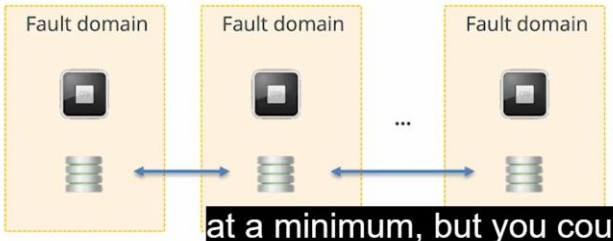
to delete the file after 360 days.

Pearson

03:18 / 03:19

High Availability (HA) Design Principles

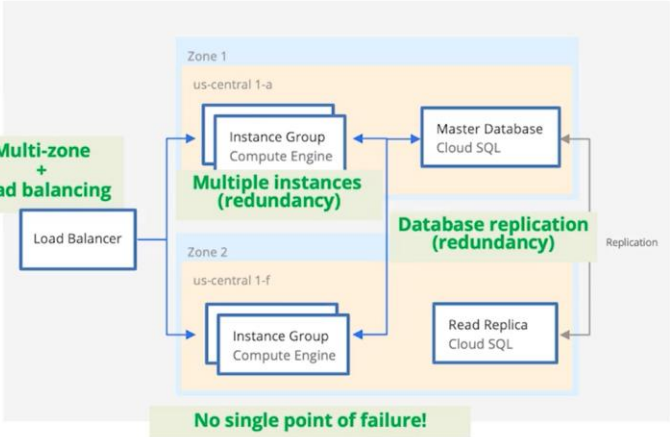
- Create redundancy
- Eliminate single points of failure
- Replicate resources across multiple fault domains



at a minimum, but you could also have resources spread

Pearson

Multi-zone HA Architecture



Pearson

here in this design.

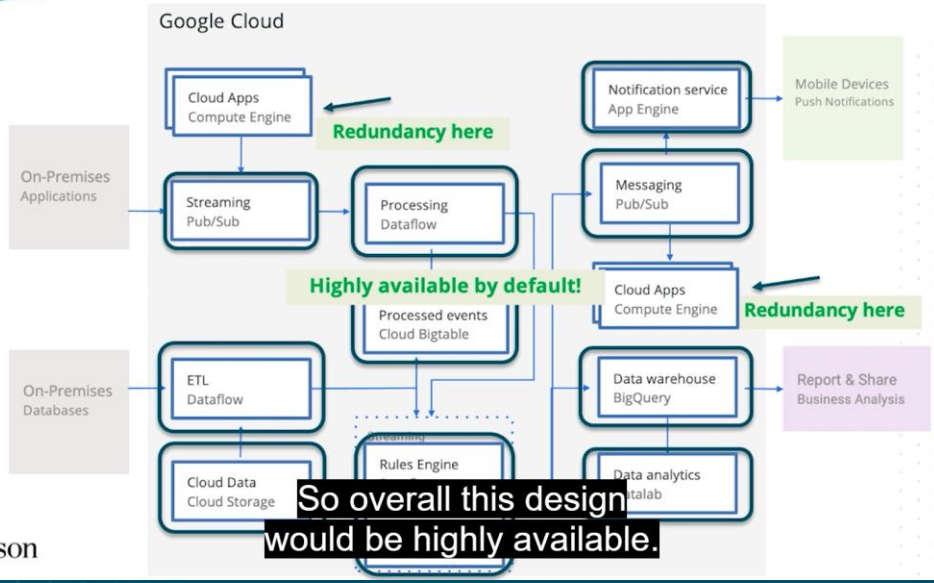
Leverage Managed Services

- Most managed services are regional (i.e., resilient against zone outages)
- Some are global or multi-region (i.e., resilient against region outages), for example:
  - Cloud Storage
  - BigQuery
  - Cloud Spanner
  - Cloud Firestore
  - HTTP(S) Load Balancer

Pearson

BigQuery, Cloud Spanner, Cloud Firestore,

High Availability Design



Pearson

Multi-Region Architecture

Regional resources should replicate data to a secondary region

- Use multi-regional storage option such as Cloud Storage
- Alternatively, use a hybrid cloud option such as Anthos
- Understand the trade-off between synchronous and asynchronous data replication (latency vs. risk of data loss)

Because if you need
synchronous data replication

Pearson

Understand the high-availability (99.9% and 99.99%) setups for Interconnect connections:

- Rule of thumb:
  - 4 connections -> **99.99%**
  - 2 connections -> **99.9%**

Medical Software: company overview

The company is a leading provider of electronic health record software to the medical industry, providing their **software as a service** to **multi-national** medical institutions.

Solution concept

The business has been growing exponentially year over year. They need to be able to **scale their environment** and **adapt their disaster recovery plan**.

Existing technical environment

The company is hosting several legacy file- and API-based integrations with insurance providers on-premises. There is **no plan to upgrade or move these systems** at the current time.

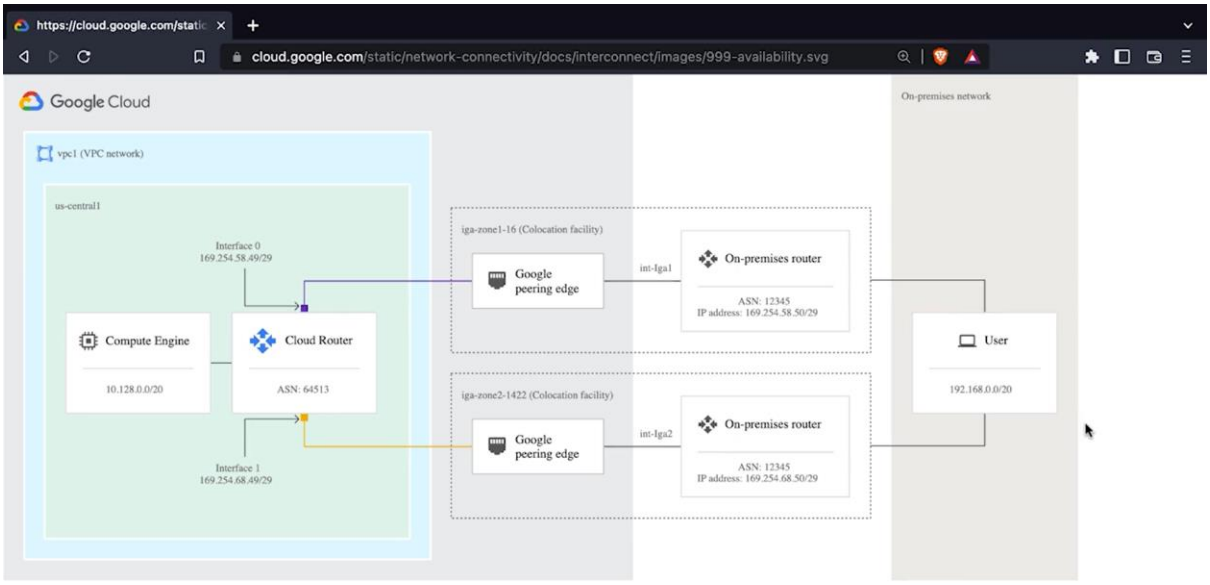
Customer-facing applications are web-based, and many have recently been **containerized to run on a group of Kubernetes clusters**. Data is stored in a mixture of relational and NoSQL databases (**MySQL, MS SQL Server, Redis, and MongoDB**)

Business Requirements

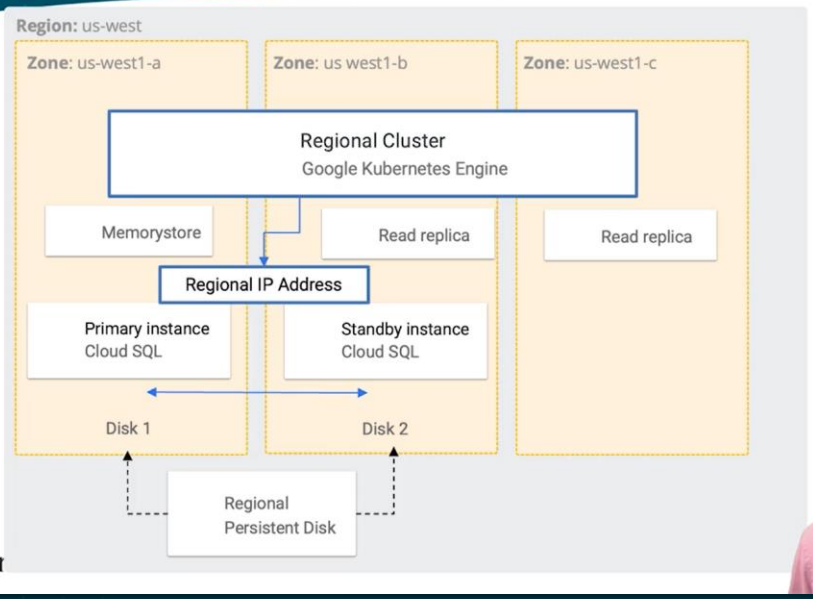
- Provide a minimum **99.9% availability** for all customer-facing systems.
- **Reduce latency** to all customers.
- Decrease infrastructure administration costs.

Technical Requirements

- Maintain legacy interfaces to insurance providers with **connectivity to both on-premises systems and cloud** providers.
- Provide a **secure and high-performance connection** between on-premises systems and Google Cloud.



Case Study Technical Solution: Web Apps



Elasticity

- Does the solution scale when busy, so that the service's availability and performance remain intact?
- Does the solution scale back when demand is low, so that infrastructure cost is reduced?

Design Strategies for Elasticity

- Autoscaling and load balancing
- Managed services & serverless

Autoscaling and Load Balancing: Compute Engine VMs

Managed instance groups (MIGs)

- Specify an instance template and optional stateful configuration
- Define an autoscaling policy
  - Target utilization metric
  - Schedules

Instance templates
Define the machine type, boot disk image or container image, labels, startup script, and other instance properties

Autoscaling and Load Balancing: Compute Engine VMs

Managed instance groups (MIGs)

- Target utilization metrics:
  - Average CPU utilization
  - HTTP load balancing serving capacity, based on either utilization or requests per second
  - Cloud Monitoring metrics

Tip: when you create an autoscaling policy with multiple signals, the autoscaler scales based on the signal that provides the largest number of virtual machine (VM) instances in the MIG

Autoscaling and Load Balancing

Kubernetes Engine containers: GKE Autoscaling

- Pod autoscaling
- Cluster autoscaling

Autoscaling and Load Balancing

GKE Autoscaling: Pod autoscaling

- Horizontal Pod autoscaling automatically increases or decreases the number of Pods based on workload's CPU, memory, or custom metrics.
- Vertical Pod autoscaling lets you analyze and set CPU and memory resources required by Pods. Can be used to provide recommended values for CPU and memory requests and limits.

Autoscaling and Load Balancing

GKE Autoscaling: Cluster autoscaling

- Automatically resizes the number of nodes in a given node pool based on the demands of workloads.
- You specify a minimum and maximum size for the node pool
- If Pods are unschedulable because there are not enough nodes in the pool, cluster autoscaler will also add nodes.
- Cluster autoscaler will automatically balance nodes across availability zones (for regional clusters)

Design Strategies for Elasticity: Serverless

A serverless architecture will be elastic by default

- Hosting applications
  - **Cloud Functions** for smaller units of code that are triggered by cloud events or HTTP requests
  - **App Engine** for larger units of code triggered by HTTP or cloud events
  - **Cloud Run** for small or large units of code that run jobs or services

Design Strategies for Elasticity: Serverless

A serverless architecture will be elastic by default

- Storing data
  - **Cloud Firestore** for a serverless NoSQL database
  - **Cloud Storage** for object storage

Design Strategies for Elasticity: Serverless

A serverless architecture will be elastic by default

- Running analytics
  - **Cloud Pub/Sub** for serverless ingestion
  - **Cloud Dataflow** for serverless stream and batch data processing
  - **Cloud Dataproc Serverless** for serverless Spark batch workloads
  - **BigQuery** for serverless data warehousing
  - **AI Platform** for serverless AI/ML

Quotas and Limits

GCP-enforced quotas:

- Used to restrict how much of a particular shared Google Cloud resource you can use
- Enforced to protect community from unforeseen spikes in usage and overloaded services

User-enforced quotas:

- You can set your own limits on service usage to avoid unexpected bills

Quotas and Limits

Types of quotas:

- **Rate quotas**: limits the number of requests you can make to an API
- **Allocation quotas**: restricts the use of resources that don't have a rate of usage

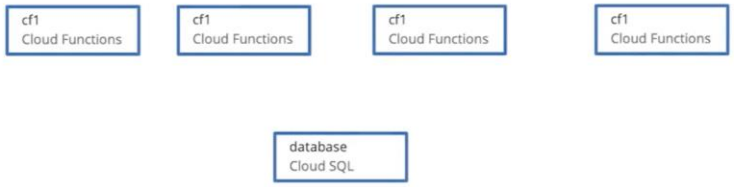
You can view all project quotas in the Google Cloud console, under the **Quotas** page

You can also manage your quota using the Service Usage API

Quotas and Limits

Rate-limiting tools to scale without hitting limits:

- Limit concurrent connections to Cloud SQL database
  - With Cloud Functions, use **max instances** setting to limit how many concurrent instances of the function are running and establishing database connections



Quotas and Limits

Rate-limiting tools to scale without hitting limits:

- Use Cloud Pub/Sub to process work in batches or control flow
  - Wait until a sufficient number of messages have accumulated before handling all of them in one batch
  - If the subscriber client processes messages more slowly than Pub/Sub sends them, use **flow control** to control the rate at which the subscriber receives messages (and spread messages over multiple subscribers)
- Use Cloud Run for heavily I/O-bound work
  - Cloud Run allows your workload to specify how many concurrent requests it can handle

Scalability for Growth

- More than just autoscaling to cover temporary demand fluctuations.
- Think about scalable design patterns:
  - Adjust capacity to meet demand with autoscaling
  - Aim for statelessness
  - Leverage serverless platform and scalable, managed services for consistent performance
  - Leverage Cloud Monitoring to make data-driven scaling decisions
  - Leverage native load balancing and multi-zone/multi-region architectures to withstand failures
  - Leverage CI/CD through native tools to help automate building and deploying apps (+ incorporate automated testing)
  - Leverage automation and loose coupling

Performance Optimization Best Practices

- Autoscaling and data processing
- GPUs and TPUs
- Application performance monitoring

Performance Optimization: GPUs and TPUs

Specialized hardware platforms

- Graphics Processing Unit (GPU)
  - Available for Compute Engine VMs
  - Can accelerate workloads such as machine learning and data processing
  - NVIDIA-based
- Tensor Processing Unit (TPU)
  - Designed by Google
  - Specifically used to accelerate machine learning workloads (with **TensorFlow**)

Performance Optimization: GPUs and TPUs

GPUs vs TPUs

When to use GPU	When to use TPU
Models with a significant number of custom TensorFlow operations	Models with no custom TensorFlow operations inside the main training loop
Models for which source code is too onerous to change	Models dominated by matrix computations
Medium-to-large models	Larger and very large models

Performance Optimization: Application Performance

Use tools to inspect and analyze performance:

- **Cloud Trace** and **OpenTelemetry**: helps you instrument code to identify latency and find bottlenecks in inter-service communications
- **Cloud Debugger**: helps you inspect and analyze production code behavior in real time without affecting its performance
- **Cloud Profiler**: helps you identify and address performance by continuously analyzing CPU and memory consumption

Case Study: Designing for Scalability

Industrial IoT: company overview

The company manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and **service centers in 100 countries**. Their mission is to build products that make their customers more productive.

Case Study: Designing for Scalability

Solution Concept

There are **2 million vehicles** in operation currently, and we see **20% yearly growth**. Vehicles collect **telemetry data from many sensors** during operation. A **small subset of critical data** is transmitted from the vehicles in **real time** to facilitate fleet management. The rest of the sensor data is **collected, compressed, and uploaded daily** when the vehicles return to home base. **Each vehicle** usually generates **200 to 500 megabytes of data per day**.

Case Study: Designing for Scalability

Existing Technical Environment

The company's vehicle data aggregation and analysis infrastructure resides **in Google Cloud** and **serves clients from all around the world**.

(...)

The **web frontend** for dealers and customers is running in Google Cloud and **allows access** to stock management and analytics.

Case Study: Designing for Scalability

Business Requirements

- Decrease cloud **operational costs** and **adapt to seasonality.**
- Increase **speed** and **reliability** of **development workflow.**
- (...)
- Create a **flexible and scalable** platform for developers to create **custom API services** for dealers and partners.

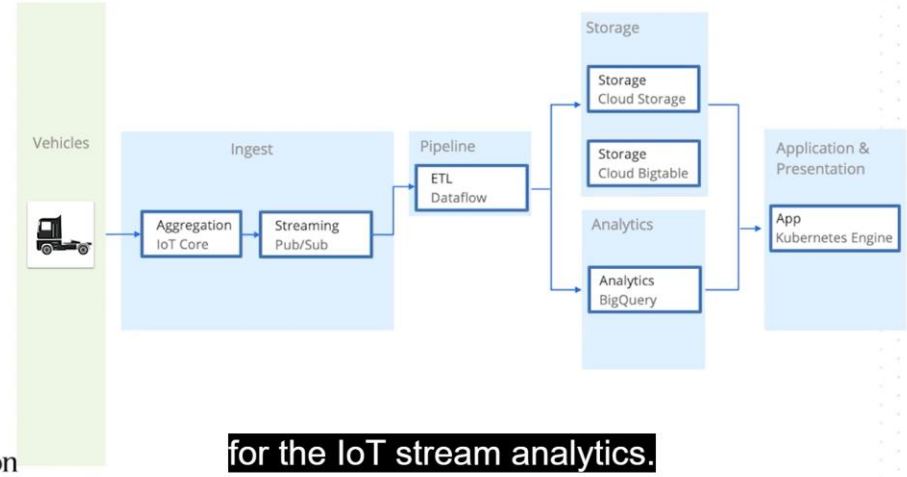
Case Study: Designing for Scalability

Technical Requirements

- (...)
- **Modernize all CI/CD pipelines** to allow developers to deploy **container-based workloads** in highly scalable environments.
- (...)
- Create a **self-service portal** for internal and **partner developers** to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- (...)

Case Study: Designing for Scalability

Technical Solution: IoT stream analytics

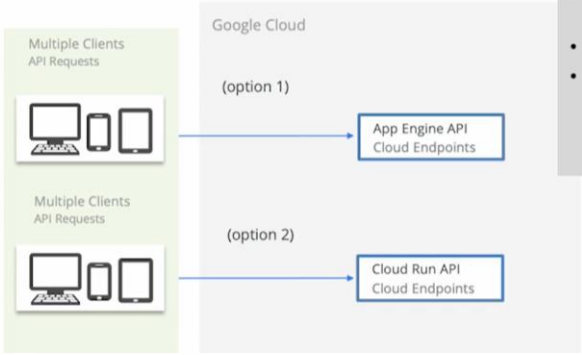


Pearson

for the IoT stream analytics.

Case Study: Designing for Scalability

Technical Solution: API



Business Requirements

- Decrease cloud **operational costs** and **adapt to seasonality.**
- (...)
- Create a **flexible and scalable** platform for developers to create **custom API services** for dealers and partners.

Your company has a web-based application hosted in a **single data center**. As the customer base grows, customers from distant locations often complain that the **website is slow**. Your company has decided to move the application to Google Cloud to benefit from its **global footprint**.

How should you design the solution to **reduce latency** for customers?

A. Deploy the application to a set of virtual machines and use DNS-based load balancing.

B. Use zonal managed instance groups in different zones with a regional load balancer.

C. Use regional managed instance groups in different regions with a global load balancer.

D. Deploy the application to a regional App Engine instance with a global load balancer.

Your company is running a **stateless** web application on two Compute Engine instances in two availability zones. The application receives **a lot of traffic during business hours** and little traffic otherwise. **During peak hours**, several users are complaining that the **application is slow and sometimes crashing**. You need to redesign the solution to **improve performance**. What should you do?

A. Deploy the application to two more instances in a separate Google Cloud region.
B. Deploy the application to an extra instance in a new availability zone. Configure startup and shutdown scripts so that the extra instance only runs during business hours.
C. Set up a Cloud Monitoring alert that triggers a Cloud Function to create a new instance if the average CPU utilization is high.
D. Create an instance template and deploy the application to a managed instance group with an autoscaling policy.

Your company has acquired another company that has a **containerized** web application running on-premises. You need to move the application to Google Cloud and redesign the solution to accommodate a **larger number of users** and **scale automatically** with usage. What should you do?

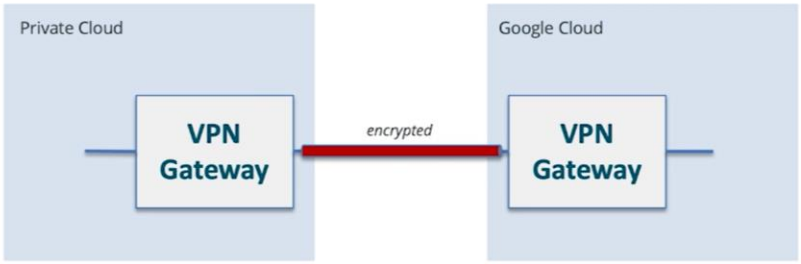
A. Host the application on Google Kubernetes Engine and enable Horizontal Pod Autoscaler and cluster autoscaling.
B. Host the application on Google Kubernetes Engine and enable Vertical Pod Autoscaler and cluster autoscaling.
C. Host the application on Compute Engine instances. Perform a load test and use Google Cloud's machine type recommender to identify the most appropriate machine type.
D. Host the application on a managed instance group with an autoscaling policy. Use Google Cloud's managed instance group machine type recommender to identify the most appropriate machine type.

Hybrid Networking Services

- Cloud VPN
- Cloud Interconnect
- Network Connectivity Center

Hybrid Networking Services: Cloud VPN

- IPSec VPN tunnel
- Traffic travels over the public internet
- Traffic is 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!)

Recommended HA configuration: Two interfaces, two external IPV4 addresses

You can configure an HA VPN gateway with only one active interface and one external IP address, but *this configuration does not provide 99.99% availability SLA!*

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

Now, each interface

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 (vendor-specific):
  - Configure two tunnels
    - GCP side: one in each Cloud VPN interface
    - Peer side: one in each device (if two devices) or interface (if single device, multiple interfaces)
  - Peer gateway must support dynamic BGP routing

Integration with On-Premises: Dedicated Interconnect

- Direct connection to Google's network
- Provides access to all GCP products and services from on-premises network, **except Google Workspace**
- Capacity starting at 50Mbps and up to hundreds of Gbps
- Connection is **not** encrypted

Integration with On-Premises: Dedicated Interconnect

- Must be able to physically meet Google's network
- 10Gbps or 100Gbps circuits with flexible VLAN attachment
- Maximum of dth) or 2x100Gbps (
- Google offer
- Must configure BGP on on-premises routers and Cloud Routers

Cloud Router is a fully distributed and fully managed software-based router that uses the Border Gateway Protocol (BGP) to advertise IP address ranges.

Integration with On-Premises: Partner Interconnect

- More points of connectivity through one of the supported service providers
- Traffic passes through service provider's network, but **not** the public internet
- Flexible VLAN attachment capacities from 50Mbps to 50Gbps
- Configure BGP only if doing layer 2 connections (configuration for layer 3 is fully automated)
- Google provides SLA for Google-Partner connection

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

Integration with On-Premises: 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)
- Connectivity can be through Cloud VPN, Dedicated/Partner Interconnect, and through a Router appliance

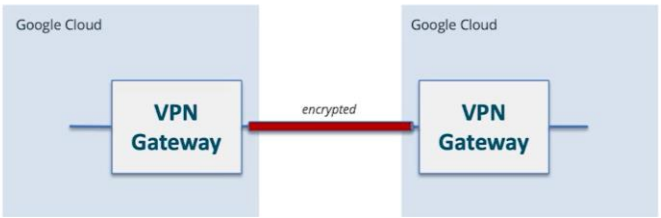
Network Connectivity Center Use Cases

- **Site-to-cloud connectivity**: Connect an external network to Google Cloud by using a third-party SD-WAN router or another virtual appliance.
- **Site-to-site data transfer**: Use Google's network as a WAN to connect sites that are outside of Google Cloud (with full mesh connectivity).
- Monitor traffic within your Google Cloud project by using a third-party firewall appliance.
- Use a third-party virtual router to connect your VPC networks to one another.

Multicloud Networking

Effectively one (private) option: Cloud VPN

- 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

- Anthos
- BigQuery Omni
- Looker

Multicloud Solutions

- Anthos
- BigQuery Omni
- Looker

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

Multicloud Solutions

- Anthos
- BigQuery Omni
- Looker

Multicloud scenario	Core Anthos components
Infrastructure, container, and cluster management	Anthos clusters on AWS / Anthos clusters on Azure
Multicluster management	Fleets and Connect
Configuration management	Anthos Config Management
Migration	Migrate to Containers
Service management	Anthos Service Mesh (AWS only)

Multicloud Solutions

- Anthos
- BigQuery Omni
- Looker

BigQuery Omni is a **multicloud analytics** solution that can access and analyze data residing in multiple cloud environments (without moving or copying the data).

Case Study: Integration with On-Premises

Medical Software: company overview

The company is a leading provider of electronic health record software to the medical industry, providing their **software as a service** to **multi-national** medical institutions.

Case Study: Integration with On-Premises

Solution concept

The business has been growing exponentially year over year. They need to be able to **scale their environment** and **adapt their disaster recovery plan**.

Case Study: Integration with On-Premises

Existing technical environment

The company is hosting several legacy file- and API-based integrations with insurance providers on-premises. There is **no plan to upgrade or move these systems** at the current time.

Case Study: Integration with On-Premises

Business Requirements

- Provide a minimum **99.9% availability** for all customer-facing systems.
- **Reduce latency** to all customers.
- Decrease infrastructure administration costs.

Case Study: Integration with On-Premises

Technical Requirements

- Maintain legacy interfaces to insurance providers with **connectivity to both on-premises systems and cloud** providers.
- Provide a **secure and high-performance connection** between on-premises systems and Google Cloud.

Case Study: Integration with On-Premises

Refer to the medical software case study. You need to design a hybrid connectivity solution that meets the business and technical requirements. The company has currently **one Dedicated Interconnect connection**. What should you do?

A. Add two VPN connections between on-premises and Google Cloud and ensure they are placed on two different physical devices
B. Add two Partner Interconnect connections in two separate metro availability zones
C. Add another Dedicated Interconnect connection in the same metro availability zone
D. Add another Dedicated Interconnect connection in a separate metro availability zone

Cloud-native Networking: VPC

Software-defined networking for:

- Compute Engine VMs,
- Google Kubernetes Engine (GKE) clusters
- App Engine flexible environment

Cloud-native Networking: VPC

- Global resource, logically isolated
- Consisting of a list of regional subnetworks (subnets)
- Implements a distributed virtual firewall
- Implements a distributed virtual router

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: Private Service Connect

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

Cloud-native Networking: Private Google Access

- Allows VMs without public IP address to reach Google APIs and services
  - Exceptions: App Engine Memcache, Filestore, Memorystore
- Enable 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

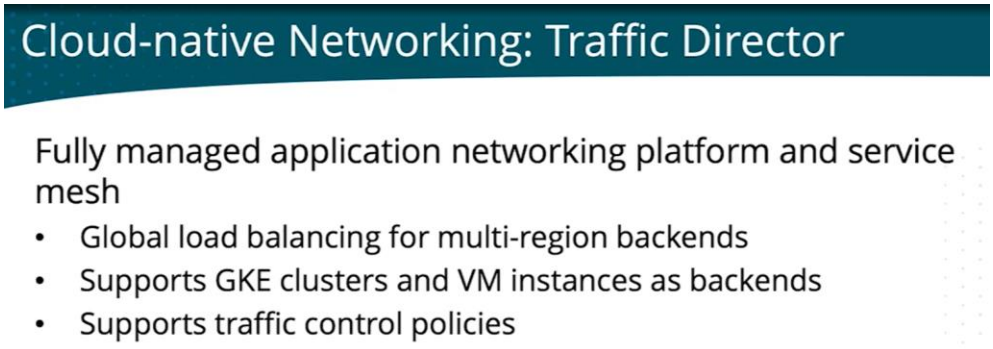
Cloud-native Networking: Load Balancing

Fully distributed, software-defined load balancing

- Supports 1 million+ queries per second
- Consistent performance and low latency
- Seamless autoscaling
- Integration with Cloud CDN and Cloud Armor

Cloud-native Networking: 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)



Cloud DNS, very basic DNS service. It's a fully managed, the only Google Cloud Service with a 100% availability SLA. Cloud CDN which is the content delivery network and cloud owner the web application firewall service which also offers DDoS protection. Then a couple of other features you mentioned here, are VPC flow logs which allows you to do troubleshooting or forensics of network traffic by analyzing flows between different components in your network. Packet mirroring allows you to mirror traffic from a network interface through another location, so that you can inspect and visualize that traffic flow.

When it comes to designing cloud-native networks, you should think first how many projects and VPC networks do you have? And then think about the inter VPC connectivity if needed. But do remember that VPCs offer a layer of isolation? So, you should choose a number of VPCs, based on how many different components you want to isolate and keep separated from each other. So, inter-VPC connectivity should be kept to a minimum, but in some cases it is necessary. Also, design hybrid network connectivity if connectivity to on-premises environments are required.

Remember that VPCs are global resources and not associated with a region. The subnets or sub-networks are associated to a region. You can use VPC to privately access managed services. For example, Cloud SQL, Cloud Storage or several other Google APIs and services with things like Private Service Connect or Private Google Access. You can secure inter VM traffic with VPC firewall rules, using network tags. You can secure network administration with Cloud IM or identity and access management. You can use shared VPC to provide centralized networking to multiple projects