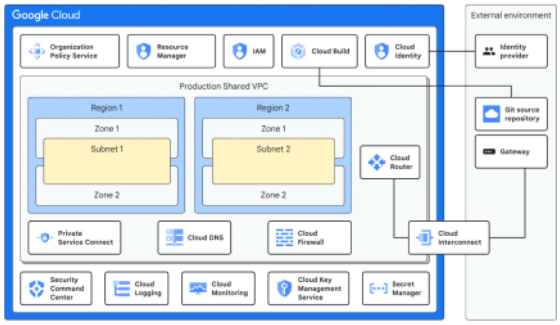
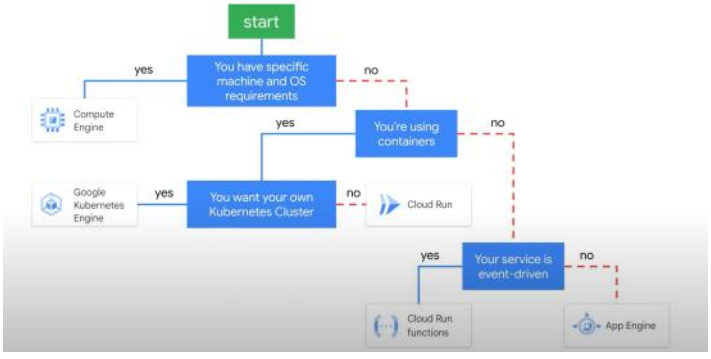
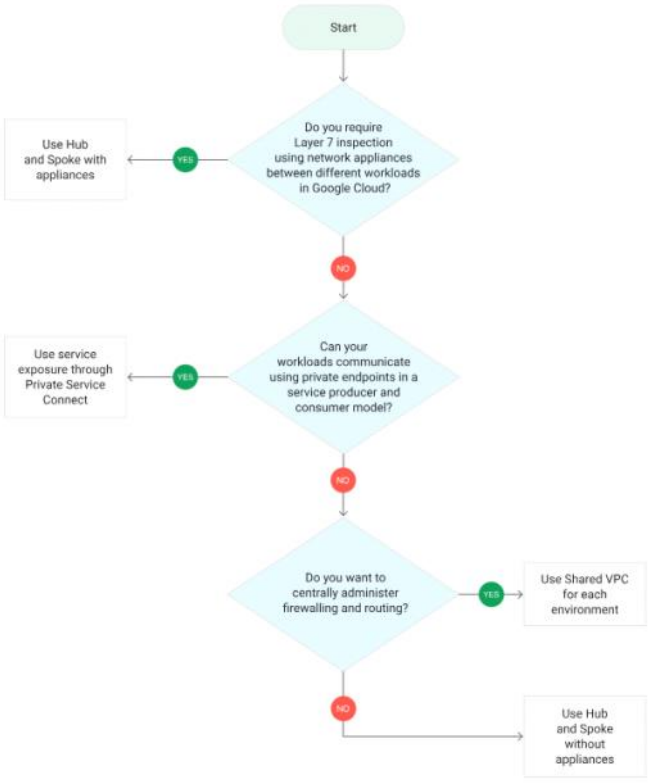
08 November 2024 10:41

E-mail id: cloud-part	ner-training@google.com

Problems with accessing Cloud Skills Boost for Partners: cloud-partner-training@google.com
Problems with a lab (locked out, etc.): support@qwiklab.com
Problems with accessing Partner Advantage: pps-support@google.com

Name	Characteristics	Use cases
Data storage product use cases		
Modelling secure flow on GCP	Client (Mobile/Laptop) -> https -> Google Cloud Armor on Google external load balancer (to block any denied IP addresses) -> Custom VPC Network each per region (and back up subnet as per availability requirements) -> Firewall rules on subnet(s) to allow/restrict SSH , HTTPS etc. -> cloud VPN Tunnels to securely communicate with on-premise networks -> Enabling "private google access" to communicate with services like cloud SQL, firestore etc. -> Private google access help reduce network costs. • Reduce risk of DDOS attacks/SQL Injection/Cross-site scripting using Google cloud armour, External Application Load Balancer, Cloud CDN. 	
Choosing platform for deployment		



• General Theme is,
- Deploy application in region closest to your users
- If high availability is needed then deploy in multiple zones within closest region (for speed)
- Frontend is typically deployed separately than backend and other services. Managed instance group can be configured to distribute VMs across zones. GKE can also be used to same effect.
- Communication between client and Frontend/UI happens over HTTPS and uses Application load balancer (HTTP). Cloud armour is used to block Ips etc.
- Communication between UI and backend services happen over TCP (typically but not necessarily) and uses network load balancer. Firewall rules are used to allow/disallow HTTP and SSH Access.
- Private google access is enabled for backend to communicate with google's managed offerings
- For Cloud SQL, the database can be configured for failover replica for high availability which provides data redundancy and a standby instance of the database server in another zone. Some data services, such as Firestore or Spanner, provide high availability by default.
- Google cloud storage can be configured for HA with multi-region storage buckets.

• Network Deployment aspects
- A Shared VPC network for each environment (production, development, and testing) connects resources from multiple projects to the VPC network.
- Virtual Private Cloud (VPC) firewall rules control connectivity to and from workloads in the Shared VPC networks.
- A Cloud NAT gateway allows outbound connections to the internet from resources in these networks without external IP addresses.
- Cloud Interconnect connects on-premises applications and users. (You can choose between different Cloud Interconnect options, including Dedicated Interconnect or Partner Interconnect.)
- Cloud VPN connects to other cloud service providers or on-prem networks.
- A Cloud DNS private zone hosts DNS records for your deployments in Google Cloud.
- Google Cloud Observability includes Cloud Monitoring for monitoring and Cloud Logging for logging. Cloud Audit Logs, Firewall Rules Logging and VPC Flow Logs help ensure all necessary data is logged and available for analysis.
- A VPC Service Controls perimeter includes Shared VPC and the on-premises environment. A security perimeter isolates service and resources, which helps to mitigate the risk of data exfiltration from supported Google Cloud services.

• Managed instance groups of VMs
- Managed instance groups create VMs based on instance templates.
- Advantages are , auto healing to recreate instances that don't respond and creating instances in multiple zones for high availability.
- Recommended to use one or more instance groups as backend for load balancers.
- Regional managed instance groups are generally recommended over zonal managed instance groups because they allow you to spread the application load across multiple zones instead of confining your application to a single zone or you having to manage multiple instance groups across different zones. This replication protects against zonal failures.

• API Mgmt products,
- Apigee - Managing high value/volume of APIs with enterprise-grade security and dev engagement
- Apigee hybrid - Maintaining and processing API traffic within your own kubernetes cluster
- Cloud endpoints - Managing gRPC services with locally hosted gateway for private networking
- API Gateway - Building proof-of-concepts or entry-level API use cases to package serverless applications running on Google Cloud

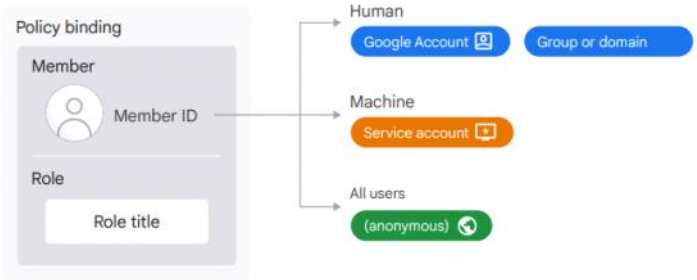
• App Engine
- a fully managed, serverless application platform designed for microservices. each Google Cloud project can contain one App Engine application, and an application has one or more services.
- Services are independently deployable and versioned.
- Applications can be scaled seamlessly from zero upward without having to worry about managing the underlying infrastructure.

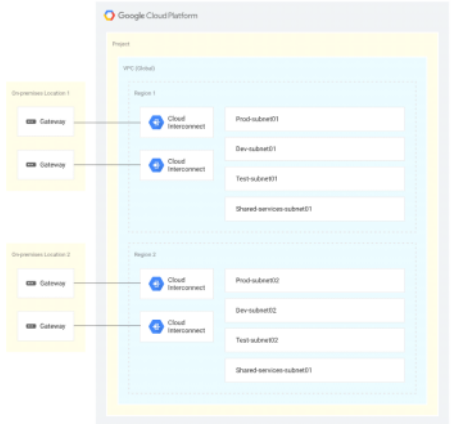
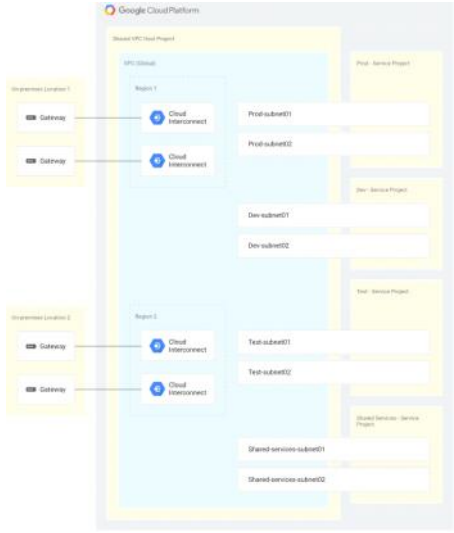
• Cloud run - Runs containers (Services or jobs).
- Criteria,
  - Serves requests, streams, or events delivered using HTTP, HTTP/2, WebSockets, or gRPC, or executes to completion.
  - Does not require a local persistent file system, but either a local ephemeral file system or a network file system.
  - Is built to handle multiple instances of the app running simultaneously.
  - Does not require more than 8 CPU and 32 GiB of memory per instance.

• Cloud run functions - Functions deployment. Event-driven or over HTTP. Short lived.
• Sole tenant mode - Lets you have a physical Compute Engine server that is dedicated to hosting only your project's VM

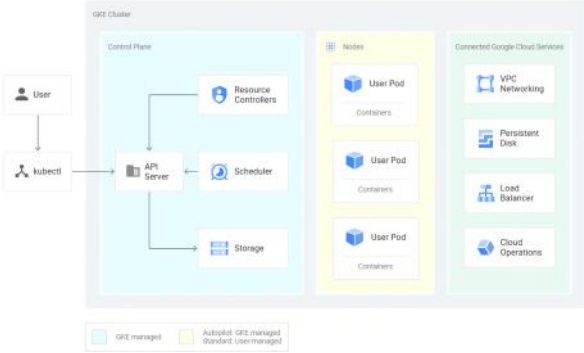
	• Shielded VM - offers verifiable integrity of your Compute Engine VM instances. Making it free from being compromised by boot or kernel level malware. The Shielded VM enables Measured Boot by performing the measurements needed to create a known good boot baseline, called the integrity policy baseline. The integrity policy baseline is used for comparison with measurements from subsequent VM boots to determine if anything has changed. • Stackdriver Trace and Logging (Google cloud operations) are tools within the Google Cloud Operations Suite that help developers monitor and debug their applications, particularly those running on Google Cloud Platform (GCP)	
Disaster Recovery	- Define service-wise objectives, - Cold Standby - requires heartbeat and snapshot system to detect failure and spin off instances in backup region, restore backups and routes transactions to new region. - Hot standby - provisions instances in multiple regions and uses global load balancer to detect and route requests. - @@@@Snapshots - Captures exact state of persistent medium (Disk) at that moment. Data can be restored to its last known good state if a disk fails or data becomes corrupted.	
DevOps	- Blue/green deployment - Only one version of a service runs at a time. New service is provisioned on new infra, tested and then made live. - Canary deployment - Deploy a new version separately and portion of traffic is diverted, behaviour is observed. - Rolling updates - Service versions are updated one at a time. There may be multiple versions of service running at a time.	
Cost savings	- VM Costs • Consider committed use discounts • Consider at least some spot VMs • Consider more small machines with auto scaling turned on - Disks Costs • Do not over-allocate. You can increase the size of your Persistent Disk when your virtual machine (VM) instance requires additional storage space or increased performance limits. You can increase the disk size at any time, whether or not the disk is attached to a running VM. • Consider standard over SSD based on I/O Requirements • Compare cost of storage alternatives (e.g. 1GB in firestore is free) - Network costs • Keep VMs close to your data - Use GKE Usage metering - Set budget and alerts	
IAM	Resource Hierarchy, - Organization -> Folders -> Projects -> Resources - Resources = represent virtual machines, Cloud Storage buckets, Virtual Private Networks (VPCs), tables in BigQuery, or anything else in Google Cloud. - Projects = resources get organized in projects - Organization = encompasses all the projects, folders, and resources in your organization • IAM Basic roles are Owner (Full Administrative Access), Editor (Modify and delete access) and Viewer (Read-only access) • Two roles created during project creation are "Workspace Super Admin" and "Cloud Organization admin". • Three types of roles in Cloud IAM: basic roles, predefined roles, and custom roles. • IAM also have pre-defined roles that offer fine-grained permissions on services. • IAM Conditions allow you to define and enforce conditional, attribute-based access control for Google Cloud resources. • A service account is an account that belongs to your application and is used for service to service interaction. • A Google group is a named collection of Google accounts and service accounts. Every group has a unique email address that is associated with the group. Google groups are a convenient way to apply an access policy to a collection of users • A policy consists of list of bindings and a binding binds members to a role. A Role is a named list of permissions defined by IAM. Organization policy is configuration of restrictions. • Google Cloud Directory Sync - synchronizes (one-way only) users and groups from your existing Active Directory or LDAP system with the users and groups in your Cloud Identity domain. • Best practices - Grant roles to google groups instead of individuals - Be careful about granting service account role - Use identity aware proxy. IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls. An IAM policy contains a list of policy bindings that bind members to a role. To authorize API and service calls, IAM reads the IAM policy that is attached to a resource.	For

Service	Scenario	Recovery Point Objective	Recovery Time Objective	Priority
Product Rating Service	Programmer deleted all ratings accidentally	24 hours	1 hour	Med
Orders service	Database server crashed	0	1 minute	High



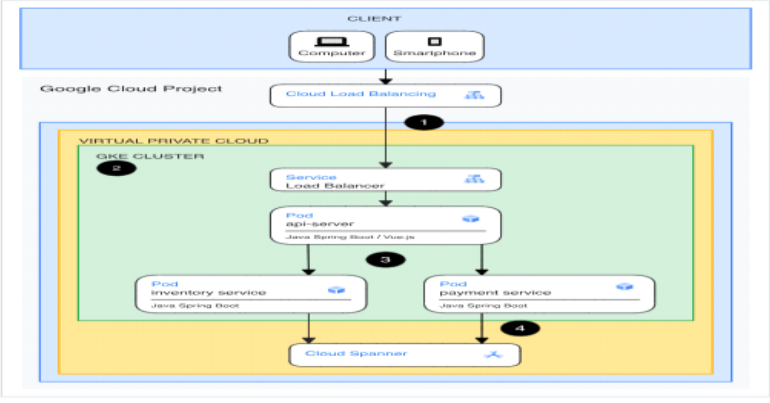
	● A service account is a type of member identity that is used by machines, applications, or services. ● Every Cloud Run service or job is linked to a service account. ● Use a user-managed service account for each Cloud Run service with a minimum set of permissions Cloud Identity, a tool for organizations to define policies and manage their users and groups using the Google Admin console. • Workload Identity Federation, you can provide on-premises or multi-cloud workloads with access to Google Cloud resources by using federated identities instead of a service account key. • @@@@ VPC Design best practices - Group applications in fewer subnets with larger address ranges - For organization with multiple teams, use shared VPC. - Principle of least privilege, Grant network user role at subnet level to user/service account/group - Use single host project, use multiple host project if separate administration policies needed for each network - Isolate sensitive data in it's own VPC network - To connect multiple VPC networks, ○ VPC Spokes (upto 250 active spokes per hub) ○ Peering ○ External routing (NAT) for private IP Address communication ○ VPN ○ Private Service Connect allows consumers to access services privately from inside their VPC network without the need for a network oriented deployment model. ○ VPC Service controls to configure service parameters around your VPC sources ○ Manage traffic with firewall rules when possible. ○ Target filtering for restricting access between application tiers (Web->App->DB) ○ Use default internet gateway for google APIs from VPC Network ○ Use VPC Flow log sampling for reduced volume • Reference architectures   Best practices and reference architectures for VPC design | Cloud Architecture Center | Google Cloud	
Generative AI (Prompt engineering)	- LLMs are large, general-purpose language models that can be pre-trained and then fine-tuned for specific purposes. - When you submit a prompt to an LLM, it calculates the probability of the correct answer from its pre-trained model. The probability is determined through a task called pre-training. - Prompts can be in the form of a question, and are categorized into four categories: zero-shot, one-shot, few-shot, and role prompts. • Zero-shot - no context or examples provided • One-shot - provide one example • Few-shot - at least 2 examples • Role prompt - Frame of reference provided to the model. - Best practices for prompts • Detailed , explicit instructions • Define boundaries for the prompt • Adopt a persona for your input • Keep each sentence concise -large language models, which are a highly sophisticated computer programs trained on gigantic amounts of data that can be text or images. -When you submit a prompt to an LLM, it calculates the probability of the correct answer from its pre-trained model. -The probability is determined through a task called pre-training.	

	-Pre-training an LLM involves feeding a massive dataset of text, images, and code to the model so that it can learn the underlying structure and patterns of the language. -In this way, the LLM works like a fancy autocomplete, suggesting the most common correct response to the prompt. -Hallucinations are words or phrases that are generated by the model that are often nonsensical or grammatically incorrect. -This happens because LLMs can only understand the information they were trained on. -This means that they might not be aware of your business's proprietary or domain-specific data. -Also, they do not have access to real-time information. -LLM does not know anything outside of what it was trained on, and it cannot truly know if that information is accurate. -Hallucinations can be caused by a number of factors, including: The model is not trained on enough data/Noisy or dirty data/not given enough context -They can also make the model more likely to generate incorrect or misleading information	
Serverless Compute	- Cloud run • Built on Knative • Can allocate up to 4 vCPUs and 8GB of memory • Supports source-based approach (i.e. deploy source code, instead of a container image. Cloud Run then builds the source and packages the application into a container Image) • Only pay for the system resources you use while a container is handling web requests, with a granularity of 100ms, and when it is starting or shutting down. • Can run any binary that is compiled for Linux 64 bit • On Cloud Run, your code can either run continuously as a service or as a job. Both services and jobs run in the same environment and can use the same integrations with other services on Google Cloud. ○ Cloud Run services are used to run code that responds to web requests, or events. ○ Cloud Run jobs are used to run code that performs work (a job) and quits when the work is done. ▪ a job can start a single container instance or multiple container instances to run your application code or job script. With multiple container instances running in parallel, the job can complete the task faster. Jobs that run multiple identical container instances are known as Array jobs. • Cloud Run works well for a broad range of applications. It lets you deploy your service with a single containerized app - Cloud functions • Cloud Run functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. Can be invoked asynchronously i.e. Events (Storage or pub/sub) or synchronously over HTTP(s) to process short-lived, event-based actions triggered from other systems such as Cloud Storage, Eventarc, or Pub/Sub etc.	
GKE/Google Container Engine	• Managed Service from GCP. • Use cases ○ AI and ML Operations ○ Data processing at scale ○ Scalable online games platforms ○ Reliable applications under heavy load • GKE has collection of nodes. Nodes are compute engine VMs. Services are deployed to Pods. • Terminologies ○ Node - Node is a Compute Engine/Physical Machine/Virtual Machine. Each node is managed by the Kubernetes control plane and has all the necessary components to run Pods. ○ Cluster - is a set of nodes that can be treated together as a single entity, on which you deploy a containerized application. ○ Pod - is the smallest deployable unit of computing that you can create and manage in Kubernetes. A Pod has one or more containers. ○ Controllers - track and manage the state of your clusters and workloads, based on the desired state ○ Deployments - represents one or more identical Pods, called replicas. ○ StatefulSet - like a Deployment but maintains a persistent unique identity for each of its Pods. ○ Daemonset - lets you add default Pods to some or all of your node ○ Namespaces - provides a mechanism for grouping and selection of resources such as Pods and Services ○ Network Policies - Allows to specify rules for traffic flow within cluster for between pods and outside e.g. control flow at IP Address/Port level ○ Service - is a method for exposing a network application that is running as one or more Pods in your cluster. A Service is an abstraction which defines a logical set of Pods and a policy by which to access them. Below are 5 types, ▪ ClusterIP (default): Internal clients send requests to a stable internal IP address. ▪ NodePort: Clients send requests to the IP address of a node on one or more nodePort values that are specified by the Service. ▪ LoadBalancer: Clients send requests to the IP address of a network load balancer. ▪ ExternalName: Internal clients use the DNS name of a Service as an alias for an external DNS name. ▪ Headless: You can use a headless service when you want a Pod grouping, but don't need a stable IP address. ○ Ingress - Makes network service (HTTP/s) available using protocol-aware configuration mechanism. Helps mapping traffic to different backends based on rules. An Ingress is an API object that defines rules which allow external access to services in a cluster. An Ingress controller fulfils the rules set in the Ingress. ○ Ingress controller - makes ingress work. Kubernetes supports and maintains GKE, GCE and Nginx ○ Edge router - Enforces firewall policy for a cluster. ○ Secrets - Kubernetes allows to create, edit, manage, and delete Secrets using kubectl or config. File or kustomize tool. ○ Kubelet - An Agent that runs on every node in the cluster. It makes sure that containers are running in pod. • Modes ○ Autopilot - built-in hardening and best practices configuration. pay only for the compute resources your running Pods request. Recommended by GCP. GKE manages the underlying infrastructure such as node configuration, autoscaling, auto-upgrades, baseline security configurations, and baseline networking configuration. ○ Standard - you pay for all resources on nodes, regardless of Pod requests. Nodes are to be managed by client. Only use Standard mode if you know you have a specific need to manually manage the node pools and clusters. • Cluster consists of ○ Control plane - Set of management nodes that run system components. Managed by GKE. ○ Nodes - set of Worker nodes. worker node(s) host the Pods that are the components of the application workload	



- GKE also runs per-node agents , called Daemonsets, that provides functionality like log collection and intra-cluster network connectivity
- Generally, you only have one container per pod, but if you have multiple containers with a hard dependency, you can package them into a single pod and share networking and storage resources between them. The Pod provides a unique network IP and set of ports for your containers and configurable options that govern how your containers should run.
- Kubernetes object spec and status : When you create an object in Kubernetes, you must provide the object spec that describes its desired state, as well as some basic information about the object (such as a name). E.g. a Deployment is an object that can represent an application running on your cluster. When you create the Deployment, you might set the Deployment spec to specify that you want three replicas of the application to be running. The Kubernetes system reads the Deployment spec and starts three instances of your desired application--updating the status to match your spec. If any of those instances should fail (a status change), the Kubernetes system responds to the difference between spec and status by making a correction--in this case, starting a replacement instance.
- Pod - Smallest deployable unit is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context.
- Controller - Kubernetes uses lots of controllers that each manage a particular aspect of cluster state. controllers are control loops that watch the state of your cluster, then make or request changes where needed.
- Typical deployment steps
  - Create a deployment using kubectl either via command line parameters or config file (yaml)
  - Create corresponding service for say expose internal IP:Port outside cluster
- GKE Networking
  - Each node gets private IP and connectivity with rest of the network. Public IP is also allocated. Cluster can be configured as private cluster to avoid public connectivity
  - Allocate IP space for nodes, pods and services upon cluster creation
  - Additional features provided to help with ip space allocation and incrementally grow over time
- GKE Autoscaling
  - Vertical - Enabled by default in Autopilot cluster. It observes Pods over time and gradually finds the optimal CPU and memory resources required by the Pods. Setting the right resources is important for stability and cost efficiency. Works by analysing CPU and memory resource consumed by Pods. Pod can only be vertically scaled by recreating it. works well for long-running workloads.
  - Horizontal - meant for scaling applications that are running in Pods based on metrics that express load. You can configure either CPU utilization or other custom metrics (for example, requests per second). In short, HPA adds and deletes Pods replicas, and it is best suited for stateless workers that can spin up quickly to react to usage spikes, and shut down gracefully to avoid workload instability.

- Best practices for operating containers
  - Use the native logging mechanisms of containers
  - Ensure that your containers are stateless and immutable
  - Avoid privileged containers
  - Avoid running as root
  - Make your application easy to monitor
  - Expose the health of your application
  - Carefully choose image version
- Typical deployment of web App,



Observability	- The Ops Agent is the primary agent for collecting telemetry data from your Compute Engine instances (VMs, not required for Standard GKE and Serverless offerings). Combining logging and metrics into a single agent, the Ops Agent uses Fluent Bit for logs, which supports high-throughput logging, and the Open Telemetry Collector for metrics. It can monitor third-party applications like databases, web servers - Google Kubernetes Engine (GKE) includes integration with Cloud Logging and Cloud Monitoring and Google Cloud Managed Service for Prometheus. When you create a GKE cluster that runs on Google Cloud, Cloud Logging and Cloud Monitoring are enabled by default and provide observability specifically tailored for Kubernetes. - VPC Flow Logs is used to monitor network by recording a portion of network flows sent and received by VM instances (including GKE nodes). These logs can be used for network monitoring, traffic analysis, forensics, real-time security analysis, and expense optimization. - Cloud logging is a fully managed service that performs at scale and can ingest application and system log data. Exporting logs to Cloud Storage makes sense for storing logs for more than 30 days. Exporting logs to BigQuery allows you to	

analyze logs and even visualize them in Looker Studio
- Cloud Trace is a distributed tracing system that collects latency data from your applications and displays it in the Google Cloud console. You can track how requests propagate through your application and receive detailed near real-time performance insights. Cloud Trace automatically analyzes all of your application's traces to generate in-depth latency reports that surface performance degradations and can capture traces from App Engine, HTTP(S) load balancers, and applications instrumented with the Cloud Trace API.
- Error Reporting counts, analyzes, and aggregates the errors in your running cloud Services and is available on App engine, compute engine, cloud functions, cloud run, GKE.
- Cloud Profiler continuously analyses the performance of CPU or memory-intensive functions executed across an application.
- Packet Mirroring clones the traffic of specific instances in your Virtual Private Cloud (VPC) network and forwards it for examination. Packet Mirroring captures all ingress and egress traffic and packet data, such as payloads and headers. The mirroring happens on the virtual machine (VM) instances, not on the network. Therefore, Packet Mirroring consumes additional bandwidth on the hosts. Filters can be used to reduce the traffic collected and hence impact on bandwidth. Typical use cases are network & App. Monitoring, Security and Compliance, Network forensics for PCI Compliance
- Performance dashboard - gives you visibility into the performance of your VPC
- Price optimization,
    - Cloud Monitoring prices are based on the:
        - Volume of chargeable metrics ingested.
        - Number of chargeable API calls.
        - Execution of Cloud Monitoring uptime checks.
        - Metrics ingested by using Google Cloud Managed Service for Prometheus.
    - Best practices
        - Use sampling
        - Reduce number of time series
        - Opt for local aggregation

SLO and SLI

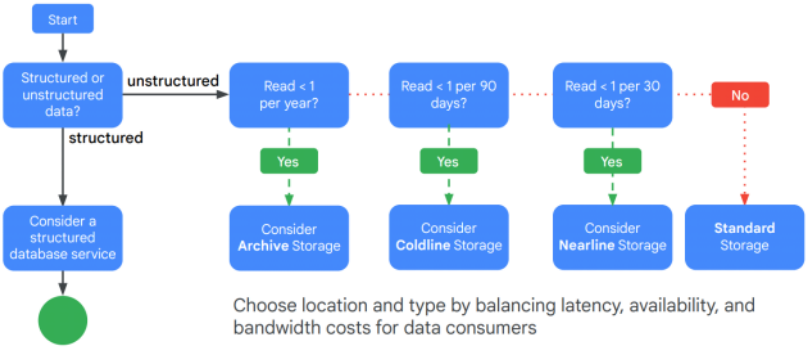
Type of service	Type of SLI	Description
Request-driven	Availability	The proportion of requests that resulted in a successful response. e.g. 99% (Any HTTP status other than 500–599 is considered successful.). Use health checks in Compute Engine or readiness and liveliness probes in GKE to enable the detection and repair of unhealthy instances.
Request-driven	Latency	The proportion of requests that were faster than some threshold. e.g. 90% of requests < 450 ms
Request-driven	Quality	If the service degrades gracefully when overloaded or when backends are unavailable, you need to measure the proportion of responses that were served in an undegraded state. For example, if the User Data store is unavailable, the game is still playable but uses generic imagery.
Pipeline	Freshness	The proportion of the data that was updated more recently than some time threshold. Ideally this metric counts how many times a user accessed the data, so that it most accurately reflects the user experience. e.g. 90% of reads use data written within the previous 1 minute.
Pipeline	Correctness	The proportion of records coming into the pipeline that resulted in the correct value coming out. E.g. 99.99999% of records injected by the prober result in the correct output.
Pipeline	Coverage	For batch processing, the proportion of jobs that processed above some target amount of data. For streaming processing, the proportion of incoming records that were successfully processed within some time window.
Storage	Durability	The proportion of records written that can be successfully read. Take particular care with durability SLIs: the data that the user wants may be only a small portion of the data that is stored. For example, if you have 1 billion records for the previous 10 years, but the user wants only the records from today (which are unavailable), then they will be unhappy even though almost all of their data is readable.

Google SRE - Continuous Improvement To Get Reliability

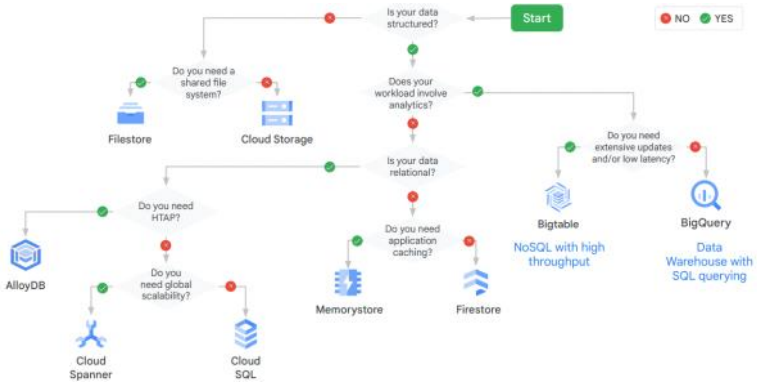
User story	SLO	SLI
Search hotel and flight	Available 99.95%	Fraction of 200 vs 500 HTTP responses from API endpoint measured per month
Search hotel and flight	95% of requests will complete in under 200 ms	Time to last byte GET requests measured every 15 seconds aggregated per 5 minutes
Supply hotel inventory	Error rate of < 0.00001%	Upload errors measured as a percentage of bulk uploads per day by custom metric
Supply hotel Inventory	Available 99.9%	Fraction of 200 vs 500 HTTP responses from API endpoint measured per month
Analyze sales performance	95% of queries will complete in under 10s	Time to last byte GET requests measured every 60 seconds aggregated per 10 minutes

Storage

- Object storage is a computer data storage architecture that manages data as “objects” and not as a file and folder hierarchy (file storage), or as chunks of a disk (block storage). Cloud Storage is Google’s object storage product, it provides immutable storage. Use cases are, BLOB storage, serving website content, storing data for archival and disaster recovery, and distributing large data objects to end users via Direct Download. @@@@GCP (Google Cloud Platform) storage, specifically Cloud Storage, can be either global, regional, or multi-regional.
    - Standard Storage - best for frequently accessed, or “hot,” data. It’s also great for data that is stored for only brief periods of time.
    - Nearline Storage - best for storing infrequently accessed data, like reading or modifying data once per month or less, on average
    - Coldline storage - a low-cost option for storing infrequently accessed data, meant for reading or modifying data, at most, once every 90 days.
    - Archive Storage - Lowest-cost option , best for data that is required to access once per year.
    - Autoclass - a feature which automatically transitions objects to appropriate storage classes based on each object's access pattern.
    - Google recommends that you use Soft Delete instead of Object Versioning to protect against permanent data loss from accidental or malicious deletions.



- Cloud SQL - Managed RDBMS Service for MySQL, PostgreSQL and SQL Server. It can scale up to 128 processor cores, 864 GB of RAM, and 64 TB of storage.
- Spanner - Managed , Horizontally scalable RDBMS Service
- Firestore - Horizontally scalable, NOSQL Cloud database for mobile, web and server deployment. It can synchronize the data on any connected device
- Bigtable - NoSQL Big data Database service. Use cases are internet of things, user analytics and financial data analysis.
- Data Ingestion
    - Dataflow - batch and stream processing framework.
    - Datafusion - ingest batch data with point-n-click interface
- Data Processing
    - Dataproc - fully managed cloud service for running Apache Spark and Apache Hadoop clusters
    - BigQuery - Used for big data analysis and interactive query capabilities. BigQuery is Google Cloud's serverless, highly scalable, and cost-effective cloud data warehouse.
- Governance
    - Dataprep - serverless service for visually exploring, cleaning, and preparing structured and unstructured data for analysis, reporting, and machine Learning. operated by Trifacta based on Trifacta wrangler.
- Storage Transfer Service enables you to import large amounts of online data into Cloud Storage quickly and cost-effectively
- Transfer Appliance, is a rackable, high-capacity storage server that you lease from Google Cloud.
- Decision matrix

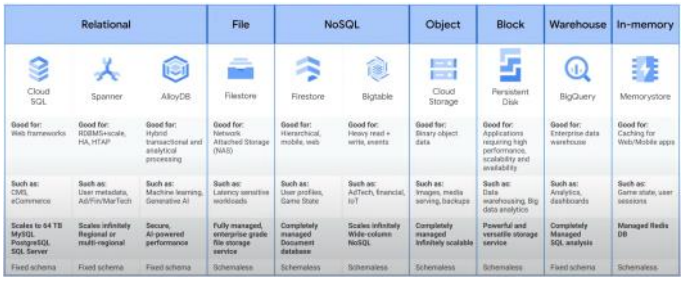


Different data storage services have different availability SLAs

Storage Choice	Availability SLA %
Cloud Storage (multi-region bucket)	>=99.95
Cloud Storage (regional bucket)	99.9
Cloud Storage (coldline)	99.0
Spanner (multi-region)	99.999
Spanner (single region)	99.99
Firestore (multi-region)	99.999
Firestore (single region)	99.99

Calculate the total cost per GB when choosing a storage service

- Bigtable and Spanner would be too expensive for storing smaller amounts of data
- Firestore is less expensive per GB, but you also pay for reads and writes
- Cloud Storage is relatively cheap, but you can't run a database in storage
- BigQuery storage is relatively cheap, but doesn't provide fast access to records and you have to pay for running queries



Persistent Disk Performance:
The performance of a Google Cloud Persistent Disk, including its throughput and IOPS, is directly related to the total capacity of the disk attached to a virtual machine (VM) instance.
Capacity and Performance:
As you increase the capacity of the Persistent Disk, the system allocates more resources to that disk, leading to higher throughput and IOPS.

Storage option	Workload types
Persistent Disk	• IOPS-intensive or latency-sensitive applications • Databases • Shared read-only storage • Rapid, durable VM backups
Hyperdisk	• IOPS-intensive or latency-sensitive applications • Databases • Shared read-only storage • Rapid, durable VM backups • Scale-out analytics
Local SSD	• Flash-optimized databases • Hot-caching for analytics • Scratch disk
Filestore	• Lift-and-shift on-premises file systems • Shared configuration files • Common tooling and utilities • Centralized logs
Managed Lustre	• AI and ML workloads • HPC
NetApp Volumes	• Lift-and-shift on-premises file systems • Shared configuration files • Common tooling and utilities • Centralized logs • Windows workloads
Cloud Storage	• Streaming videos • Media asset libraries • High-throughput data lakes • Backups and archives • Long-tail content

Firewall best practices	- Implement least-privilege principles. Block all traffic by default and only allow the specific traffic you need. This includes limiting the rule to just the protocols and ports you need. - Use hierarchical firewall policy rules to block traffic that should never be allowed at an organization or folder level. - For "allow" rules, restrict them to specific VMs by specifying the service account of the VMs. - If you need to create rules based on IP addresses, try to minimize the number of rules. It's easier to track one rule that allows traffic to a range of 16 VMs than it is to track 16 separate rules - Turn on Firewall Rules Logging and use Firewall Insights to verify that firewall rules are being used in the intended way. Firewall Rules Logging can incur costs, so you might want to consider using it selectively.	
Identity aware proxy	IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls. IAP policies scale across your organization. You can define access policies centrally and apply them to all of your applications and resources. When you assign a dedicated team to create and enforce policies, you protect your project from incorrect policy definition or implementation in any application. When? -Use IAP when you want to enforce access control policies for applications and resources. With IAP, you can set up group-based application access: a resource could be accessible for employees and inaccessible for contractors, or only accessible to a specific department.	

Load balancer and latency	

Load balancer	Deployment mode	Traffic type	Network Service Tier	Load-balancing scheme
Application Load Balancers	Global external	HTTP or HTTPS	Premium	EXTERNAL_MANAGED
	Regional external	HTTP or HTTPS	Standard	EXTERNAL_MANAGED
	Classic	HTTP or HTTPS	Global in Premium Regional in Standard	EXTERNAL
	Internal Always regional	HTTP or HTTPS	Premium	INTERNAL_MANAGED
Proxy Network Load Balancers	Global external	TCP with optional SSL offload	Global in Premium Regional in Standard	EXTERNAL
	Regional external	TCP	Standard only	EXTERNAL_MANAGED
	Internal Always regional	TCP without SSL offload	Premium only	INTERNAL_MANAGED
Passthrough Network Load Balancers	External Always regional	TCP, UDP, ESP, GRE, ICMP, and ICMPv6	Premium or Standard	EXTERNAL
	Internal Always regional	TCP or UDP	Premium only	INTERNAL

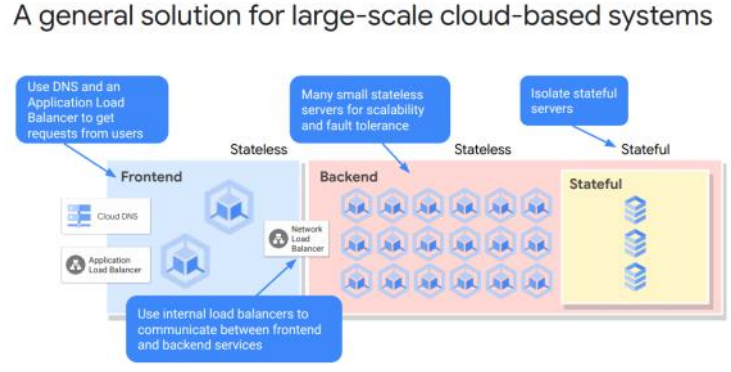
- Internal load balancers and Cloud Service Mesh don't support user-facing traffic
- External Application load balancer is global in scope and pass through network load balancer is regional in scope. For premium tier, traffic enters at closest point of presence and then onto google's network.
- SSL proxy is a global load balancing service for encrypted, non-HTTP traffic
- Network load balancing is a regional, non-proxied load balancing service. In other words, all traffic is passed through the load balancer, instead of being proxied, and traffic can only be balanced between VM instances that are in the same region, unlike a global load balancer.
- The internal TCP/UDP load balancer is a regional, private load balancing service for TCP- and UDP-based traffic. In other words, this load balancer enables you to run and scale your services behind a private load balancing IP address This means that it is only accessible through the internal IP addresses of virtual machine instances that are in the same region.
- Google Cloud Internal HTTP(S) Load Balancing is a proxy-based, regional Layer 7 load balancer that also enables you to run and scale your services behind an internal load balancing IP address. Backend services support the HTTP, HTTPS, or HTTP/2 protocols.
- Network access latency improves when using external load balancer as,
    - It maintains persistent connections open to serving backends.
    - Does ssl offload
    - Automatic upgrade to HTTP/2.HTTP/2 can reduce the number of packets needed, by using improvements in binary protocol, header compression, and connection multiplexing.
- Additional measures for improvement in latency
    - Use cloud CDN for cacheable content
    - serving static content directly from Cloud Storage through the external Application Load Balancer
    - Deploying web servers close to user's region
- In GKE, the internal Application Load Balancer is a proxy-based, regional, Layer 7 load balancer that enables you to run and scale your services behind an internal load balancing IP address. GKE Ingress objects support the internal Application Load Balancer natively through the creation of Ingress objects on GKE clusters.
- HTTP Load Balancer flow,
    - A global forwarding rule directs incoming requests from the internet to a target HTTP proxy.
    - The target HTTP proxy checks each request against a URL map to determine the appropriate backend service for the request. For example, you can send requests for www.example.com/audio to one backend service, which contains instances configured to deliver audio files, and requests for www.example.com/video to another backend service, which contains instances configured to deliver video files.
    - The backend service directs each request to an appropriate backend based on serving capacity, zone, and instance health of its attached backends.

Resource Management	

- The resource manager lets you hierarchically manage resources by project, folder, and organization.
- Project accumulates the consumption of all its resources so can be used for tracking
- Project quotas prevent runaway consumption in case of an error or malicious attack. Quotas do not guarantee that resources will be available at all times
- Labels are a utility for organizing Google Cloud resources. They are attached to VM, disk, snapshot. e.g. create a label to define the environment of your virtual machines
- Network tags are applied to instances only and mainly used for networking such as firewall rules and custom routes.
- Budget can be used to track spending and received alerts on breach via email/pub-sub
- **Recommendation: Labelling all your resources and exporting your billing data to BigQuery to analyze your spend.

Microservices design	

General blueprint,



12 factors

- 1 -> The codebase should be tracked in a version control such as Git.
- 2 -> Dependencies should be declared explicitly and stored in version control. Dependency tracking is performed by language-specific tools such as Maven for Java and Pip for Python
- 3 -> Every application has a configuration for different environments like test, production, and development. This configuration should be external to the code and usually kept in environment variables for deployment flexibility

- 4 -> Every backing service such as a database, cache, or message service, should be accessed via URLs and set by configuration. The backing services act as abstractions for the underlying resource. The aim is to be able to swap one backing service for a different implementation easily.
- 5 -> The software deployment process should be broken into three distinct stages: build, release, and run. Each stage should result in an artifact that's uniquely identifiable.
- 6 -> each service should have its own datastore and caches using, for example, Memorystore to cache and share common data between services used.
- 7 -> Services should be exposed using a port number.
- 8 -> The application should be able to scale out by starting new processes and scale back in as needed to meet demand/load.
- 9 -> Applications should be written to be more reliable than the underlying infrastructure they run on. This means they should be able to handle temporary failures in the underlying infrastructure and gracefully shut down and restart quickly.
- 10 -> The aim should be to have the same environments used in development and test/staging as are used in production. Infrastructure as code and Docker containers make this easier
- 11 -> decouple the collection, processing, and analysis of logs from the core logic of your apps. Logging should be to standard output and aggregating into a single source.
- 12 -> Admin processes (cron jobs, cloud tasks etc.) should be automated and repeatable, not manual, processes

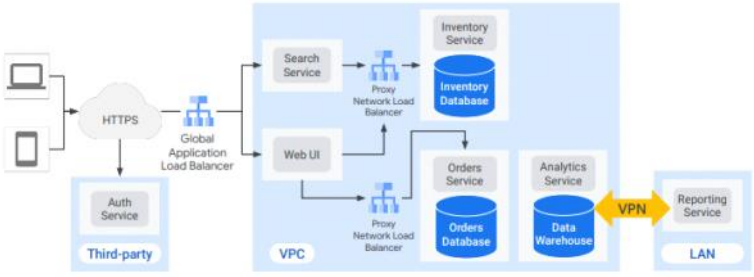
Service specific requirements gathering and considerations,

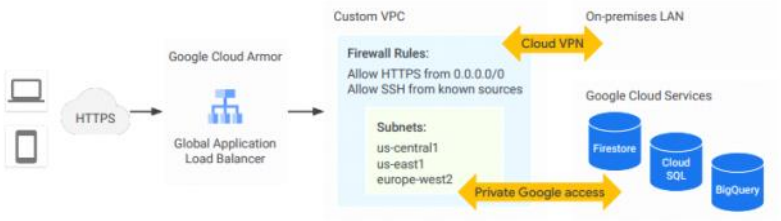
Service	Structured or Unstructured	SQL or NoSQL	Strong or Eventual Consistency	Amount of Data (MB, GB, TB, PB, ExB)	Read only or Read/Write
Inventory	Structured	NoSQL	Strong	GB	Read/Write
Inventory uploads	Unstructured	N/A	N/A	GB	Read only
Orders	Structured	SQL	Strong	TB	Read/Write
Analytics	Structured	SQL	Eventual	TB	Read only

Service	Persistent Disk	Cloud Storage	Cloud SQL	Firestore	Bigtable	Spanner	BigQuery
Inventory				X			
Inventory uploads		X					
Orders			X				
Analytics							X

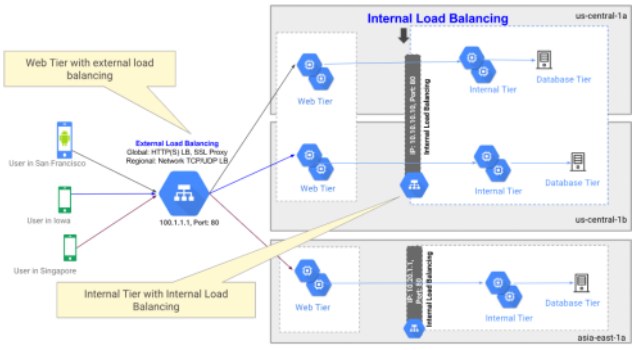
Google Cloud

Service	Internet facing or Internal only	Application Load Balancer	Proxy Network Load Balancer	Passthrough Network Load Balancer	Multi-Regional?
Search	Internet facing	X			Yes
Inventory	Internal		X		No
Analytics	Internet facing	X			No
Web UI	Internet facing	X			Yes
Orders	Internal		X		No





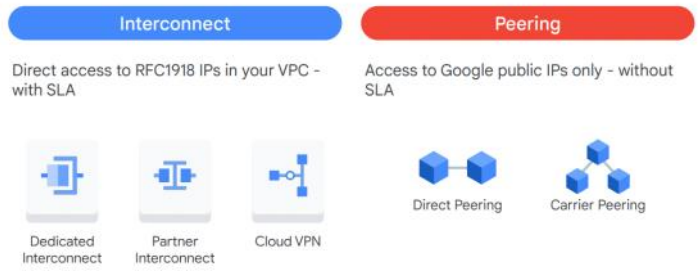
Deployment Blueprint



- A Shared VPC network for each environment (production, development, and testing) connects resources from multiple projects to the VPC network.
- VPC peering is a method for connecting two VPCs so that they can communicate as if they were part of the same network, without the need for traffic to traverse the public internet. VPNs, on the other hand, create secure, encrypted connections over the public internet.
- Carrier peering - service that allows you to connect your Google Cloud VPC network to networks outside of Google Cloud, primarily to networks in Azure or AWS. It's essentially a private, dedicated connection between your GCP network and another cloud provider's network, providing significantly better performance, security, and control compared to traditional internet-based connections.

Comparison of Interconnect options

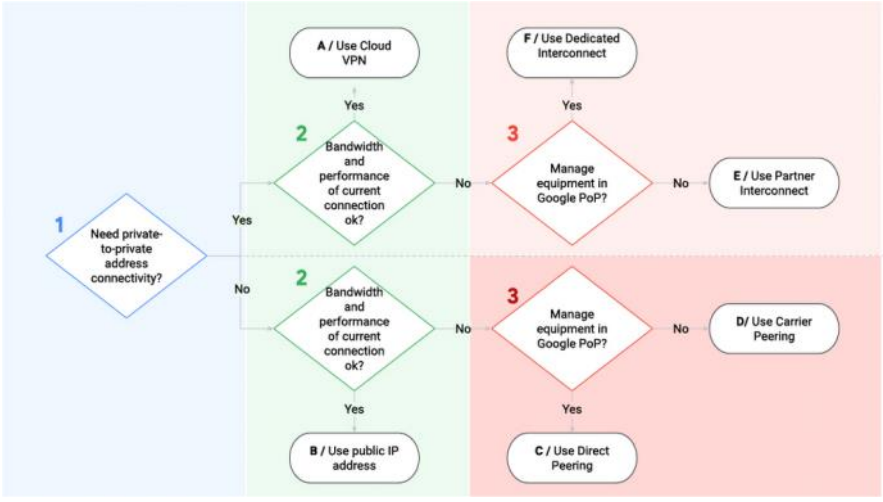
Connection	Provides	Capacity	Requirements	Access Type
VPN tunnel	Encrypted tunnel to VPC networks through the public internet	1.5–3 Gbps per tunnel	Remote VPN gateway	Internal IP addresses
Dedicated Interconnect	Dedicated, direct connection to VPC networks	10 Gbps or 100 Gbps per link	Connection in colocation facility	
Partner Interconnect	Dedicated bandwidth, connection to VPC network through a service provider	50 Mbps – 50 Gbps per connection	Service provider	
Cross-Cloud Interconnect	Dedicated physical connection between VPC network and network hosted by service provider	10 Gbps or 100 Gbps per connection	Primary and redundant ports (Google Cloud and remote cloud service provider)	



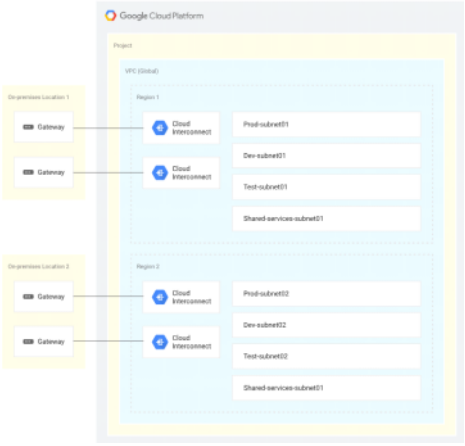
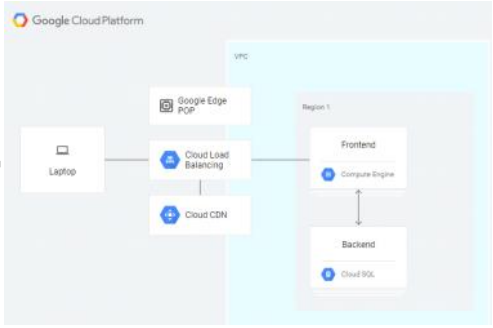
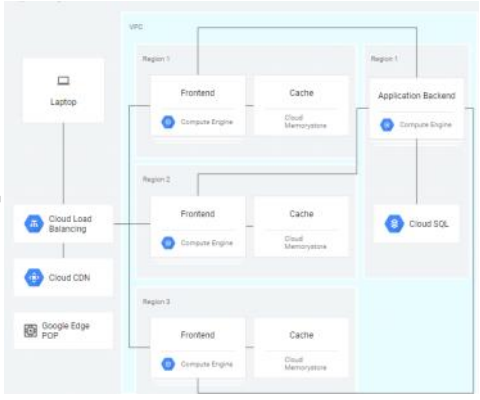
- VPC Service control benefits,
  - Access from unauthorized networks using stolen credentials
  - Data exfiltration by malicious insiders or compromised code
  - Public exposure of private data caused by misconfigured IAM policies
  - Monitoring access to services
  - A service perimeter creates a security boundary around Google Cloud resources. A service

perimeter allows free communication within the perimeter but, by default, blocks communication to Google Cloud services across the perimeter.

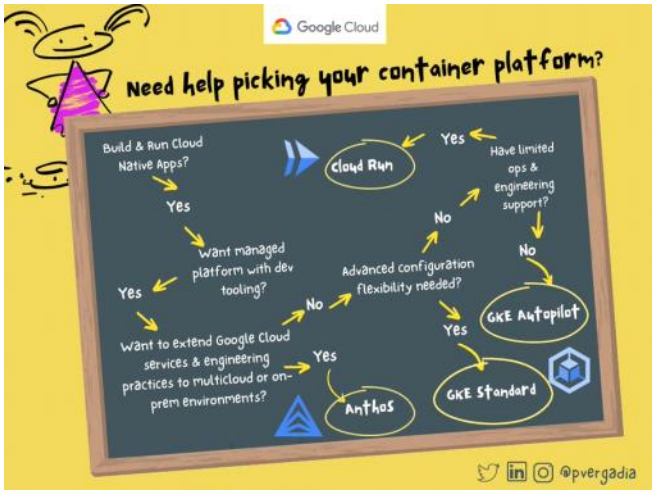
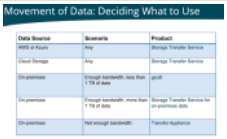
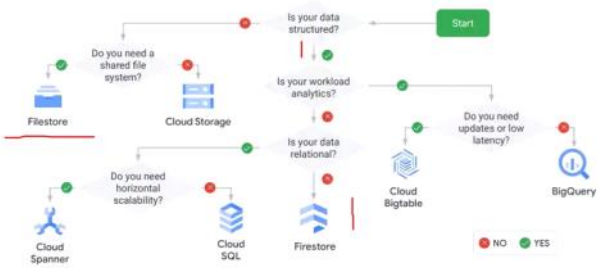
- Networking decision tree
  - Private to Private - VPN or interconnect
  - **Private/Public to Public - Public IP/Peering

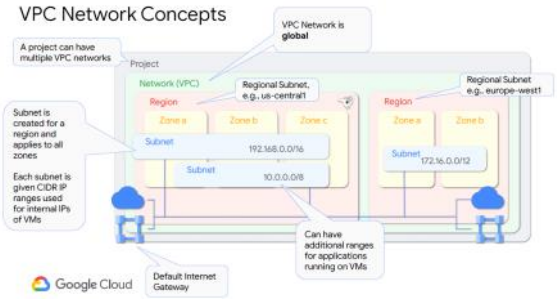


- Virtual Private Cloud (VPC) firewall rules control connectivity to and from workloads in the Shared VPC networks.
- A Cloud NAT gateway allows outbound connections to the internet from resources in these networks without external IP addresses.
- Cloud Interconnect provides connectivity between your on-premises network and your Virtual Private Cloud (VPC) network through a supported service provider(You can choose between different Cloud Interconnect options, including Dedicated Interconnect or Partner Interconnect.)
- Cloud VPN connects to other cloud service providers.
- A Cloud DNS private zone hosts DNS records for your deployments in Google Cloud
- VPC Design best practices (ref: Best practices and reference architectures for VPC design | Cloud Architecture Center | Google Cloud)
  - Virtual Private Cloud (VPC): A virtual system that provides global, scalable networking functionality for your Google Cloud workloads. VPC includes VPC Network Peering, Private Service Connect, private services access, and Shared VPC.
  - Network Connectivity Center : provides full bandwidth between workload VPCs and provides transitivity between workload VPCs. An orchestration framework that simplifies network connectivity among spoke resources that are connected to a central management resource called a hub.
  - Cloud Interconnect: A service that extends your external network to the Google network through a high-availability, low-latency connection.
  - Cloud VPN: A service that securely extends your peer network to Google's network through an IPsec VPN tunnel.
  - Cloud Router: **A distributed and fully managed offering that provides Border Gateway Protocol (BGP) speaker and responder capabilities. Cloud Router works with Cloud Interconnect, Cloud VPN, and Router appliances to create dynamic routes in VPC networks based on BGP-received and custom learned routes.
  - Use custom mode VPC networks (You can't connect two auto mode VPC networks together using VPC Network Peering because their subnets use identical primary IP ranges.;Auto mode subnets all have the same name as the network.;
  - Group applications into fewer subnets with larger address ranges
  - Start with a single VPC network for resources that have common requirements
  - Use Shared VPC for administration of multiple working groups
  - Grant the network user role at the subnet level
  - Use a single host project if resources require multiple network interfaces
  - Use multiple host projects if resource requirements exceed the quota of a single project
  - Use multiple host projects if you need separate administration policies for each VPC network
  - Isolate sensitive data in its own VPC network
  - Choose the VPC connection method that meets your cost, performance, and security needs
    - **Network peering - Recommended to connect VPN networks; No additional charges
    - External routing
    - Cloud VPN (Cloud to Cloud) - a managed service to connect VPC networks by creating IPsec tunnels between sets of endpoints
    - Cloud Interconnect (On-prem to GCP) - extends your on-premises network to Google's network

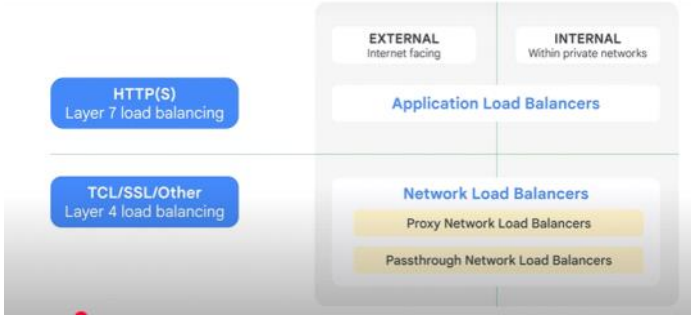
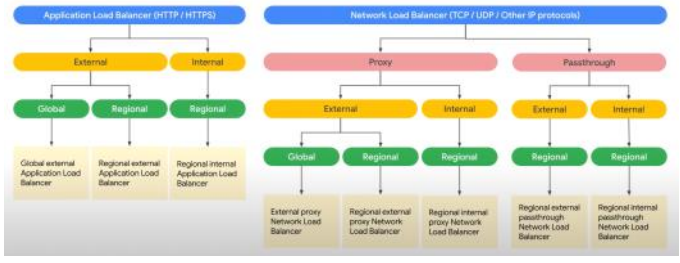
	through a highly available, low-latency connection. You can use Dedicated Interconnect to connect directly to Google or use Partner Interconnect to connect to Google through a supported service provider ○ Multiple network interfaces • Create a shared services VPC if multiple VPC networks need access to common resources but not each other • When connecting with on-premise environment, use dynamic routing when possible • Use a connectivity VPC network to scale a hub-and-spoke architecture with multiple VPC networks • Typical VPC Layout  • App deployment ○ Single Region  ○ Multi region  • VPC Networks are global. Need to create subnet in for the region. A project can have multiple networks. Resources across regions can reach each other without interconnect • IP Address ranges can not overlap. Subnets are expandable without downtime. Maximum of 8 interfaces per VM.	
Hierarchy	Project -> Network -> Region -> Zone	
Block Storage	Design an optimal storage strategy for your cloud workload | Cloud Architecture Center | Google Cloud	
Cloud HSM	host encryption keys and perform cryptographic operations in a cluster of FIPS 140-2 Level 3 certified HSMs	
Cloud DLP	- DLP (Data loss prevention) - Scans data in cloud storage, BigQuery etc. and detects sensitive data like emails, cards etc. - Fully managed service to discover, classify and protect sensitive data - Use case : protecting PII data. (De-identification is the process of removing identifying information from data.) - Cloud DLP provides fast scalable classification and redaction for sensitive data elements like credit card numbers, names, social security numbers, US and selected international identifier numbers, phone numbers, and Google Cloud credentials.	

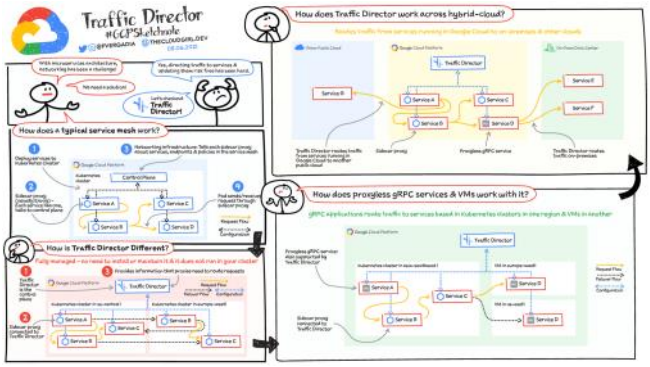
<table>
<tr>
<td>Cloud Storage</td>
<td>- No Allocation; Pay for storage consumed
- Max object size: 5TB
- Uses the key to encrypt object's data, checksum and CRC
- Standard server-side keys to encrypt object metadata
-
<table>
<tr><th></th><th>Standard</th><th>Nearline</th><th>Coldline</th><th>Archive</th></tr>
<tr><td>Use case</td><td>"Hot" data and/or stored for only brief periods of time like data-intensive computations</td><td>Infrequently accessed data like data backup, long-tail multimedia content, and data archiving</td><td>Infrequently accessed data that you read or modify at most once a quarter</td><td>Data archiving, online backup, and disaster recovery</td></tr>
<tr><td>Minimum storage duration*</td><td>None</td><td>30 days</td><td>90 days</td><td>365 days</td></tr>
<tr><td>Retrieval cost</td><td>None</td><td>$0.01 per GB</td><td>$0.02 per GB</td><td>$0.05 per GB</td></tr>
<tr><td>Availability SLA</td><td>99.95% (multi/dual)
99.90% (region)</td><td colspan="2">99.90% (multi/dual)
99.00% (region)</td><td>None</td></tr>
<tr><td>Durability</td><td colspan="4">99.999999999%</td></tr>
</table>
</td>
<td>• Streaming videos
• Media asset libraries
• High-throughput data lakes
• Backups and archives
• Long-tail content</td>
</tr>
<tr>
<td>ClearBlade IOT Core</td>
<td></td>
<td>- Data ingestion from IOT Devices</td>
</tr>
<tr>
<td>Cloud Datafusion</td>
<td>- Powered by open source project CDAP
- Visual data wrangling and data pipelining tool</td>
<td>- Code-free ETL
- End-to-end data lineage for root cause and impact analysis</td>
</tr>
<tr>
<td>Cloud DataPrep</td>
<td>- Used to normalize data before processing</td>
<td>- For analysis, reporting and machine learning</td>
</tr>
<tr>
<td>Cloud DataProc</td>
<td>- Managed Apache Spark and Hadoop service
- Runs serverless or on Kubernetes/VMs
- Built-in integration with cloud storage, BigQuery and BigTable (Datastore)</td>
<td>- Large scale batch processing, querying, streaming and machine learning</td>
</tr>
<tr>
<td>Cloud Dataflow</td>
<td>- fully managed service that can be used to process both streams and batches of data</td>
<td></td>
</tr>
<tr>
<td>Cloud NAT</td>
<td>- Outbound Internet Access
- Allow resources in VPC to create outbound connections without requiring external IP Addresses</td>
<td></td>
</tr>
<tr>
<td>Cloud Armor</td>
<td>- DDOS defence service and WAF
- Supports layer 7 WAF Rules.
- Pre-defined rules for common attacks like DDOS, SQL Injection and cross-site scripting</td>
<td></td>
</tr>
<tr>
<td>NGFW Appliance</td>
<td>- To control traffic between resources
- Deep packet inspection
- FQDN filtering
- TSL/SSL traffic inspection</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>- Cloud Endpoints
• API management gateway that helps you develop, deploy, and manage APIs on any Google Cloud backend. It runs on Google Cloud and leverages a lot of Google's underlying infrastructure.Access can be controlled by IAM
• Three endpoint options like OpenAPI,gRPC, framework for Python & Java
• Protect and monitor public APIs
• Validate each call with JWT and google API keys
- APIgee
• Apigee is an API management platform built for enterprises, with deployment options on cloud, on-premises, or hybrid. The feature set includes an API gateway, customizable portal for onboarding partners and developers, monetization, and deep analytics around APIs. You can use Apigee for any http/https backends, no matter where they are running (on-premises, any public cloud, etc.).
• Provides monetization, traffic control, throttling, security and hybrid (third -parties) integration.</td>
<td></td>
</tr>
<tr>
<td>Kubernetes</td>
<td>- Node pool - create pods with same configuration. Ability to configure nodes
- 3 types of services (Pods in a deployment are regularly created and destroyed, causing their IP addresses to change constantly, Makes it difficult for frontend applications to identify which pods to connect to)
• ClusterIP
• NodePort
• Load balancer
- Modes
• Autopilot - pre-configured cluster's underlying infrastructure with an optimized cluster configuration that is ready for production workloads
• Standard - Provides advanced configuration flexibility over cluster's underlying infrastructure
- Statefulset - Each pod has attached data volume. If Pod is restarted then its existing volume is reattached.
- Binary authorization ensures that internal processes that safeguard the quality and integrity of your software have been successfully completed before an application is deployed to your production environment. Binary authorization is a Google Cloud service
- Google Cloud's operations suite offers a fully managed logging, metrics collection, monitoring, dashboarding, and alerting solution that watches all sides of your hybrid or multi-cloud network
 
</td>
<td></td>
</tr>
</table>

		
Storage transfer appliance	- High capacity, ruggedized, tamper resistant storage device to ship data to upload facility - Data encrypted using AES 256 and uploaded to Cloud storage	- Good fit for data > 10TB
Storage transfer service	- Can be used to transfer large amounts of data from on-premises. Tens of gbps of network connection is needed	
Managed Instance Groups	- Compute Engine VMs - Target Utilization metrics • Average CPU Utilization • HTTP Load balancing based either on request per second or utilization - Cloud monitoring metrics - Autoscaling cannot be used if MIG has stateful configuration - Preemptible instances - offered at discount for a period (Max 24 hours). Not covered by any SLA. - Spot instances - pre-emption (release) can be handled via script (for cleanup etc.)  When converting physical server CPUs to Google Compute Engine vCPUs, it is important to remember that vCPU on a Compute Engine instance is implemented as a single hyper-thread on an Intel Xeon Processor. Since each Xeon Processor has two hyper-threads, that means you need to multiply the number of cores by two to get the number of threads, and thus the number of vCPUs. Here are the steps to calculate this: 1.Take the number of servers (4) 2.Multiply that number by the number of cores per server (4) 3.Multiply that number by the number of hyper-threads (2) 4 servers x 4 cores per server x 2 hyper-threads = 32 vCPUs	
GKE	- Managed Kubernetes service - Declarative configuration - using YAML - Imperative Configuration - using commands (CLI) - Pod Autoscaling • Horizontal - automatically increases or decreases Pods based on CPU, memory or custom metrics • Vertical - lets you analyse and set CPU and memory resourced for POD. - Cluster autoscaling • Automatically resizes node pool on demands of workload • Automatically balances nodes across availability zones - Advantages over self-managed K8S • Node pools to designate subsets of nodes within a cluster • Node auto-repair to maintain node health and availability • Logging and monitoring with google cloud's operation suite	- Containerized workloads of sufficient complexity - Stateful microservices - Hybrid and multi-cloud workloads - Application Modernization
Serverless NEG	**A network endpoint group (NEG) specifies a group of backend endpoints for a load balancer. A serverless NEG is a backend that points to a Cloud Run, App Engine, Cloud Run functions, or API Gateway service.	
App Engine	- Standard ○ Containers are preconfigured with runtimes ○ Instance class determines amount of memory and CPU ○ Can scale to zero if no traffic ○ Cannot SSH or use custom libraries	Larger units of code triggered by cloud events or http requests -Web Server -Django App -Web and Mobile backend

	- Flexible ○ Greater CPU And memory ○ Custom libraries ○ Custom docker containers ○ Access resources in same network	-Microservices
Cloud run	-Serverless container platform. Pay for what you use. -Supports API Endpoints -Request based Auto scaling and scale to zero -Built-in traffic mgmt.	- Runs scheduled jobs - Useful for heavily I/O-bound work - Can control # of concurrent tasks • Websites and Web Applications • APIs and Microservices • Processing streaming data from Pub/Sub • Custom runtimes
Cloud pub/sub	Serverless	- Ingestion - To process work in batches or control flow
Dataflow	Serverless	Stream and batch data processing
Dataproc	Serverless	Spark batch uploads
Networking	 - Best practices • Start with a single VPC network for resources that have common requirements. • Use Shared VPC for administration of multiple working groups. • Grant the network user role at the subnet level. • Use a single host project if resources require multiple network interfaces. • Use multiple host projects if resource requirements exceed the quota of a single project. • Use multiple host projects if you need separate administration policies for each VPC. • Use custom mode subnets in your enterprise VPC networks. • Group applications into fewer subnets with larger address ranges. • Start with a single VPC network for resources that have common requirements. • Use Shared VPC for administration of multiple working groups. • Grant the network user role at the subnet level. • Use a single host project if resources require multiple network interfaces. • Use multiple host projects if resource requirements exceed the quota of a single project. • Use multiple host projects if you need separate administration policies for each VPC. • Create a single VPC network per project to map VPC network quotas to projects. • Create a VPC network for each autonomous team, with shared services in a common VPC network. • Create VPC networks in different projects for independent IAM controls. • Isolate sensitive data in its own VPC network. • Choose the VPC connection method that meets your cost, performance, and security needs. • Use Network Connectivity Center VPC spokes. • Use VPC Network Peering if you need to insert NVAs or if your application doesn't support Private Service Connect. • Use external routing if you don't need private IP address communication. • Use Cloud VPN to connect VPC networks that host service access points that are not transitively reachable over Network Connectivity Center. • Use multi-NIC virtual appliances to control traffic between VPC networks through a cloud device. • Use dynamic routing when possible. • Use a connectivity VPC network to scale a hub-and-spoke architecture with multiple VPC networks. • Identify clear security objectives. • Limit external access. • Define service perimeters for sensitive data. • Manage traffic with Google Cloud firewall rules when possible. • Use fewer, broader firewall rule sets when possible. • Isolate VMs using service accounts when possible. • Use automation to monitor security policies when using tags. • Use additional tools to help secure and protect your apps. • Use fixed external IP addresses with Cloud NAT. • Reuse IP addresses across VPCs with Cloud NAT. • Use Private DNS zones for name resolution. • Use the default internet gateway where possible. • Add explicit routes for Google APIs if you need to modify the default route. • Deploy instances that use Google APIs on the same subnet. • Tailor logging for specific use cases and intended audiences. • Increase the log aggregation interval for VPC networks with long connections. • Use VPC Flow Logs sampling to reduce volume. • Remove additional metadata when you only need IP and port data. Use Network Intelligence Center to get insights into your networks - VMs must have internal IP and can have external IP. - **VPC Network Peering allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization.	

	- For this, 10.0.0.0/24 -> 24-16 -> (2^8 -4) -> Max. 252 IP Addresses are allowed since first & last 2 are reserved by GCP - A Project can have one or more networks. A Network can span multiple zones within region (VMs on same subnet but different zones) • Increased availability by placing VMs in multiple zones within same subnetwork . Using single subnetwork allows to create firewall rule. - Tags are way to allocate strings to infra like VMs and then can use them in firewall rules. - Default behaviour of network is to disallow all incoming traffic but allow outgoing - VM needs external IP to communicate with external world. With Private google access, it can access GCP Services. - Cloud NAT (Network Address translation) provides internet access (Non google) to private VM (those without external IP). This allows only ingress as external services only get Cloud NAT IP and not the Internal IP of private VM. - Firewall rules - protect your virtual machine instances from unapproved connections, both inbound and outbound, known as ingress and egress, respectively. Essentially, every VPC network functions as a distributed firewall. - Interconnect - VM from GCP to On-prem VM. - Peering - access google sources from on-prem, • VPC Peering - both networks on GCP. • Shared VPC - both networks on GCP; need to be from same org but different projects. Shared VPC vs. VPC peering Consideration / Shared VPC / VPC Network Peering Across Organizations / No / Yes Within Project / No / Yes Network Administration / Centralized / Decentralized - VPN - to securely connect on-premise network to GCP VPC Network • Class VPN - 99.9% availability • HA VPN - 99.99% availability - Bastion host - a temporary VM with external IP. A bastion host provides an external facing point of entry into a network containing private network instances. This host provides a single point of secure access and can be stopped to disable inbound SSH.	
VPC (Best practices and reference architectures for VPC design | Cloud Architecture Center | Google Cloud)	Virtual Private Cloud (VPC) provides networking functionality to Compute Engine virtual machine (VM) instances, Google Kubernetes Engine (GKE) containers, and serverless workloads. VPC provides networking for your cloud-based services that is global, scalable, and flexible.	-Start with a single VPC network for resources that have common requirements. -Use Shared VPC for administration of multiple working groups. -Grant the network user role at the subnet level. -Use a single host project if resources require multiple network interfaces. -Use multiple host projects if resource requirements exceed the quota of a single project. -Use multiple host projects if you need separate administration policies for each VPC.
Private Service Connect	- Capability of Google Cloud networking that allows consumers to access managed services privately from inside their VPC network. With Private Service Connect, consumers can use their own internal IP addresses to access services without leaving their VPC networks. Traffic remains entirely within Google Cloud.	
Cloud NAT	- Cloud NAT provides network address translation (NAT) for outbound traffic to the internet, Virtual Private Cloud (VPC) networks, on-premises networks, and other cloud provider networks.	
Network Service tier	Network Service Tiers lets you optimize connectivity between systems on the internet and your Google Cloud instances. Premium Tier delivers traffic on Google's premium backbone, while Standard Tier uses regular ISP networks.	
VPC Flow logs	VPC Flow Logs records a sample of packets sent from and received by virtual machine (VM) instances, including instances used as Google Kubernetes Engine nodes, and packets sent through VLAN attachments for Cloud Interconnect and Cloud VPN tunnels.	- Network Monitoring - Understanding network usage and optimizing network traffic expenses - Network forensics (Incident diagnostics)
VPC service controls	**Mitigate data exfiltration risks by enforcing a security perimeter to isolate resources of multi-tenant Google Cloud services. Configure private communications between cloud resources from VPC networks spanning cloud and on-premise deployments. Keep sensitive data private and take advantage of the fully managed storage and data processing capabilities.	- Mitigate threats such as data exfiltration - Isolate parts of the environment by trust level - Secure access to multi-tenant services
Network connectivity center	Network connectivity Center - Connecting enterprise sites by using Google's network as WAN - Hub and Spoke model - on-premises networks (spokes) connect to Network connectivity center (hub) - Connectivity can be through cloud VPN, interconnect and through a router appliance	
Interconnect	- Dedicated - Physical connection to Google's network in a supported co-location facility - High bandwidth needs (10s of Gbps)	

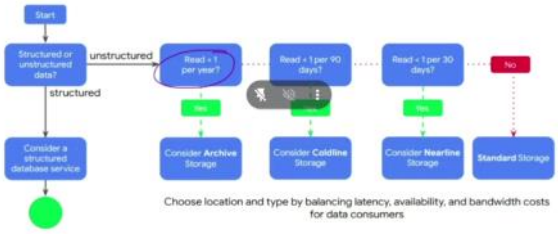
	- Can reach google's network directly to colocation - Don't want traffic to pass through a service provider network - Partner - Bands needs are in the 100s of Mbps or low Gbps - Not able to reach Google's network directly - Don't want to setup and maintain routing equipment at colocation facility - VLAN attachments (also known as interconnect Attachments) determine which Virtual Private Cloud (VPC) networks can reach your on-premises network through a Dedicated Interconnect connection.	
Cloud VPN	securely extends your peer network to Google's network through an IPsec VPN tunnel. Traffic is encrypted and travels between the two networks over the public internet. Cloud VPN is useful for low-volume data connections.	
Cloud router	Cloud Router enables you to dynamically exchange routes between your Virtual Private Cloud (VPC) and peer network by using Border Gateway Protocol (BGP). For example, if you use a Cloud VPN tunnel to connect your networks, you can use Cloud Router to establish a BGP session with a router in your peer network over a Cloud VPN tunnel. The peer network can be an on-premises network, multicloud network, or another VPC network. Cloud Router automatically learns new subnet IP address ranges in your VPC network and can announce them to your peer network.	
Multicloud offerings	Multicloud solutions - Anthos ○ Multicluster mgmt - fleets and connects ○ Config. Mgmt - Anthos config. Mgmt. ○ Service Mgmt - Anthos Service Mesh (AWS Only) ▪ Service mesh allows to use simple YAML config file to define how containers interact and external network connections - Bigquery omni - multicloud analytics solution that access and analyze data residing in multiple cloud environments - Looker - multicloud business intelligence Anthos has a Policy Controller that checks configuration files and enforces their rules against every Kubernetes API request. With this, we can create guardrails for our applications by defining security rules that are enforced on all of our containers across all of our Anthos deployments.	alo
Load Balancers	- Internal load balancer - specific to region - Global load balancer - spans across regions 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) - external HTTP(s) load balancer improves access latency and Cloud Armor can be configured to block the Distributed Denial-of-Service (DDoS) attack.  	
Web Security	Web Security Scanner identifies security vulnerabilities in your App Engine, Google Kubernetes Engine (GKE),	

scanner	and Compute Engine web applications. It crawls your application, following all links within the scope of your starting URLs, and attempts to exercise as many user inputs and event handlers as possible. Web Security Scanner only supports public URLs and IPs that aren't behind a firewall.	
Traffic Director		
Containers Best practices	- Use native logging mechanisms - Ensure that containers are stateless and immutable - Avoid privileged containers - Avoid running as root - Make your application easy to monitor - Expose the health of your application - Carefully choose image version	
Google Recommender service	Recommender allows you to retrieve recommendations for Cloud resources, helping you to improve security, save costs, and more. Each recommendation includes a suggested action, its justification, and its impact. Recommendations are grouped into a per-resource collection.	
Service account insights	service account insights, which are findings about which service accounts in your project have not been used in the past 90 days	
Traffic Director	- Managed application networking platform and service mesh - Supports GKE Clusters and VM instances	
Global External HTTP(s) load Balancer	- Used for load balancing across multiple regions and zones.	
VPC Service Controls	Controls to enable a network perimeter lets you restrict access to services behind a private endpoint. You can restrict access to specific network ranges. Gated egress topology lets APIs in on-premise environments be available only to processes inside Google Cloud without direct public internet access.	
Identity Aware Proxy	provides secure access for valid accounts. Provides Authentication (Oauth), DDOS, and context-aware access controls. IAP- Identity-Aware Proxy is a service that lets you use SSH and RDP on your GCP VMs from the public internet, wrapping traffic in HTTPS and validating user access with IAM	
Cloud Security Scanner	a web vulnerability scanner that automatically scans web applications hosted on GCP for security vulnerabilities	
Network Tags	-Used with firewall rules	
Labels	Budgets are useful for visibility into the amount of money spent and can even alert you when the budget is exceeded. You can use labels to organize your resources and define the limits you alert on. Budgets don't enforce your spending. Enforcing spending limits is your responsibility, and for good reason.	
Databases		
Firebase	Firebase storage for "Mobile apps"/"User generated content"/"Robust uploads over mobile networks"	-
Cloud SQL	- Managed relational DB supporting MySQL, PG and SQL Server - Automated backups, maintenance, replication and failover - Vertical scaling - Horizontal scaling only for reads - SLA - 99.95% - On creation, allows to choose SSD/HDD, Capacity, Automatic storage increase (Y/N), not allowed to decrease storage - Automatic encryption for data at rest using AES-256 in DEK-KEK concept - Data can be encrypted in transit using SSL or by using cloud sql auth proxy (Auth proxy automatically encrypts traffic to and from DB using TLS 1.3 with 256 bit cipher and uses IAM permissions)	Heterogenous migrations Legacy applications Enterprise workloads Hybrid cloud, multicloud, and edge
Spanner	- Managed relational DB with consistency at global scale - Automatic, synchronous replication for HA. - Supports Google Standard SQL and PostgreSQL - SLA : 99.99% for regional and 99.999% for multi-regional - Defined by "Compute capacity" on instance creation (either as processing units or number of nodes) - To create an instance with size of 300GB; set compute capacity to 100 processing units; Add another 100 processing units if storage needs grow beyond 409.6 GB. - Billing on actuals and not on allotment - Data is encrypted by default using gcp managed keys.	Gaming Retail Global financial ledger Supply chain/inventory management
Firestore	○ Document DB ○ No Allocation, scales with usage ○ Pay for amount of data consumed ○ Data is encrypted by default using gcp managed keys. ○ Performance ▪ Choose location that is nearest to users ▪ Use asynchronous calls over synchronous calls ▪ Prefer SDK over REST API	Used For "user profiles"/"session mgmt"/"real-time capabilties"
BigQuery	- Serverless	- Data warehousing, analytics and BI

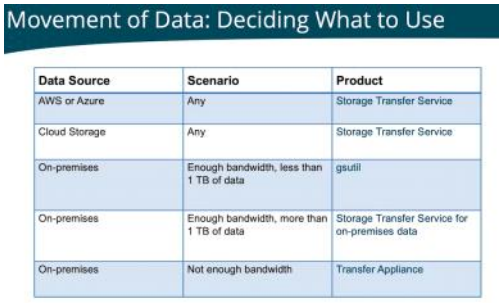
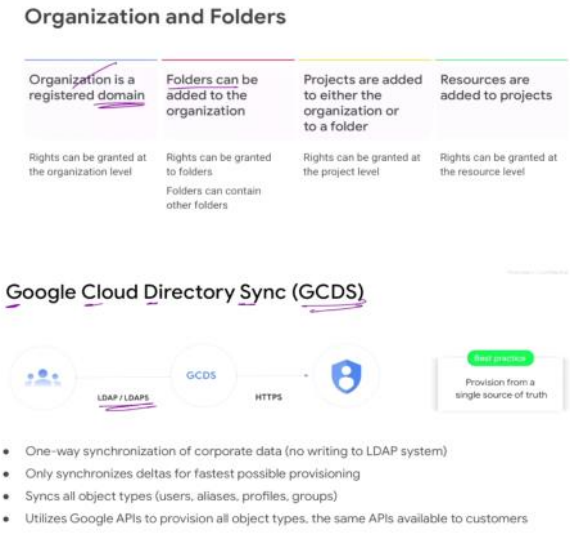
<table>
<tr><td></td><td>- Has transfer service to move data from cloud storage, S3 , teradata, RedShift</td><td>Workloads
- Availability SLA of 99.99%
- Bigquery for "analytics"/"DWH"/ML</td></tr>
<tr><td>BigTable
(Datastore)</td><td>Typical Criteria for Use,
➢ Working with more than 1TB of semi-structured or structured data.
➢ Data is fast with high throughput, or it's rapidly changing.
➢ They're working with NoSQL data. This usually means transactions where strong relational semantics are not required.
➢ Data is a time-series or has natural semantic ordering.
➢ They're working with big data, running asynchronous batch or synchronous real-time processing on the data.
➢ Or they're running machine learning algorithms on the data</td><td>Personalization

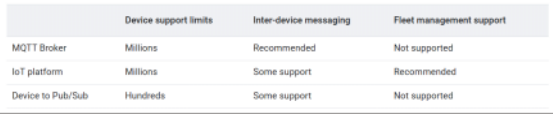
Adtech

Recommendation engines

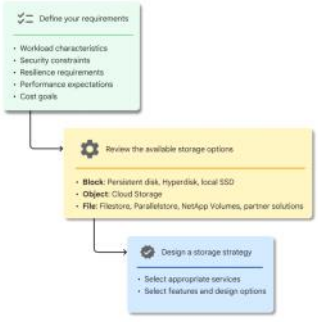
Fraud detection</td></tr>
<tr><td>MemoryStore</td><td>○ Allocation during instance creation
○ Data at rest encrypted by google
○ In-transit encryption can be enabled</td><td></td></tr>
<tr><td>Cloud Storage</td><td>- No Allocation; Pay for storage consumed
- Max object size: 5TB
- Uses the key to encrypt object's data, checksum and CRC
- Standard server-side keys to encrypt object metadata
- Various timelines
• Standard - No Minimum duration
• Nearline - Minimum duration is 30 days. suitable if data is accessed every Month
• Coldline - Minimum duration is 90 days
• Archive - Minimum duration is 365 days
 

Choose location and type by balancing latency, availability, and bandwidth costs for data consumers
- File storage on compute engines
• Persistent disks and local SSDs
• Managed and partner Solutions
<table><tr><th>Solution</th><th>Optimal dataset</th><th>Throughput</th><th>Managed support</th><th>Export protocols</th></tr><tr><td>Filestore Basic</td><td>1 TiB to 64 TiB</td><td>Up to 1.2 GiB/s</td><td>Fully managed by Google</td><td>NFSv3</td></tr><tr><td>Filestore Zonal</td><td>1 TiB to 100 TiB</td><td>Up to 26 GiB/s</td><td>Fully managed by Google</td><td>NFSv4.1</td></tr><tr><td>Filestore Regional</td><td>1 TiB to 100 TiB</td><td>Up to 26 GiB/s</td><td>Fully managed by Google</td><td>NFSv4.1</td></tr><tr><td>Google Cloud NetApp Volumes</td><td>1 GiB to 100 TiB</td><td>MBs/s to 4.5 GiB/s</td><td>Fully managed by Google</td><td>NFSv3, NFSv4.1, SMB2, SMB3</td></tr><tr><td>NetApp Cloud Volumes ONTAP</td><td>1 GiB to 1 PiB</td><td>varies</td><td>Customer-managed</td><td>NFSv3, NFSv4.1, SMB2, SMB3, iSCSI</td></tr><tr><td>Nasuni</td><td>10s of TB to > 1 PB</td><td>Up to 1.2 GBps</td><td>Nasuni- and customer-managed</td><td>NFSv3, NFSv4, NFSv4.1, NFSv4.2, SMB2, SMB3</td></tr><tr><td>Read-only Persistent Disk</td><td>< 64 TB</td><td>240 to 1,200 MBps</td><td>No</td><td>Direct attachment</td></tr></table></td><td></td></tr>
<tr><td>Cloud billing budgets</td><td>- Used to monitor and track to monitor all of your Google Cloud charges in one place. Setting a budget does not automatically cap Google Cloud or Google Maps Platform usage or spending. Budgets trigger alerts to inform you of how your usage costs are trending over time.
- Budgets are useful for visibility into the amount of money spent and can even alert you when the budget is exceeded. You can use labels to organize your resources and define the limits you alert on. Budgets don't enforce your spending. Enforcing spending limits is your responsibility, and for good reason.</td><td></td></tr>
<tr><td>Data Movement</td><td>  </td><td></td></tr>
</table>

		
Design Considerations	- Zones and Regions - Deploy over multiple regions, select regions based on geographic proximity - Security - Use Organization policy to enforce guardrails, cloud IAM and least privilege and data encryption at REST and in-transit - Scalability - MIGs to support VM Management, POD Auto-scalers, Managed services	
Availability Metrics	RTO - Maximum acceptable length of time your application can be offline RPO - Maximum acceptable length of time during which data might be lost SLO - Defines target value for a specific measurable characteristics of SLA SLA - agreement between two parties on specifics of service like times, locations, costs etc.	
DR Building blocks	- Traffic director - used to perform service health checks and initiate failover to redirect traffic to healthy instances - Automate failover and recovery as much as possible - Design for end to end recovery - Use monitoring to detect issues - Document every step of recovery process - Test DR Plan regularly	
Budgeting	- Cloud billing - allows to define budget scoped to entire account, one or more projects , one or more products. Alerts can be based on threshold value. Resources are not shutdown when threshold is reached and continue to incur costs. - **Data can be exported to bigquery for analysis. Labels can be used for better cost mgmt. - Committed use discounts - ideal for predictable and steady state usage. Recommender can be used to generate recommendations.	
Compute	- TPU - tensor processing unit - google's custom-developed, domain specific circuits for machine learning workloads. - Spot VMs - ideal for fault-tolerant, non time-critical applications. Do not have min or max runtime. They are not always available	
Data migration service	- To migrate from Mysql/SQL Server/PostgreSQL to Cloud SQL - It can be used to migrate Oracle to PostgreSQL - Also supports continuous migration (Change capture)	
Storage transfer service	- to Move files to GCP - Can be scheduled	
Cloud shell	- Provided on gcp itself - Cloud storage utility - gsutil - CLI - gcloud - Bigquery command line	
IAM	 -Service accounts - when using service accounts within Google Cloud (for example, from Compute Engine or App Engine) Google automatically manages the keys for service accounts. However, if you want to be able to use service accounts outside of Google Cloud, or want a different rotation period, it is possible to also manually create and manage your own service account keys -Best practices, • Use projects to group resources that share the same trust boundary. • Check the policy granted on each resource and make sure you understand the inheritance. • Use “principles of least privilege” when granting roles. • Audit policies in Cloud Audit Logs: setiampolicy.	

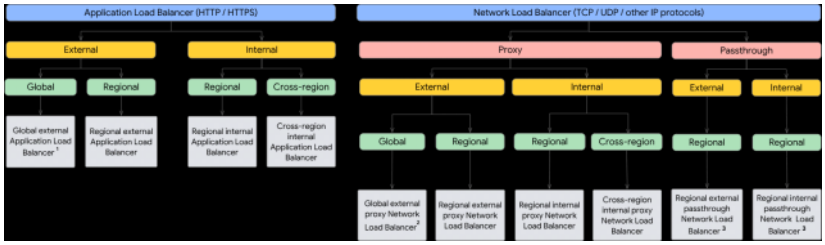
	• Audit membership of groups used in policies. • Grant roles to Google groups instead of individuals • Be very careful granting serviceAccountUser role. • For service account, establish key rotation policies and methods and audit keys. • Use IAP (Identity aware proxy) to establish central authorisation layer. Application level access control instead of relying on firewalls.	
Monitoring	- A *scoping project* hosts a metrics scope. The scoping project stores the alerting policies, uptime checks, dashboards, synthetic monitors, services, and monitoring groups that you configure. Because every Google Cloud project hosts a metrics scope, every project is also a scoping project. When you use the Google Cloud console, the scoping project is the project selected by the Google Cloud console project picker. - Precision in alerting policies is "The proportion of events detected that were significant." - Error budget "The proportion of alerts detected that were relevant to the sum of relevant alerts and missed alerts." - Trace is used to diagnose latency in http requests - "Heap" signifies amount of memory allocated. - best practices to reduce monitoring costs -> Reduce Ops Agents usage	
IOT Approaches on GCP		
Video Intelligence API, HLS Protocol	**GCP Video Intelligence API Streaming API allows real-time analysis of live video streams, supporting features like live label detection and shot change detection	
Web Security Scanner	Web Security Scanner identifies security vulnerabilities and misconfigurations in your App Engine, Google Kubernetes Engine (GKE), and Compute Engine web applications. It crawls your application, following all links within the scope of your starting URLs, and attempts to exercise as many user inputs and event handlers as possible. Web Security Scanner only supports public URLs and IPs that aren't behind a firewall.	
Traffic Director	Managed global load balancing with capacity, health awareness Support for multi-environment service meshes spanning across multi-cluster Kubernetes, hybrid cloud, VMs, gRPC services, and more.	
Cloud Composer	Cloud Composer, also known as BigQuery Engine for Apache Airflow, is a managed Apache Airflow service that helps you create, schedule, monitor and manage workflows. Cloud Composer automation helps you create Airflow environments quickly and use Airflow-native tools, such as the powerful Airflow web interface and command line tools, so you can focus on your workflows and not your infrastructure	

- Cloud storage design approach



- Load Balancer Types (Ref: Choose a load balancer | Load Balancing | Google Cloud)-
  - **External load balancers** distribute traffic that comes from the internet to your Google Cloud Virtual Private Cloud (VPC) network.
  - **Internal load balancers** distribute traffic that comes from clients in the same VPC network as the load balancer or clients connected to your VPC network by using VPC Network Peering, Cloud VPN, or Cloud Interconnect
- Use VPC Peering to connect networks when both of them are in GCP.
- Cloud VPN to connect on-premise with GCP network. Useful for low-volume data connections.
  - As a managed service, Cloud VPN provides an SLA of 99.9 percent monthly uptime for the classic VPN Configuration and 99.99 percent monthly uptime for the high availability VPN configuration.
  - HA VPN is high availability option providing 99.99% uptime. Multiple tunnels are configured from remote network for HA.A Cloud Router can manage routes for a Cloud VPN tunnel using Border Gateway Protocol or BGP.
- Cloud interconnect
  - Dedicated high speed connection
  - Two types
    - Dedicated Interconnect provides a direct connection to a co-location facility. Need to provision a cross connect between the Google network and your own network.
    - Partner Interconnect provides a connection through a service provider. useful for lower bandwidth requirements, starting from 50 megabits per second.
- Connectivity between google cloud and on-prem,
  - Carrier Peering exists outside of Google Cloud. Instead of Carrier Peering, the recommended methods of access to Google Cloud are Partner Interconnect, which uses a service provider, or Dedicated Interconnect, which provides a direct connection to Google.
  - If used with Google Cloud, Carrier Peering doesn't produce any custom routes in a VPC network. Traffic sent from resources in a VPC network leaves by way of a route whose next hop is either a default internet gateway (a default route, for example) or a Cloud VPN tunnel.
  - To send traffic through Carrier Peering by using a route whose next hop is a Cloud VPN tunnel, the IP address of your on-premises network's VPN gateway must be in your configured destination range.
- Service perimeter - To protect Google Cloud services in your projects and mitigate the risk of data exfiltration, you can specify service perimeters at the project or VPC network level.
- Beyondcorp - BeyondCorp is Google's implementation of the zero trust model. It builds upon a decade of experience at Google, combined with ideas and best practices from the community. By shifting access controls from the network perimeter to individual users, BeyondCorp enables secure work from virtually any location without the need for a traditional VPN.

- Choosing load balancer



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

Cloud Migration Strategies

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

- **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)

Movement of Data: BigQuery Data Transfer Service

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

System Design Considerations

- Compute Resources
  - Choose compute platform based on technical requirements of the workload, lifecycle automation processes, regionalization, and security
- Network
  - Choose VPC topology to support application communication and security requirements
  - Choose hybrid or multicloud connectivity to support external integrations
- Storage Resources
  - Choose appropriate storage type based on data type and requirements

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

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

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)

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
- Leverage committed use discounts
  - Ideal for workloads with predictable resource needs, available as 1- or 3-year term(s).

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

Compliance on GCP: HIPAA

Health Insurance Portability and Accountability Act

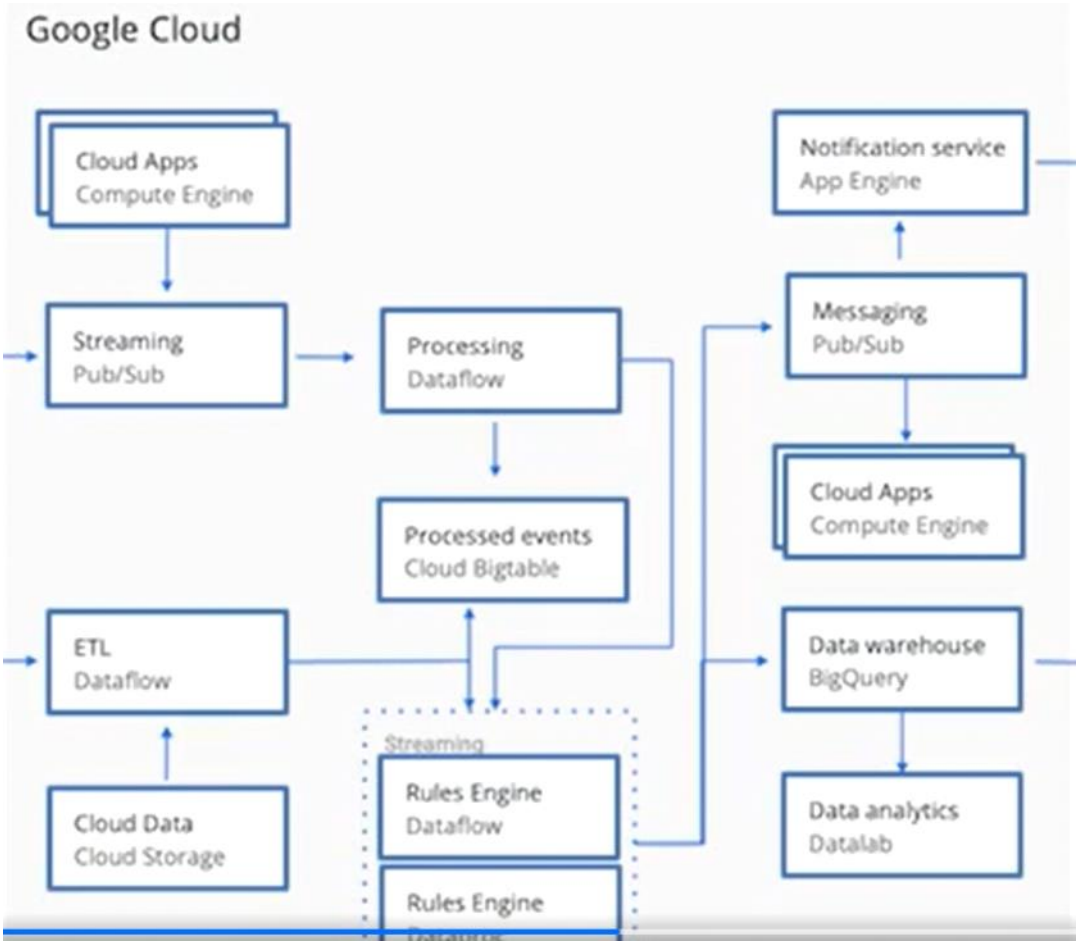
- U.S. standard to protect sensitive patient health information from being disclosed
- There is no certification: complying with HIPAA is a shared responsibility between Google and customer
- HIPAA demands compliance with the "Security Rule", the "Privacy Rule", and the "Breach Notification Rule"
- Google will enter into Business Associate Agreements (BAA) with customers as necessary under HIPAA. Customers can request BAA directly from their account manager

High Availability (HA) Design Principles

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

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



Design Strategies for Elasticity

- Autoscaling and load balancing
- Managed services & serverless

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)

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

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: 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

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

Cloud-native Networks: Design Considerations

- VPCs are global resources and not associated with a region. Subnetworks are regional.
- You can use VPC to privately access managed services (Cloud SQL, Cloud Storage, etc.)
- You can secure inter-VM traffic with VPC firewall rules using network tags.
- You can secure network administration with Cloud IAM.
- You can use Shared VPC to provide centralized networking to multiple projects.

- Storage
  - Persistent disks for "Disks forVM"/"Storage for databases"/"Sharing read-only data across VMs"/"Backups"
  - Local SSD for "Flash optimized databases"/"Hot caching layer for analytics"/"Application scratch disk"
  - Firebase storage for "Mobile apps"/"User generated content"/"Robut uploads over mobile networks"
  - Bigtable for "Adtect"/"IOT"
  - Bigquery for "analytics"/"DWH"/ML
  - Cloud firestoreMemorystore for "Application caching"/"Stream processing"

IaaS: Compute Engine instances

You're responsible for:

- Patching and upgrading the operating system (OS)
- Configuring various OS settings
- Installing and maintaining application libraries and runtime
- Manually setting up export or streaming of application logs
- (If using custom OS) Creating and maintaining VM image
- Architecting for fault-tolerance and scalability

- Managed instance groups (MIGs) for,
  - Stateless workload
  - Stateless batch compute workload

Deployment option	Scenario
Zonal MIG	Your VMs can be deployed to a single zone (note: not HA)
Regional MIG	Distribute VMs across multiple zones within a region. Tolerance against zonal failures
MIG with autoscaling	You want your MIG to automatically add VMs when demand increases, and remove them when demand drops
MIG with preemtible VMs	Your workload can tolerate disruptions and you want to reduce costs
Stateful MIG	Your workload needs to retain state whenever VMs are autohealed, updated, or recreated

IaaS: Compute Engine instances

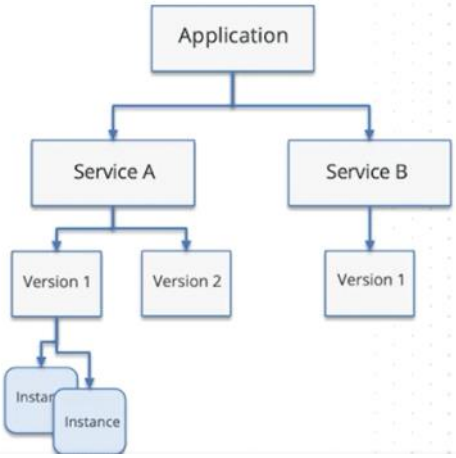
When to choose Compute Engine instances?

- 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

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



- App Engine Standards vs App Engine Flexible
- Platform as a Service - Customers are responsible for Application and Data

Migration Phases

Assess

Thorough discovery of existing environment

Identifying app dependencies and requirements

TCO Calculations

Performance benchmarks

Plan

Foundational cloud infrastructure

Identity management

Organization and project structure

Networking

Deploy

Implement and execute a deployment process

Move workloads

Refine cloud infrastructure

Optimize

Adopt cloud-native Technologies

Scalability, DR

Cost optimization

Training

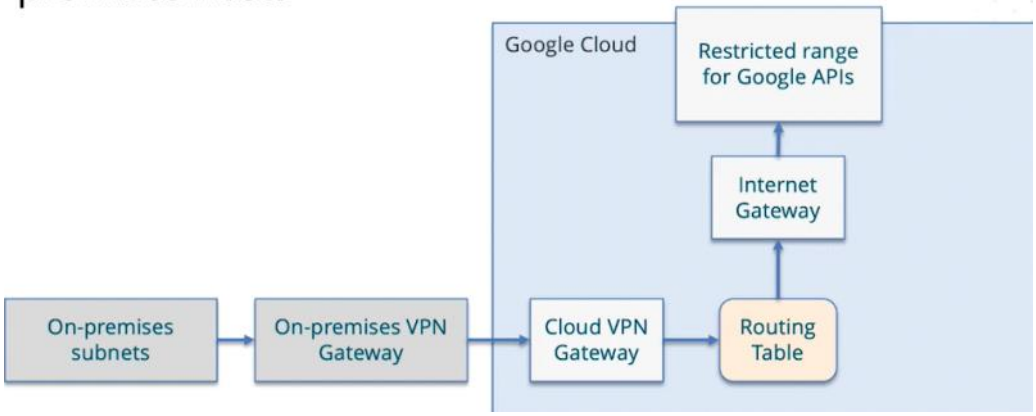
AI/ML and insights

Accessing Google Cloud Services: Networking

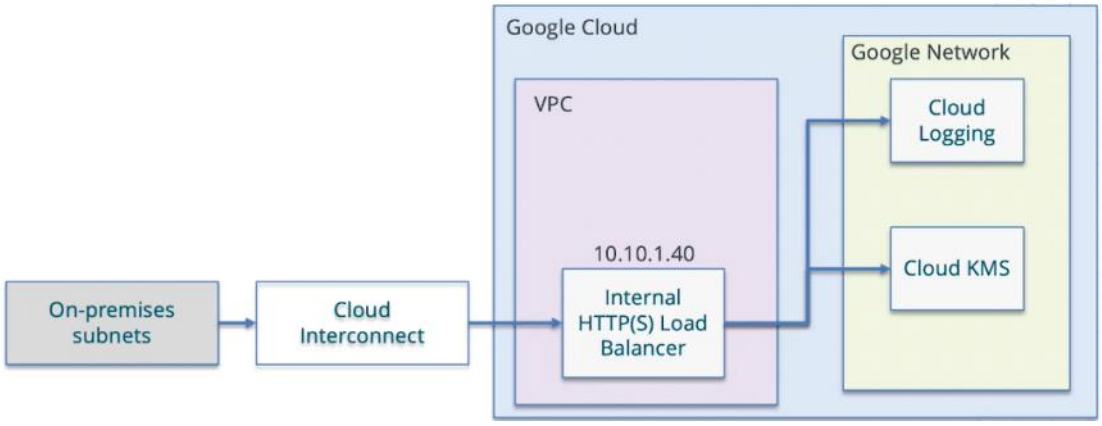
- For private networking between on-premises and Google Cloud:
  - Cloud VPN or Cloud Interconnect
  - Private Google Access for on-premises hosts
  - Private Service Connect
    - (Optional) with consumer HTTP(S) service controls

Note: DNS, firewall rules, and routes must be properly configured on-premises

Example: Cloud VPN + Private Google Access for on-premises hosts



Example: Cloud Interconnect + Private Service Connect with consumer HTTP(S) service controls



Service Account Keys

- RSA key pairs
- Lets you authenticate as the service account by having access to the private key
- User-managed key pairs are a security risk

Workload Identity Federation

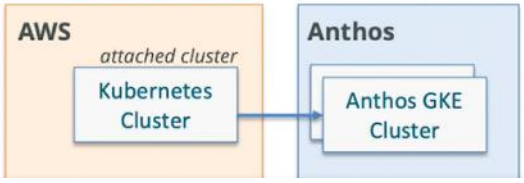
- Identity federation with AWS, Azure, or any identity provider that supports OpenID Connect / SAML 2.0.
- Use IAM to grant external identities IAM roles, including ability to impersonate service accounts

Integrating with Existing Systems with Anthos

- Anthos attached clusters
- Multicluster management with Fleets and Connect
- Configuration management with Anthos Config Management
- Service management with Anthos Service Mesh

Anthos Attached Clusters

- You can manage any existing standard, CNCF-compliant Kubernetes installation
- Anthos automatically installs **Config Management** on clusters for consistent configurations and security policies across all Kubernetes environments
- You can use **Connect Gateway** to connect to clusters across environments without proxies, inbound firewall rules, or bastion hosts

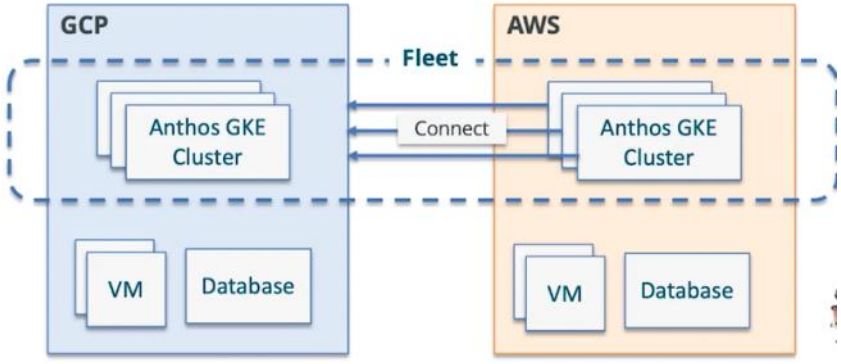


Anthos Fleets and Connect

- A fleet is a logical grouping of Kubernetes cluster and other resources that can be managed together
- Simplifies the management of multi-cluster deployments
- Creating a fleet involves registering the clusters you want to manage together to a fleet
- With fleets, you can choose two options for authenticating to clusters:
  - Connect Gateway

Anthos Fleets and Connect

Example

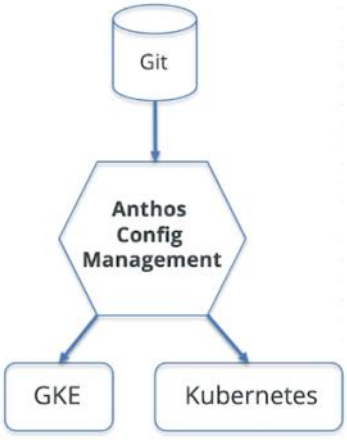


- A fleet is a logical grouping of Kubernetes cluster and other resources that can be managed together
- Simplifies the management of multi-cluster deployments
- Creating a fleet involves registering the clusters you want to manage together to a fleet
- With fleets, you can choose two options for authenticating to clusters:
  - Connect Gateway
  - Anthos Identity Service

- 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
- This enables 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

- How will on-premises connect to cloud?
  - **Cloud VPN**: supports up to 3Gbps
  - **Cloud Interconnect**: supports up to hundreds of Gbps

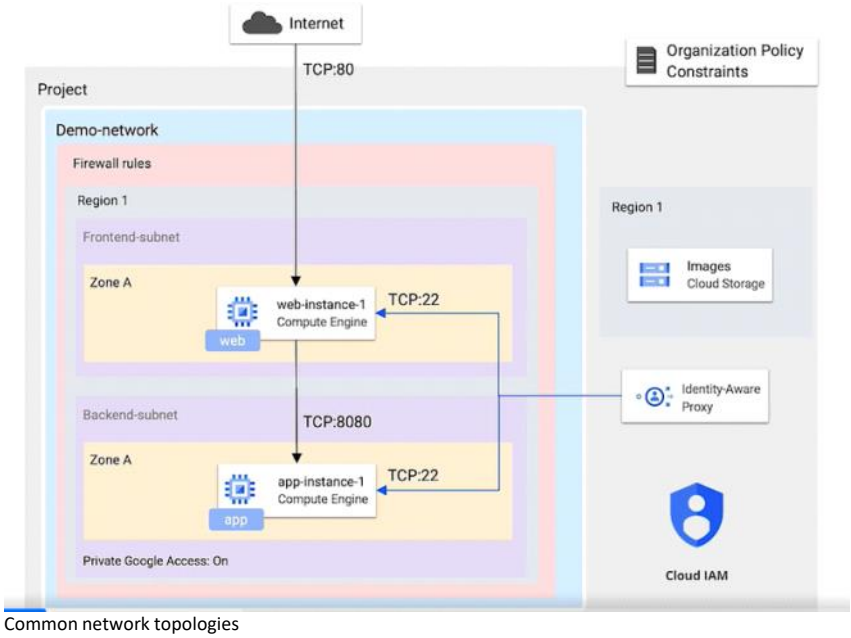
Dedicated Interconnect	Partner Interconnect
High bandwidth needs (tens of Gbps)	Bandwidth needs are in the hundreds of Mbps or low Gbps
Can reach Google's network directly at a colocation facility	Not able to reach Google's network directly
Don't want traffic to pass through a service provider network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

- VPC Design
  - Start with a single VPC network
  - Use custom mode to restrict traffic and customize address scheme
  - Group applications into fewer subnets with larger address ranges
  - Centralized vs. decentralized control
    - Use Shared VPC to maintain centralized control of the network
  - Secure and limit external connectivity
  - Layer 7 inspection required? Use hub-and-spoke design with appliances

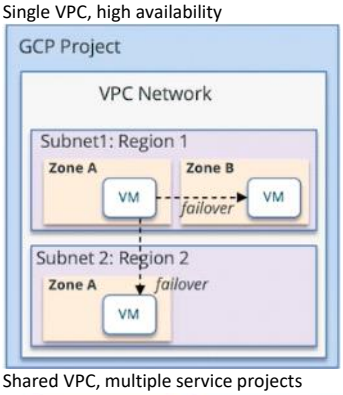
  - Define proof-of-concepts (PoCs) to validate that the cloud meets all use cases and requirements
  - Implement and run the PoCs; Examples:
    - Implement firewall rules for a complex workload
    - Compare performance of an on-premises relational database against Cloud SQL
    - Test the internal and external network latency for your apps on GCP
    - Transfer data to BigQuery and run business critical queries

- Each experiment should have a precise scope, expected outputs, and measurable business impact
- Example measures:
  - % reduction of launch time
  - % reduction in set up time of disaster recovery environment
  - % increase in performance
  - % reduction in latency
  - Projected increase in reliability (through use of a globally available service)

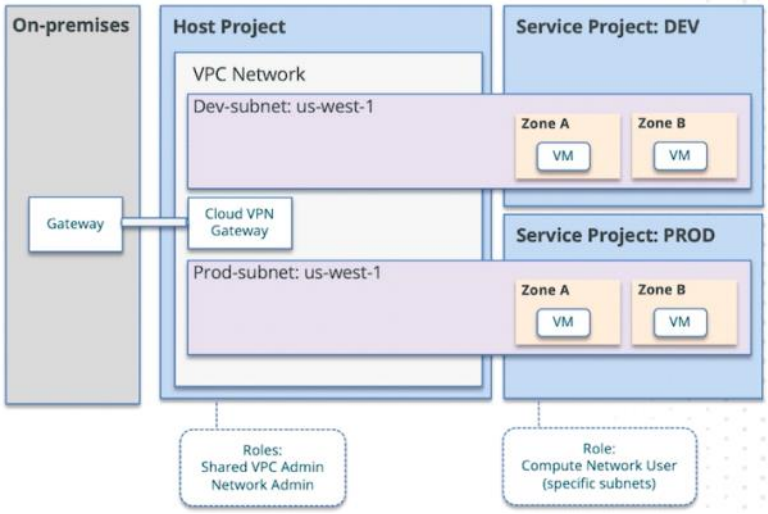
- Continuous Optimization and improvement
  - Optimize infrastructure
    - Use Managed services
    - Adopt cloud-native architecture
    - Reduce Costs
  - Monitor everything
    - Define metrics that are important to assess the health
    - Setup alerts in cloud monitoring
    - Export logs to storage and run analytics
  - Automate everything
  - Codify everything - capture aspects of solution in code. Make environment fully auditable and repeatable.
- Steps in setting up hybrid/multi-cloud network
  - Use shared VPCs to centrally manage peering, subnets, firewall rules and permissions
  - Avoid using VPCs to isolate workloads
  - Consider using TLS Encryption for cloud interconnect
  - When managing firewall rules, prefer service-account based filtering over network tag-based filtering
  - Determine redundancy type on non-gcp gateway
  - Create/use cloud router
  - Create HA VPN Gateway
  - Create vpn tunnels between gateway and non-gcp gateway
  - Create BGP Sessions
  - Apply corresponding config on non-gcp network
  - Interconnect
    - Dedicated - Create interconnect (dedicated or partner)
    - Partner
      - Establish connectivity with service provider
      - Create VLAN attachment
      - Config on-premise routers
- Guidance on securing networks
  - Disable default networks
  - Secure hybrid/multi-cloud connectivity
    - IPSec VPN
    - TLS Encryption over interconnect
    - Configure private service connect
  - Deploy zero trust networks
  - Configure vpc secure controls to secure perimeter
  - Cloud armor configuraiton against web attacks
- VPC firewall rules
  - Can not control connections to/from serverless offerings (Cloud storage, bigquery etc.)
  - Can not control access to Fully-qualified domain names (FQDN) like www.google.com
- **Sample Network Diagram



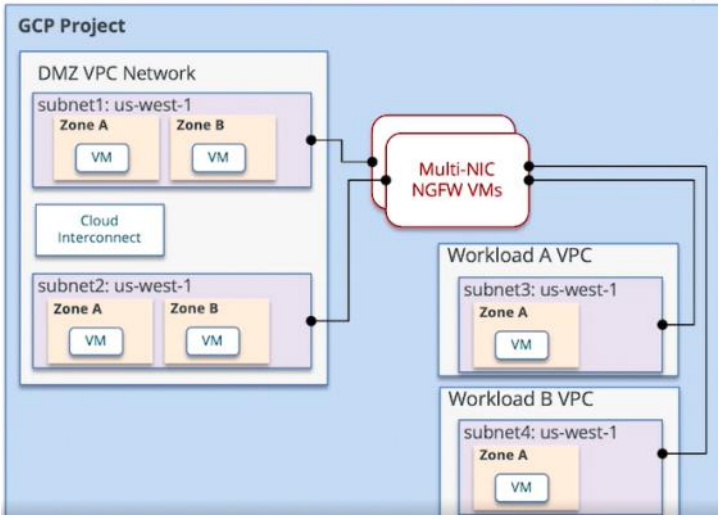
- Common network topologies
  - Single VPC, high availability



  - Shared VPC, multiple service projects



  - Multiple VPC, bridged by firewall appliance



- Data storage

- Serverless data processing
  - Cloud dataflow
  - Dataproc serverless
  - Pub/sub
  - Dataprep
- Storage
  - Managed instance groups (MIG)
    - Autoscaling can not be used if MIG has stateful configuration
    - Preemptible instances - offered as discount for a period (Max 24 hours). Not covered by any SLA.
    - Spot instances - preemption (release) can be handled via script (for cleanup etc.)

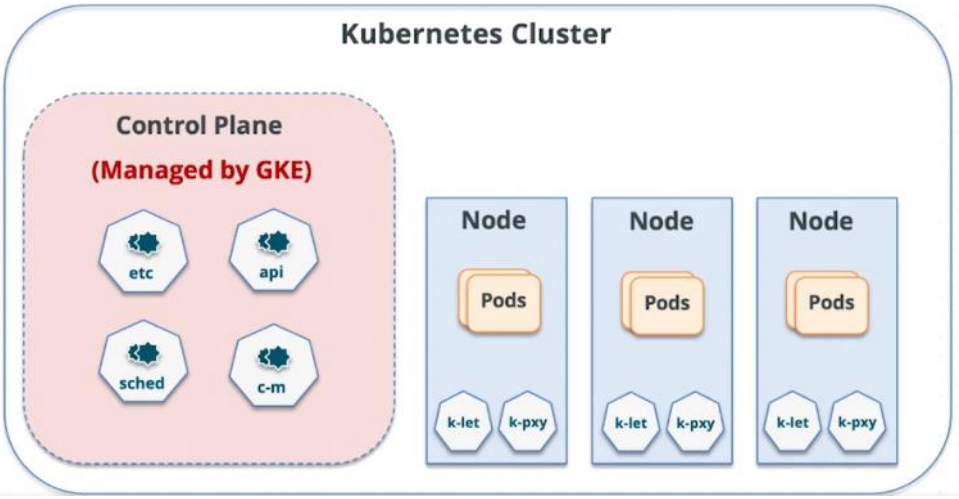
- Load balancing
  - HTTP(S) load balancing
  - TCP/UDP load balancing
  - TCP/SSL Proxy load balancing
  - Generic template is
    - Assign Static IP
    - Create Health checks
    - Create backend service(s)
  - Cloud NAT uses cloud router only as control plane and does not add any routes. Traffic does not pass through cloud router or any intermediate proxy in the data path
- Gcloud commands to setup network port access rules

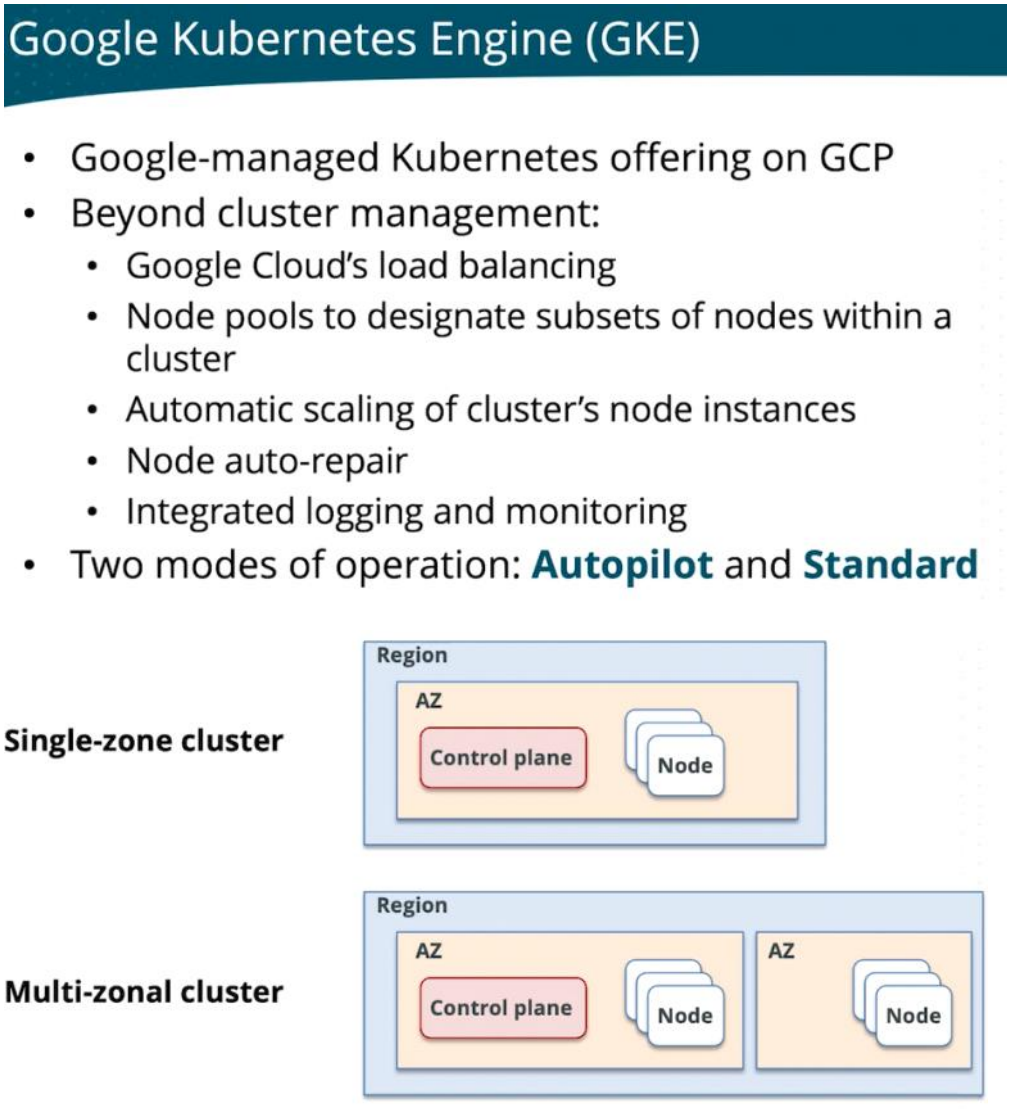
```
gcloud compute firewall-rules create vm1-allow-
egress-tcp-port443-to-192-0-2-5 \
   --network NETWORK_NAME \
   --action allow \
   --direction egress \
   --rules tcp:443 \
   --destination-ranges 100.50.1.21/32 \
   --priority 70 \
   --target-tags app
```

Type	Protocol	Port
SSH	TCP	22
RDP	TCP	3389
HTTPS	TCP	443
HTTP	TCP	80
DNS	TCP/UDP	53
FTP	TCP	20/21
LDAP	TCP/UDP	389

- VM Manager - OS Configuration Mgmt.
  - Installing and maintaining agents (for monitoring/logging)

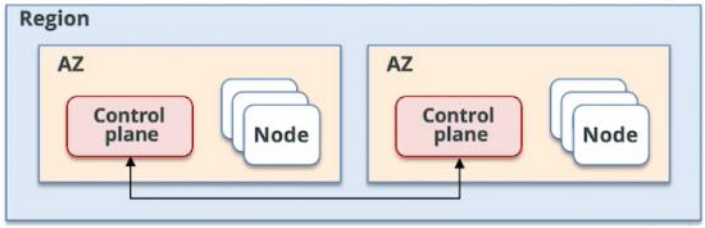
- Deploying security agents and ensuring agent is runnig on all VMs
- Running OS Policy compliance checks
- Modify existing startup scripts
- Adding repositories for software packages
- Managing files on VMs

- K8S
  - 

  - GKE
  - 

  -

Regional cluster



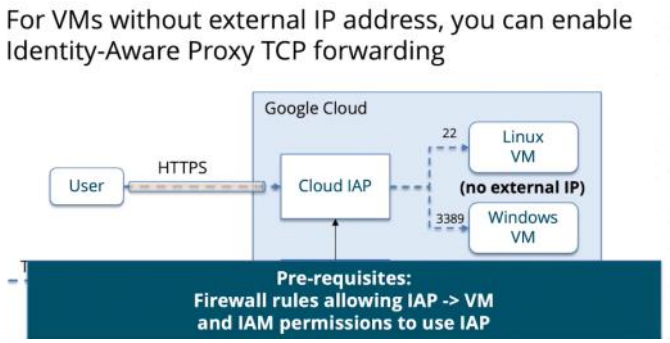
- Supports vertical and horizontal pod autoscaling
- Steps to deploy stateless app
  - Create a deployment config/service manifest (with 'kind: Deployment')
  - Use "kubectl apply"
- Service types
  - Clusterip (Kind: Service, type: clusterIP)
  - Nodeport (Kind: Service, type: clusterIP) - the service is reachable by using the IP Address of any node with nodeport value
  - LoadBalancer - L4, pass through LB
  - ExternalName - provides internal alias for external DNS Name
  -

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

- Storing Data
  - ConfigMaps - k8s native objects for non-sensitive data
- Steps to deploy stateful apps
  - statefulsets - repreents set of pods with unique, persistent identities and stable hostnames maintained by k8s regardless of scheduled
  - Deployment steps
    - Manifest with 'kind: statefulset'

- Benefits
  - Increase development velocity
  - Running efficient services with autoscaling
  - Deploying applications anywhere (portability)

- Security
  - Typical hierarchy is Organization -> Folder -> Project -> Resource
  - IAM roles can be granted at any of above levels. Roles granted at level are inherited by all resources below
  - IAM Effective allow policy for a resource is the union of inherited allow policy and its own allow policy
  - Best practices
    - Mirror resource hierarchy structure to organization structure
    - Use projects to group resources that share trust boundry
    - On every project, ensure that at least 2 principles have owner role
    - Limit project creation by granting the project creator role to a single group
  - All data at rest is always encrypted using AES256
  - Securing data in transit
    - Google managed certificates are supported with google external HTTPS LB and external proxy load balancer
    - For regional HTTP LB and internal LB, self-managed SSL Certificates are only way
    - Anthos service mesh - provides TLS Certificates to every workload in mesh. It encrypts all TCP communication. Uses google-managed CA.
  - Customer supplied encryption key (CSEK) - used only for cloud storage. Used for encryption object's data, CRC32c checksum, MD5 Hash but not metadata.
  - Good practices via Organization policy constraints
    - Resource location restriction - defines set of locations where resources can be created
    - Resource service usage - defines types of resources that can be used
    - Allowed KMS key types (e.g. HSM)
    - Restriction on access to Cloud SQL either at authorized networks level or Public IPs
  - Identity aware proxy (IAP)
    - Manage access to applications in app engine, compute engine and GKE
    - Central authorization layer for applications
    - Application-level access control



    - PCI DSS
        - Requirements
            - Install and maintain firewall configuration
            - Do not use vendor-supplied defaults for system passwords and other parameters
            - Protect stored cardholder data
            - Encrypt transmission of cardholder data across open, public networks
            - Use and regularly update anti-virus software or programs
            - Develop and maintain secure systems and applications
            - Restrict access to cardholder data by business need to know
            - Assign a unique ID to each person with computer access
            - Restrict physical access to network resources and cardholder data
            - Regularly test security systems and processes
            - Maintain a policy that address information security of all personnel
        - Design considerations
            - Isolate payment processing in separate GCP a/c or project
            - Configure ingress/egress firewall rules and controls
            - Configure/use hardened images as base and secure package manager
            - Use IAM roles to restrict access and enforce principle of least privilege
            - Enforce strong password guidelines and multi-factor authentication
            - Configure monitoring, logging and audit log exports
            - Use cloud data loss prevention to filter sensitive information before displaying it in any tool
- Cloud dataflow - Apache beam pipeline runner… Batch and Streaming. Unify stream and batch data processing that's serverless, fast, and cost-effective.
    - Cloud Pub/sub - A highly scalable messaging service for publishing and subscribing.
    - Vertex AI/Bigquery ML - Train your machine learning models at scale, to host your trained model in the cloud, and to use your model to make predictions about new data.
    - Cloud endpoints/Apigee - An API management system that helps you secure, monitor, analyze, and set quotas on your APIs using the same infrastructure that Google uses for its own APIs.
    - Cloud KMS - Key management
    - Cloud SQL
    - Cloud bigtable -datawarehouse
- Uptime checks
    - Using Pub/Sub is only one example of how to integrate a third-party tool to handle sending notifications to the person on-call. You can use Cloud Functions, App Engine, or scripts on a server, or the service used to manage on-call might be able to subscribe to the Pub/Sub topic directly. Pub/Sub is a good option for a notification channel if a standard one, like email, isn't suitable. Uptime checks are from a user's perspective, and only check external IP addresses. Uptime checks are only one of many metrics you could use.
- Partner Interconnect and Dedicated Interconnects provide the high performance and high availability compared to cloud VPN
- Interrogation on Architectural Decisions
    - Will this compute/storage/network design enable you to refactor your design over time?
    - How well would this scale if you got 10x or 100x the traffic you expected?
    - Are there big data products or services you could use now and grow into?
    - Does your VPC Network need to be accessed from somewhere else? By who?
    - Could you implement your solution as a fully managed pipeline?
    - How will your service be accessed from the internet?
    - What if any/compliance requirements exist?
    - What is likely disaster? would your environment survive?
    - How do you now your service is healthy? How do you troubleshoot and fix if its not?
    - How do you deploy new versions of service and roll them back ?

- GCP Case studies --> non-functional points,
- TerramEarth
    1. **Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.**
        - **GCP Solution:**
            - **Data Ingestion & Storage:** Utilize **Cloud Pub/Sub** for real-time ingestion of critical telemetry data from vehicles. Store this data in **Bigtable** for low-latency, high-throughput access suitable for real-time analysis. Batch sensor data uploaded daily can be stored cost-effectively in **Cloud Storage**.
            - **Predictive Analytics & Machine Learning:** Leverage **Vertex AI** to build and deploy machine learning models for anomaly detection and predictive maintenance. These models can analyze real-time and historical telemetry data to identify potential malfunctions early.
            - **Real-time Alerting & Notification:** Use **Cloud Functions** or **Cloud Run** triggered by Vertex AI predictions to send alerts to relevant dealerships and internal teams via **Cloud Tasks** and services like **SendGrid** or **Twilio**.
            - **Inventory & Logistics Integration:** Integrate Vertex AI predictions with your legacy inventory and logistics management systems (via the API abstraction layer discussed in technical requirements) to automate part ordering and optimize shipping to dealerships.
    2. **Decrease cloud operational costs and adapt to seasonality.**
        - **GCP Solution:**
            - **Autoscaling:** Implement autoscaling for compute resources like **Compute Engine**, **Google Kubernetes Engine (GKE)**, and **Cloud Run** to automatically adjust capacity based on demand, optimizing costs during low-usage periods.
            - **Serverless Computing:** Utilize serverless services like **Cloud Functions** and **Cloud Run** for event-driven workloads, eliminating the need to

manage underlying infrastructure and paying only for actual usage.
        - **Cost Optimization Tools:** Leverage **Cloud Billing reports and dashboards**, **Billing export to BigQuery**, and **Recommendations** to gain visibility into spending and identify cost optimization opportunities. Consider using **Committed Use Discounts** for predictable workloads.
        - **Storage Tiering:** Implement **Cloud Storage lifecycle policies** to automatically move less frequently accessed data to lower-cost storage tiers (Nearline, Coldline, Archive), balancing cost and access needs.
3. **Increase speed and reliability of development workflow.**
    - **GCP Solution:**
        - **Containerization:** Embrace **Docker** and **Google Kubernetes Engine (GKE)** for containerizing applications, ensuring consistent deployments across environments and improving scalability and reliability.
        - **Managed Databases:** Utilize managed database services like **Cloud SQL** and **Cloud Spanner** to offload database administration tasks, improving developer productivity and application reliability.
        - **Cloud Build:** Implement **Cloud Build** for fully managed CI/CD pipelines, automating the build, test, and deploy processes for containerized and other workloads.
        - **Artifact Registry:** Use **Artifact Registry** to securely store and manage container images and other build artifacts.
4. **Allow remote developers to be productive without compromising code or data security.**
    - **GCP Solution:**
        - **Identity and Access Management (IAM):** Implement granular IAM policies using the principle of least privilege to control access to resources based on roles and responsibilities.
        - **Virtual Private Cloud (VPC) Service Controls:** Define security perimeters around your Google Cloud resources to prevent data exfiltration and unauthorized access, even from within your organization.
        - **Cloud VPN or Interconnect:** Securely connect remote developers to your Google Cloud environment using **Cloud VPN** or dedicated **Interconnects**, depending on bandwidth and latency requirements.
        - **BeyondCorp Enterprise:** Consider implementing **BeyondCorp Enterprise** for zero-trust access, providing secure access to applications and resources based on user and device identity and context, regardless of location.
        - **Secret Manager:** Securely store and manage sensitive information like API keys and passwords using **Secret Manager**, preventing them from being hardcoded in code.
5. **Create a flexible and scalable platform for developers to create custom API services for dealers and partners.**
    - **GCP Solution:**
        - **API Gateway:** Implement **API Gateway** (part of Apigee or Cloud Endpoints) to manage, secure, monitor, and analyze your APIs. This provides a central point of control for your API ecosystem, enabling developers to easily discover and consume services.
        - **Cloud Functions & Cloud Run:** Empower developers to build and deploy lightweight, scalable API services using serverless compute options like **Cloud Functions** for event-driven APIs or **Cloud Run** for containerized APIs.
        - **Service Mesh (Istio on GKE):** For more complex microservices architectures, consider using a service mesh like **Istio** on GKE to manage traffic, enforce security policies, and provide observability across services.
        - **Pub/Sub & Eventarc:** Enable asynchronous communication between services using **Cloud Pub/Sub** for decoupling and scalability. **Eventarc** can further simplify event-driven architectures by routing events from various GCP services to your custom API services.

Technical Requirements

1. **Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.**
    - **GCP Solution:**
        - **API Gateway (Apigee or Cloud Endpoints):** Utilize **API Gateway** to create a consistent and secure interface for accessing your legacy systems. This allows you to:
            - Abstract away the complexities of your backend systems.
            - Implement security policies (authentication, authorization, rate limiting).
            - Transform data formats if needed.
            - Monitor API usage.
        - **Cloud Functions or Cloud Run:** Develop lightweight API proxies using **Cloud Functions** or **Cloud Run** that sit between the API Gateway and your legacy systems. These proxies can handle protocol translation, data mapping, and basic business logic.
        - **Hybrid Connectivity:** Ensure reliable connectivity between Google Cloud and your private data centers using **Cloud Interconnect** or **Cloud VPN**.
2. **Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.**
    - **GCP Solution:**
        - **Cloud Build:** Implement **Cloud Build** as your central CI/CD platform. It integrates seamlessly with source code repositories (Cloud Source Repositories, GitHub, GitLab, Bitbucket), builds container images using **Cloud Buildpacks** or custom Dockerfiles, runs tests, and pushes images to **Artifact Registry**.
        - **Artifact Registry:** Use **Artifact Registry** to store and manage Docker container images, Helm charts, and other build artifacts in a private and secure repository.
        - **Cloud Deploy:** For advanced deployment orchestration to GKE, Cloud Run, or Compute Engine, consider **Cloud Deploy** for progressive rollouts, rollback capabilities, and approval workflows.
        - **GitOps with Anthos Config Management:** For more complex multi-cluster GKE environments, explore GitOps practices using **Anthos Config Management** to manage infrastructure and application configurations declaratively through Git.
3. **Allow developers to run experiments without compromising security and governance requirements.**
    - **GCP Solution:**
        - **Separate Projects:** Create distinct Google Cloud projects for development, staging, and production environments. This provides strong isolation for security and resource management.
        - **IAM Policies:** Implement strict IAM policies at the project level to control access to resources in each environment. Grant developers the necessary permissions within their designated development projects without affecting production.
        - **VPC Networks:** Utilize separate VPC networks for each environment to isolate network traffic. Consider using VPC Network Peering or Shared VPC for controlled connectivity between environments.
        - **Resource Manager (Folders & Organizations):** Organize your projects within folders and under a Google Cloud Organization to enforce consistent policies and governance at scale.
        - **Policy Controller (part of Anthos Config Management):** Define and enforce organizational policies as code to ensure compliance and security across all projects.
4. **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.**
    - **GCP Solution:**
        - **Cloud Foundation Toolkit or Terraform:** Use Infrastructure-as-Code (IaC) tools like **Cloud Foundation Toolkit** or **Terraform** to define and provision standardized project templates and resource configurations.
        - **Cloud Functions or Cloud Run with a Web UI:** Develop a custom self-service portal using **Cloud Functions** or **Cloud Run** for the backend logic and a web framework (e.g., React, Angular) for the frontend. This portal can interact with the Google Cloud APIs to automate project creation and resource provisioning based on predefined templates.

        - **Service Catalog (part of Google Cloud Marketplace):** Explore using **Service Catalog** to curate and share approved service offerings (including project templates and data analytics environments) within your organization.
        - **API Gateway Integration:** The self-service portal should integrate with the **API Gateway** to allow developers to discover and request access to API endpoints. This can involve automated workflows for access approvals.
        - **Identity Platform or Firebase Authentication:** Implement a robust authentication and authorization mechanism for the self-service portal to manage user access.
5. **Use cloud-native solutions for keys and secrets management and optimize for identity-based access.**
    - **GCP Solution:**
        - **Secret Manager:** Utilize **Secret Manager** as the central service for securely storing and managing sensitive information like API keys, passwords, and certificates. It offers versioning, access control, and audit logging.
        - **Cloud KMS (Key Management Service):** Use **Cloud KMS** to manage encryption keys. You can use KMS-managed keys to encrypt data at rest in various GCP services (Cloud Storage, BigQuery, etc.).
        - **Workload Identity:** For applications running on GKE, leverage **Workload Identity** to allow your Kubernetes pods to authenticate to Google Cloud services using their Kubernetes service account, eliminating the need to manage service account keys manually.
        - **IAM Service Accounts:** Utilize **IAM service accounts** with the principle of least privilege for non-human entities (e.g., VMs, Cloud Functions) to access other GCP resources.
        - **Identity Platform or Firebase Authentication:** For authenticating users (dealers, partners) accessing your applications and APIs, use **Identity Platform** or **Firebase Authentication** for secure and scalable identity management.
6. **Improve and standardize tools necessary for application and network monitoring and troubleshooting.**
    - **GCP Solution:**
        - **Cloud Monitoring:** Use **Cloud Monitoring** for collecting and visualizing metrics, logs, and traces from your applications and infrastructure. Set up alerts for critical events and create custom dashboards for comprehensive observability.
        - **Cloud Logging:** Centralize and analyze logs from all your GCP services and applications using **Cloud Logging**. Utilize features like log-based metrics and log analytics with BigQuery.
        - **Cloud Trace:** Implement **Cloud Trace** to understand the latency of your requests as they propagate through your distributed systems, aiding in performance troubleshooting.
        - **Cloud Profiler:** Use **Cloud Profiler** to identify performance bottlenecks in your applications by analyzing CPU and memory usage.
        - **Network Intelligence Center:** Leverage **Network Intelligence Center** for network monitoring, analysis, and troubleshooting. This includes tools like Network Topology, Performance Dashboard, and Firewall Insights.
        - **Service Directory:** For service discovery within your cloud environment, consider **Service Directory** to easily locate and connect to services.

- Ehr health

The proposed architecture centres around a hybrid cloud model initially, transitioning towards a cloud-native approach as legacy systems are retired. It prioritizes managed services for reduced operational overhead, scalability, and built-in high availability.
Here's a breakdown of the key components:

1. **Foundation & Networking:**
    - **VPC Network(s):** Set up a robust Virtual Private Cloud (VPC) network in GCP, potentially using a Shared VPC model to centralize network management and allow different service teams (e.g., application, data) to share network resources securely. Use multi ple regions for high availability and disaster recovery.
    - **Cloud Interconnect:** Establish a dedicated, high-bandwidth, low-latency connection between EHR Healthcare's on-premises data centers (specifically for the legacy integrations) and the GCP VPC. This provides a more reliable and secure connection than VPN over the public internet. Redundancy should be built-in with multiple connections.
    - **Global Load Balancing (GCLB):** Provide a single, global entry point for customer-facing web applications. GCLB automatically routes user traffic to the nearest available backend (GKE cluster), reducing latency and improving availability. It can span multipl e regions.
    - **Cloud CDN:** Integrate Cloud CDN with GCLB to cache static assets closer to users, further reducing latency and load on the backend applications.
    - **Private Google Access:** Configure subnets to allow instances (like GCE VMs or GKE nodes) without public IP addresses to access Google APIs and services securely within the GCP network.
    - **VPC Service Controls:** Implement a security perimeter around sensitive data services (like BigQuery, Cloud Storage) to mitigate data exfiltration risks, adding an extra layer of protection for healthcare data.
2. **Compute:**
    - **Google Kubernetes Engine (GKE):** This will be the core platform for hosting the containerized customer-facing applications.
        - Deploy GKE clusters across multiple zones within a region for high availability (regional clusters). For disaster recovery, deploy clusters in multiple regions.
        - Leverage GKE Autopilot or Node Pools with auto-scaling to handle traffic spikes and scale environments dynamically based on demand.
        - Use GKE's built-in features for service discovery, load balancing, and rolling updates.
    - **Compute Engine (GCE):** Potentially used for specific legacy applications not yet containerized or for running third-party software that requires a VM. Managed Instance Groups (MIGs) can provide auto-scaling and auto-healing for VM-based workloads.
3. **Data Services:**
    - **Cloud SQL:** Managed service for MySQL and MS SQL Server databases. Choose appropriate machine types, configure high availability (synchronous replication), and enable automated backups with point-in-time recovery. Use Data Migration Service (DMS) for online migration from on-premises databases.
    - **AlloyDB for PostgreSQL:** If future applications or migration of MySQL/SQL Server databases benefit from PostgreSQL compatibility and require extreme performance and availability, AlloyDB is a strong candidate (though MySQL/SQL Server are current).
    - **Memorystore:** Managed service for Redis. Configure high availability using the Standard Tier with replication.
    - ****Firestore / Datastore:** For the MongoDB workload, consider migrating to Firestore or Datastore if the document model aligns well and eventual consistency is acceptable for certain use cases. Alternatively, run a managed MongoDB service on GCE/GKE, though thi s increases operational overhead compared to fully managed GCP services. Dataflow or custom scripts can help with migration.
    - **BigQuery:** A serverless, highly scalable data warehouse. This is central to providing insights into healthcare trends, predictions, and reporting.
        - Ingest data from operational databases (Cloud SQL, Firestore/Datastore) via ETL/ELT pipelines (Dataflow, Dataproc, or simple scheduled queries).
        - Ingest data from new providers (files on Cloud Storage, streaming via Pub/Sub) directly into BigQuery.
    - **Cloud Storage:** Scalable and durable object storage. Use for:
        - Landing zone for file-based data ingestion from providers (both legacy on-prem via Cloud Interconnect and new cloud-based sources).
        - Storing database backups.
        - Archiving historical data.
        - Storing container images (via Artifact Registry).
4. **Integration & Messaging:**

- **Pub/Sub:** A scalable, serverless messaging service. Use to:
  - Decouple services.
  - Ingest streaming data from providers.
  - Trigger downstream processes (e.g., data processing pipelines).
- **Cloud Functions / Cloud Run:** Serverless options for processing incoming files, transforming data, or building small API endpoints for integrations without managing servers. Can be triggered by Pub/Sub or Cloud Storage events.
- **API Gateway / Apigee:** If exposing APIs to external partners or providers becomes complex, these services can provide centralized API management, security, and analytics.

5. **Data Processing & Analytics:**
   - **Dataflow:** A fully managed service for stream and batch data processing. Ideal for complex ETL/ELT pipelines to clean, transform, and load data into BigQuery.
   - **Dataproc:** Managed Spark and Hadoop service. Useful if existing data processing jobs rely on these frameworks.
   - **Vertex AI:** Unified platform for building and deploying Machine Learning models.
     - Train models on data stored in BigQuery.
     - Deploy models for making predictions on industry trends or provider-specific data.
     - Integrate predictions back into reporting or operational systems.
6. **Management, Monitoring & Operations:**
   - **Cloud Operations (Monitoring, Logging, Error Reporting, Trace, Profiler):** Centralized, integrated suite for observing GCP resources and applications.
     - **Cloud Logging:** Collect logs from all applications and services (GKE, Cloud SQL, GCE, etc.). Centralized storage and querying. Set appropriate log retention policies.
     - **Cloud Monitoring:** Collect metrics (CPU usage, network traffic, error rates, etc.). Create custom dashboards for centralized visibility. Set up robust alerting policies (beyond email, e.g., using Pub/Sub to trigger automated remediation or integration with incident management tools).
     - **Error Reporting, Trace, Profiler:** Help identify and diagnose application performance issues and errors within the GKE environment.
   - **Cloud Build:** Serverless CI/CD service. Automate the build, test, and containerization of applications.
   - **Artifact Registry:** Securely store and manage container images and other build artifacts. Integrated with Cloud Build and GKE.
   - **Cloud Deploy:** Managed service for continuous delivery to GKE. Orchestrate deployments across different environments (dev, staging, prod) and regions.
   - **Infrastructure as Code (IaC):** Use tools like Terraform or Cloud Deployment Manager to define and provision GCP resources programmatically. This ensures consistency, repeatability, and reduces manual configuration errors.
   - **Cloud Identity / Identity Platform:** Integrate with Microsoft Active Directory using Google Cloud Directory Sync or a third-party solution to synchronize users and groups. Manage access to GCP resources using IAM roles and permissions based on synchronized identities. This provides centralized user management and single sign-on capabilities.
7. **Security & Compliance:**
   - **IAM:** Implement fine-grained access control using the principle of least privilege. Assign specific roles to users and service accounts for accessing resources.
   - **Security Best Practices:** Adhere to GCP security best practices, including using service accounts with minimal permissions, network segmentation, encryption at rest and in transit (default for most GCP services), and regular security audits.
   - **Regulatory Compliance (HIPAA/HITRUST):** GCP provides a HIPAA-compliant infrastructure. EHR Healthcare must ensure their *use* of GCP services and application design also meet regulatory requirements. This includes proper data handling, access controls, auditing (via Cloud Audit Logs), and privacy measures within their application layer. Sign a Business Associate Addendum (BAA) with Google.

How the GCP Solution Addresses Requirements:

Business Requirements:

- **On-board new insurance providers quickly:**
  - **Technical:** Cloud Storage + Dataflow/Cloud Functions/Pub/Sub pipelines enable rapid setup of data ingestion streams from new sources (files, APIs). Scalable compute (GKE) and databases scale automatically to handle increased load.
  - **Business:** Reduced infrastructure provisioning time allows focus on data integration logic.
- **Minimum 99.9% availability:**
  - **Technical:** Multi-zone/multi-region GKE clusters, High Availability Cloud SQL/AlloyDB/Memorystore, Global Load Balancing, redundant Cloud Interconnect connections. Managed services reduce the likelihood of outages due to misconfiguration or hardware failure.
  - **Business:** Improved service reliability leads to increased customer trust and satisfaction.
- **Centralized visibility and proactive action:**
  - **Technical:** Cloud Operations (Monitoring, Logging) provides a unified view of system health and performance. Robust alerting allows for proactive intervention.
  - **Business:** Operations teams can quickly identify and resolve issues, reducing downtime and improving system stability. Insights from monitoring data can inform capacity planning.
- **Increase ability to provide insights into healthcare trends:**
  - **Technical:** BigQuery provides a scalable platform for aggregating and analyzing large datasets. Dataflow/Dataproc enables complex data processing.
  - **Business:** Enables data scientists and analysts to query vast amounts of data quickly and derive meaningful insights.
- **Reduce latency to all customers:**
  - **Technical:** Global Load Balancing routes users to the closest region. Cloud CDN caches static content. Placing GKE clusters in multiple strategic regions reduces network hops.
  - **Business:** Provides a faster, more responsive user experience for global customers.
- **Maintain regulatory compliance:**
  - **Technical:** Leveraging GCP's compliant infrastructure (HIPAA BAA), implementing strong IAM, VPC Service Controls, encryption, and comprehensive audit logging (Cloud Audit Logs).
  - **Business:** Ensures continued adherence to critical healthcare regulations, protecting patient data and avoiding penalties.
- **Decrease infrastructure administration costs:**
  - **Technical:** Transitioning to managed services (GKE, Cloud SQL, BigQuery, etc.) shifts the burden of patching, maintenance, and scaling from EHR's team to Google. Automation via IaC and CI/CD reduces manual effort.
  - **Business:** Allows the IT team to focus on higher-value tasks like application development and innovation rather than routine infrastructure management. Optimized scaling reduces wasteful over-provisioning.
- **Make predictions and generate reports on industry trends:**
  - **Technical:** BigQuery + Vertex AI provides the platform for data storage, analysis, model training, and deployment.
  - **Business:** Enables data-driven strategic decisions, potential new data products for customers (insights-as-a-service), and improved operational forecasting.

Technical Requirements:

- **Maintain legacy interfaces to insurance providers:**
  - **Technical:** Cloud Interconnect provides secure, high-performance connectivity to on-premises systems hosting these interfaces. Cloud Storage can act as an intermediary for file-based transfers. Pub/Sub or Cloud Functions can trigger processing when new data arrives.

- **Provide a consistent way to manage container-based applications:**
    - **Technical:** GKE is the central, managed Kubernetes platform. Cloud Build/Artifact Registry/Cloud Deploy provide a standardized CI/CD pipeline for deploying to GKE clusters.
- **Provide a secure and high-performance connection between on-premises systems and Google Cloud:**
    - **Technical:** Cloud Interconnect is specifically designed for this, offering dedicated bandwidth and lower latency compared to VPN. Robust network security configurations within the VPC.
- **Provide consistent logging, log retention, monitoring, and alerting capabilities:**
    - **Technical:** Cloud Operations suite centralizes these functions across all GCP services and GKE. Configurable log retention policies and flexible alerting channels.
- **Maintain and manage multiple container-based environments:**
    - **Technical:** GKE supports multiple clusters for different environments (dev, staging, prod). IaC (Terraform/Deployment Manager) ensures consistency across these environments. Cloud Build/Deploy simplifies deployments to these different targets.
- **Dynamically scale and provision new environments:**
    - **Technical:** GKE auto-scaling handles traffic spikes within an environment. IaC templates allow for rapid and consistent provisioning of entirely new environments when needed (e.g., for a large new client or specific project).
- **Create interfaces to ingest and process data from new providers:**
    - **Technical:** Cloud Storage (file drop), Pub/Sub (streaming), Cloud Functions/Cloud Run/Dataflow (processing logic) provide flexible building blocks for creating tailored ingestion pipelines based on provider format and delivery method.

Addressing Executive Statement Concerns:

- **Investment in training on different systems:** Focusing on a single cloud platform (GCP) and its managed services reduces the need for expertise in diverse, disparate on-premises hardware and software. Training can be centralized on GCP skills.
- **Managing similar but separate environments:** Using IaC and a standardized platform like GKE and Cloud Operations allows environments to be provisioned and managed consistently, reducing configuration drift and complexity.
- **Responding to outages:** Managed services have built-in reliability and failover. Cloud Operations provides proactive monitoring and alerting to identify potential issues early. Automated scaling handles traffic spikes that previously caused outages due to inadequate capacity. Consistent monitoring provides better visibility into the root cause of issues.
- **Leveraging a scalable, resilient platform:** The multi-region GCP architecture with managed services inherently provides this.
- **Span multiple environments seamlessly:** Shared VPC, Global Load Balancer, and consistent CI/CD pipelines facilitate this.
- **Consistent and stable user experience:** Achieved through reduced latency (GCLB, CDN), high availability (multi-zone/region deployments), and proactive monitoring/scaling preventing overload-related instability.
- **Positions us for future growth:** The scalable and flexible nature of cloud services allows EHR Healthcare to quickly adapt to increasing demand and onboard new providers/data sources without major infrastructure overhauls.

Migration Strategy Considerations:

- **Phased Approach:** Migrate critical components incrementally (e.g., databases first, then applications).
- **Data Migration Service (DMS):** Utilize DMS for online, minimal-downtime migration of MySQL and SQL Server databases.
- **Containerization:** Ensure all customer-facing applications are properly containerized and compatible with Kubernetes.
- **Testing:** Thoroughly test applications and integrations in the GCP environment before cutting over production traffic.
- **Legacy Decommissioning:** Plan the retirement of on-premises systems once their functionality is fully replaced or integrated with cloud services.

Security and Compliance (Deep Dive):

Given the healthcare context, a detailed compliance strategy is paramount. Beyond the BAA and basic security services, EHR Healthcare must:

- Map HIPAA/HITRUST controls to specific GCP service configurations.
- Implement strong access controls (IAM) down to the resource level.
- Ensure data is encrypted at rest (managed services provide this) and in transit (TLS/SSL for application traffic, IPsec/encryption for Cloud Interconnect/VPN).
- Configure Cloud Audit Logs to capture all API calls and access events, retaining logs as required by regulations.
- Establish data loss prevention (DLP) scans if handling unstructured sensitive data.
- Regularly review security configurations and conduct vulnerability assessments.
- Train staff on cloud security best practices and compliance requirements.

- Mountkirk

1. Support multiple gaming platforms:

- **Solution:**
    - **Game Client Agnostic Backend:** By building the game backend on Google Kubernetes Engine (GKE), Mountkirk Games can create a set of core game logic and APIs that are independent of the client platform (mobile, web, console, etc.).
    - **API Gateway (Cloud Endpoints or API Gateway):** Implement an API gateway to manage and secure access to the backend services. This provides a consistent interface for all game clients, handling authentication, request routing, and potentially rate lim iting.
    - **Server-Side Rendering (Leveraging GPUs):** As mentioned in the technical requirements, using GPU processing on the server -side (via Compute Engine with NVIDIA GPUs) allows for rendering game frames and streaming them to various client devices. This shi fts the heavy lifting from the client to the cloud, enabling support for less powerful devices.
    - **WebRTC for Streaming:** For web-based platforms, WebRTC can provide low-latency, real-time communication for streaming game visuals and handling player input.

2. Support multiple regions:

- **Solution:**
    - **GKE Regional Clusters:** Deploy GKE clusters in multiple Google Cloud regions geographically close to their player base. This ensures low latency for players in different parts of the world.
    - **Global Load Balancer:** Utilize the Global HTTP(S) Load Balancer to intelligently route players to the nearest available and healthy regional GKE cluster. This provides automatic failover and optimal performance.
    - **Multi-Region Spanner:** As planned, the multi-region Spanner cluster will provide a globally consistent and highly available database for the global leaderboard and potentially other game state that needs to be synchronized across regions.

3. Support rapid iteration of game features:

- **Solution:**
    - **Containerization (Docker & GKE):** GKE provides an agile environment for deploying and managing containerized game backend services. This allows for faster build, test, and release cycles for new features and bug fixes.
    - **CI/CD Pipelines (Cloud Build, Cloud Deploy):** Implement robust CI/CD pipelines to automate the process of building, testing, and deploying new versions of the game backend to the GKE clusters.
    - **Blue/Green Deployments or Canary Releases:** Leverage GKE's deployment strategies to roll out new features to a small subset of users (canary) or deploy a new version alongside the old one (blue/green) for testing and easy rollback if needed.
    - **Feature Flags:** Implement feature flags to enable or disable new features without requiring a full deployment, allowing for controlled rollout and A/B testing.

4. Minimize latency:

- **Solution:**
    - **Regional GKE Clusters & Global Load Balancer:** As mentioned above, deploying regionally and using the Global Load Balancer is crucial for routing players to the closest game server, minimizing network latency.
    - **Low-Latency Network:** Google Cloud's global network is designed for high throughput and low latency.

    - **Optimized Game Server Code:** Efficient game server code and network protocols are essential for minimizing processing and communication delays.
    - **Cloud CDN (Content Delivery Network):** While primarily for static content, Cloud CDN could potentially be used to cache game assets closer to players, reducing download times.
    - **Server-Side Rendering with Low-Latency Streaming:** If adopting server-side rendering, the streaming technology (like WebRTC) needs to be optimized for low latency to provide a responsive gaming experience.

5. Optimize for dynamic scaling:

- **Solution:**
    - **GKE Horizontal Pod Autoscaler (HPA):** Automatically scale the number of game server pods within each GKE cluster based on resource utilization (CPU, memory, custom metrics).
    - **GKE Cluster Autoscaler:** Automatically adjust the size of the underlying Compute Engine node pools in the GKE clusters based on the demand for pods.
    - **Multi-Region Spanner:** Spanner's auto-scaling capabilities ensure that the database can handle fluctuating loads without manual intervention.
    - **Cloud Functions or Cloud Run (for event-driven tasks):** For less latency-sensitive backend tasks (e.g., processing game logs), consider using serverless options that scale automatically based on events.

6. Use managed services and pooled resources:

- **Solution:**
    - **GKE:** A fully managed Kubernetes service that offloads the operational burden of managing the underlying infrastructure.
    - **Global Load Balancer:** A managed load balancing service.
    - **Multi-Region Spanner:** A fully managed, globally distributed database service.
    - **Cloud Storage:** For storing game assets, logs, and backups, providing scalable and cost-effective storage.
    - **Cloud Monitoring and Cloud Logging:** Managed services for monitoring the health and performance of the game and its infrastructure.
    - **Cloud Build and Cloud Deploy:** Managed services for CI/CD pipelines.

7. Minimize costs:

- **Solution:**
    - **Autoscaling:** Dynamically scaling resources up and down based on demand ensures that Mountkirk Games only pays for what they use.
    - **Preemptible VMs (for non-critical workloads):** For development, testing, or even some batch processing tasks, preemptible VMs can significantly reduce costs.
    - **Rightsizing Recommendations:** Google Cloud provides recommendations for optimizing the size of Compute Engine instances.
    - **Committed Use Discounts:** For predictable workloads, committing to a certain level of resource usage can lead to significant discounts.
    - **Spot VMs (for flexible workloads):** Similar to preemptible VMs but can be interrupted with a short notice. Suitable for fault-tolerant workloads.
    - **Serverless Options (Cloud Functions, Cloud Run):** For event-driven or stateless backend logic, serverless can be more cost-effective as you only pay for execution time.
    - **Cost Monitoring and Analysis (Cloud Billing):** Regularly monitor billing data and use tools like Cloud Billing reports and dashboards to identify cost optimization opportunities.

Addressing Technical Requirements

Here's a detailed look at how GCP services can address the technical needs:

1. Dynamically scale based on game activity:

- **Solution:**
    - **GKE with HPA and Cluster Autoscaler:** As mentioned earlier, these Kubernetes features will automatically adjust the number of game server instances and the underlying compute resources based on metrics like CPU utilization, memory usage, or custom metrics related to player concurrency.
    - **Spanner Auto-Scaling:** Spanner automatically scales its processing and storage capacity based on demand, ensuring the global leaderboard remains responsive even with a large number of concurrent players.

2. Publish scoring data on a near real-time global leaderboard:

- **Solution:**
    - **Game Server Integration:** Game servers running in GKE will need to publish scoring events to Spanner in near real-time.
    - **Multi-Region Spanner:** The multi-region Spanner cluster provides the low-latency, high-availability, and global consistency required for a real-time global leaderboard. Data written in one region is quickly replicated to others, ensuring all players see an up-to-date view.
    - **Leaderboard Service (on GKE):** A dedicated service running on GKE can query Spanner and potentially perform aggregations or ranking calculations before serving the leaderboard data to the game clients. This service can also leverage caching mechanisms (e.g., Memorystore) for faster read access.
    - **WebSockets or Server-Sent Events (SSE):** To provide a near real-time experience on the client-side, the leaderboard service can use WebSockets or SSE to push updates to connected game clients whenever the leaderboard changes.

3. Store game activity logs in structured files for future analysis:

- **Solution:**
    - **Cloud Logging:** Configure the game servers and other backend components to write logs to Cloud Logging.
    - **Structured Logging:** Implement structured logging (e.g., using JSON format) to make the logs easier to query and analyze.
    - **Log Sinks to Cloud Storage:** Create a Cloud Logging sink to automatically export relevant logs to Cloud Storage buckets.
    - **BigQuery for Analysis:** Link the Cloud Storage bucket to BigQuery, allowing data analysts to run complex SQL queries on the structured log data for player behavior analysis, game telemetry, and identifying trends.

4. Use GPU processing to render graphics server-side for multi-platform support:

- **Solution:**
    - **Compute Engine with NVIDIA GPUs:** Provision Compute Engine instances equipped with powerful NVIDIA GPUs in the regional GKE clusters.
    - **GPU Node Pools in GKE:** Configure specific node pools within the GKE clusters that utilize these GPU-enabled Compute Engine instances.
    - **Containerization of Rendering Software:** Containerize the game server software along with the necessary rendering libraries and drivers to run on the GPU nodes.
    - **Streaming Technology (WebRTC, etc.):** Implement a low-latency streaming solution (like WebRTC) to capture the rendered frames from the server-side GPUs and transmit them to the various client platforms. Player input needs to be sent back to the server in real-time as well.

5. Support eventual migration of legacy games to this new platform:

- **Solution:**
    - **API-Driven Architecture:** The new platform's API-driven design will make it easier to integrate with legacy game systems.
    - **"Strangler Fig" Pattern:** Gradually replace components of the legacy games with new services running on the GKE platform. This could involve creating new APIs for game logic, data storage, or matchmaking and slowly redirecting traffic from the old systems.
    - **Hybrid Connectivity (Cloud Interconnect or VPN):** Ensure secure and low-latency connectivity between the existing lift-and-shifted

VMs and the new GKE environment during the migration process.
    - **Data Migration Strategies:** Plan and execute data migration strategies to move relevant game data from the legacy systems to the new platform's data stores (potentially Spanner or other appropriate GCP databases).
    - **Containerization of Legacy Backends (where feasible):** If the legacy game backends can be containerized, they could potentially be run on GKE alongside the new game, simplifying management and providing a consistent scaling environment. However, this might require significant refactoring.

By strategically leveraging these Google Cloud services and following cloud-native design principles, Mountkirk Games can build a highly scalable, low-latency, and globally accessible retro-style FPS game that meets both their business and technical requirements. The existing Google Cloud foundation and their experience with cloud analytics will be valuable assets in this ambitious endeavor.

- Helicopter racing

Business Requirements:

- **Support ability to expose the predictive models to partners.**
    - **How to meet:** Develop well-documented APIs (Application Programming Interfaces) using a service like Google Cloud's API Gateway. This allows controlled and secure access to the prediction models for authorized partners. Consider different API key managem ent strategies and potentially offer tiered access based on partnership levels.
- **Increase predictive capabilities during and before races:**
    - **Race results:**
        - **How to meet:** Leverage machine learning services like Vertex AI to train more sophisticated models using historical race data, pilot performance, helicopter specifications, and weather conditions. Implement real-time data ingestion pipelines to feed live race telemetry into the models for dynamic predictions.
    - **Mechanical failures:**
        - **How to meet:** Integrate sensor data from the helicopters (e.g., engine temperature, rotor speed, hydraulic pressure) and use anomaly detection ML models to predict potential mechanical issues. This could involve time-series analysis and predictive maintenance techniques within Vertex AI.
    - **Crowd sentiment:**
        - **How to meet:** Employ Natural Language Processing (NLP) models through services like Vertex AI Natural Language API to analyze social media feeds, live chat during streams, and potentially fan surveys to gauge real-time and overall crowd sentiment.
- **Increase telemetry and create additional insights.**
    - **How to meet:** Implement a robust data ingestion pipeline using services like Pub/Sub and Dataflow to collect a wider range of telemetry data from helicopters (e.g., precise location, altitude, speed, acceleration, pilot biometrics if available). Utilize data pr ocessing and analytics services like BigQuery and Looker to analyze this data and generate new insights, such as optimal racing lines, pil ot fatigue patterns, and helicopter performance under different conditions.
- **Measure fan engagement with new predictions.**
    - **How to meet:** Integrate analytics into the streaming platform to track how viewers interact with the prediction features (e.g., toggling predictions on/off, viewing prediction probabilities, comparing predictions to actual events). Use services like Google Analy tics or Firebase Analytics to capture this data and visualize it in dashboards using Looker to understand user adoption and preferenc es. A/B testing different prediction features can also provide valuable insights.
- **Enhance global availability and quality of the broadcasts.**
    - **How to meet:** Utilize a Content Delivery Network (CDN) like Cloud CDN to cache video content closer to users globally, reducing latency and improving streaming quality. Implement multi-region deployment for critical services to ensure high availability and resilience.
- **Increase the number of concurrent viewers.**
    - **How to meet:** Employ scalable infrastructure services like Compute Engine autoscaling for the streaming platform and prediction services. Optimize the video encoding and delivery pipeline for efficiency. Leverage load balancing services to distribute tr affic across multiple instances.
- **HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets.**
    - **How to meet:** Continue to invest in and refine the prediction models using Vertex AI. The mobile data centers with local compute can be strategically deployed in emerging markets to serve the low-latency video content. While the race prediction services are hosted in the existing public cloud, ensure efficient data transfer between the mobile data centers (for telemetry) and the cloud for r eal-time predictions and insights.

Technical Requirements:

- **Maintain or increase prediction throughput and accuracy.**
    - **How to meet:** Optimize the existing TensorFlow models for performance using techniques like quantization and distributed training on Vertex AI. Continuously monitor model performance and retrain with new data to improve accuracy. Design the prediction pipeli ne to handle a high volume of real-time data with low latency.
- **Reduce viewer latency.**
    - **How to meet:** As mentioned earlier, implement Cloud CDN. Optimize the video encoding process to reduce the time it takes to prepare streams. Ensure low-latency data pipelines for delivering real-time predictions to the streaming platform.
- **Increase transcoding performance.**
    - **How to meet:** Migrate the transcoding process to a more scalable and managed service like Google Cloud's Transcoder API. This service is designed for high-performance video processing and can handle varying resolutions and formats efficiently. It also reduces the operational overhead of managing transcoding VMs.
- **Create real-time analytics of viewer consumption patterns and engagement.**
    - **How to meet:** Implement a real-time data pipeline using Pub/Sub to ingest viewer activity data from the streaming platform. Use Dataflow for real-time stream processing and analysis. Store the processed data in a low-latency data store like Bigtable or Firestore for fast querying and visualization with Looker dashboards.
- **Create a data mart to enable processing of large volumes of race data.**
    - **How to meet:** Utilize BigQuery as the data warehouse to build the data mart. Ingest historical race data, telemetry data, prediction results, and viewer engagement data into BigQuery. Leverage BigQuery's SQL capabilities for efficient querying and analysis o f large datasets.
- **Minimize operational complexity.**
    - **How to meet:** Embrace managed services across the platform (e.g., Vertex AI, Transcoder API, Pub/Sub, Dataflow, BigQuery, Cloud CDN). This reduces the need for manual infrastructure management. Implement Infrastructure-as-Code (IaC) using tools like Terraform to automate infrastructure provisioning and management. Utilize monitoring and logging services like Cloud Monitoring and Clo ud Logging for proactive issue detection and troubleshooting.
- **Ensure compliance with regulations.**
    - **How to meet:** Understand the relevant data privacy and security regulations in the regions where HRL operates. Implement appropriate security measures, including data encryption at rest and in transit, access controls (IAM), and audit logging. Leverage Googl e Cloud's compliance certifications and features to meet regulatory requirements.
- **Create a merchandising revenue stream.**
    - **How to meet:** Integrate the prediction capabilities and real-time insights into the fan experience to drive engagement. This increased engagement can then be leveraged to promote merchandise. Consider:

- **Personalized recommendations:** Based on fan preferences and predictions they engage with.
- **Exclusive merchandise:** Tied to specific predictions or race outcomes.
- **Interactive experiences:** Where fans can use predictions to earn discounts or exclusive items.
- **Data-driven insights for marketing:** Understanding fan behaviour through the analytics platform to target merchandising efforts effectively.

By addressing these business and technical requirements with a focus on leveraging managed services, robust data pipelines, and scalable infrastructure, the Helicopter Racing League can enhance its offerings, reach a global audience, and create new revenue streams.

70) A global insurance company that uses Google Cloud Platform plans for its employees to work from home. It has requested a scalable, cost-effective solution that can enable encrypted desktop streaming so that employees can access corporate resources. Which of the following would you recommend?

A. Google workspace
B. Google Cloud Virtual Desktop
C. Create an encrypted connection to the office network
D. Enable Remote Desktop Protocol (RDP) to connect to remote desktops

Correct Answer: *B*

74) You are building a continuous deployment pipeline for a project stored in a Git source repository and want to ensure that code changes can be verified before deploying to production. What should you do?

A. Use Spinnaker to deploy builds to production using the red/black deployment strategy so that changes can easily be rolled back.
B. Use Spinnaker to deploy builds to production and run tests on production deployments.
C. Use Jenkins to build the staging branches and the master branch. Build and deploy changes to production for 10% of users before doing a complete rollout.
D. Use Jenkins to monitor tags in the repository. Deploy staging tags to a staging environment for testing. After testing, tag the repository for production and deploy that to the production environment.

Correct Answer: *D*

73) You are analyzing and defining business processes to support your startup's trial usage of GCP, and you don't yet know what consumer demand for your product will be. Your manager requires you to minimize GCP service costs and adhere to Google best practices. What should you do?

A. Utilize free tier and sustained use discounts. Provision a staff position for service cost management.
B. Utilize free tier and sustained use discounts. Provide training to the team about service cost management.
C. Utilize free tier and committed use discounts. Provision a staff position for service cost management.
D. Utilize free tier and committed use discounts. Provide training to the team about service cost management.

Correct Answer: *B*

78) You are working as a Security Consultant in an organization. Your manager has asked you to ensure that packages in container images are updated, and any images with security issues should not be pushed to your live Google Kubernetes Engine environment. Which of the following services would you use?

A. Deployment strategies
B. Binary Authorization
C. Vulnerability Scanning
D. Vulnerability Scanning with Binary Authorization

Correct Answer: *D*

80) Google Cloud Platform resources are managed hierarchically using organization, folders, and projects. When Cloud Identity and Access Management (IAM) policies exist at these different levels, what is the effective policy at a particular node of the hierarchy?

A. The effective policy is determined only by the policy set at the node
B. The effective policy is the policy set at the node and restricted by the policies of its ancestors
C. The effective policy is the union of the policy set at the node and policies inherited from its ancestors
D. The effective policy is the intersection of the policy set at the node and policies inherited from its ancestors

Correct Answer: *C*

87) You have an application deployed on Google Kubernetes Engine using a Deployment named echo-deployment. The deployment is exposed using a Service called echo-service. You need to perform an update to the application with minimal downtime to the application. What should you do?

A. Use kubectl set image deployment/echo-deployment <new-image>
B. Use the rolling update functionality of the Instance Group behind the Kubernetes cluster
C. Update the deployment yaml file with the new container image. Use kubectl delete deployment/echo-deployment and kubectl create "€xf <yaml-file>
D. Update the service yaml file which the new container image. Use kubectl delete service/echo-service and kubectl create "€xf <yaml-file>

Correct Answer: *A*

88) Your company is using BigQuery as its enterprise data warehouse. Data is distributed over several Google Cloud projects. All queries on BigQuery need to be billed on a single project. You want to make sure that no query costs are incurred on the projects that contain the data. Users should be able to query the datasets, but not edit them.
How should you configure users' access roles?

A. Add all users to a group. Grant the group the role of BigQuery user on the billing project and BigQuery dataViewer on the projects that contain the data.
B. Add all users to a group. Grant the group the roles of BigQuery dataViewer on the billing project and BigQuery user on the projects that contain the data.
C. Add all users to a group. Grant the group the roles of BigQuery jobUser on the billing project and BigQuery dataViewer on the projects that contain the data.
D. Add all users to a group. Grant the group the roles of BigQuery dataViewer on the billing project and BigQuery jobUser on the projects that contain the data.

Correct Answer: *C*

90) Your web application must comply with the requirements of the European Union's General Data Protection Regulation (GDPR). You are responsible for the technical architecture of your web application. What should you do?

A. Ensure that your web application only uses native features and services of Google Cloud Platform, because Google already has various certifications and provides €œpass-on€ compliance when you use native features.
B. Enable the relevant GDPR compliance setting within the GCPConsole for each of the services in use within your application.
C. Ensure that Cloud Security Scanner is part of your test planning strategy in order to pick up any compliance gaps.
D. Define a design for the security of data in your web application that meets GDPR requirements.

Correct Answer: *D*

94) You are creating an App Engine application that uses Cloud Datastore as its persistence layer. You need to retrieve several root entities for which you have the identifiers. You want to minimize the overhead in operations performed by Cloud Datastore. What should you do?

A. Create the Key object for each Entity and run a batch get operation
B. Create the Key object for each Entity and run multiple get operations, one operation for each entity
C. Use the identifiers to create a query filter and run a batch query operation
D. Use the identifiers to create a query filter and run multiple query operations, one operation for each entity

Correct Answer: *A*

100) Your company is running a stateless application on a Compute Engine instance. The application is used heavily during regular business hours and lightly outside of business hours. Users are reporting that the application is slow during peak hours. You need to optimize the application's performance. What should you do?

A. Create a snapshot of the existing disk. Create an instance template from the snapshot. Create an autoscaled managed instance group from the instance template.
B. Create a snapshot of the existing disk. Create a custom image from the snapshot. Create an autoscaled managed instance group from the custom image.
C. Create a custom image from the existing disk. Create an instance template from the custom image. Create an autoscaled managed instance group from the instance template.
D. Create an instance template from the existing disk. Create a custom image from the instance template. Create an autoscaled managed instance group from the custom image.

Correct Answer: *C*

113) Your web application uses Google Kubernetes Engine to manage several workloads. One workload requires a consistent set of hostnames even after pod scaling and relaunches.
Which feature of Kubernetes should you use to accomplish this?

A. StatefulSets
B. Role-based access control
C. Container environment variables
D. Persistent Volumes

Correct Answer: *A*

115) Your architecture calls for the centralized collection of all admin activity and VM system logs within your project.
How should you collect these logs from both VMs and services?

A. All admin and VM system logs are automatically collected by Stackdriver.
B. Stackdriver automatically collects admin activity logs for most services. The Stackdriver Logging agent must be installed on each instance to collect system logs.
C. Launch a custom syslogd compute instance and configure your GCP project and VMs to forward all logs to it.
D. Install the Stackdriver Logging agent on a single compute instance and let it collect all audit and access logs for your environment.

Correct Answer: *B*

116) You have an App Engine application that needs to be updated. You want to test the update with production traffic before replacing the current application version.
What should you do?

A. Deploy the update using the Instance Group Updater to create a partial rollout, which allows for canary testing.
B. Deploy the update as a new version in the App Engine application, and split traffic between the new and current versions.
C. Deploy the update in a new VPC, and use Google's global HTTP load balancing to split traffic between the update and current applications.
D. Deploy the update as a new App Engine application, and use Google's global HTTP load balancing to split traffic between the new and current applications.

Correct Answer: *B*

117) All Compute Engine instances in your VPC should be able to connect to an Active Directory server on specific ports Any other traffic emerging from your instances is not allowed. You want to enforce this using VPC firewall rules. How should you configure the firewall rules?

A. Create an egress rule with priority 1000 to deny all traffic for all instances. Create another egress rule with priority 100 to allow the Active Directory traffic for all instances.
B. Create an egress rule with priority 100 to deny all traffic for all instances. Create another egress rule with priority 1000 to allow the Active Directory traffic for all instances.
C. Create an egress rule with priority 1000 to allow the Active Directory traffic. Rely on the implied deny egress rule with priority 100 to block all traffic for all instances.
D. Create an egress rule with priority 100 to allow the Active Directory traffic. Rely on the implied deny egress rule with priority 1000 to block all traffic for all instances.

Correct Answer: *A*

119) You need to design a solution for global load balancing based on the URL path being requested. You need to ensure operations reliability and end-to-end in- transit encryption based on Google best practices. What should you do?

A. Create a cross-region load balancer with URL Maps.
B. Create an HTTPS load balancer with URL Maps.
C. Create appropriate instance groups and instances. Configure SSL proxy load balancing.
D. Create a global forwarding rule. Configure SSL proxy load balancing.

Correct Answer: *B*

124) You need to ensure reliability for your application and operations by supporting reliable task scheduling for compute on GCP. Leveraging Google best practices, what should you do?

A. Using the Cron service provided by App Engine, publish messages directly to a message-processing utility service running on Compute Engine instances.
B. Using the Cron service provided by App Engine, publish messages to a Cloud Pub/Sub topic. Subscribe to that topic using a message-processing utility service running on Compute Engine instances.
C. Using the Cron service provided by Google Kubernetes Engine (GKE), publish messages directly to a message-processing utility service running on Compute Engine instances.
D. Using the Cron service provided by GKE, publish messages to a Cloud Pub/Sub topic. Subscribe to that topic using a message-processing utility service running on Compute Engine instances.

Correct Answer: *B*

129) You are implementing a single Cloud SQL MySQL second-generation database that contains business-critical transaction data. You want to ensure that the minimum amount of data is lost in case of catastrophic failure. Which two features should you implement? (Choose two.)

A. Sharding
B. Read replicas
C. Binary logging
D. Automated backups
E. Semisynchronous replication

Correct Answer: *CD*

130) You are working at a sports association whose members range in age from 8 to 30. The association collects a large amount of health data, such as sustained injuries. You are storing this data in BigQuery. Current legislation requires you to delete such information upon request of the subject. You want to design a solution that can accommodate such a request. What should you do?

A. Use a unique identifier for each individual. Upon a deletion request, delete all rows from BigQuery with this identifier.
B. When ingesting new data in BigQuery, run the data through the Data Loss Prevention (DLP) API to identify any personal information. As part of the DLP scan, save the result to Data Catalog. Upon a deletion request, query Data Catalog to find the column with personal information.
C. Create a BigQuery view over the table that contains all data. Upon a deletion request, exclude the rows that affect the subject's data from this view. Use this view instead of the source table for all analysis tasks.
D. Use a unique identifier for each individual. Upon a deletion request, overwrite the column with the unique identifier with a salted SHA256 of its value.

Correct Answer: *B*

Update type

Managed instance groups support two types of update:

- Automatic, or **proactive**, updates
- Selective, or **opportunistic**, updates

If you want to apply updates automatically, set the type to **proactive**.

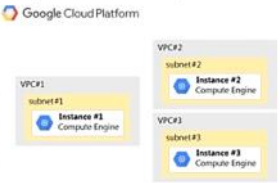
Alternatively, if an automated update is potentially too disruptive, you can choose to perform an **opportunistic** update. The MIG applies an opportunistic update only when you manually initiate the update on selected instances or when new instances are created. New instances can be created when you or another service, such as an autoscaler, resizes the MIG. Compute Engine does not actively initiate requests to apply opportunistic updates on existing instances.

138) Your company is designing its application landscape on Compute Engine. Whenever a zonal outage occurs, the application should be restored in another zone as quickly as possible with the latest application data. You need to design the solution to meet this requirement. What should you do?

A. Create a snapshot schedule for the disk containing the application data. Whenever a zonal outage occurs, use the latest snapshot to restore the disk in the same zone.
B. Configure the Compute Engine instances with an instance template for the application, and use a regional persistent disk for the application data. Whenever a zonal outage occurs, use the instance template to spin up the application in another zone in the same region. Use the regional persistent disk for the application data.
C. Create a snapshot schedule for the disk containing the application data. Whenever a zonal outage occurs, use the latest snapshot to restore the disk in another zone within the same region.
D. Configure the Compute Engine instances with an instance template for the application, and use a regional persistent disk for the application data. Whenever a zonal outage occurs, use the instance template to spin up the application in another region. Use the regional persistent disk for the application data.

Correct Answer: *B*

141) Your company has a project in Google Cloud with three Virtual Private Clouds (VPCs). There is a Compute Engine instance on each VPC. Network subnets do not overlap and must remain separated. The network configuration is shown below.



Instance #1 is an exception and must communicate directly with both Instance #2 and Instance #3 via internal IPs. How should you accomplish this?

A. Create a cloud router to advertise subnet #2 and subnet #3 to subnet #1.
B. Add two additional NICs to Instance #1 with the following configuration: €¢ ›NIC1 ‹ –›VPC: VPC #2 ‹ –›SUBNETWORK: subnet #2 €¢ ›NIC2 ‹ –›VPC: VPC #3 ‹ –›SUBNETWORK: subnet #3 Update firewall rules to enable traffic between instances.
C. Create two VPN tunnels via CloudVPN: 1€¢ ›between VPC #1 and VPC #2. 1€¢ ›between VPC #2 and VPC #3. Update firewall rules to enable traffic between the instances.
D. Peer all three VPCs: €¢ ›Peer VPC #1 with VPC #2. €¢ ›Peer VPC #2 with VPC #3. Update firewall rules to enable traffic between the instances.

Correct Answer: *B*

This page provides an overview of Cloud Storage FUSE, a FUSE adapter that lets you mount and access Cloud Storage buckets as local file systems, so applications can read and write objects in your bucket using standard file system semantics.

This documentation always reflects the latest version of Cloud Storage FUSE. For details on the latest version, see Cloud Storage FUSE releases on GitHub.

148) You are developing an application using different microservices that should remain internal to the cluster. You want to be able to configure each microservice with a specific number of replicas. You also want to be able to address a specific microservice from any other microservice in a uniform way, regardless of the number of replicas the microservice scales to. You need to implement this solution on Google Kubernetes Engine. What should you do?

A. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using a Service, and use the Service DNS name to address it from other microservices within the cluster.
B. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using an Ingress, and use the Ingress IP address to address the Deployment from other microservices within the cluster.
C. Deploy each microservice as a Pod. Expose the Pod in the cluster using a Service, and use the Service DNS name to address the microservice from other microservices within the cluster.
D. Deploy each microservice as a Pod. Expose the Pod in the cluster using an Ingress, and use the Ingress IP address name to address the Pod from other microservices within the cluster.

Correct Answer: *A*

156) For this question, refer to the Mountkirk Games case study. Mountkirk Games' gaming servers are not automatically scaling properly. Last month, they rolled out a new feature, which suddenly became very popular. A record number of users are trying to use the service, but many of them are getting 503 errors and very slow response times.
What should they investigate first? (Case Study 1)

A. Verify that the database is online.
B. Verify that the project quota hasn't been exceeded.
C. Verify that the new feature code did not introduce any performance bugs.
D. Verify that the load-testing team is not running their tool against production.

Answer(s): B

Explanation:
503 is service unavailable error. If the database was online everyone would get the 503 error.
https://cloud.google.com/docs/quota#capping_usage

157) For this question, refer to the Mountkirk Games case study Mountkirk Games needs to create a repeatable and configurable mechanism for deploying isolated application environments. Developers and testers can access each other's environments and resources, but they cannot access staging or production resources. The staging environment needs access to some services from production. (Case Study 1)
What should you do to isolate development environments from staging and production?

A. Create a project for development and test and another for staging and production.
B. Create a network for development and test and another for staging and production.
C. Create one subnetwork for development and another for staging and production.
D. Create one project for development, a second for staging and a third for production.

195) For this question, refer to the TerramEarth case study. TerramEarth has decided to store data files in Cloud Storage. You need to configure Cloud Storage lifecycle rule to store 1 year of data and minimize file storage cost.
Which two actions should you take? (Case Study 6)

A. Create a Cloud Storage lifecycle rule with Age: "30", Storage Class: "Standard", and Action: "Set to Coldline", and create a second GCS life-cycle rule with Age: "365", Storage Class: "Coldline", and Action: "Delete".
B. Create a Cloud Storage lifecycle rule with Age: "30", Storage Class: "Coldline", and Action: "Set to Nearline", and create a second GCS life-cycle rule with Age: "91", Storage Class: "Coldline", and Action: "Set to Nearline".
C. Create a Cloud Storage lifecycle rule with Age: "90", Storage Class: "Standard", and Action: "Set to Nearline", and create a second GCS life-cycle rule with Age: "91", Storage Class: "Nearline", and Action: "Set to Coldline".
D. Create a Cloud Storage lifecycle rule with Age: "30", Storage Class: "Standard", and Action: "Set to Coldline", and create a second GCS life-cycle rule with Age: "365", Storage Class: "Nearline", and Action: "Delete".

Answer(s): A

204) You have an application that runs in Google Kubernetes Engine (GKE). Over the last 2 weeks, customers have reported that a specific part of the application returns errors very frequently. You currently have no logging or monitoring solution enabled on your GKE cluster. You want to diagnose the problem, but you have not been able to replicate the issue. You want to cause minimal disruption to the application. What should you do?

A. 1. Update your GKE cluster to use Cloud Operations for GKE. 2. Use the GKE Monitoring dashboard to investigate logs from affected Pods.
B. 1. Create a new GKE cluster with Cloud Operations for GKE enabled. 2. Migrate the affected Pods to the new cluster, and redirect traffic for those Pods to the new cluster. 3. Use the GKE Monitoring dashboard to investigate logs from affected Pods.
C. 1. Update your GKE cluster to use Cloud Operations for GKE, and deploy Prometheus. 2. Set an alert to trigger whenever the application returns an error.
D. 1. Create a new GKE cluster with Cloud Operations for GKE enabled, and deploy Prometheus. 2. Migrate the affected Pods to the new cluster, and redirect traffic for those Pods to the new cluster. 3. Set an alert to trigger whenever the application returns an error.

Correct Answer: *A*

207) You are working in an Infrastructure team where you are using Google Cloud Platform as your Cloud Provider. Your boss ha asked you to configure a compute engine to host an application that needs access to a recently created Storage Bucket. Your bo tells you to use a new Service Account to do this.
How would you achieve the given requirement in the most reliable way?

A. Create a new Service Account with a role as Storage Admin
While configuring Compute Engine, Choose the newly created Service Account and Set the scope to Storage API
Set of path of service account keys in your application

B. Create a Service Account with role as Storage Admin
While configuring Compute Engine, Choose the newly created Service Account and Set the scope to Storage API
No need to Set the path of service account keys in your application

C. Create a Service Account with role as Storage Admin
While configuring Compute Engine, Choose the newly created Service Account.
Set of path of service account keys in your application

D. Create a Service Account with role as Storage Admin
While configuring Compute Engine, Choose the newly created Service Account.
No need to Set the path of service account keys in your application

Correct Answer: *D*
You have chosen the required privileges for the Service account, and if your application is running in the compute engine, you are not required to specify credentials explicitly.

212) For this question, refer to the TerramEarth case study.
(Case Study 6)
You start to build a new application that uses a few Cloud Functions for the backend. One use case requires a Cloud Function func_display to invoke another Cloud Function func_query. You want func_query only to accept invocations from func_display You also want to follow Google's recommended best practices.
What should you do?

A.Create a token and pass it in as an environment variable to func_display.
When invoking func_query, include the token in the request Pass the same token to func _query and reject the invocation if the tokens are different.
B.Make func_query 'Require authentication.' Create a unique service account and associate it to func_display. Grant the service account invoker role for func_query. Create an id token in func_display and include the token to the request when invoking func_query.
C.Make func _query 'Require authentication' and only accept internal traffic. Create those two functions in the same VP Create an ingress firewall rule for func_query to only allow traffic from func_display.
D.Create those two functions in the same project and VPC. Make func_query only accept internal traffic. Create an ingress firewall for func_query to only allow traffic from func_display. Also, make sure both functions use the same service account.

Answer(s): B
Explanation:
https://cloud.google.com/functions/docs/securing/authenticating#authenticating_function_to_func tion_calls

236) For this question, refer to the EHR Healthcare case study. You are a developer on the EHR customer portal team. Your team recently migrated the customer portal application to Google Cloud. The load has increased on the application servers, and now the application is logging many timeout errors. You recently incorporated Pub/Sub into the application architecture, and the application is not logging any Pub/Sub publishing errors. You want to improve publishing latency.
What should you do? (Case Study 9)

A.Increase the Pub/Sub Total Timeout retry value.
B.Move from a Pub/Sub subscriber pull model to a push model.
C.Turn off Pub/Sub message batching.
D.Create a backup Pub/Sub message queue.

Answer(s): C

Explanation:
https://cloud.google.com/pubsub/docs/publisher?hl=en#batching

238) For this question, refer to the EHR Healthcare case study. You are responsible for designing the Google Clou network architecture for Google Kubernetes Engine. You want to follow Google best practices. Considering the EHR Healthcare business and technical requirements, what should you do to reduce the attack surface? (Case Study 9)

A.Use a private cluster with a private endpoint with master authorized networks configured.
B.Use a public cluster with firewall rules and Virtual Private Cloud (VPC) routes.
C.Use a private cluster with a public endpoint with master authorized networks configured.
D.Use a public cluster with master authorized networks enabled and firewall rules.

Answer(s): A

"master authorized networks" is a feature that enables you to restrict access to the Kubernetes control plane in a GKE (Google Kubernetes Engine) cluster to specific IP address ranges.

258) You have created several preemptible Linux virtual machine instances using Google Compute Engine. You want to properly shut down your application before the virtual machines are preempted.
What should you do?

A.Create a shutdown script named k99.shutdown in the /etc/rc.6.d/ directory.
B.Create a shutdown script registered as a xinetd service in Linux and configure a Stackdnver endpoint check to call the service.
C.Create a shutdown script and use it as the value for a new metadata entry with the key shutdown- script in the Cloud Platform Console when you create the new virtual machine instance.
D.Create a shutdown script, registered as a xinetd service in Linux, and use the gcloud compute instances add-metadata command to specify the service URL as the value for a new metadata entry with the key shutdown-script-url

Answer(s): C

266) You are helping the QA team to roll out a new load-testing tool to test the scalability of your primary cloud services that run on Google Compute Engine with Cloud Bigtable.
Which three requirements should they include? Choose 3 answers

A.Ensure that the load tests validate the performance of Cloud Bigtable.
B.Create a separate Google Cloud project to use for the load-testing environment.
C.Schedule the load-testing tool to regularly run against the production environment.
D.Ensure all third-party systems your services use are capable of handling high load.
E.Instrument the production services to record every transaction for replay by the load-testing tool.
F.Instrument the load-testing tool and the target services with detailed logging and metrics collection.

Answer(s): A,B,F

268) Your company runs several databases on a single MySQL instance. They need to take backups of a specific database at regular intervals. The backup activity needs to complete as quickly as possible and cannot be allowed to impact disk performance. How should you configure the storage?

A.Configure a cron job to use the gcloud tool to take regular backups using persistent disk snapshots.
B.Mount a Local SSD volume as the backup location. After the backup is complete, use gsutil to move the backup to Google Cloud Storage.
C.Use gcsfuse to mount a Google Cloud Storage bucket as a volume directly on the instance and write backups to the mounted location using mysqldump
D.Mount additional persistent disk volumes onto each virtual machine (VM) instance in a RAID10 array and use LVM to create snapshots to send to Cloud Storage.

Answer(s): B

272) You want to enable your running Google Container Engine cluster to scale as demand for your application changes.
What should you do?

A.Add additional nodes to your Container Engine cluster using the following command:
gcloud container clusters resize CLUSTER_NAME --size 10
B.Add a tag to the instances in the cluster with the following command:
gcloud compute instances add-tags INSTANCE --tags enable --autoscaling max-nodes-10
C.Update the existing Container Engine cluster with the following command:
gcloud alpha container clusters update mycluster --enable-autoscaling --min-nodes=1 --max- nodes=10
D.Create a new Container Engine cluster with the following command:
gcloud alpha container clusters create mycluster --enable-autocaling --min-nodes=1 --max-nodes=10 and redeploy your application.

Answer(s): B