Google Cloud

Preparing for Your Associate Cloud Engineer Journey

Section 2: Planning and Configuring a Cloud Solution

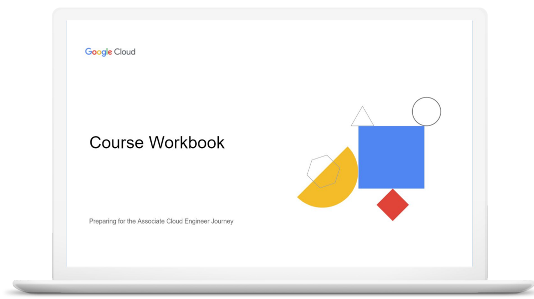
Welcome to section 2 of the exam guide: Planning and Configuring a Cloud Solution.

Review and study planning

Now let's review the diagnostic questions together, paying attention to the areas where you need to focus your study time. Some of this may be content you're already very familiar with, while other aspects may be new to you.

Your study plan:

Planning and configuring cloud solutions



2.1	Planning and configuring compute resources
2.2	Planning and configuring data storage options
2.3	Planning and configuring network resources

Google Cloud

Just like we did with the previous module, we'll approach this review by looking at the objectives of this exam section and the questions you just answered about each one. We'll introduce an objective, briefly review the answers to the related questions, then talk about where you can find out more in the learning resources and/or in Google Cloud documentation. As we go through each section objective, use the page in your workbook to mark the specific documentation, courses (and modules!), and skill badges you'll want to emphasize in your study plan.

2.1 Planning and configuring compute resources

Considerations include:

- Selecting appropriate compute choices for a given workload (e.g., Compute Engine, Google Kubernetes Engine, Cloud Run, Cloud Run functions)
- Using spot VM instances and custom machine types as appropriate

A cloud architect will typically design a cloud solution. As an Associate Cloud Engineer, it’s important for you to be able to plan and configure compute resources to align with Google recommended practices. That means you should be comfortable with the different compute options available in Google Cloud and when to use each one.

You encountered these types of considerations in the following questions:

- Question 1: Differentiate between available compute options in Google Cloud.
- Question 2: Determine when to use serverless solutions such as Cloud Run, e.g. code focus, quick development, minimal overhead.
- Question 3: Determine when to use Compute Engine, e.g. more control, OS level dependencies, customization.
- Question 4: Determine when to use Google Kubernetes Engine, e.g. platform independence, deployment velocity, container management.

2.1 Diagnostic Question 01 Discussion

Cymbal Superstore decides to migrate their supply chain application to Google Cloud. You need to configure specific operating system dependencies.

What should you do?

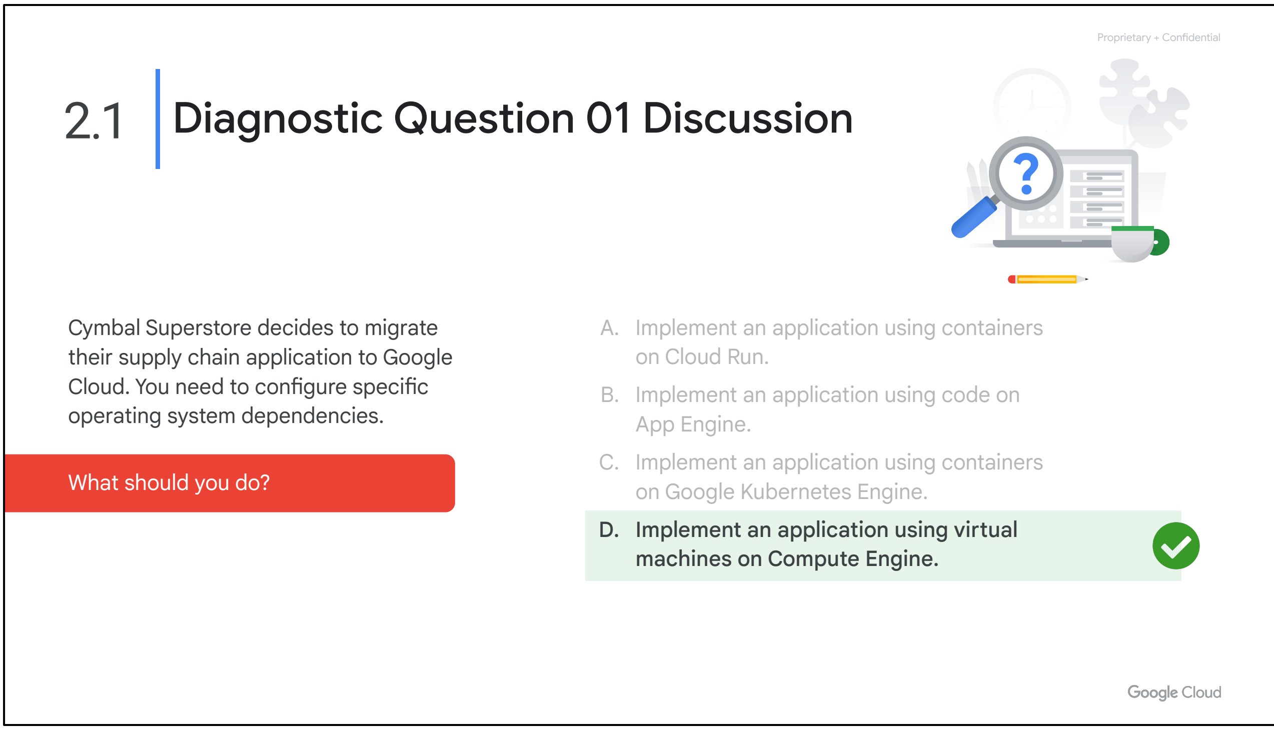
A. Implement an application using containers on Cloud Run.

B. Implement an application using code on App Engine.

C. Implement an application using containers on Google Kubernetes Engine.

D. Implement an application using virtual machines on Compute Engine.

Question:
Cymbal Superstore decides to migrate their supply chain application to Google Cloud. You need to configure specific operating system dependencies. What should you do?



Feedback:
A. Implement an application using containers on Cloud Run.
Feedback: Incorrect. Cloud Run deploys containers in Google Cloud without you specifying the underlying cluster or deployment architecture.

B. Implement an application using code on App Engine.
Feedback: Incorrect. App Engine is a platform as a service for deployment of your code on infrastructure managed by Google. You don’t manage operating system dependencies with App Engine.

C. Implement an application using containers on Google Kubernetes Engine.
Feedback: Incorrect. Google Kubernetes Engine is a container management platform as a service and doesn’t give you control over operating system dependencies.

* D. Implement an application using virtual machines on Compute Engine.
Feedback: Correct! Compute Engine gives you full control over operating system choice and configuration.

Where to look:
https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree

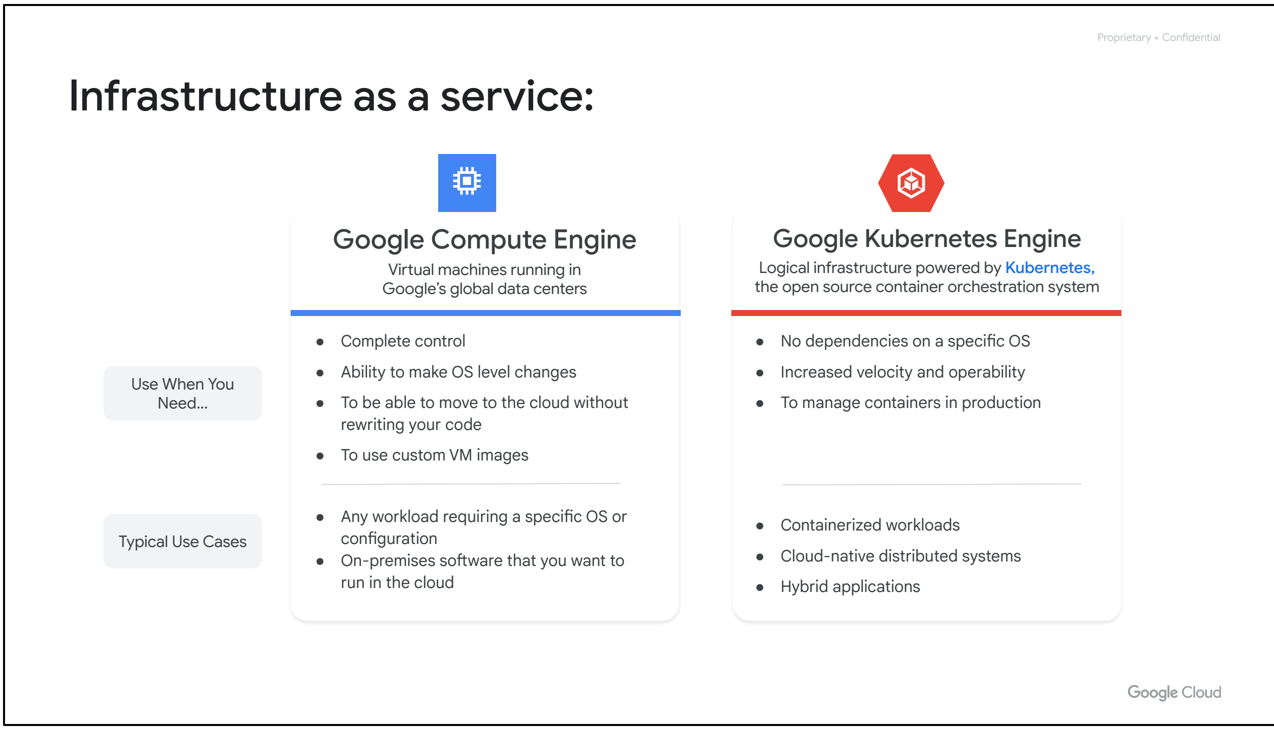
Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
    - M3 Virtual Machines and Networks in the Cloud

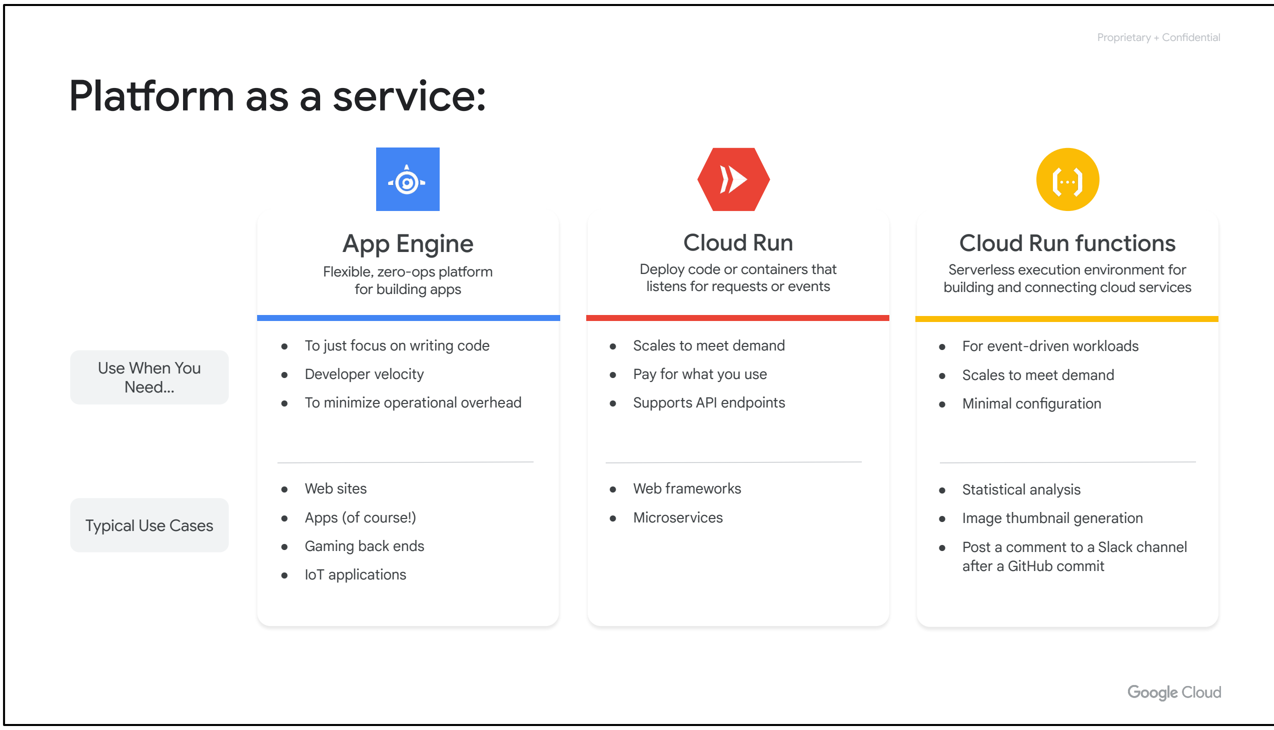
- Architecting with Google Compute Engine (ILT)
    - M3 Virtual Machines

- Essential Google Cloud Infrastructure: Foundation (On-demand)
    - M3 Virtual Machines

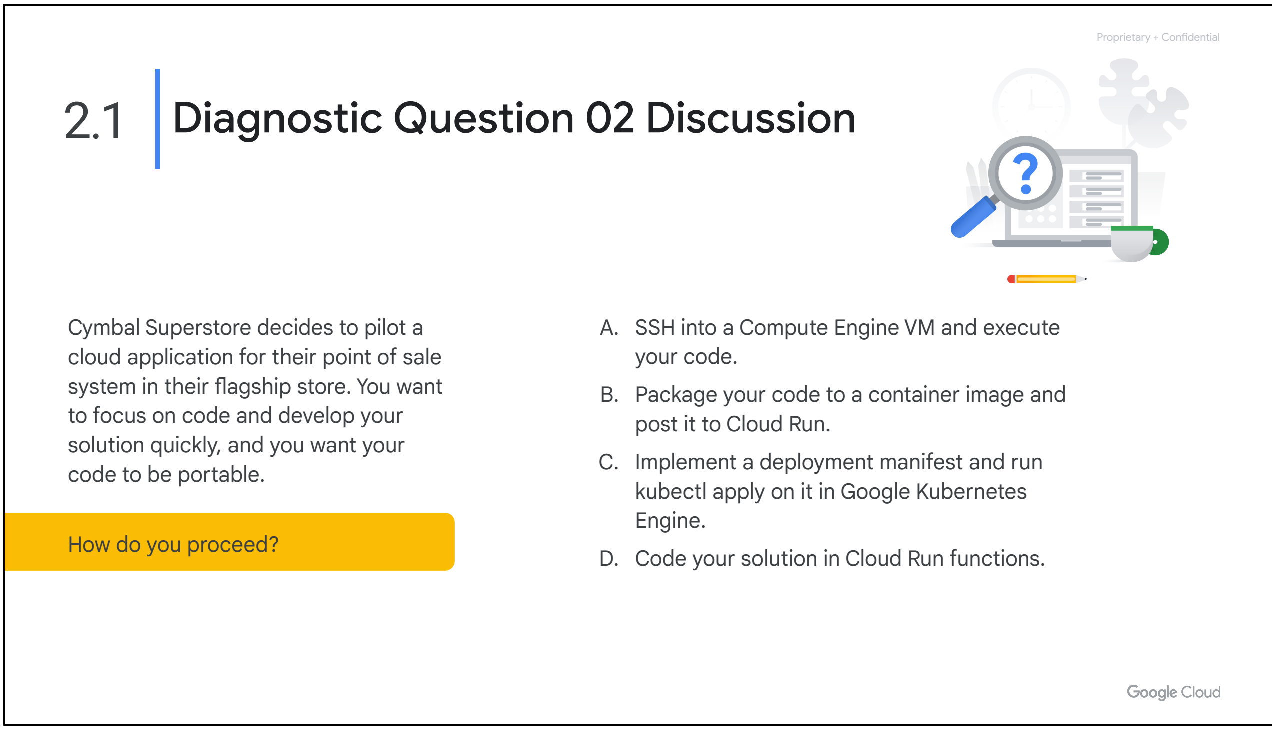
Summary:
There are five distinct ways to interact with compute resources in Google Cloud. They can be divided into server-based services, where manage and pay for infrastructure, and serverless options, where you just pay for execution time.



Compute Engine and Google Kubernetes Engine are server-based.



App Engine, Cloud Run, and Cloud Run functions are serverless options, where you focus on code and Google manages the underlying hardware and operating system for you.



Question:
Cymbal Superstore decides to pilot a cloud application for their point of sale system in their flagship store. You want to focus on code and develop your solution quickly, and you want your code to be portable. How do you proceed?

Proprietary + Confidential

2.1 Diagnostic Question 02 Discussion

Cymbal Superstore decides to pilot a cloud application for their point of sale system in their flagship store. You want to focus on code and develop your solution quickly, and you want your code to be portable.

How do you proceed?

A. SSH into a Compute Engine VM and execute your code.

B. Package your code to a container image and post it to Cloud Run.

C. Implement a deployment manifest and run kubectl apply on it in Google Kubernetes Engine.

D. Code your solution in Cloud Run functions.

Google Cloud

Feedback:
A. SSH into a Compute Engine VM and execute your code.
Feedback: Incorrect. Configuring SSH connectivity to a Compute Engine VM does not meet the focus on code requirement of this scenario.

*B. Package your code to a container image and post it to Cloud Run.
Feedback: Correct! Cloud Run provides serverless container management. It lets you focus on code and you can deploy your solution quickly.

C. Implement a deployment manifest and run kubectl apply on it in Google Kubernetes Engine.
Feedback: Incorrect. Google Kubernetes Engine requires you to build and manage resources of a cluster to host your container in GKE. This does meet the requirement of focusing on code.

D. Code your solution in Cloud Run functions.
Feedback: Incorrect. Cloud Run functions manages your code as short, executable functions and does not manage your code in containers, which are more portable.

Where to look:
https://cloud.google.com/hosting-options

Content Mapping:
- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)

- M6 Applications in the Cloud

Summary:
The three serverless compute options available in Google Cloud are App Engine, Cloud Run, and Cloud Run functions. All of these services abstract the underlying infrastructure so you can focus on code. You only pay for how long your application runs.

This is different than Compute Engine and GKE. In Compute Engine you implement and manage virtual machines that your apps run on. With GKE you implement and manage clusters of compute nodes you deploy your container images to.

App Engine has two environments: standard and flexible. Standard provides a sandbox environment and totally abstracts the infrastructure for you. The flexible environment gives you more choices for deploying your app. It supports more languages, supports different runtimes, and lets you load dependencies you need in the underlying architecture.

Cloud Run, which is also serverless, enables you to run stateless containers via web requests and Google Cloud service events. Cloud Run operates using Knative, an open-source, Kubernetes-based platform. It builds, deploys, and manages modern serverless workloads. Cloud Run gives you the choice of running your containers either fully-managed or in your own GKE cluster.

2.1 Diagnostic Question 03 Discussion

An application running on a highly-customized version of Ubuntu needs to be migrated to Google Cloud. You need to do this in the least amount of time with minimal code changes.

How should you proceed?

A. Create Compute Engine Virtual Machines and migrate the app to that infrastructure.

B. Deploy the existing application to App Engine.

C. Deploy your application in a container image to Cloud Run.

D. Implement a Kubernetes cluster and create pods to enable your app.

Question:
An application running on a highly-customized version of Ubuntu needs to be migrated to Google Cloud. You need to do this in the least amount of time with minimal code changes. How should you proceed?

Proprietary + Confidential

2.1 Diagnostic Question 03 Discussion

An application running on a highly-customized version of Ubuntu needs to be migrated to Google Cloud. You need to do this in the least amount of time with minimal code changes.

How should you proceed?

A. Create Compute Engine Virtual Machines and migrate the app to that infrastructure.

B. Deploy the existing application to App Engine.

C. Deploy your application in a container image to Cloud Run.

D. Implement a Kubernetes cluster and create pods to enable your app.

Google Cloud

Feedback:
*A. Create Compute Engine Virtual Machines and migrate the app to that infrastructure
Feedback: Correct! Compute Engine is a great option for quick migration of traditional apps. You can implement a solution in the cloud without changing your existing code.

B. Deploy the existing application to App Engine.
Feedback: Incorrect. You would need to change your code to run it on App Engine.

C. Deploy your application in a container image to Cloud Run.
Feedback: Incorrect. You would need to re-engineer the current app to work in a container environment.

D. Implement a Kubernetes cluster and create pods to enable your app.
Feedback: Incorrect. You would need to build and manage your Kubernetes cluster, and re-engineer the current app to work in a container environment.

Where to look:
https://cloud.google.com/hosting-options,
https://cloud.google.com/compute/docs/tutorials

Content mapping:
- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)

    - M3 Virtual Machines and Networks in the Cloud

- Architecting with Google Compute Engine (ILT)
    - M3 Virtual Machines

- Essential Google Cloud Infrastructure: Foundation (On-demand)
    - M3 Virtual Machines

Summary:
Compute Engine is Google Cloud's infrastructure-as-a-service offering. It gives you maximum flexibility of developing on a virtual machine (VM). It does require more management than serverless options, though.

A VM has an operating system. You choose how and if it autoscales. Autoscaling can add more machines based on monitored performance thresholds.

A common use case for Compute Engine is migrating an enterprise application designed to run on a server infrastructure. If you set up an architecture similar to your on-premise solution you can port your code quite easily.

To monitor performance you can connect Cloud Logging and Monitoring from Google Cloud Observability.

2.1 Diagnostic Question 04 Discussion

You want to deploy a microservices application. You need full control of how you manage containers, reliability, and autoscaling, but don't want or need to manage the control plane.

A. Cloud Run
B. App Engine
C. Google Kubernetes Engine
D. Compute Engine

Which compute option should you use?

Question:
You want to deploy a microservices application. You need full control of how you manage containers, reliability, and autoscaling, but don't want or need to manage the control plane. Which compute option should you use?

2.1 Diagnostic Question 04 Discussion

You want to deploy a microservices application. You need full control of how you manage containers, reliability, and autoscaling, but don't want or need to manage the control plane.

Which compute option should you use?

A. Cloud Run
B. App Engine
C. Google Kubernetes Engine
D. Compute Engine

Google Cloud

Feedback:
A. Cloud Run
Feedback: Incorrect. Cloud Run does not give you full control over your containers.

B. App Engine
Feedback: Incorrect. App Engine does not give you full control over your containers.

*C. Google Kubernetes Engine
Feedback: Correct! Google Kubernetes Engine gives you full control of container orchestration and availability.

D. Compute Engine
Feedback: Incorrect. Deploying in Compute Engine would require you to load and manage your own container management software.

Where to look:
https://cloud.google.com/docs/choosing-a-compute-option

Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
  - M5 Containers in the Cloud

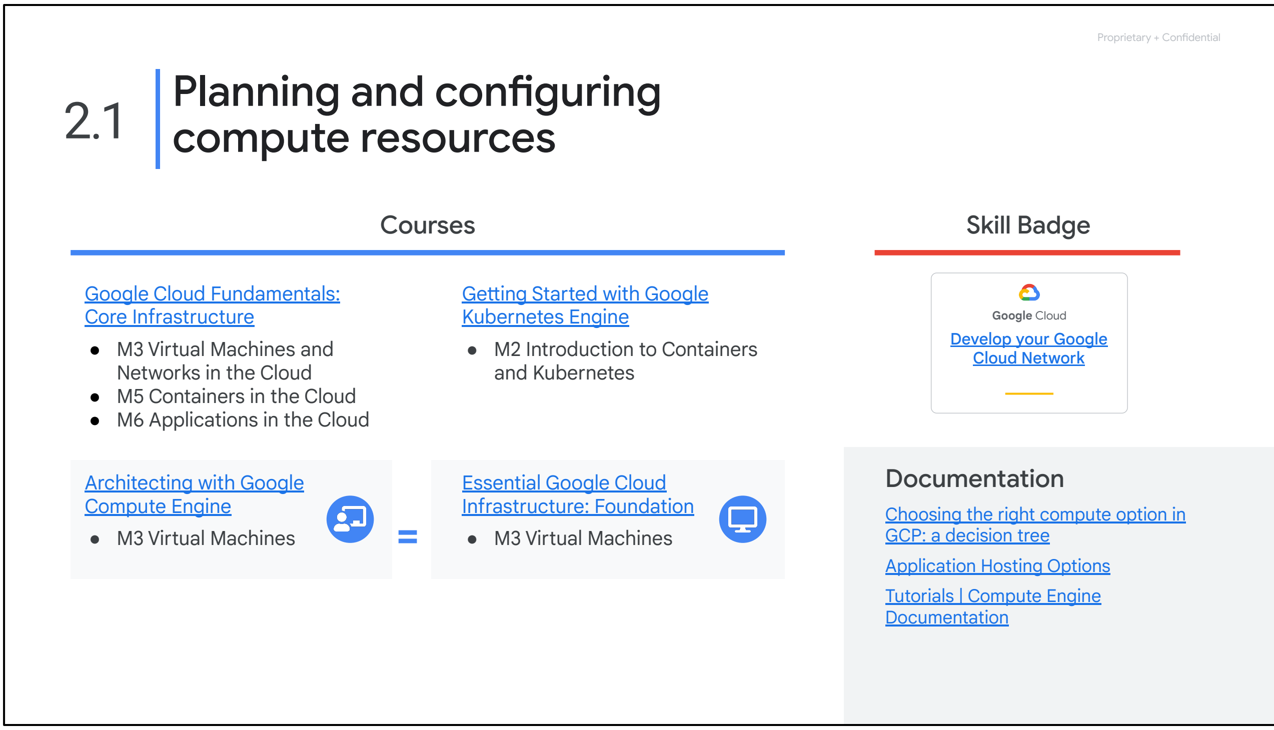
- Getting Started with GKE (ILT and On-demand)

    - M2 Introduction to Containers and Kubernetes

- Skill Badge
    - Develop your Google Cloud Network (https://www.cloudskillsboost.google/course_templates/625)

Summary:
GKE is a platform-as-a-service offering for running containerized applications in the cloud. Google manages the control plane for you, under your administrative control. Containers abstract application dependencies from the host operating system. This makes container architectures highly portable. It saves costs compared to implementing multiple VMs on a host hypervisor, which each requiring a copy of the operating system. Kubernetes lets you orchestrate code in containers.

If you have containerized applications that use a native Kubernetes architecture in your on-premise environment, it can be straightforward to migrate to Google Cloud.



Let's take a moment to consider resources that can help you build your knowledge and skills in this area.

The concepts in the diagnostic questions we just reviewed are covered in these modules, skill badge, and documentation. You'll find this list in your workbook so you can take a note of what you want to include later when you build your study plan. Based on your experience with the diagnostic questions, you may want to include some or all of these in your plan.

Google Cloud Fundamentals: Core Infrastructure (On-demand)
Getting Started with Google Kubernetes Engine (On-demand)
Architecting with Google Compute Engine (ILT)
Essential Google Cloud Infrastructure: Foundation (On-demand)

Develop your Google Cloud Network (Skill Badge)

https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree
https://cloud.google.com/hosting-options
https://cloud.google.com/compute/docs/tutorials

Proprietary + Confidential

2.2 Planning and configuring data storage options

Considerations include:

- Product choice
  (e.g., Cloud SQL, BigQuery, Firestore, Spanner, Bigtable)
- Choosing storage options
  (e.g., zonal Persistent Disk, regional Persistent Disk, Standard, Nearline, Coldline, Archive)

Google Cloud

Along with compute resources, an Associate Cloud Engineer needs to be able to plan and configure data storage in Google Cloud. That requires understanding of the different product choices and storage options, and how to make recommended choices for a given use case.

You encountered these types of considerations in the following questions:
Question 5: Associate Google Cloud data services with modern data constructs, e.g. relational, non-relational or no-sql, object or warehouse.
Question 6: List the storage classes available for cloud storage including regional, multiregional, near and cold line.
Question 7: Differentiate between relational databases and data warehouses, e.g. relational databases can be transactional, while data warehouses are analytical and based on historical data.

Proprietary + Confidential

2.2 Diagnostic Question 05 Discussion

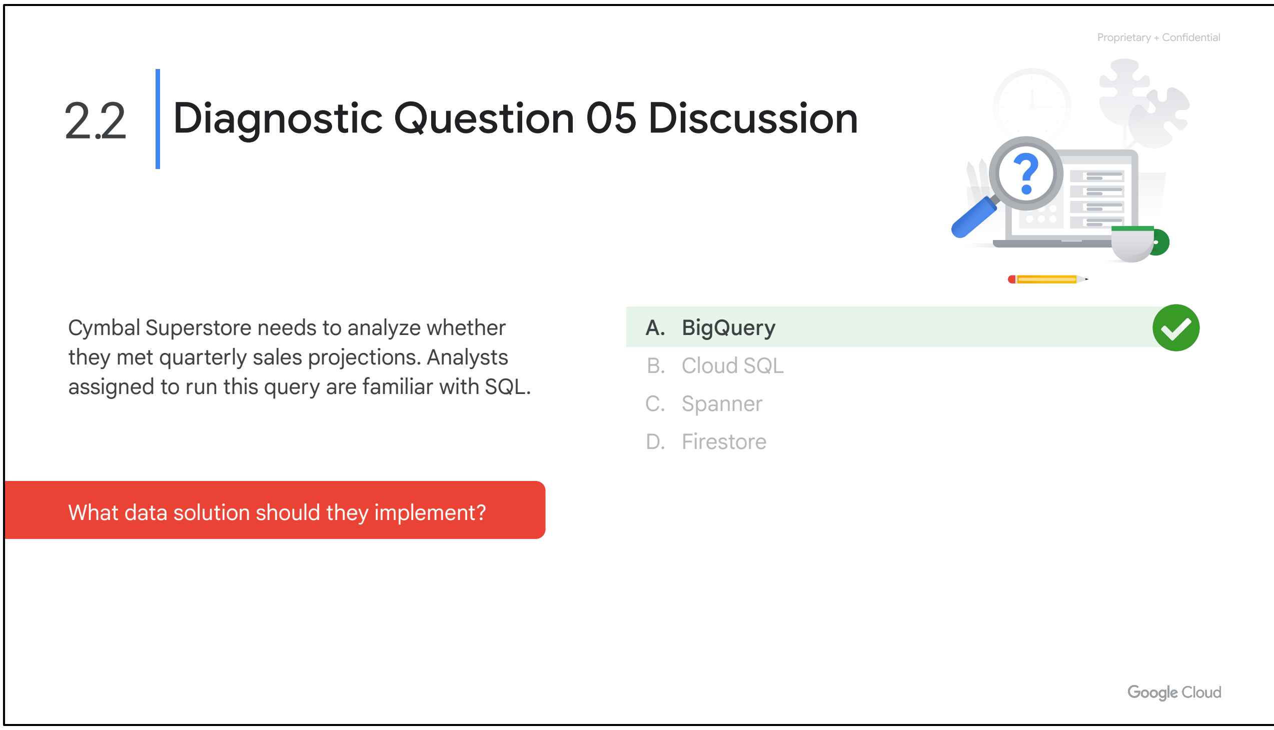
Cymbal Superstore needs to analyze whether they met quarterly sales projections. Analysts assigned to run this query are familiar with SQL.

A. BigQuery
B. Cloud SQL
C. Spanner
D. Firestore

What data solution should they implement?

Google Cloud

Question:
Cymbal Superstore needs to analyze whether they met quarterly sales projections. Analysts assigned to run this query are familiar with SQL. What data solution should they implement?



Feedback:
*A. BigQuery
Feedback: Correct! BigQuery is Google Cloud’s implementation of a modern data warehouse. BigQuery analyzes historical data and uses a SQL query engine.

B. Cloud SQL
Feedback: Incorrect. Cloud SQL is optimized for transactional reads and writes. It is not a good candidate for querying historical data as described in the scenario.

C. Spanner
Feedback: Incorrect. Spanner is an SQL-compatible relational database, but it is not built for analyzing historical data.

D. Firestore
Feedback: Incorrect. Firestore is a NoSQL document database used to define entities with attributes. It is not a good choice for the analysis of historical data as described in the scenario.

Where to look:
https://cloud.google.com/storage-options/

Content mapping:
- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)

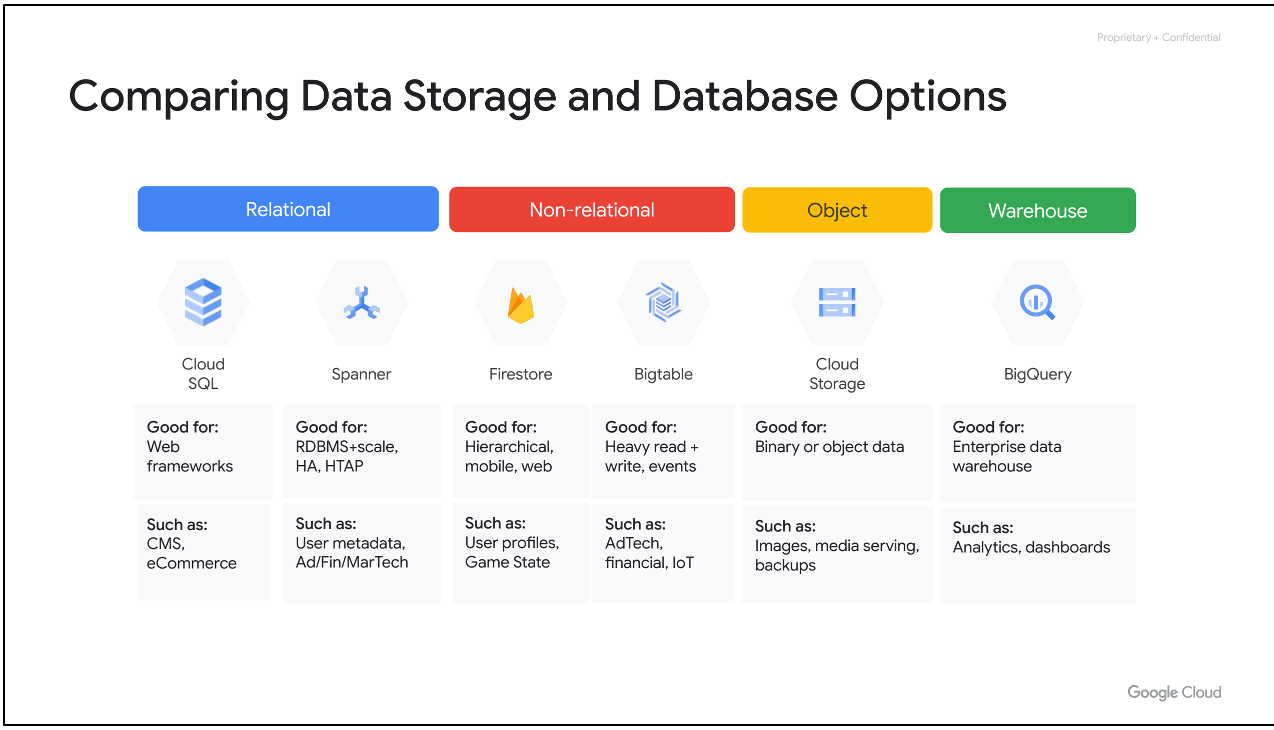
    - M4 Storage in the Cloud

- Architecting with Google Compute Engine (ILT)
    - M5 Storage and Database Services

- Essential Google Cloud Infrastructure: Core Services (On-demand)
    - M2 Storage and Database Services

- Skill Badge
    - Set Up an App Dev Environment on Google Cloud (https://www.cloudskillsboost.google/course_templates/637)

Summary:
Explanation/summary on the following slide.



This question relates to Google Cloud data services and what data construct they are based on.

Firestore and Bigtable are NoSQL implementations. Firestore is a document database that supports entities and attributes. Bigtable is based on column families where rows of data are referenced by a key that combines commonly queried columns. Related columns can additionally be organized into column families such as username and address.

Cloud Storage is Google Cloud’s recommended object storage service. Think of pictures and videos, as well as file objects with an implicit schema, such as logs and csv files.

Google’s relational database offerings include Cloud SQL and Spanner. Use them when you need a transactional processing system you can query with SQL. Cloud SQL is a managed version of databases you can implement on-premises, while Spanner is horizontally scalable and globally available.

BigQuery is a serverless distributed query engine that is primarily used as a modern data warehouse. It does have a native storage format but can also query external data where it resides. You interact with it by using a form of SQL. Keep in mind its native storage format is not a good solution for a backend store for an application. It does, however, improve performance of analytical queries you run against it using the query engine.

2.2 Diagnostic Question 06 Discussion

Cymbal Superstore’s supply chain application frequently analyzes large amounts of data to inform business processes and operational dashboards.

What storage class would make sense for this use case?

A. Archive
B. Coldline
C. Nearline
D. Standard

Question:
Cymbal Superstore’s supply chain application frequently analyzes large amounts of data to inform business processes and operational dashboards. What storage class would make sense for this use case?

2.2 Diagnostic Question 06 Discussion

Cymbal Superstore's supply chain application frequently analyzes large amounts of data to inform business processes and operational dashboards.

What storage class would make sense for this use case?

A. Archive
B. Coldline
C. Nearline
D. Standard

Feedback:
A. Archive
Feedback: Incorrect. Archive storage is the best choice for data that you plan to access less than once a year.

B. Coldline
Feedback: Incorrect. Dashboards need current data to analyze. Coldline is good for storing data accessed only every 90 days.

C. Nearline
Feedback: Incorrect. Dashboards need current data to analyze. Nearline is good for storing data accessed only every 30 days.

*D. Standard. Correct. Standard storage is best for data that is frequently accessed ("hot" data) and/or stored for only brief periods of time. In addition, co-locating your resources by selecting the regional option maximizes the performance for data-intensive computations and can reduce network charges.

Where to look:
https://cloud.google.com/storage/docs/storage-classes

Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
    - M4 Storage in the Cloud

- Architecting with Google Compute Engine (ILT)
    - M5 Storage and Database Services

- Essential Google Cloud Infrastructure: Core Services (On-demand)
    - M2 Storage and Database Services

- Skill Badge
    - Set Up an App Dev Environment on Google Cloud (https://www.cloudskillsboost.google/course_templates/637)

Summary:
Explanation/summary on the following slide.

Storage Classes and use cases summary

	Standard	Nearline	Coldline	Archive
Use when you need...	• No retrieval cost • No minimum storage duration	• Very low cost per GB stored and can accept higher per-operation costs • 30-day minimum storage duration	• Even lower cost per GB stored and can accept higher per-operation costs • 90-day minimum storage duration	• Lowest cost per GB stored and can accept the highest per-operation costs • 365-day minimum storage duration
Typical use cases	“Hot” data and/or stored for only brief periods of time like data-intensive computations	Infrequently (i.e., no more than once per month) accessed data. Ideal for back-up and serving long-tail multimedia content.	Very infrequently accessed data - ie, once a year. Typically this is for disaster recovery, or for financial data that has to be kept for a certain length of time to meet regulatory needs.	Data archiving, online backup, and disaster recovery

Data location and storage class affect the availability and cost of storing your data in Cloud Storage. You can choose regional, dual-region, and multi-regional location options. Storage classes include Standard, Nearline, Coldline and Archive storage. The different storage classes determine pricing based on how long your data is stored and how often you access it.

Standard storage is the default storage class. Data stored using this class is immediately available. It is the recommended storage class for frequently accessed data. You should locate your data in the same region as the services you are going to use to ingest and analyze the data to reduce latency as much as possible. Specifying a dual-region location that includes the region where your application resides will still give you low latency, but your data will also be available in another region in case of an outage. Extending your storage settings to a multi-region will make data available over a large geographic area such as US, Europe, or Asia.

The other storage classes implement ways to store infrequently accessed data. Nearline storage is for data that is only accessed around every 30 days. Coldline storage is for data that is only accessed around once every quarter, or 90 days. Archive storage is long-term storage for data accessed only once a year. These storage classes have optimized pricing, but also expect you to keep your data in them for the minimum limits specified above. If you access your data before the minimum amount of time you will be charged a data access fee.

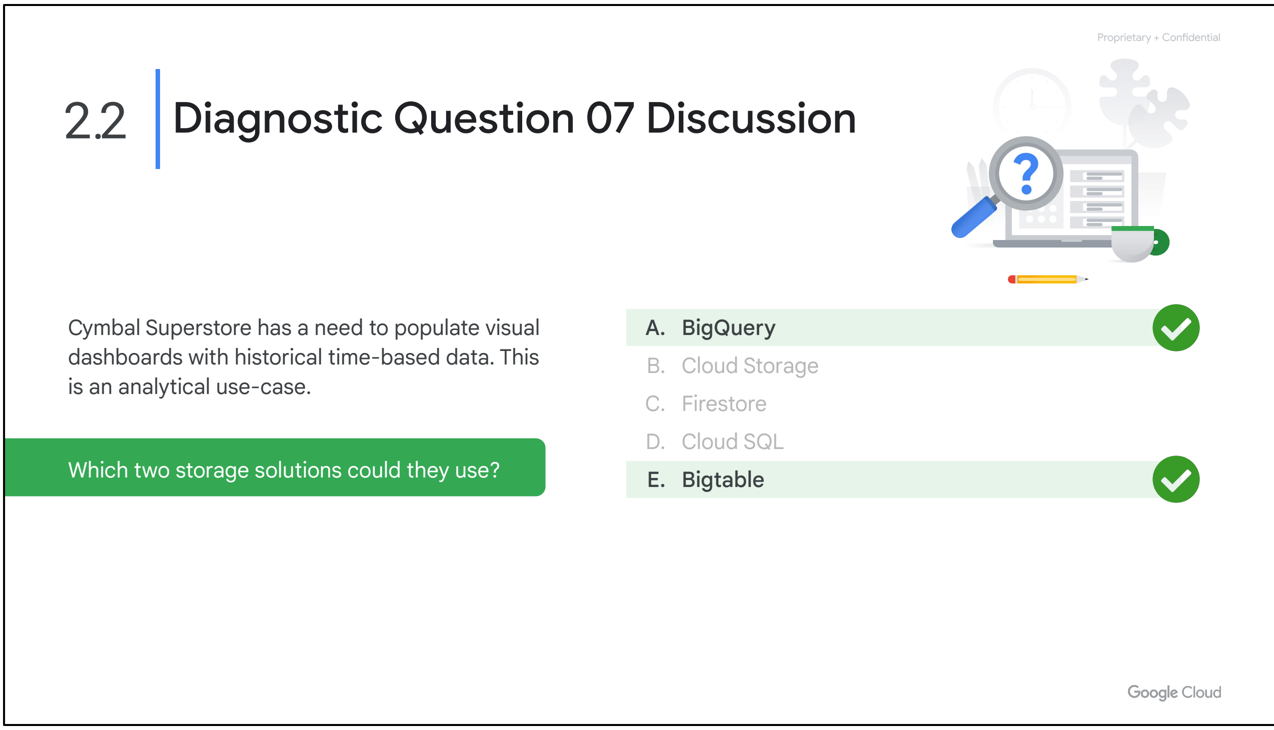
2.2 Diagnostic Question 07 Discussion

Cymbal Superstore has a need to populate visual dashboards with historical time-based data. This is an analytical use-case.

Which two storage solutions could they use?

A. BigQuery
B. Cloud Storage
C. Firestore
D. Cloud SQL
E. Bigtable

Question:
Cymbal Superstore has a need to populate visual dashboards with historical time-based data. This is an analytical use-case. Which two storage solutions could they use?



Feedback:

*A. BigQuery

Feedback: Correct! BigQuery is a data warehouse offering optimized to query historical time-based data. BigQuery can run queries against data in its own column-based store or run federated queries against data from other data services and file stores.

B. Cloud Storage

Feedback: Incorrect. Cloud Storage is a large object store and is not queryable. It is not transactional or analytical.

C. Firestore

Feedback: Incorrect. Firestore is a transactional NoSQL store where you define attribute key-value pairs describing an entity.

D. Cloud SQL

Feedback: Incorrect. Cloud SQL is a transactional relational database optimized for both reads and writes used in an operational context, but not for analyzing historical data.

*E. Bigtable

Feedback: Correct! Bigtable is a petabyte scale, NoSQL, column family database with row keys optimized for specific queries. It is used to store historic, time-based data and answers the need for this requirement.

Where to look:
https://cloud.google.com/load-balancing

Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
  - M4 Storage in the Cloud

- Architecting with Google Compute Engine (ILT)
  - M5 Storage and Database Services

- Essential Google Cloud Infrastructure: Core Services (On-demand)
  - M2 Storage and Database Services

Summary:
Explanation/summary on the following slide.

Comparing storage options: use cases

	Firestore	Bigtable	Cloud Storage	Cloud SQL	Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Best for	Storing, syncing, and querying data	"Flat" data, Heavy read/write, events, analytical data	Structured and unstructured binary or object data	Web frameworks, existing applications	Large-scale database applications (> ~2 TB)	Interactive querying, offline analytics
Use cases	Mobile, web, and server development	AdTech, Financial and IoT data	Images, large media files, backups	User credentials, customer orders	Whenever high I/O, global consistency is needed	Data warehousing

Two common workloads required in a modern business environment are transactional workloads and analytical workloads.

Transactional workloads are optimized for more writes and updates than reads. Transactional means either all parts of an update happen or none of them do. For example, think of the importance of making sure deposits and withdrawals are recorded in a financial system. Both of these are part of one transaction.

Relational database management systems are commonly used for applications that are transactional in nature. Relational database services used to support transactional systems in Google Cloud include Cloud SQL and Spanner.

Cloud SQL is a managed database service that gives you access to common database types you might implement in your own infrastructure, like MySql or PostGre. It is implemented on virtual machines in the cloud with different options for size and availability.

Spanner shards your database across a cluster of database nodes, offering strong consistency and global availability. It is fully managed service, so you don't need to worry about underlying virtual machines.

The other type of workload is analytical. It is based on querying historical data that doesn't change often, and is optimized for writes. BigQuery is a good option for this kind of workload.

2.2 Planning and configuring data storage options

Courses

[Google Cloud Fundamentals: Core Infrastructure]
- M4 Storage in the Cloud

[Architecting with Google Compute Engine]
- M5 Storage and Database Services

=

[Essential Google Cloud Infrastructure: Core Services]
- M2 Storage and Database Services

Skill Badge

Google Cloud
Set Up an App Dev Environment on Google Cloud

Documentation

Cloud Storage Options
Storage classes
Data lifecycle | Cloud Architecture Center

Let's take a moment to consider resources that can help you build your knowledge and skills in this area.

The concepts in the diagnostic questions we just reviewed are covered in these modules, skill badge, and documentation. You'll find this list in your workbook so you can take a note of what you want to include later when you build your study plan. Based on your experience with the diagnostic questions, you may want to include some or all of these in your plan.

Google Cloud Fundamentals: Core Infrastructure (On-demand)
Architecting with Google Compute Engine (ILT)
Essential Google Cloud Infrastructure: Core Services (On-demand)

Set Up an App Dev Environment on Google Cloud (Skill Badge)

https://cloud.google.com/storage-options/
https://cloud.google.com/storage/docs/storage-classes
https://cloud.google.com/architecture/data-lifecycle-cloud-platform

Proprietary + Confidential

2.3 Planning and configuring network resources

Considerations include:

- Load balancing
- Availability of resource locations in a network
- Network Service Tiers

Google Cloud

Together with compute and storage decisions, an associate cloud architect should be able to plan and configure network resources in Google Cloud - including load balancing, resource locations, and Network Service Tiers.

You explored these types of tasks in questions 8 and 9: Match Google Cloud load balancing options to the appropriate TCP layer: Global https: Layer 7, Global SSL: Layer 4: Global TCP Proxy: Layer 4.

2.3 Diagnostic Question 08 Discussion

Cymbal Superstore is piloting an update to its ecommerce app for the flagship store in Minneapolis, Minnesota. The app is implemented as a three-tier web service with traffic originating from the local area and resources dedicated for it in us-central1. You need to configure a secure, low-cost network load-balancing architecture for it.

How do you proceed?

A. Implement a premium tier global external Application Load Balancer connected to the web tier as the frontend, and a regional internal Application Load Balancer between the web tier and backend.

B. Implement a global external proxy Network Load Balancer connected to the web tier as the frontend, and a premium tier passthrough Network Load Balancer between the web tier and the backend.

C. Configure a standard tier regional external Application Load Balancer connected to the web tier as a frontend and a regional internal Application Load Balancer between the web tier and the backend.

D. Configure a regional internal proxy Network Load Balancer connected to the web tier as the frontend and a standard tier internal proxy Network Load Balancer between the web tier and the backend.

Question:
Cymbal Superstore is piloting an update to its ecommerce app for the flagship store in Minneapolis, Minnesota. The app is implemented as a three-tier web service with traffic originating from the local area and resources dedicated for it in us-central1. You need to configure a secure, low-cost network load-balancing architecture for it. How do you proceed?

2.4 Diagnostic Question 09 Discussion

Cymbal Superstore is piloting an update to its ecommerce app for the flagship store in Minneapolis, Minnesota. The app is implemented as a three-tier web service with traffic originating from the local area and resources dedicated for it in us-central1. You need to configure a secure, low-cost network load-balancing architecture for it.

How do you proceed?

A. Implement a premium tier global external Application Load Balancer connected to the web tier as the frontend, and a regional internal Application Load Balancer between the web tier and backend.

B. Implement a global external proxy Network Load Balancer connected to the web tier as the frontend, and a premium tier passthrough Network Load Balancer between the web tier and the backend.

C. Configure a standard tier regional external Application Load Balancer connected to the web tier as a frontend and a regional internal Application Load Balancer between the web tier and the backend.

D. Configure a regional internal proxy Network Load Balancer connected to the web tier as the frontend and a standard tier internal proxy Network Load Balancer between the web tier and the backend.

Google Cloud

Feedback:

A. Incorrect. A premium global external Application Load Balancer is more expensive, and the scenario calls for a low cost solution. Also, all the resources for the scenario are in the same region.

B. Incorrect. TCP protocol lacks the security features of HTTP(S), making it less suitable for an ecommerce application. A premium tier passthrough Network Load Balancer is global and is not the proper solution between web and backend within a region.

C. Correct! All of the resources are in the same region, so this is the most cost effective way to ensure secure communication between the Application Load Balancer and the backed instances.

D. Incorrect. A Network Load Balancer is not a good solution for web front ends. For a web frontend, you should use an Application Load Balancer (layer 7) whenever possible.

Where to look:
https://cloud.google.com/load-balancing/docs/load-balancing-overview

Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
    - M3 Virtual Machines and Networks in the Cloud

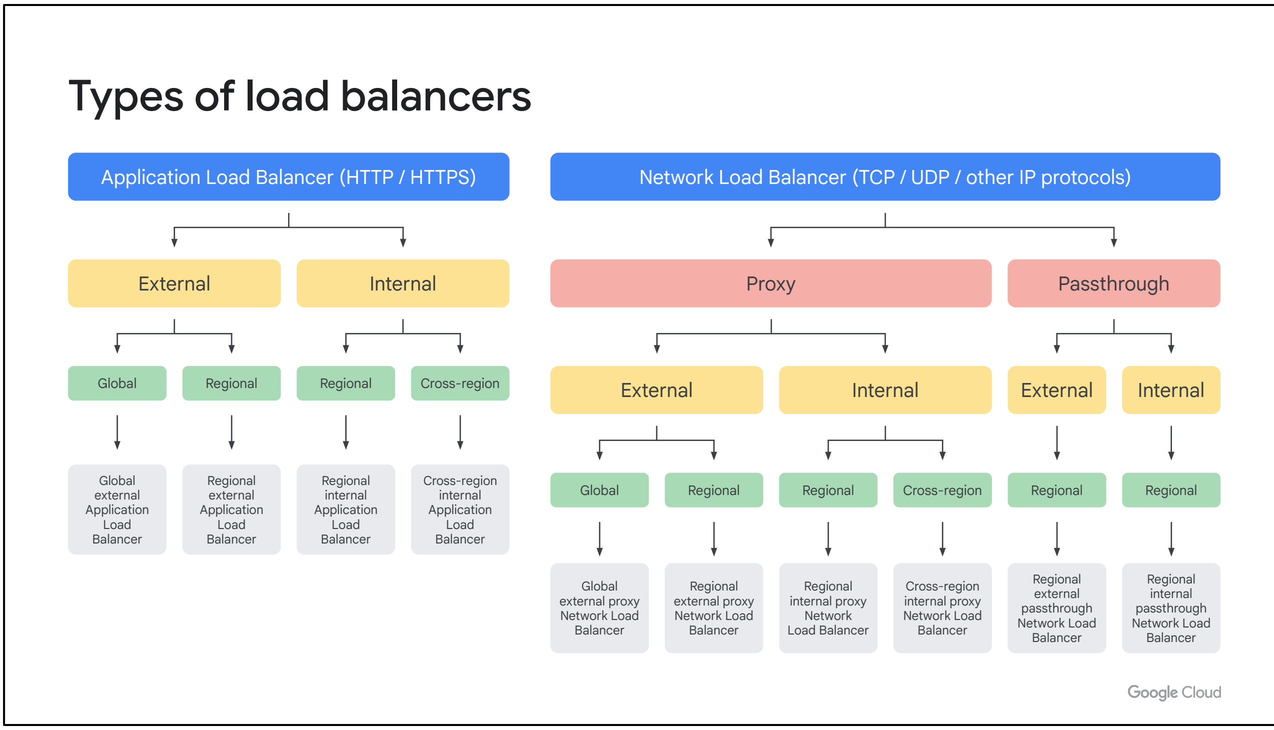
- Architecting with Google Compute Engine (ILT)

    - M2 Virtual Networks
    - M9 Load Balancing and Autoscaling

- Essential Google Cloud Infrastructure: Foundations (On-demand)
    - M2 Virtual Networks

- Essential Google Cloud Infrastructure: Scaling and Automation (On-demand)
    - M2 Load Balancing and Autoscaling

Summary:
Explanation/summary on the following slide.



Google Cloud offers a range of load balancing solutions that can be classified based on the OSI model layer they operate at and their specific functionalities.

Application Load Balancers

These load balancers operate at the application layer and are designed to handle HTTP and HTTPS traffic, making them ideal for web applications and services that require advanced features like content-based routing and SSL/TLS termination. Application Load Balancers operate as reverse proxies, distributing incoming traffic across multiple backend instances based on rules you define. They are highly flexible and can be configured for both internet-facing (external) and internal applications.

Network Load Balancers

Network Load Balancers operate at the transport layer and efficiently handle TCP, UDP, and other IP protocols. They can be further classified into two types:

Proxy Load Balancers: These also function as reverse proxies, terminating client connections and establishing new ones to backend services. They offer advanced traffic management capabilities and support backends located both on-premises and in various cloud environments.

Passthrough Load Balancers: Unlike proxy load balancers, these do not modify or terminate connections. Instead, they directly forward traffic to the backend while preserving the original source IP address. This type is well-suited for applications that require direct server return or need to handle a wider range of IP protocols.

2.3 Diagnostic Question 09 Discussion

What Google Cloud load balancing option runs at Layer 7 of the TCP stack?

A. Global Application Load Balancer
B. Global proxy Network Load Balancer
C. Regional passthrough Network Load Balancer
D. Regional internal proxy Network Load Balancer

Google Cloud

Question:
What Google Cloud load balancing option runs at Layer 7 of the TCP stack?

Proprietary + Confidential

2.3 Diagnostic Question 09 Discussion

What Google Cloud load balancing option runs at Layer 7 of the TCP stack?

A. Global Application Load Balancer
B. Global proxy Network Load Balancer
C. Regional passthrough Network Load Balancer
D. Regional internal proxy Network Load Balancer

Google Cloud

Feedback:

A. Correct! A Global Application Load Balancer operates at layer 7 of the OSI model. This means it can inspect and route traffic based on application-layer information, such as HTTP headers, URL paths, and cookies.
B. Incorrect. Global Proxy Network Load Balancers are layer 4 reverse proxy load balancers that distribute TCP traffic to backends in your Google Cloud Virtual Private Cloud (VPC) network or in other cloud environments.
C. Incorrect. Regional passthrough Network Load Balancers are Layer 4 regional, passthrough load balancers. These load balancers distribute traffic among backends in the same region as the load balancer.
D. Incorrect. Regional Internal passthrough Network Load Balancers are layer 4 load balancers and distribute traffic among internal virtual machine (VM) instances in the same region in a Virtual Private Cloud (VPC) network.

Where to look:
https://cloud.google.com/load-balancing/docs/load-balancing-overview

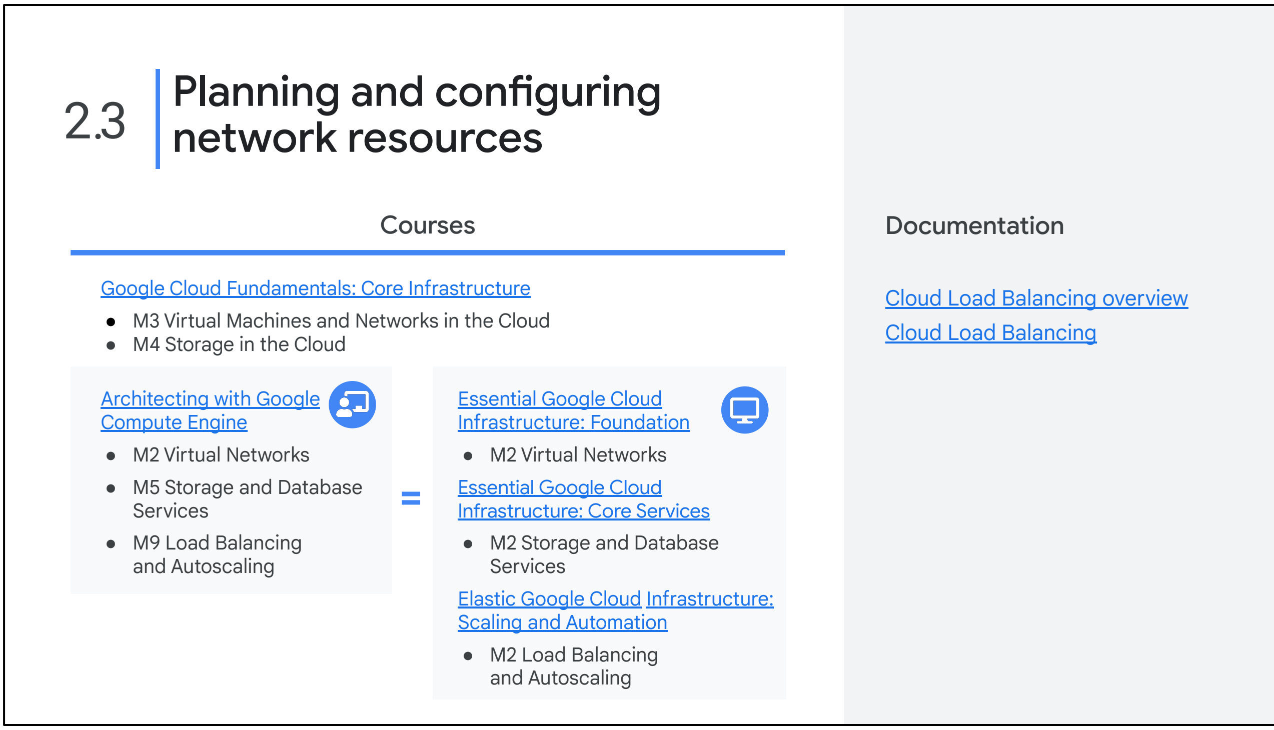
Content mapping:

- Google Cloud Fundamentals: Core Infrastructure (ILT and On-demand)
  - M4 Storage in the Cloud

- Architecting with Google Compute Engine (ILT)
  - M5 Storage and Database Services

- Essential Google Cloud Infrastructure: Core Services (On-demand)
    - M2 Storage and Database Services

Summary:
The Application Load Balancer is a proxy-based Layer 7 load balancer that lets you run and scale your services. The Application Load Balancer distributes HTTP and HTTPS traffic to backends hosted on a variety of Google Cloud platforms—such as Compute Engine, Google Kubernetes Engine (GKE), Cloud Storage, and Cloud Run—as well as external backends connected over the internet or by using hybrid connectivity.



Let’s take a moment to consider resources that can help you build your knowledge and skills in this area.

The concepts in the diagnostic questions we just reviewed are covered in these modules and in this documentation. You’ll find this list in your workbook so you can take a note of what you want to include later when you build your study plan. Based on your experience with the diagnostic questions, you may want to include some or all of these in your plan.

Google Cloud Fundamentals: Core Infrastructure (On-demand)
Architecting with Google Compute Engine (ILT)
Essential Google Cloud Infrastructure: Foundation (On-demand)
Essential Google Cloud Infrastructure: Core Services (On-demand)
Elastic Google Cloud Infrastructure: Scaling and Automation (On-demand)

https://cloud.google.com/load-balancing/docs/load-balancing-overview
https://cloud.google.com/load-balancing