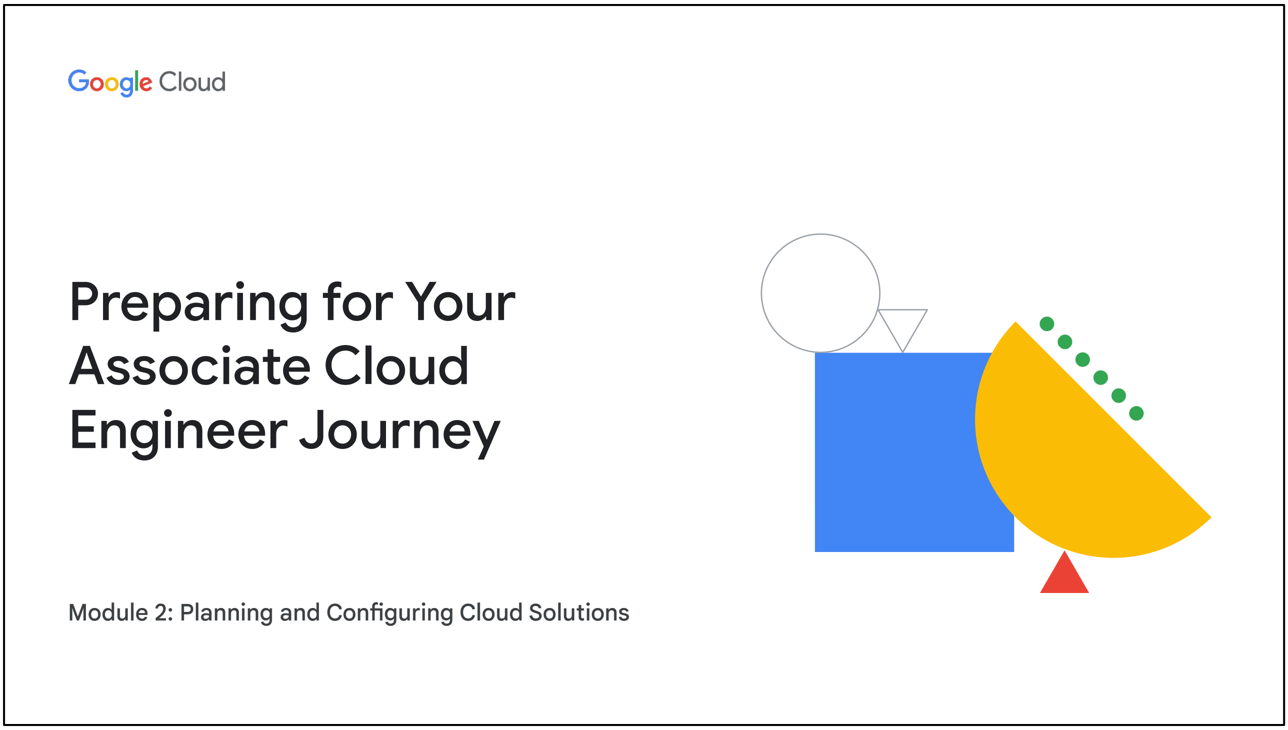


Google Cloud

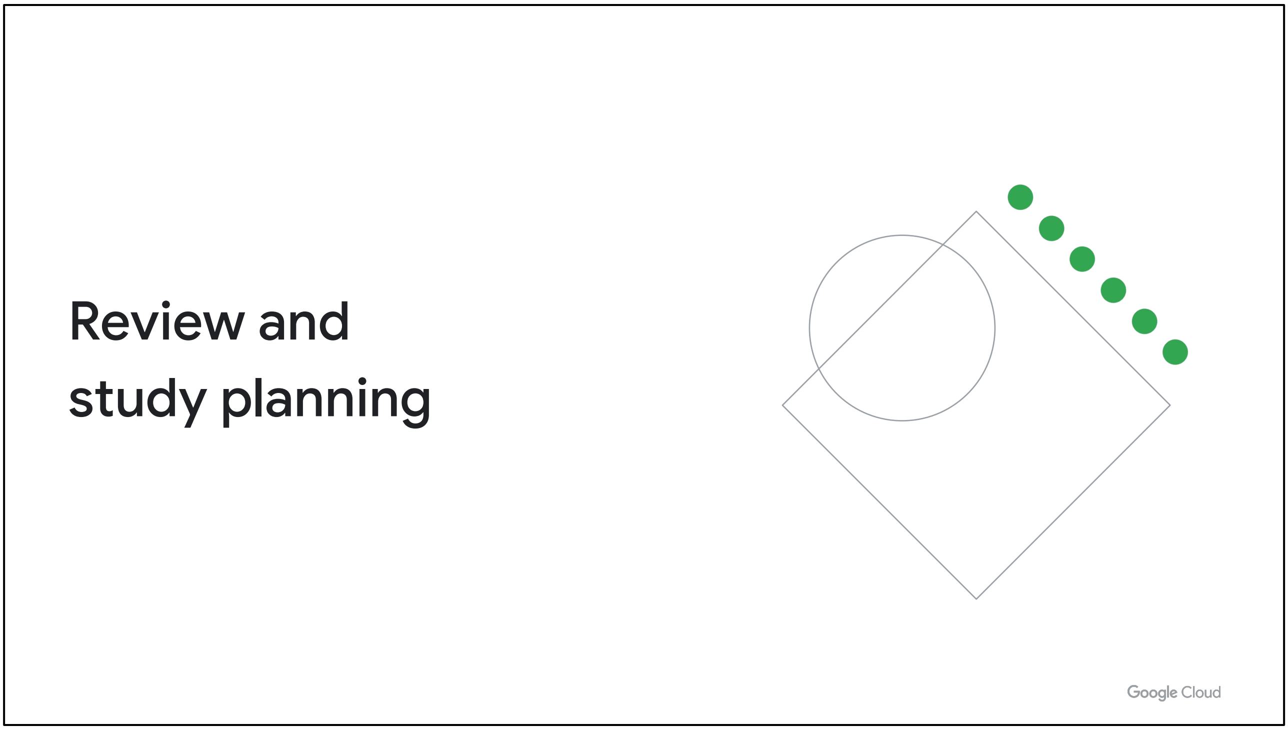
Preparing for Your Associate Cloud Engineer Journey

Module 2: Planning and Configuring Cloud Solutions

Welcome to Module 2: Planning and Configuring Cloud Solutions.

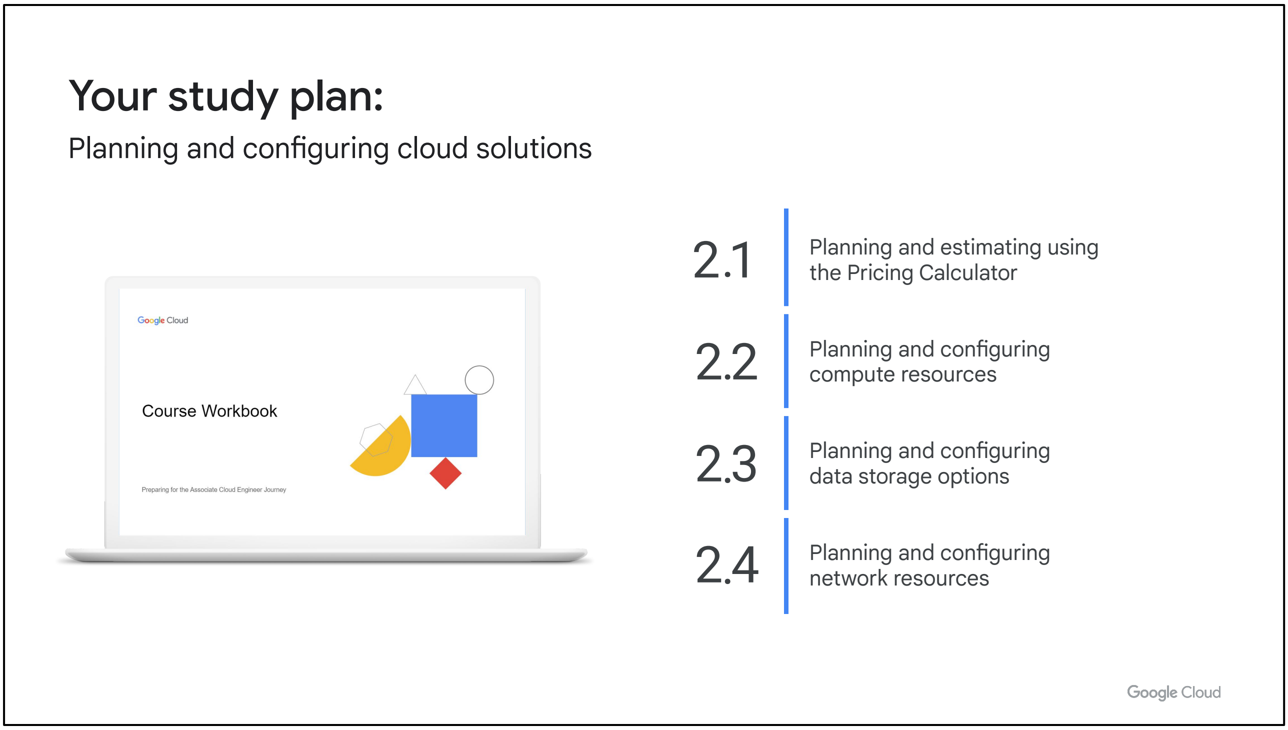
Please complete the diagnostic questions now

- Forms are provided for you to answer the diagnostic questions
- The instructor will provide you a link to the forms
- The diagnostic questions are also available in the workbook



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

Google Cloud
Course Workbook
Preparing for the Associate Cloud Engineer Journey

2.1 Planning and estimating using the Pricing Calculator

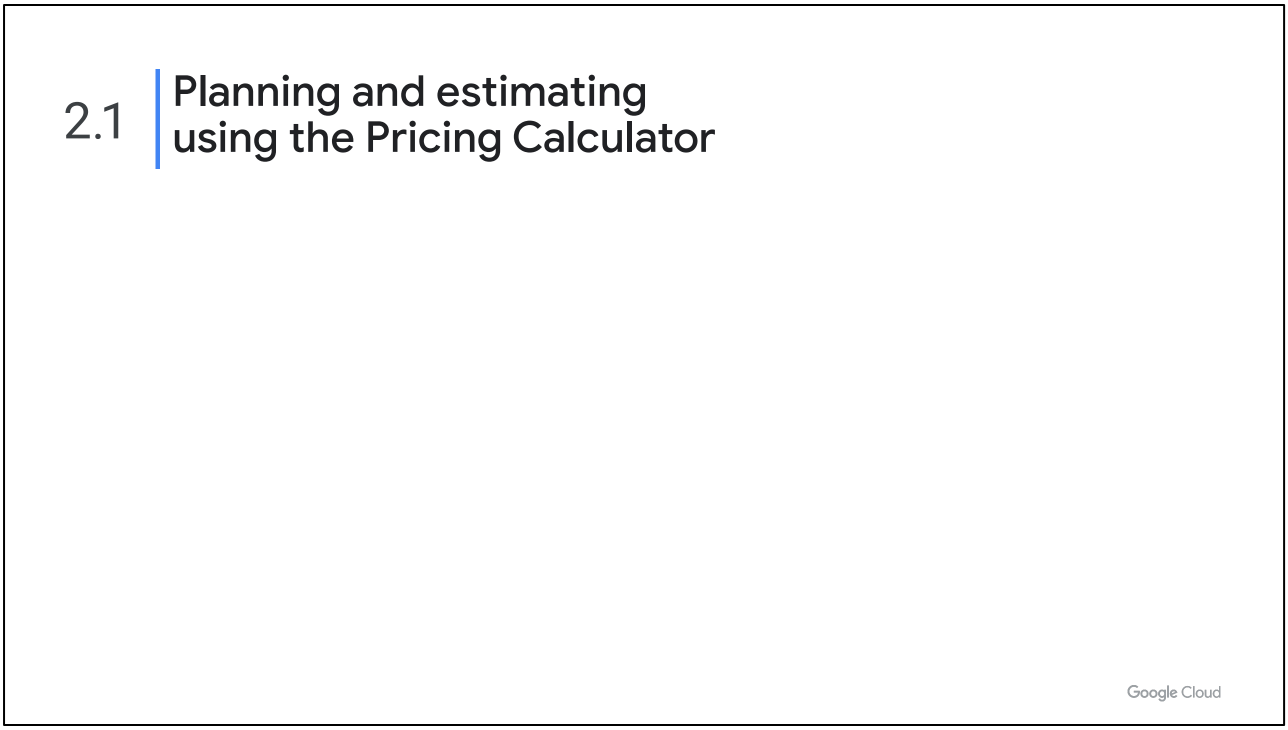
2.2 Planning and configuring compute resources

2.3 Planning and configuring data storage options

2.4 Planning and configuring network resources

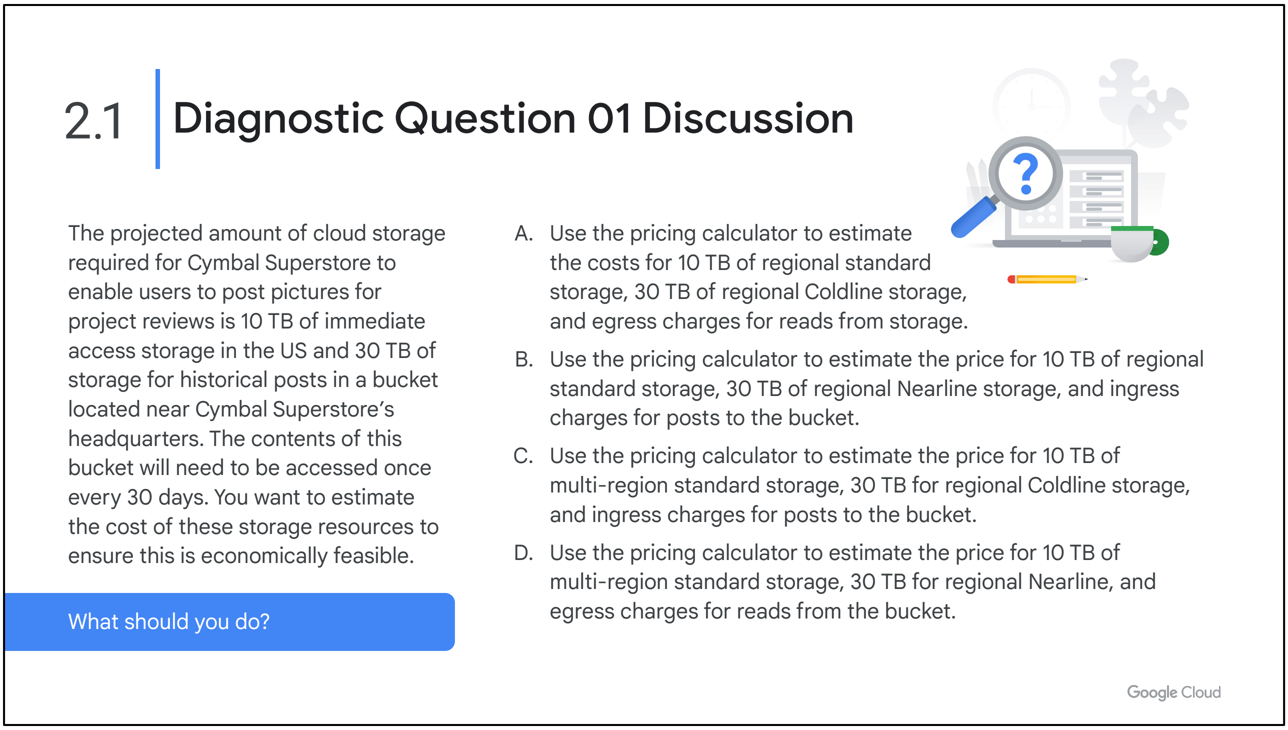
Google Cloud

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 quests you'll want to emphasize in your study plan.



2.1 Planning and estimating using the Pricing Calculator

At the start of this module we discussed how an Associate Cloud Engineer would plan and configure Cymbal Superstore’s cloud applications and some of the considerations involved. Google’s pricing calculator can help you estimate costs and make decisions about compute and storage choices, as you saw in the first diagnostic question.



2.1 Diagnostic Question 01 Discussion

The projected amount of cloud storage required for Cymbal Superstore to enable users to post pictures for project reviews is 10 TB of immediate access storage in the US and 30 TB of storage for historical posts in a bucket located near Cymbal Superstore's headquarters. The contents of this bucket will need to be accessed once every 30 days. You want to estimate the cost of these storage resources to ensure this is economically feasible.

What should you do?

A. Use the pricing calculator to estimate the costs for 10 TB of regional standard storage, 30 TB of regional Coldline storage, and egress charges for reads from storage.

B. Use the pricing calculator to estimate the price for 10 TB of regional standard storage, 30 TB of regional Nearline storage, and ingress charges for posts to the bucket.

C. Use the pricing calculator to estimate the price for 10 TB of multi-region standard storage, 30 TB for regional Coldline storage, and ingress charges for posts to the bucket.

D. Use the pricing calculator to estimate the price for 10 TB of multi-region standard storage, 30 TB for regional Nearline, and egress charges for reads from the bucket.

Question:
The projected amount of cloud storage required for Cymbal Superstore to enable users to post pictures for project reviews is 10 TB of immediate access storage in the US and 30 TB of storage for historical posts in a bucket located near Cymbal Superstore's headquarters. The contents of this bucket will need to be accessed once every 30 days. You want to estimate the cost of these storage resources to ensure this is economically feasible. What should you do?
A. Use the pricing calculator to estimate the costs for 10 TB of regional standard storage, 30 TB of regional Coldline storage, and egress charges for reads from storage.
Feedback: Incorrect. The storage is US which indicates multi-region storage instead of regional standard storage. The 30-day requirement points to Nearline storage, not Coldline.

B. Use the pricing calculator to estimate the price for 10 TB of regional standard storage, 30 TB of regional Nearline storage, and ingress charges for posts to the bucket.
Feedback: Incorrect. The storage is US which indicates multi-region storage instead of regional standard storage and ingress (data writes) is free. There are no costs associated with ingress.

C. Use the pricing calculator to estimate the price for 10 TB of multi-region standard storage, 30 TB for regional Coldline storage, and ingress charges for posts to the bucket.
Feedback: Incorrect. The 30-day requirement points to Nearline storage, not Coldline and ingress (data writes) is free, there are no costs associated with ingress.

*D. Use the pricing calculator to estimate the price for 10 TB of multi-region standard storage, 30 TB for regional Nearline, and egress charges for reads from the bucket.
Feedback: Correct! Data storage pricing is based on the amount of data and storage type. Standard storage is immediately available. Nearline storage is for data accessed roughly every 30 days. Egress is the amount of data read from the bucket and is also chargeable.

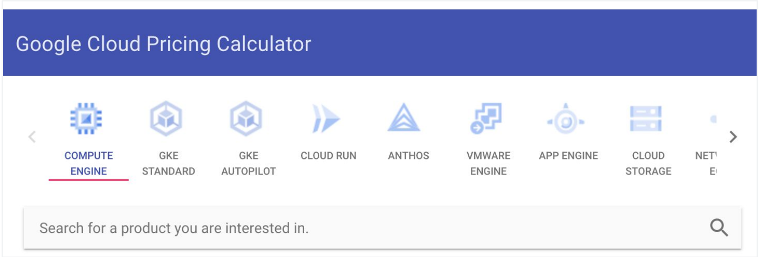
Where to look: https://cloud.google.com/products/calculator/

Content mapping:

- Instructor-led Training/OnDemand
    - Essential Google Cloud Infrastructure
        - M3 Virtual Machines
    - Architecting with Google Compute Engine
        - M6 Resource Management

Summary:
Explanation/summary on the following slides.

Pricing Calculator



https://cloud.google.com/products/calculator/

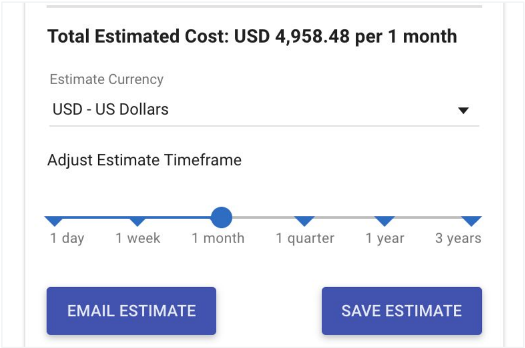
1. Select a product from scrolling list at top of the form
2. Form for each product will show expense variables
3. Enter target configuration in form
4. Submit each section to add to your overall estimate

https://cloud.google.com/products/calculator/

The Pricing Calculator is a multi-section form that lets you estimate the costs of different cloud services based on how you are going to use and configure them. For example, you can estimate costs of implementing a database in Cloud SQL, object storage in Cloud Storage, data warehouse needs using BigQuery, among others.

Once you have some possible configurations in mind, you can use the pricing calculator to estimate costs for the different products you will be using.

Pricing Calculator



Total Estimated Cost: USD 4,958.48 per 1 month

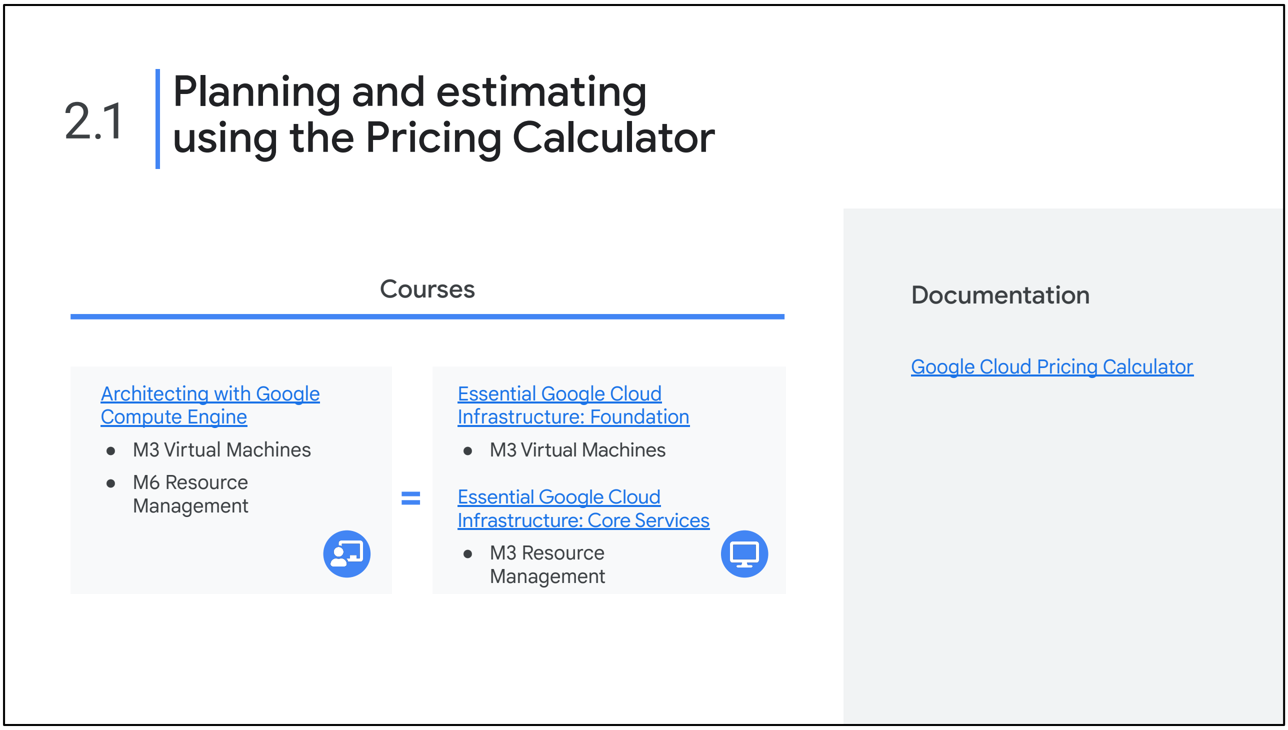
Estimate Currency

USD - US Dollars

Adjust Estimate Timeframe

- Total estimated cost: daily, weekly, monthly, quarterly, yearly and 3-year increments
- Cost is only an estimate
- Question to ask: How closely your estimated usage matches your actual usage?
- Not a binding contract, just a planning tool

Keep in mind pricing estimates generated by the calculator are not binding in any way. Please just use them as a planning tool. Real costs are going to be determined by actual usage.



2.1 Planning and estimating using the Pricing Calculator

Courses

Architecting with Google Compute Engine

- M3 Virtual Machines
- M6 Resource Management

=

Essential Google Cloud Infrastructure: Foundation

- M3 Virtual Machines

Essential Google Cloud Infrastructure: Core Services

- M3 Resource Management

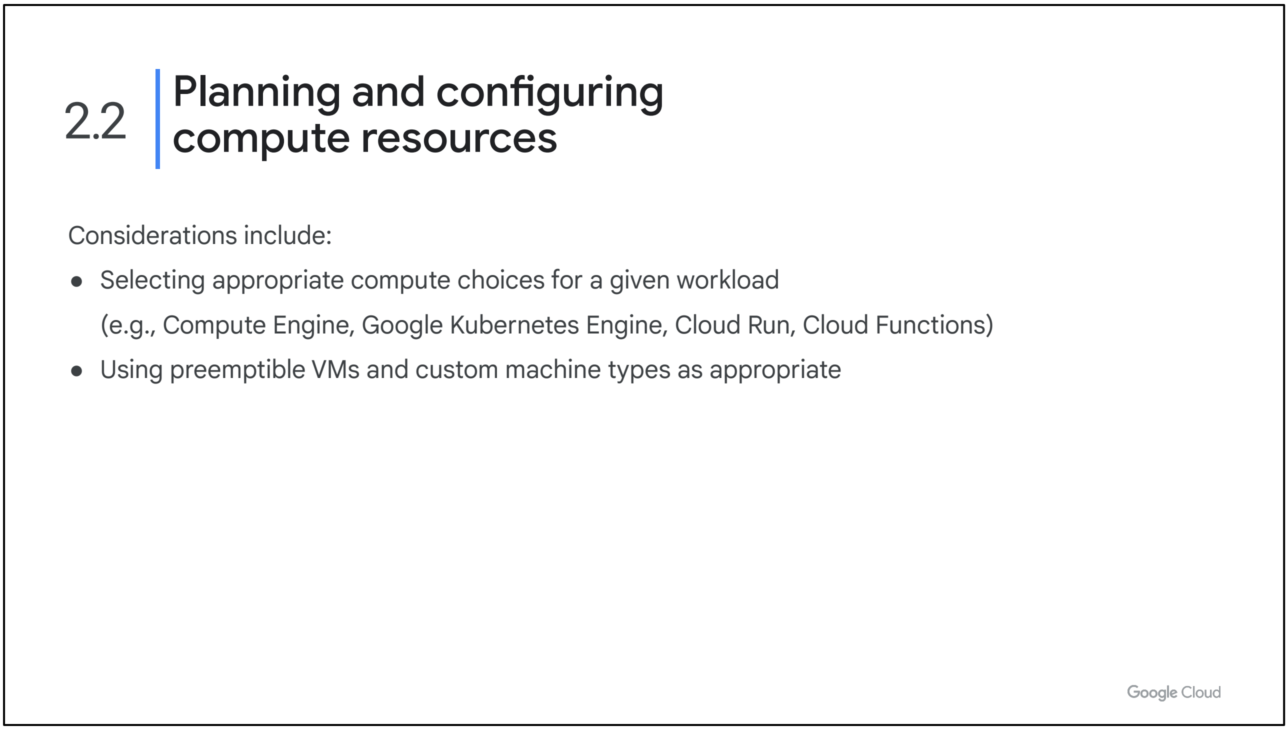
Documentation

Google Cloud Pricing Calculator

Now that we’ve reviewed the diagnostic question related to Section 2.1, 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.

https://cloud.google.com/products/calculator/



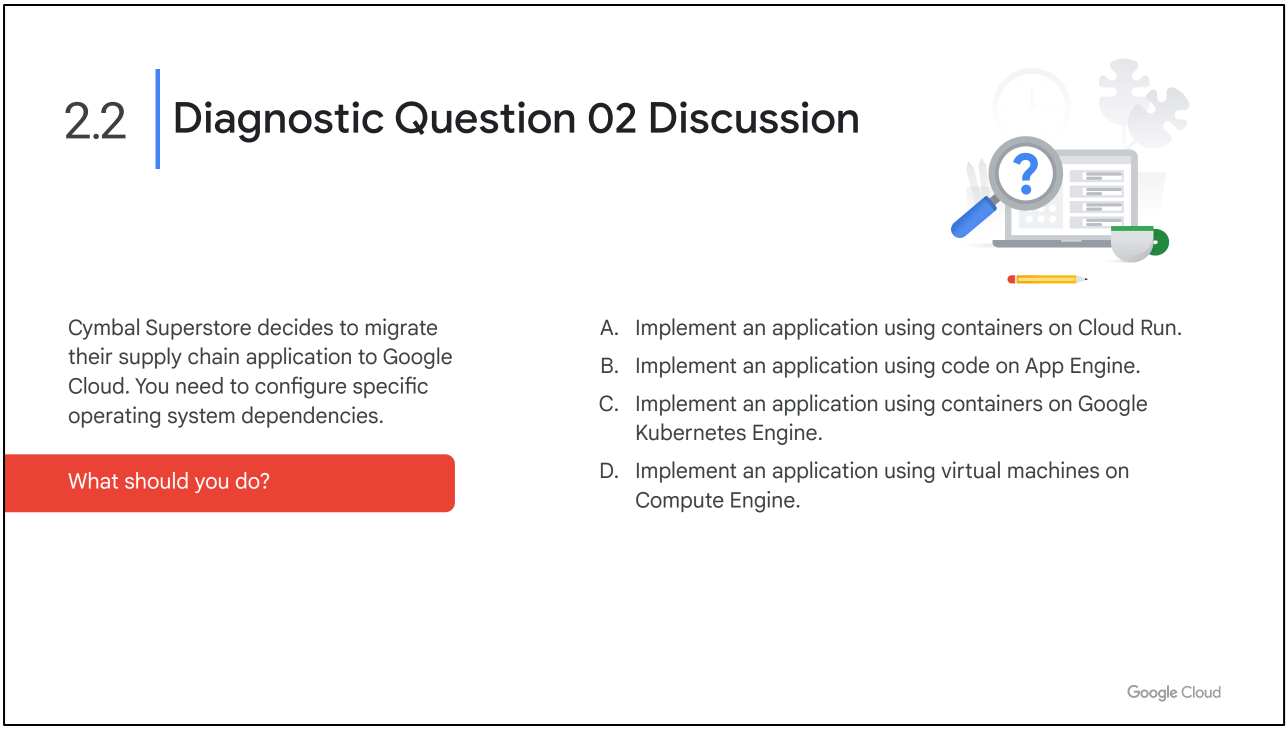
2.2 Planning and configuring compute resources

Considerations include:

- Selecting appropriate compute choices for a given workload (e.g., Compute Engine, Google Kubernetes Engine, Cloud Run, Cloud Functions)
- Using preemptible VMs 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 2: Differentiate between available compute options in Google Cloud.
Question 3: Determine when to use serverless solutions such as Cloud Run, e.g. code focus, quick development, minimal overhead.
Question 4: Determine when to use Google Compute Engine, e.g. more control, OS level dependencies, customization.
Question 5: Determine when to use Kubernetes Engine, e.g. platform independence, deployment velocity, container management.



2.2 Diagnostic Question 02 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?
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:

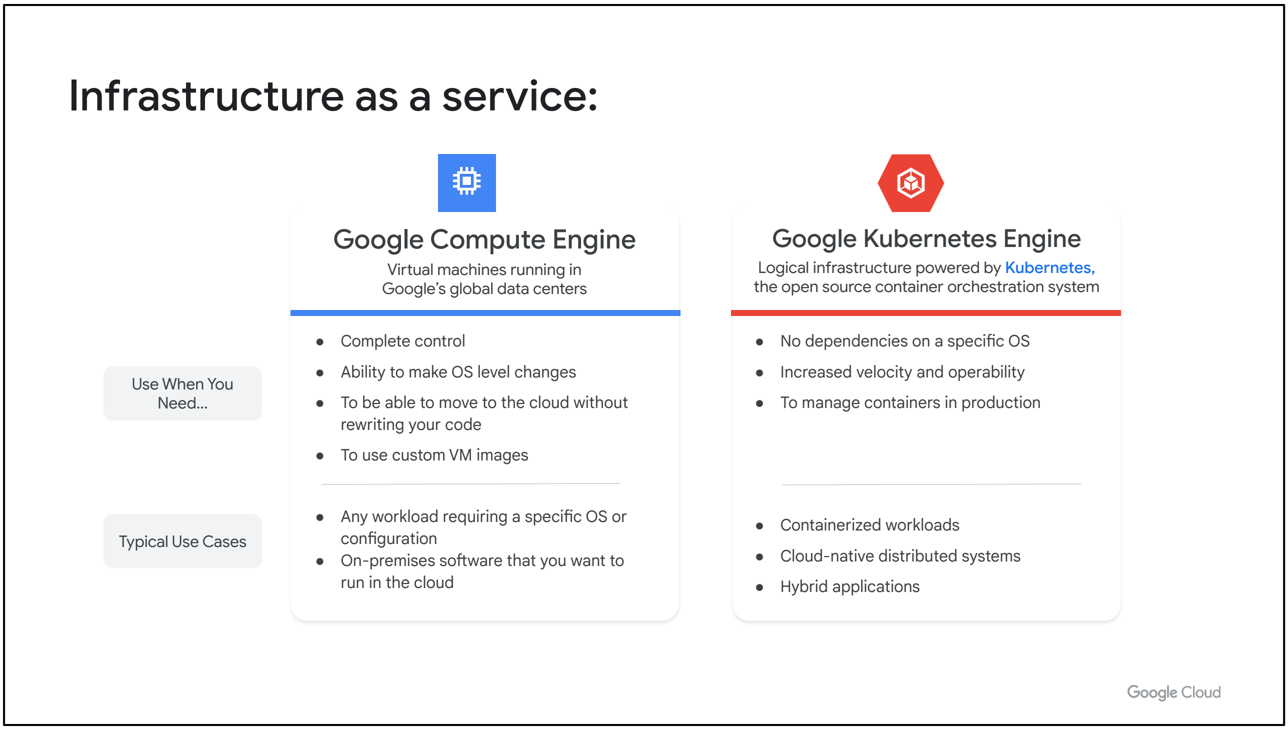
https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree

Content mapping:

- Instructor-led Training/OnDemand
  - Google Cloud Fundamentals: Core Infrastructure
    - M3 Virtual Machines in the Cloud
  - Architecting with Google Compute Engine
    - 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.

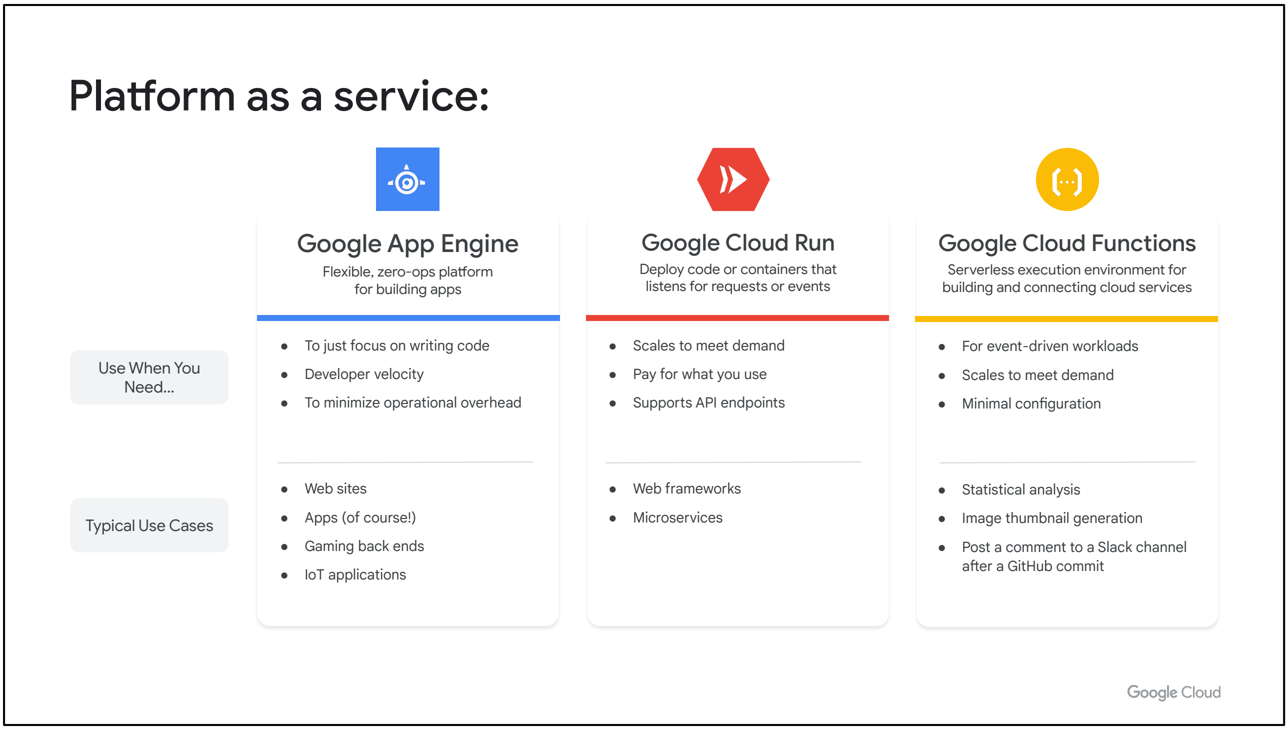


Infrastructure as a service:

	Google Compute Engine Virtual machines running in Google's global data centers	Google Kubernetes Engine Logical infrastructure powered by Kubernetes, the open source container orchestration system
Use When You Need...	• Complete control • Ability to make OS level changes • To be able to move to the cloud without rewriting your code • To use custom VM images	• No dependencies on a specific OS • Increased velocity and operability • To manage containers in production
Typical Use Cases	• Any workload requiring a specific OS or configuration • On-premises software that you want to run in the cloud	• Containerized workloads • Cloud-native distributed systems • Hybrid applications

Google Cloud

Compute Engine and Google Kubernetes Engine are server-based.

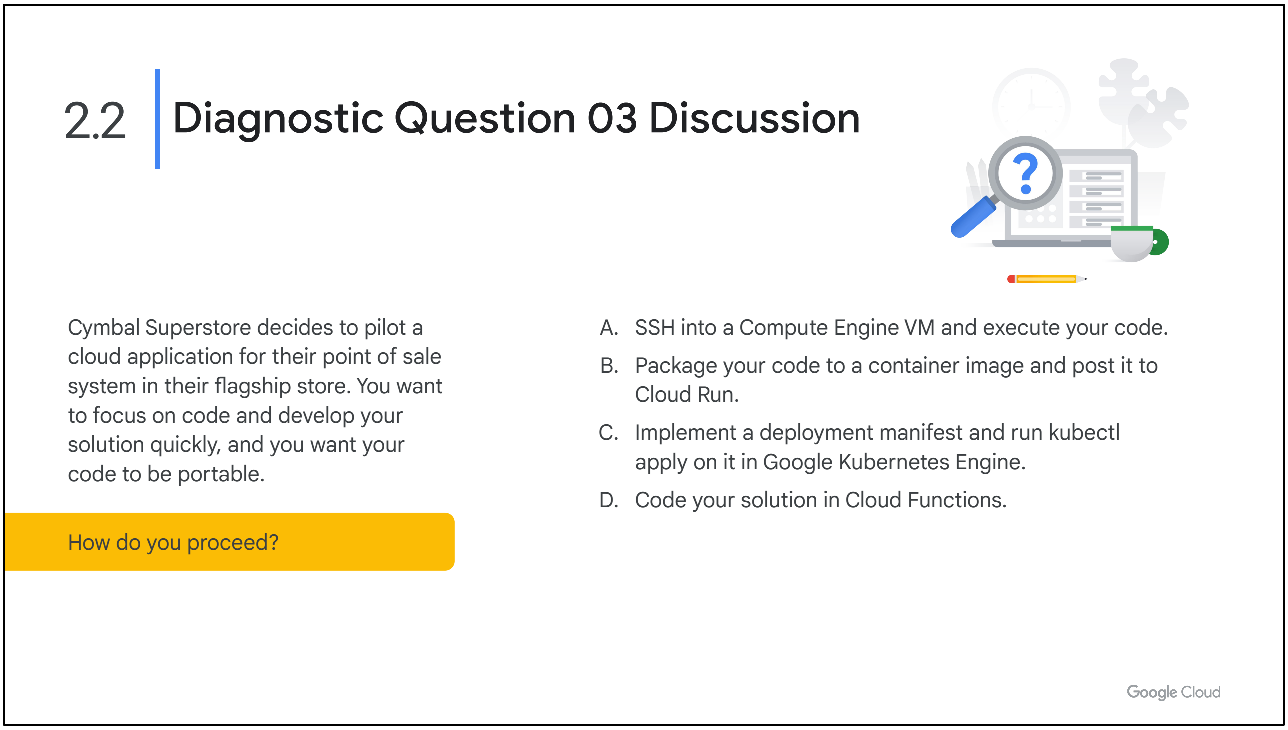


Platform as a service:

	Google App Engine	Google Cloud Run	Google Cloud Functions
	Flexible, zero-ops platform for building apps	Deploy code or containers that listens for requests or events	Serverless execution environment for building and connecting cloud services
Use When You Need...	• To just focus on writing code • Developer velocity • To minimize operational overhead	• Scales to meet demand • Pay for what you use • Supports API endpoints	• For event-driven workloads • Scales to meet demand • Minimal configuration
Typical Use Cases	• Web sites • Apps (of course!) • Gaming back ends • IoT applications	• Web frameworks • Microservices	• Statistical analysis • Image thumbnail generation • Post a comment to a Slack channel after a GitHub commit

Google Cloud

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



2.2 Diagnostic Question 03 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 Functions.

Google Cloud

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?

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 Functions.
Feedback: Incorrect. Cloud 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:

- Instructor-led Training/OnDemand
  - Google Cloud Fundamentals: Core Infrastructure
    - M6 Applications in the Cloud

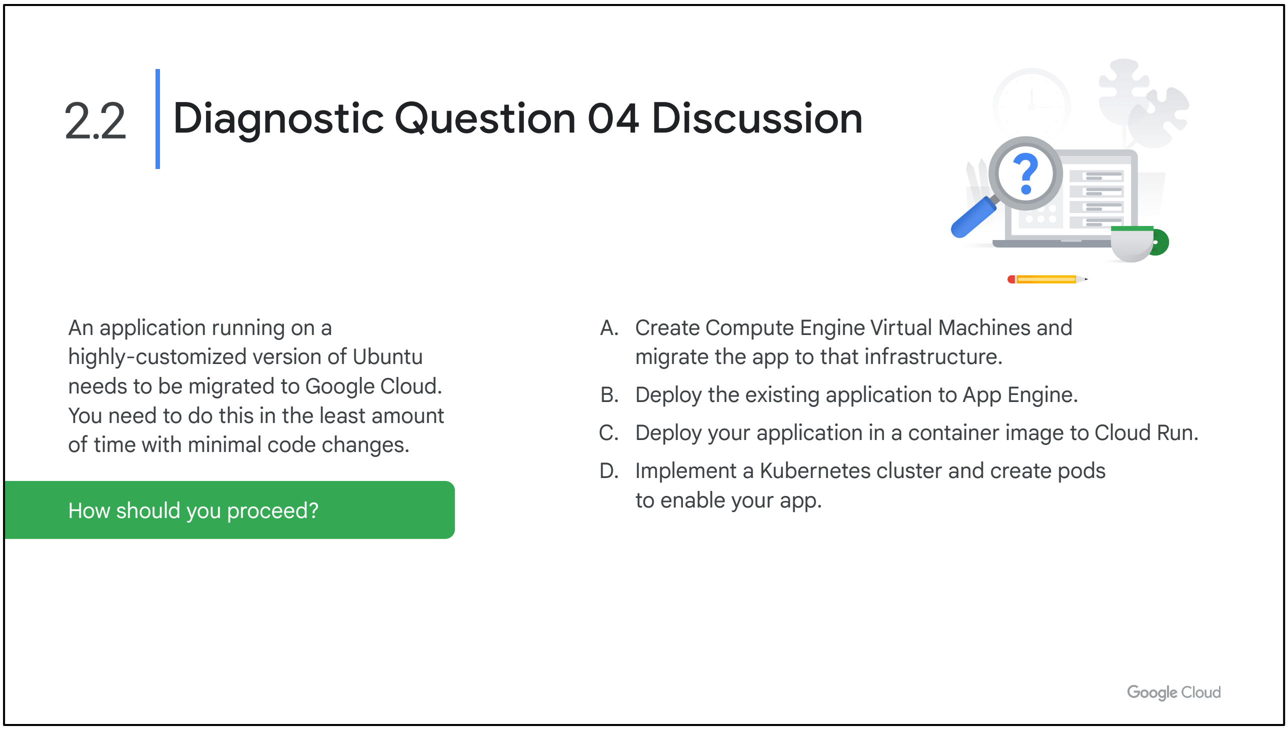
Summary:

The three serverless compute options available in Google Cloud are App Engine, Cloud Run, and Cloud 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.



2.2 Diagnostic Question 04 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

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?

*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:

- Instructor-led Training/OnDemand
  - Google Cloud Fundamentals: Core Infrastructure
    - M3 Virtual Machines in the Cloud
  - Architecting with Google Compute Engine
    - M3 Virtual Machines

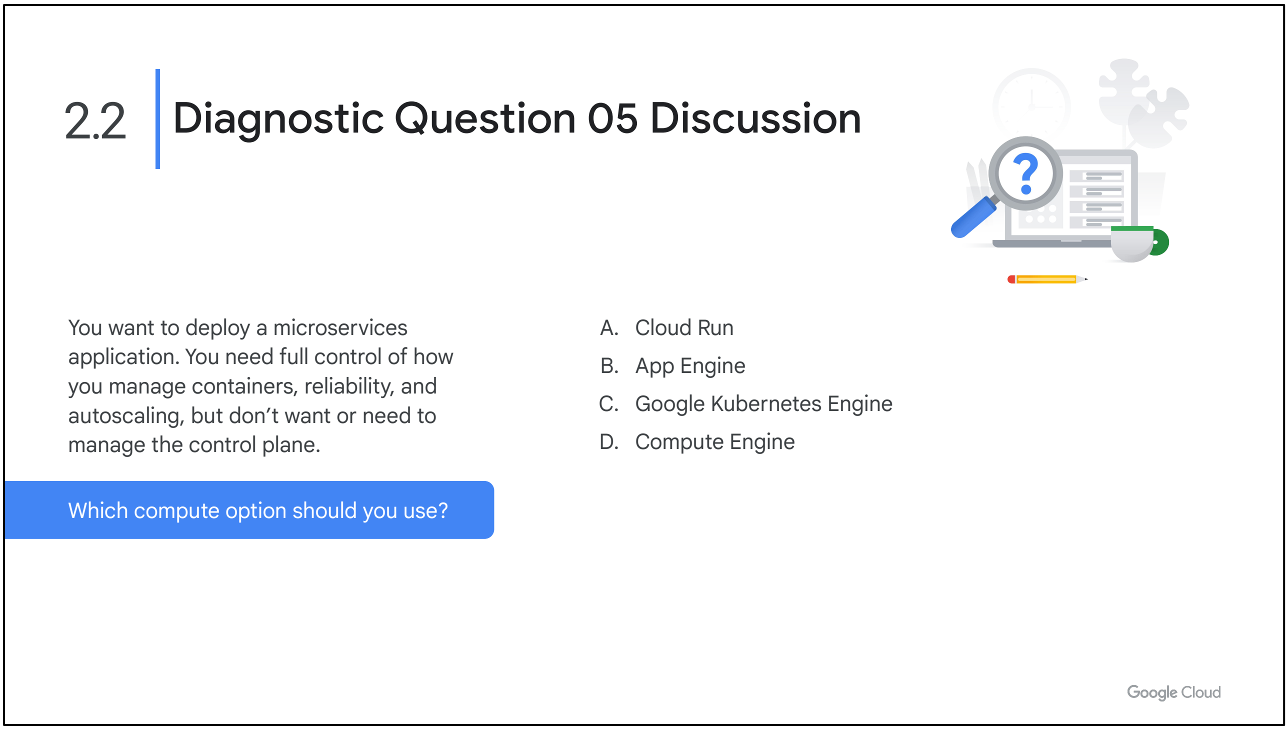
Summary:

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 the operations suite.



2.2 Diagnostic Question 05 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

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?

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

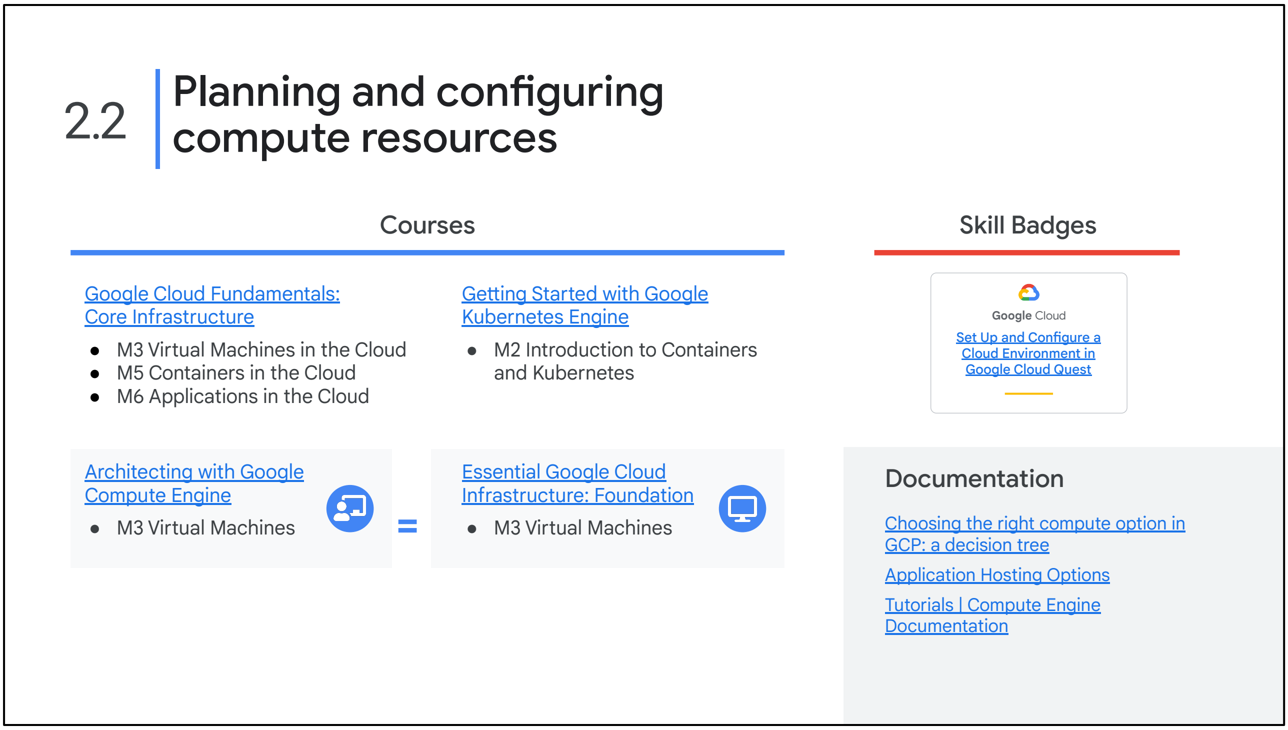
Content mapping:

- Instructor-led Training/OnDemand

    - Google Cloud Fundamentals: Core Infrastructure
        - M5 Containers in the Cloud
    - Getting Started with GKE
        - M2 Introduction to Containers and Kubernetes
- Quests
    - Set Up and Configure a Cloud Environment in Google Cloud (https://www.qwiklabs.com/quests/119)

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.



2.2 Planning and configuring compute resources

Courses

[Google Cloud Fundamentals: Core Infrastructure](#)

- M3 Virtual Machines in the Cloud
- M5 Containers in the Cloud
- M6 Applications in the Cloud

[Getting Started with Google Kubernetes Engine](#)

- M2 Introduction to Containers and Kubernetes

[Architecting with Google Compute Engine](#)

- M3 Virtual Machines

=

[Essential Google Cloud Infrastructure: Foundation](#)

- M3 Virtual Machines

Skill Badges

Google Cloud

[Set Up and Configure a Cloud Environment in Google Cloud Quest](#)

Documentation

[Choosing the right compute option in GCP: a decision tree](#)

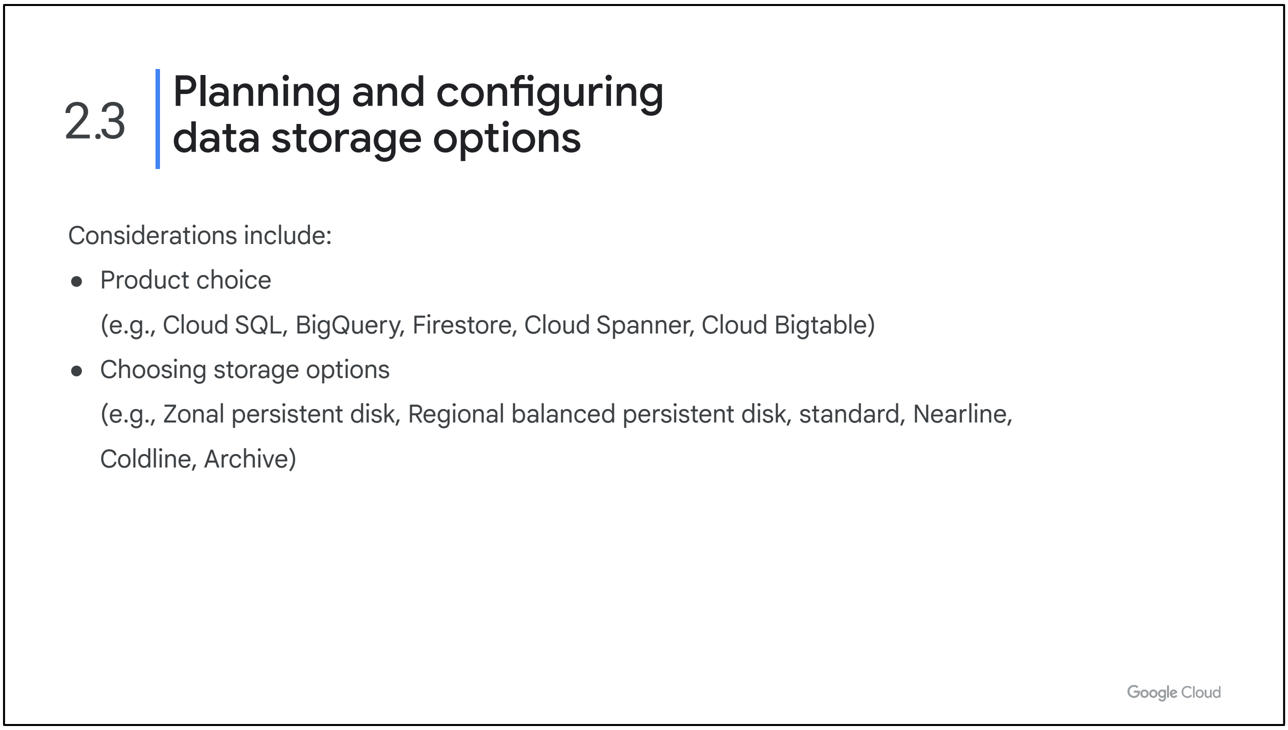
[Application Hosting Options](#)

[Tutorials | Compute Engine Documentation](#)

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.

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



2.3 Planning and configuring data storage options

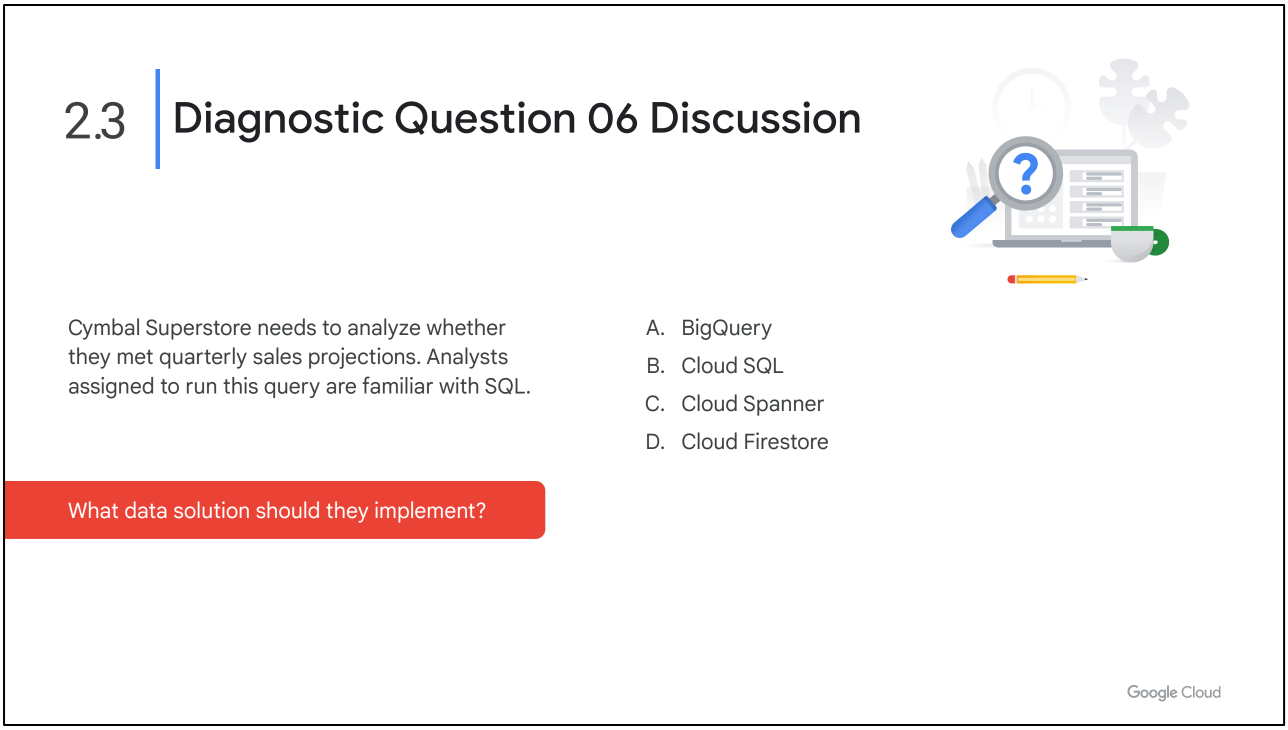
Considerations include:

- Product choice
  (e.g., Cloud SQL, BigQuery, Firestore, Cloud Spanner, Cloud Bigtable)
- Choosing storage options
  (e.g., Zonal persistent disk, Regional balanced 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 6: Associate Google Cloud data services with modern data constructs, e.g. relational, non-relational or no-sql, object or warehouse.
Question 7: List the storage classes available for cloud storage including regional, multiregional, near and cold line.
Question 8: Differentiate between relational databases and data warehouses, e.g. relational databases can be transactional, while data warehouses are analytical and based on historical data.



2.3 Diagnostic Question 06 Discussion

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?

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

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?

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

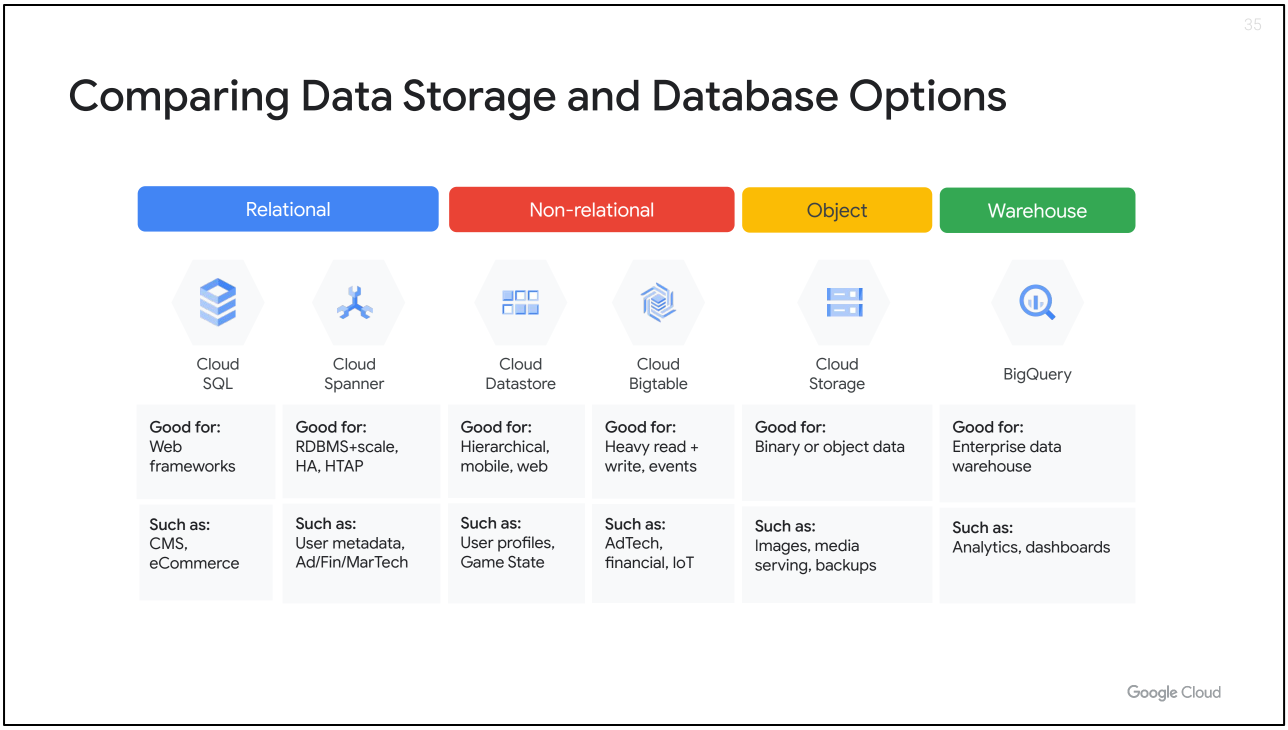
D. Cloud Firestore
Feedback: Incorrect. Cloud Firestore is a NoSQL document database used to define entities with attributes. It is not a good choice for the analysis of historical data as described in the scenario.

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

Content mapping:

- Instructor-led Training/OnDemand
  - Google Cloud Fundamentals: Core Infrastructure
    - M4 Storage in the Cloud
  - Architecting with Google Compute Engine
    - M5 Storage and Database Services
- Quests
  - Perform Foundational Infrastructure Tasks in Google Cloud (https://www.qwiklabs.com/quests/118)

Summary:
Explanation/summary on the following slide.



Comparing Data Storage and Database Options

Relational		Non-relational		Object	Warehouse
Cloud SQL	Cloud Spanner	Cloud Datastore	Cloud Bigtable	Cloud Storage	BigQuery
Good for: Web frameworks	**Good for:** RDBMS+scale, HA, HTAP	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events	**Good for:** Binary or object data	**Good for:** Enterprise data warehouse
Such as: CMS, eCommerce	**Such as:** User metadata, Ad/Fin/MarTech	**Such as:** User profiles, Game State	**Such as:** AdTech, financial, IoT	**Such as:** Images, media serving, backups	**Such as:** Analytics, dashboards

Google Cloud

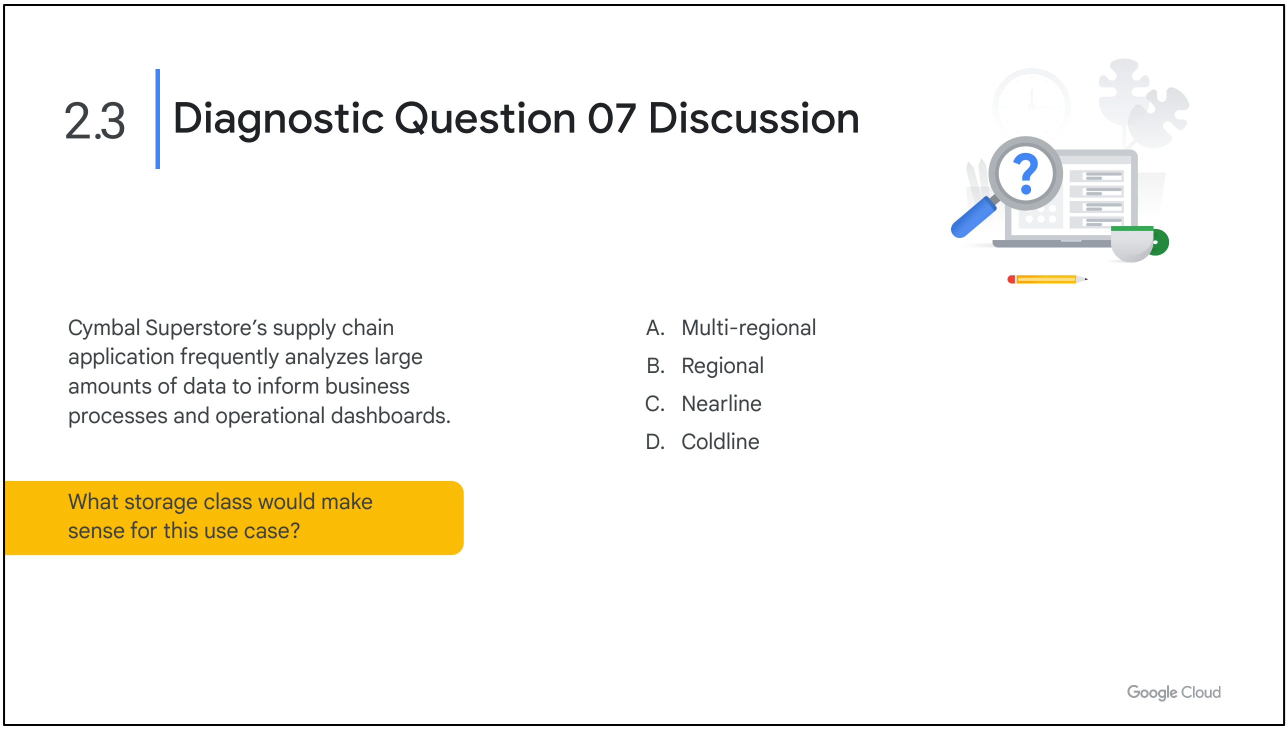
This question relates to Google Cloud data services and what data construct they are based on.

Firestore and Cloud Bigtable are NoSQL implementations. Firestore is a document database that supports entities and attributes. Cloud Big Table 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 Cloud 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 Cloud 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.



2.3 Diagnostic Question 07 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. Multi-regional
B. Regional
C. Nearline
D. Coldline

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?

A. Multi-regional
Feedback: Incorrect. For performance reasons, storage is required close to where analysis and reporting is happening. Multi-regional is not a good storage type for this use case.

*B. Regional
Feedback: Correct! In this case, for performance reasons, you want the storage assets to be close to where the processing is taking place, so a regional storage class makes sense.

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

D. Coldline
Feedback: Incorrect. Dashboards need current data to analyze. Coldline is good for

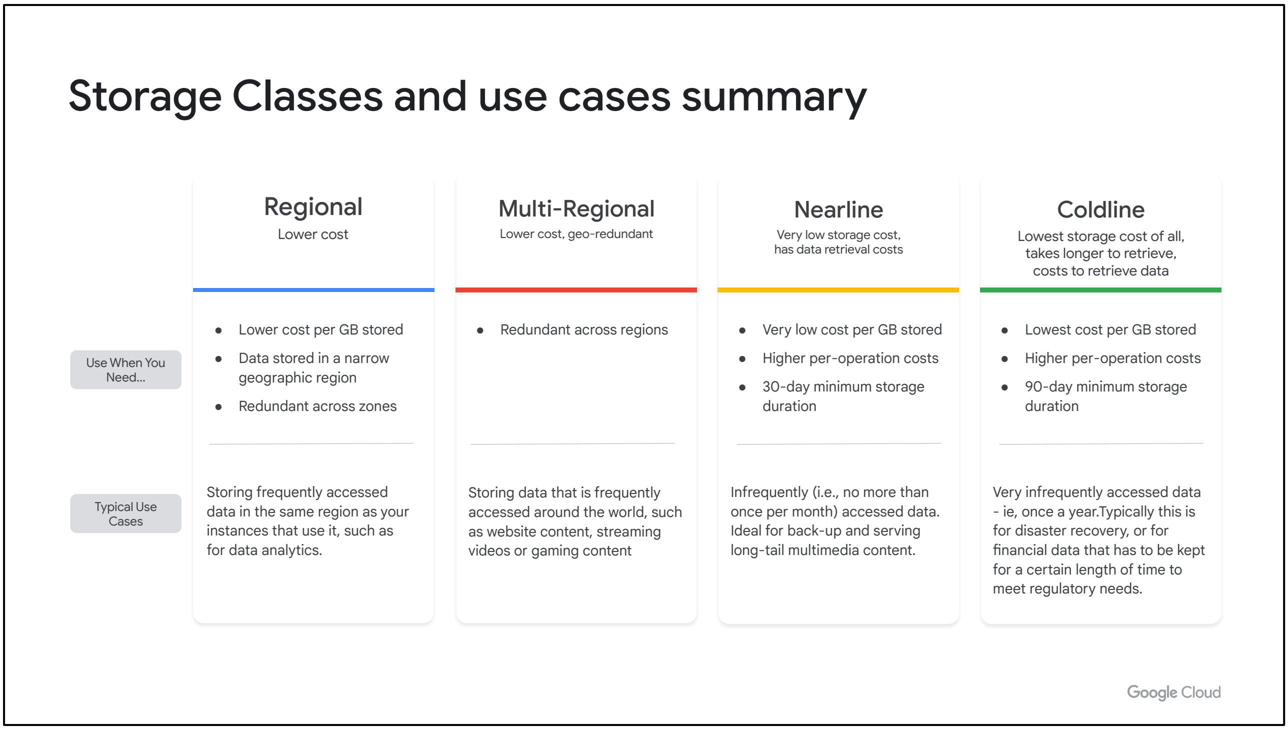
storing data accessed only every 90 days.

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

Content mapping:

- Instructor-led Training/OnDemand
    - Google Cloud Fundamentals: Core Infrastructure
        - M4 Storage in the Cloud
    - Architecting with Google Compute Engine
        - M5 Storage and Database Services
- Quests
    - Perform Foundational Infrastructure Tasks in Google Cloud (https://www.qwiklabs.com/quests/118)

Summary:
Explanation/summary on the following slide.



Storage Classes and use cases summary

	Regional Lower cost	Multi-Regional Lower cost, geo-redundant	Nearline Very low storage cost, has data retrieval costs	Coldline Lowest storage cost of all, takes longer to retrieve, costs to retrieve data
Use When You Need...	• Lower cost per GB stored • Data stored in a narrow geographic region • Redundant across zones	• Redundant across regions	• Very low cost per GB stored • Higher per-operation costs • 30-day minimum storage duration	• Lowest cost per GB stored • Higher per-operation costs • 90-day minimum storage duration
Typical Use Cases	Storing frequently accessed data in the same region as your instances that use it, such as for data analytics.	Storing data that is frequently accessed around the world, such as website content, streaming videos or gaming content	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.

Google Cloud

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. Cold line 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