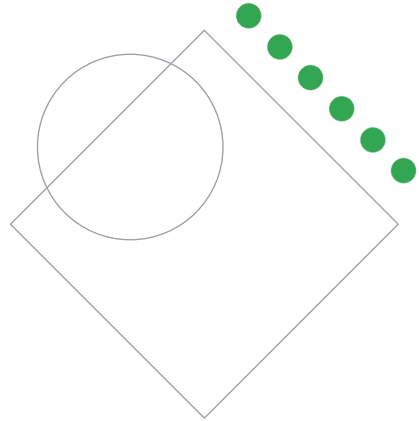Google Cloud

Preparing for Your Professional Cloud Architect Journey

Module 1: Designing and Planning a Cloud Solution Architecture

Welcome to Module 1: Designing and Planning a Cloud Solution Architecture.
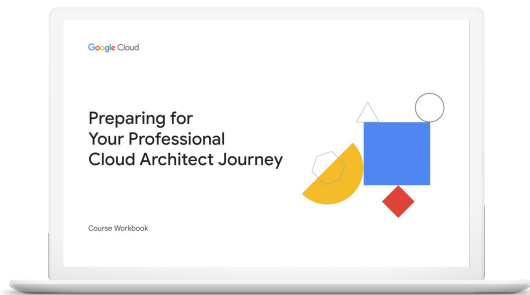
Review and study planning

You'll now review the diagnostic questions and your answers to help you identify what to include in your study plan.

Your study plan:

Designing and planning a cloud solution architecture

1.1	Designing a solution infrastructure that meets business requirements
1.2	Designing a solution infrastructure that meets technical requirements
1.3	Designing network, storage, and compute resources
1.4	Creating a migration plan
1.5	Envisioning future solution improvements

Google Cloud

The diagnostic questions align with these objectives of this exam section. Use the PDF resource that follows to review the questions and how you answered them. Pay specific attention to the rationale for both the correct and incorrect answers. Use the resources detailed under **Where to look** and **Content mapping** to build a study plan that meets your learning needs.

1.1 | Designing a solution infrastructure that meets business requirements

Considerations include:
- Business use cases and product strategy
- Cost optimization
- Supporting the application design
- Integration with external systems
- Movement of data
- Design decision trade-offs
- Build, buy, modify, or deprecate
- Success measurements (e.g., key performance indicators [KPI], return on investment [ROI], metrics)
- Compliance and observability

A Professional Cloud Architect should understand the business requirements driving the solution. Ultimately technology is just a tool you can use to help businesses achieve their objectives. For example, there are a number of things you can do to reduce costs, but if your instances take a long time to start and performance is more important than cost, you may decide to leave some instances running even if they're not being used and not to follow recommendations to resize your instances.

Question 1 asked you to explore meeting business requirements by using managed services for data processing in Google Cloud. Question 2 required you to evaluate key performance indicators (KPIs) to establish a service level objective (SLO).

Diagnostic Question 01 Discussion

Cymbal Direct drones continuously send data during deliveries. You need to process and analyze the incoming telemetry data. **After processing, the data should be retained, but it will only be accessed once every month or two.** Your CIO has issued a directive to incorporate managed services wherever possible. You want a **cost-effective solution to process the incoming streams of data.**

What should you do?

A. Ingest data with ClearBlade IoT Core, process it with **Dataprep,** and store it in a **Coldline Cloud Storage bucket.**

B. Ingest data with ClearBlade IoT Core, and then publish to Pub/Sub. Use **Dataflow** to process the data, and store it in a **Nearline Cloud Storage bucket.**

C. Ingest data with ClearBlade IoT Core, and then publish to Pub/Sub. Use **BigQuery** to process the data, and store it in a **Standard Cloud Storage bucket.**

D. Ingest data with ClearBlade IoT Core, and then store it in **BigQuery.**

Google Cloud

Feedback:
A. Incorrect. Dataprep is used to normalize data before processing, if necessary. Coldline could be used, but Nearline is probably a better fit because the data could be accessed every month. Coldline has a higher cost for data access than Nearline, which makes it a poor choice for data accessed "every month or two."
B. Correct! Dataflow is a fully managed service that can be used to process both streams and batches of data. Nearline is a good fit because the data could be accessed every month.
C. Incorrect. BigQuery is Google Cloud's serverless data warehousing tool tool, which is not used for processing real-time data. Standard could be used but is not the most cost-effective solution. Nearline would be more cost-effective, because the data is only accessed once per month at most. Standard storage has a lower access cost, but has higher storage costs, so it's a better fit for frequently accessed data which will not be retained for long periods of time.
D. Incorrect. BigQuery is Google Cloud's serverless data warehousing tool, which is not used for processing real-time data.

Where to look:
https://cloud.google.com/architecture/data-lifecycle-cloud-platform?hl=en#process_and_analyze
https://www.clearblade.com/iot-core/

Content mapping:
- Architecting with Google Compute Engine (ILT)

- - ○ M11 Managed Services

- ● Elastic Google Cloud Infrastructure: Scaling and Automation (On-demand)
  - ○ M4 Managed Services

Summary:
Taking business requirements into account when designing solutions is important: in this case, favoring managed services. Dataflow is a great solution for streaming data (or batch processing). Because drones are essentially IoT devices streaming telemetry data, ClearBlade IoT Core is the best choice to ingest data. Pub/Sub acts as the glue between ClearBlade IoT Core and Dataflow by providing asynchronous communication between services. Storing the data in Nearline Cloud Storage is a good choice because of its low cost and ability to scale well to large amounts of data. It's important to understand the attributes of the different storage classes so you can select the correct one for your solution.

1.1 | Diagnostic Question 02 Discussion

Customers need to have a g**ood experience** when accessing your web application so they will continue to use your service. You want to **define key performance indicators (KPIs)** to establish a service level objective (SLO).

Which KPI could you use?

A. Eighty-five percent of customers are **satisfied users**
B. Eighty-five percent of **requests succeed when aggregated over 1 minute**
C. **Low latency** for > 85% of requests when aggregated over 1 minute
D. Eighty-five percent of **requests are successful**

Google Cloud

Feedback:
A. Incorrect. KPIs must be specific and measurable. 'Satisfied' is not specific, and the KPI is not bounded in a period of time.
B. Correct! This is specific, and you can reasonably expect to meet this KPI.
C. Incorrect. KPIs must be specific and measurable. The word 'low' does not provide enough information to determine whether latency is acceptable. You should be able to compare your measurement against a threshold value to determine whether it is acceptable.
D. Incorrect. This KPI is not bounded by time. You could have 100% success during one period of time and 75% during another, but the average (87.5) uptime would exceed the KPI.

Where to look:
https://sre.google/sre-book

Content mapping:
- ● Architecting with Google Cloud: Design and Process (ILT)
  - ○ M1 Defining Services

- ● Reliable Google Cloud Infrastructure: Design and Process (On-demand)
  - ○ M1 Defining Services

Summary:
KPIs should be SMART: Specific, Measurable, Achievable, Relevant, and

Time-Bound. When you determine a KPI, the indicator chosen must be specific and measurable. A web server responding "quickly" isn't specific. You can measure latency as a value, so measuring average latency over a period of time is a better indicator. KPIs must also be relevant: for example, there is no benefit to measuring the size of a request if it doesn't affect a user in some way. You also want goals to be achievable. An SLO of 100% uptime isn't realistically achievable, so it wouldn't make sense to set 100% uptime as a goal.

1.1 Designing a solution infrastructure that meets business requirements

Resources to start your journey

Google Cloud Architecture Framework: System design

SRE Books

Google Cloud

Understanding business requirements is a critical part of the Professional Cloud Architect's role. You need to design solutions that meet both business and technical needs of an organization. The Google Cloud Architecture Center provides one place to get started. Google's Site Reliability Engineering books are another resource.

https://cloud.google.com/architecture/framework/system-design
https://sre.google/books/

| Designing a solution infrastructure that meets technical requirements

Considerations include:

- High availability and failover design
- Elasticity of cloud resources with respect to quotas and limits
- Scalability to meet growth requirements
- Performance and latency

As a Professional Cloud Architect, you must design solutions that meet technical requirements. For example, if a client needs to have zero data loss, you have a number of tools you can use to help provide that, but there might be a trade off. You need to know that the requirement exists, so you can recommend the right tool for the job, but also justify why those tradeoffs might be necessary. For example, your organization may be using a traditional SQL database, but you are reaching the upper limits of its scalability. You could consider using Spanner or BigQuery to address this problem. However, you may need to make some changes to the application. The changes will probably be fairly minor, but you will need to be aware of them.

Question 3 tested your knowledge of how to use the CPU and memory options for virtual machines to meet requirements.

Diagnostic Question 03 Discussion

Cymbal Direct developers have written a new application. Based on initial usage estimates, you decide to run the application on **Compute Engine instances with 15 Gb of RAM and 4 CPUs.** These instances **store persistent data locally.** After the application runs for several months, historical data indicates that the **application requires 30 Gb of RAM**. Cymbal Direct management wants you to make adjustments that will **minimize costs.**

What should you do?

A. Stop the instance, and then use the command gcloud compute instances **set-machine-type VM_NAME --machine-type e2-standard-8.** Start the instance again.

B. Stop the instance, and then use the command gcloud compute instances **set-machine-type VM_NAME --machine-type e2-standard-8.** Set the instance's metadata to: **preemptible: true**. Start the instance again.

C. Stop the instance, and then use the command gcloud compute instances **set-machine-type VM_NAME --machine-type 2-custom-4-30720**. Start the instance again.

D. Stop the instance, and then use the command gcloud compute instances **set-machine-type VM_NAME --machine-type 2-custom-4-30720**. Set the instance's metadata to: **preemptible: true.** Start the instance again.

Google Cloud

Feedback:
A. Incorrect. An e2-standard-8 instance will have the appropriate amount of memory. However, this instance type will have more CPU than necessary and incur additional unnecessary costs.
B. Incorrect. An e2-standard-8 instance will have the appropriate amount of memory. However, this instance type will have more CPU than necessary and incur additional unnecessary costs. Although preemptible instances can save substantial money, they are not appropriate for instances that need to store persistent data locally.
C. Correct! Custom instances are a good way to optimize costs. You don't have to pay for resources you don't need.
D. Incorrect. Although preemptible instances can save substantial money, they are not appropriate for instances that need to store persistent data locally.

Where to look:
https://cloud.google.com/compute/docs/instances/changing-machine-type-of-stopped-instance

Content mapping:
- ● Architecting with Google Compute Engine (ILT)
  - ○ M3 Virtual Machines

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

Summary:

Cost is often a factor when architecting a solution, and you can use tools to optimize your spending in Google Cloud. Use the pricing calculator to calculate the cost of resources, and check for recommendations in the console for over-sized instances. Use preemptible and committed-use instances where appropriate. Remember that the standard instance sizes are based on what makes sense for most general purpose applications, but your environment may differ. You can use custom instances to make sure you only pay for the resources you need.

1.2 | Designing a solution infrastructure that meets technical requirements

Resources to start your journey

Google Cloud Architecture Framework: System design

You only had one diagnostic question that specifically addressed this section objective. In order to address technical requirements, you need a thorough understanding of the available options in Google Cloud. You can learn more about specific products and services in the other courses in the Professional Cloud Architect learning path. This link is one place to get started.

https://cloud.google.com/architecture/framework/system-design

1.3 | Designing network, storage, and compute resources

Considerations include:

- Integration with on-premises/multicloud environments
- Cloud-native networking (VPC, peering, firewalls, container networking)
- Choosing data processing technologies
- Choosing appropriate storage types (e.g., object, file, databases)
- Choosing compute resources (e.g., preemptible, custom machine type, specialized workload)
- Mapping compute needs to platform products

Every cloud solution environment will include some type of networking, storage, and compute resources. Most environments will use databases.As a Professional Cloud Architect, you need a thorough understanding of the solutions available in Google Cloud for these resources. You should be extremely comfortable with how to decide which compute, network, data processing technology, storage type and tier to use.

Question 4 challenged you to differentiate between the different types and usage of VPC networks. Question 5 asked you to evaluate when to use Shared VPC and when to use VPC Network Peering to configure private communication between VPC networks. Question 6 tested your knowledge of using Google Cloud Interconnect and Peering services to connect an on-premises infrastructure to Google Cloud. Question 7 tested your knowledge of storage and database solutions in Google Cloud. Question 8 examined your understanding of Kubernetes concepts and commands.

Diagnostic Question 04 Discussion

You are creating a new project. You plan to set up a Dedicated interconnect between two of your data centers in the near future and want to ensure that your resources are only deployed to the **same regions** where your data centers are located. You need to make sure that you **don't have any overlapping IP addresses** that could cause conflicts when you set up the interconnect. You want to use **RFC 1918 class B address space.**

What should you do?

A. Create a new project, **leave the default network in place**, and then use the default 10.x.x.x network range to create subnets in your desired regions.

B. Create a new project, delete the default VPC network, **set up an auto mode VPC network**, and then use the default 10.x.x.x network range to create subnets in your desired regions.

C. Create a new project, delete the default VPC network, **set up a custom mode VPC network**, and then use IP addresses in the **172.16.x.x address range** to create subnets in your desired regions.

D. Create a new project, delete the default VPC network, **set up the network in custom mode**, and then use IP addresses in the **192.168.x.x address range** to create subnets in your desired zones. **Use VPC Network Peering** to connect the zones in the same region to create regional networks.

Google Cloud

Feedback:
A. Incorrect. Default mode networks create subnets for you automatically in each zone and could allow people to accidentally provision resources in other regions.
B. Incorrect. Auto mode networks create subnets for you automatically in each zone and could allow people to accidentally provision resources in other regions.
C. Correct! Custom networks give you full control.
D. Incorrect. Subnets are regional, not zonal.

Where to look:
https://cloud.google.com/vpc/docs/vpc

Content mapping:
- ● Architecting with Google Compute Engine (ILT)
  - ○ M2 Virtual Networks

- ● Essential Google Cloud Infrastructure: Foundation (On-demand)
  - ○ M2 Virtual Networks

Summary:
A custom mode VPC network does not automatically create subnets. This type of network provides you with complete control over its subnets and IP address ranges. You decide which subnets to create, in regions you choose, and which IP address ranges you use for subnets, but only if they fall within the RFC 1918 address space. RFC 1918 Class B uses the 172.16.x.x address space. Default networks are created when you set up a new project but are a form of auto mode VPC network, and thus do

not give you the ability to specify where your subnets are created. Subnets that use default IP address ranges are created automatically in all regions.

Diagnostic Question 05 Discussion

Cymbal Direct is working with Cymbal Retail, a separate, autonomous division of Cymbal with different staff, networking teams, and data center. Cymbal Direct and Cymbal Retail are **not in the same Google Cloud organization**. Cymbal Retail needs access to Cymbal Direct's web application for making bulk orders, but the **application will not be available on the public internet**. You want to ensure that **Cymbal Retail has access to your application with low latency**. You also want to avoid **egress network charges** if possible.

What should you do?

A. Verify that the subnet range Cymbal Retail is using **doesn't overlap** with Cymbal Direct's subnet range, and then **enable VPC Network Peering for the project.**

B. If Cymbal Retail does not have access to a Google Cloud data center, **use Carrier Peering** to connect the two networks.

C. Specify Cymbal Direct's project as the **Shared VPC** host project, and then configure Cymbal Retail's project as a service project.

D. Verify that the subnet Cymbal Retail is using has the **same** IP address range with Cymbal Direct's subnet range, and then **enable VPC Network Peering for the project.**

Google Cloud

Feedback:
A. Correct! VPC Peering allows for shared networking between organizations.
B. Incorrect. Carrier Peering lets you connect to Google's public infrastructure when you can't satisfy the peering requirements yourself.
C. Incorrect. Because the partner is in a different organization, you can't use a Shared VPC.
D. Incorrect. If both subnets use the same IP address range, there could be IP address conflicts and issues with routing.

Where to look:
https://cloud.google.com/vpc/docs/vpc-peering

Content mapping:
- Architecting with Google Compute Engine (ILT)
  - M8 Interconnecting Networks

- Elastic Google Cloud Infrastructure: Scaling and Automation (On-demand)
  - M1 Interconnecting Networks

Summary:
Use VPC Network Peering to configure private communication between VPC networks in different organizations. You can use it for different projects within the same organization, but that's less common. Make sure that there aren't overlapping IP address ranges. Shared VPC only works within the same organization, so that's an

important piece of information you can use to help determine which solution is more appropriate.

Diagnostic Question 06 Discussion

Cymbal Direct's employees will use **Google Workspace**. Your current on-premises network **cannot meet the requirements to connect** to Google's public infrastructure.

What should you do?

A. Order a **Dedicated Interconnect** from a Google Cloud partner, and ensure that proper routes are configured.

B. Connect the network to a Google point of presence, and enable **Direct Peering.**

C. Order a **Partner Interconnect** from a Google Cloud partner, and ensure that proper routes are configured.

D. Connect the on-premises network to Google's public infrastructure via a partner that supports **Carrier Peering.**

Google Cloud

Feedback:
A. Incorrect. A Dedicated Interconnect connects a data center to your VPC.
B. Incorrect. Cymbal Direct's on-premises network cannot meet the requirements for peering.
C. Incorrect. A Partner Interconnect allows access to your VPC network via a partner.This would allow access between on-premises and your internal IP addresses, which greatly exceeds the scope and does not follow the principle of least privilege.
D. Correct! Carrier Peering allows using a Google partner to connect to Google public infrastructure, such as Workspace or YouTube.

Where to look:
https://cloud.google.com/network-connectivity/docs/carrier-peering

Content mapping:
- Architecting with Google Compute Engine (ILT)
  - M8 Interconnecting Networks

- Elastic Google Cloud Infrastructure: Scaling and Automation (On-demand)
  - M1 Interconnecting Networks

Summary:
Carrier Peering allows you to connect to Google Workspace and YouTube without meeting the peering requirements yourself. With Carrier Peering, you can use a Google partner to provision dedicated bandwidth to connect to the public

infrastructure.

| Diagnostic Question 07 Discussion

Cymbal Direct is evaluating database options to store the **analytics data** from its experimental drone deliveries. You're currently using a small cluster of MongoDB NoSQL database servers. You want to move to a **managed NoSQL database service with consistent low latency that can scale throughput seamlessly and can handle the petabytes of data** you expect after expanding to additional markets.

What should you do?

A. Extract the data from MongoDB. Insert the data into **Firestore** using Datastore mode.

B. Create a **Bigtable** instance, extract the data from MongoDB, and insert the data into Bigtable.

C. Extract the data from MongoDB. Insert the data into **Firestore** using Native mode.

D. Extract the data from MongoDB, and insert the data into **BigQuery**.

Google Cloud

Feedback:
A. Incorrect. Firestore does not meet the requirements for consistent low latency, scaling throughput seamlessly, and petabyte-scaling data.
B. Correct! Bigtable is ideal for IoT, gives consistently sub-10ms latency, and can be used at a petabyte scale.
C. Incorrect. Firestore does not meet the requirements for consistent low latency, scaling throughput seamlessly, and petabyte-scale.
D. Incorrect. BigQuery is used for Enterprise data warehouse and building reports and extracting insights. Bigtable meets the requirements for consistent low latency, scaling throughput seamlessly, and petabyte-scale.

Where to look:
https://cloud.google.com/bigtable/

Content mapping:
- 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:
Bigtable is often used as a solution when working with IoT and analytics because it's very good at storing heavy read/write data and scaling while maintaining linear performance. Bigtable is the basis for several Google products because it scales into

the petabyte range.

You are working with a client who is using **Google Kubernetes Engine (GKE)** to migrate applications from a virtual machine–based environment to a microservices-based architecture. Your client has a **complex legacy application that stores a significant amount of data on the file system of its VM.** You do not want to re-write the application to use an external service to store the file system data.

What should you do?

A. In Cloud Shell, create a YAML file defining your **Deployment** called deployment.yaml. Create a Deployment in GKE by running the command kubectl apply -f deployment.yaml

B. In Cloud Shell, create a YAML file defining your **Container** called build.yaml. Create a Container in GKE by running the command gcloud builds submit –config build.yaml .

C. In Cloud Shell, create a YAML file defining your **StatefulSet** called statefulset.yaml. Create a StatefulSet in GKE by running the command kubectl apply -f statefulset.yaml

D. In Cloud Shell, create a YAML file defining your **Pod** called pod.yaml. Create a Pod in GKE by running the command kubectl apply -f pod.yaml

Google Cloud

Feedback:
A. Incorrect. A Deployment represents multiple identical Pods. A Deployment is used to ensure that Pods are available, but it does not attempt to maintain state. Containers are considered disposable. The kubectl command can be used from gcloud to create or modify deployments defined in a YAML file or manifest.
B. Incorrect. A container must be run within a pod. Cloud Build is used to build a container image using either a YAML file or a Dockerfile. The container can then be run in Kubernetes by referencing that image.
C. Correct! A StatefulSet represents a group of persistent Pods. The YAML file will define a PersistentVolumeClaim (PVC) that allows for an application to retain state. A StatefulSet is commonly used with applications like databases.
D. incorrect. A Pod is the smallest unit of Deployment and contains one or more containers. A Pod doesn't define how many containers must be running, so they're generally managed by the Deployment or StatefulSet.

Where to look:
https://cloud.google.com/kubernetes-engine/docs/concepts/pod

Content mapping:
Getting Started with Google Kubernetes Engine, M3

Summary:
As a Cloud Architect, you need to understand the relationships between a Deployment, Container, StatefulSet, and Pod. A Container must run in an

environment, and that environment is the Pod. A Deployment is multiple instances of a Pod, and a StatefulSet ensures that the Pods will persist when state information must be retained. Most containers externalize state so that they can be destroyed and/recreated on demand. A StatefulSet ensures the containers persist so they can maintain state. Although as a Professional Cloud Architect it generally won't be your responsibility to manage Kubernetes, you should be familiar with the basic Kubernetes commands and concepts.

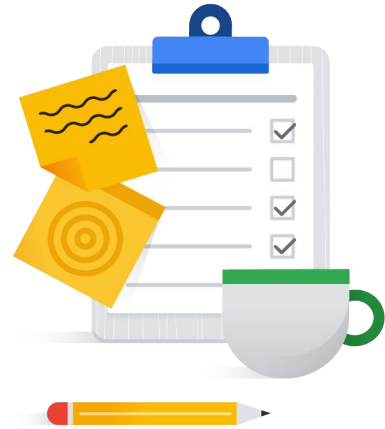## 1.3 | Designing network, storage, and compute resources

Resources to start your journey

Choose and manage compute | Architecture Framework | Google Cloud

Design your network infrastructure | Architecture Framework | Google Cloud

Select and implement a storage strategy | Architecture Framework | Google Cloud

Google Cloud documentation

Google Cloud

The diagnostic questions you just took addressed some of the many network, storage, and compute resources available in Google Cloud. You can learn about these products and services in more detail in the other courses in the Professional Cloud Architect learning path. You can also explore the options in Google Cloud's documentation. These are some links to get you started. You'll find this list in your workbook.

https://cloud.google.com/architecture/framework/system-design/compute
https://cloud.google.com/architecture/framework/system-design/networking
https://cloud.google.com/architecture/framework/system-design/storage
https://cloud.google.com/docs

1.4 | Creating a migration plan

Considerations include:

- Integrating solutions with existing systems
- Migrating systems and data to support the solution
- Software license mapping
- Network planning
- Testing and proofs of concept
- Dependency management planning

The world is still very much in transition from traditional on-premises environments to cloud-based environments. Additionally, more and more people are migrating to Google Cloud from other vendors, or are adopting a multi-cloud approach. As a Professional Cloud Architect, you need to understand how to create and implement a migration plan, and the considerations involved.

While this is not a complete list of all of the things you need to consider, these are a good starting point. You would want to consider this list when planning a migration, or how you would apply these considerations to the case studies, and options you have to address them in Google Cloud.

Question 9 asked you to evaluate approaches for migrating legacy applications.

You are working in a mixed environment of VMs and Kubernetes. **Some of your resources are on-premises, and some are in Google Cloud.** Using containers as a part of your CI/CD pipeline has sped up releases significantly. You want to start **migrating some of those VMs to containers** so you can get similar benefits. You want to **automate the migration process** where possible.

What should you do?

A. **Manually create a GKE cluster**, and then use **Migrate to Containers (Migrate for Anthos)** to set up the cluster, import VMs, and convert them to containers.

B. Use **Migrate to Containers (Migrate for Anthos)** to automate the creation of **Compute Engine instances** to import VMs and convert them to containers.

C. **Manually create a GKE cluster.** Use **Cloud Build** to import VMs and convert them to containers.

D. Use **Migrate for Compute Engine** to import VMs and convert them to containers.

Google Cloud

Feedback:
A. Correct. You must initially create a GKE cluster. Then you can use Migrate to Containers (Migrate for Anthos) to set up the cluster and import the VMs.
B. Incorrect. Migrate to Containers (Migrate for Anthos) uses containers in GKE to migrate the VMs; It does not use Compute Engine instances.
C. Cloud Build lets you build Docker-compatible containers, but you can't use it to automate the importing of VMs.
D. Incorrect. Use Migrate for Compute Engine to import VMs and create Compute Engine instances. You can't use Migrate for Compute Engine to create containers.

Where to look:
https://cloud.google.com/migrate/containers

Content mapping:
- Getting Started with Google Kubernetes Engine (ILT and On-demand)
  - M3 Kubernetes Architecture

Summary:
Migrate for Anthos is a very powerful tool that can be used to import VMs to GKE. It automates the configuration of a GKE cluster and the importing of the VMs. Migrate for Compute Engine is also a useful tool, but instead converts VMs to Compute Engine instances.

Creating a migration plan

Resources to start your journey

Migrate to Containers | Google Cloud
Migration to Google Cloud: Choosing your migration path
Migrating to the cloud: a guide and checklist
Cloud Migration Products & Services
Application Migration | Google Cloud

Google Cloud

Creating a migration plan involves many considerations. Remember, the diagnostic question we just reviewed only covers one scenario. Here are some links to resources that can help you get started learning about migration plans. You'll find this list in your workbook.

https://cloud.google.com/migrate/containers
https://cloud.google.com/resources/cloud-migration-checklist
https://cloud.google.com/products/cloud-migration
https://cloud.google.com/solutions/application-migration
https://cloud.google.com/architecture/migration

1.5 | Envisioning future solution improvements

Considerations include:

- Cloud and technology improvements
- Evolution of business needs
- Evangelism and advocacy

Another critical part of the Professional Cloud Architect role involves envisioning future solution improvements.

Maybe the environment you're working in is new to the cloud, and you want to keep things simple with a "lift-and-shift" approach, and you can't move to a managed database service right away. You can still highlight the business requirement of protecting that database, highlight the improvement you would get by moving to a managed service, and evangelize for that as a future improvement.

Question 10 tested your knowledge of using best practices to optimize scalability and reliability, which can be one aspect of envisioning improvements.

Cymbal Direct has created a proof of concept for a social integration service that highlights images of its products from social media. **The proof of concept is a monolithic application** running on a single SuSE Linux virtual machine (VM). **The current version requires increasing the VM's CPU and RAM in order to scale.** You would like to **refactor the VM so that you can scale out instead of scaling up.**

What should you do?

A. **Move the existing codebase and VM provisioning scripts to git**, and attach external persistent volumes to the VMs.

B. Make sure that the application declares any **dependent requirements** in a requirements.txt or equivalent statement so that they can be referenced in a startup script. Specify the startup script in a **managed instance group** template, and use an autoscaling policy.

C. Make sure that the application declares any **dependent requirements** in a requirements.txt or equivalent statement so that they can be referenced in a startup script, and **attach external persistent volumes to the VMs**.

D. **Use containers instead of VMs**, and use a **GKE autoscaling deployment.**

Google Cloud

Feedback:
A. Incorrect. Version control allows for change control. Backing services allow for flexibility in design, not concurrency.
B. Incorrect. Concurrency will help, but dependencies refer to declaring and isolating code dependencies.
C. Incorrect. Backing services allow for flexibility in design, but not concurrency. Dependencies refer to declaring and isolating code dependencies.
D. Correct! Treating each app as one or more stateless processes means externalizing state to a separate database service. This allows for more concurrent processing.

Where to look:
https://cloud.google.com/architecture/twelve-factor-app-development-on-gcp

Content mapping:
- Architecting with Google Cloud: Design and Process (ILT)
  - M2 Microservice Design and Architecture

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
  - M2 Microservice Design and Architecture

Summary:
Containers are a common and standard tool used to isolate processes. Using "12 factor" application development best practices can be a useful check when designing

or redesigning applications. Following best practices such as the 12 factors allows for process isolation by externalizing state. This approach lets you use GKE to automatically scale instances based on CPU load.

The diagnostic question we just reviewed covers one scenario where you could envision how to improve a solution in the future. Following the "12 factor" application development best practices can help you design or redesign applications. This link is a good starting point for envisioning future solution improvements.

https://cloud.google.com/architecture/twelve-factor-app-development-on-gcp