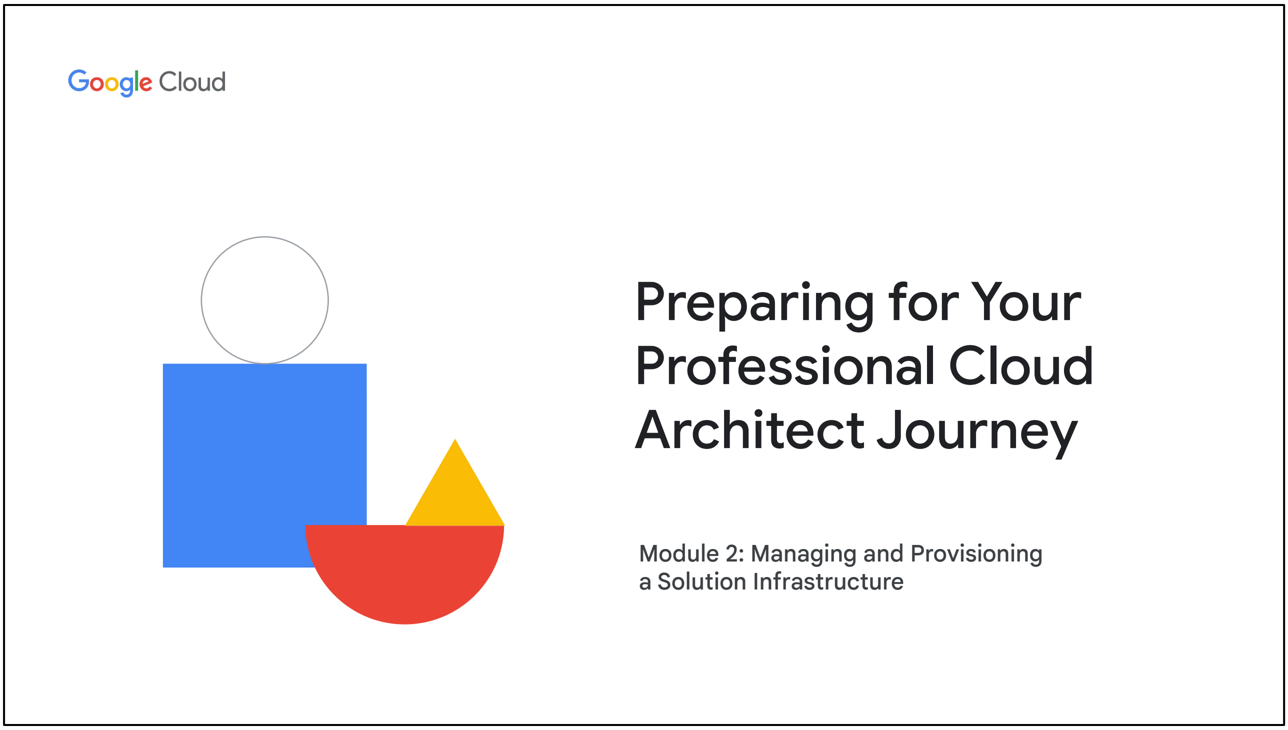
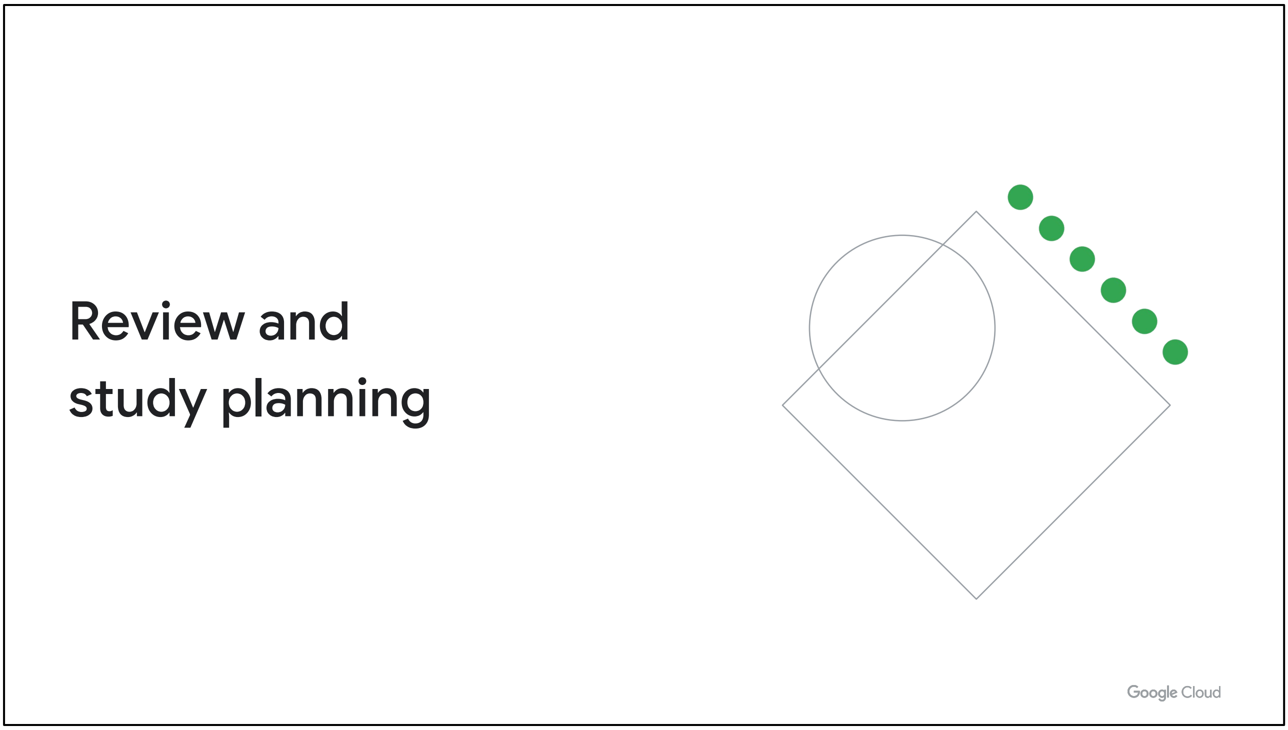


Google Cloud

Preparing for Your Professional Cloud Architect Journey

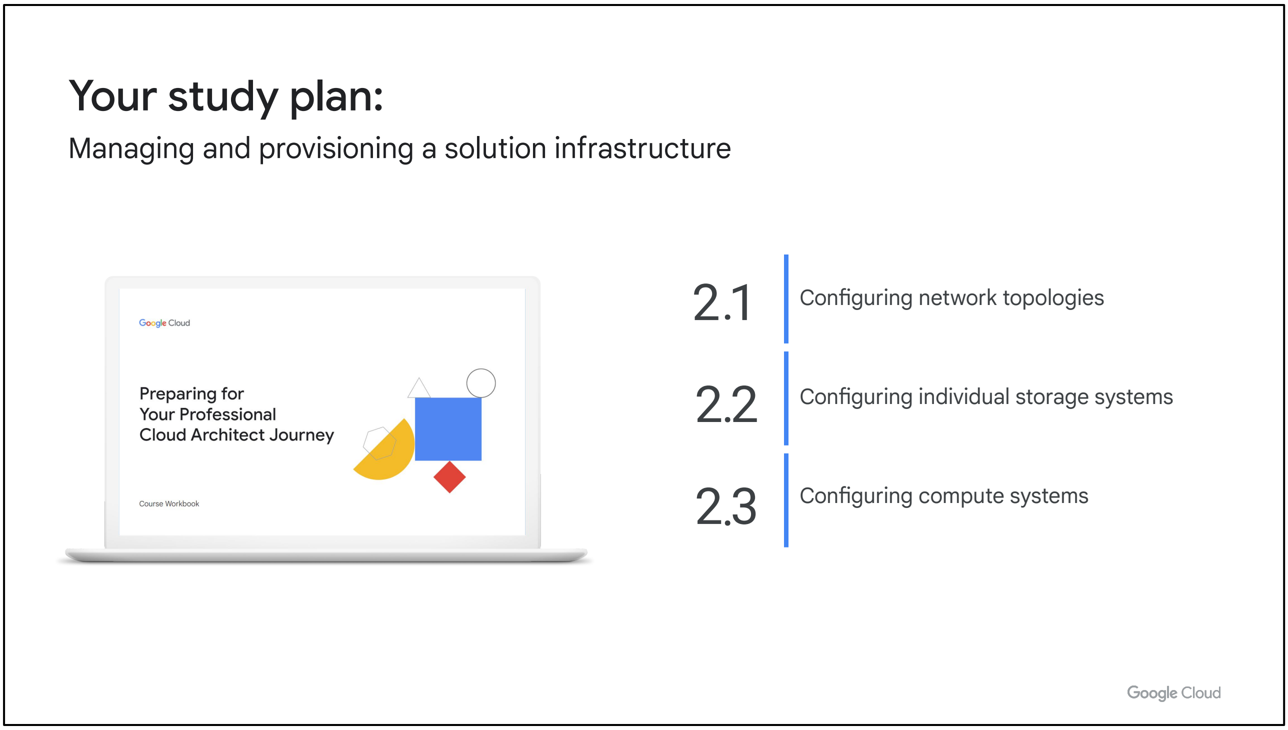
Module 2: Managing and Provisioning
a Solution Infrastructure

Welcome to Module 2: Managing and Provisioning a Solution Infrastructure.



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:

Managing and provisioning a solution infrastructure

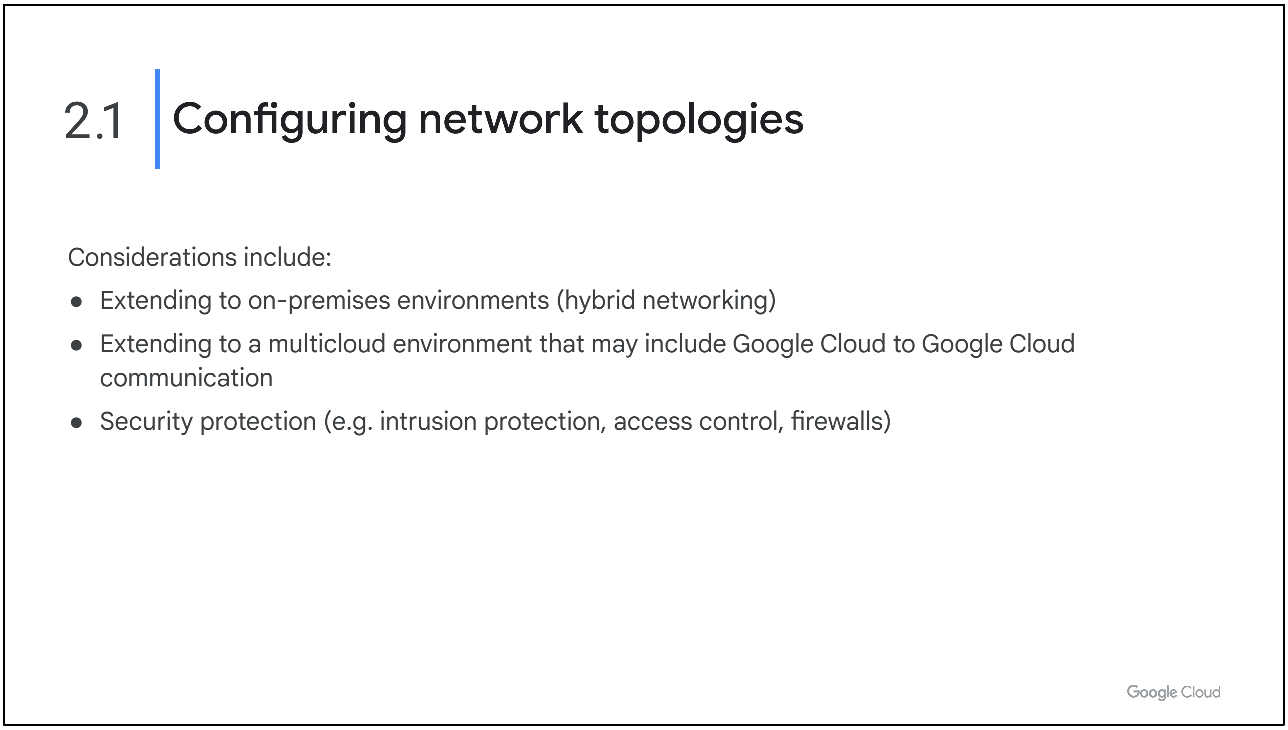
Google Cloud
Preparing for Your Professional Cloud Architect Journey
Course Workbook

2.1 Configuring network topologies

2.2 Configuring individual storage systems

2.3 Configuring compute systems

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.



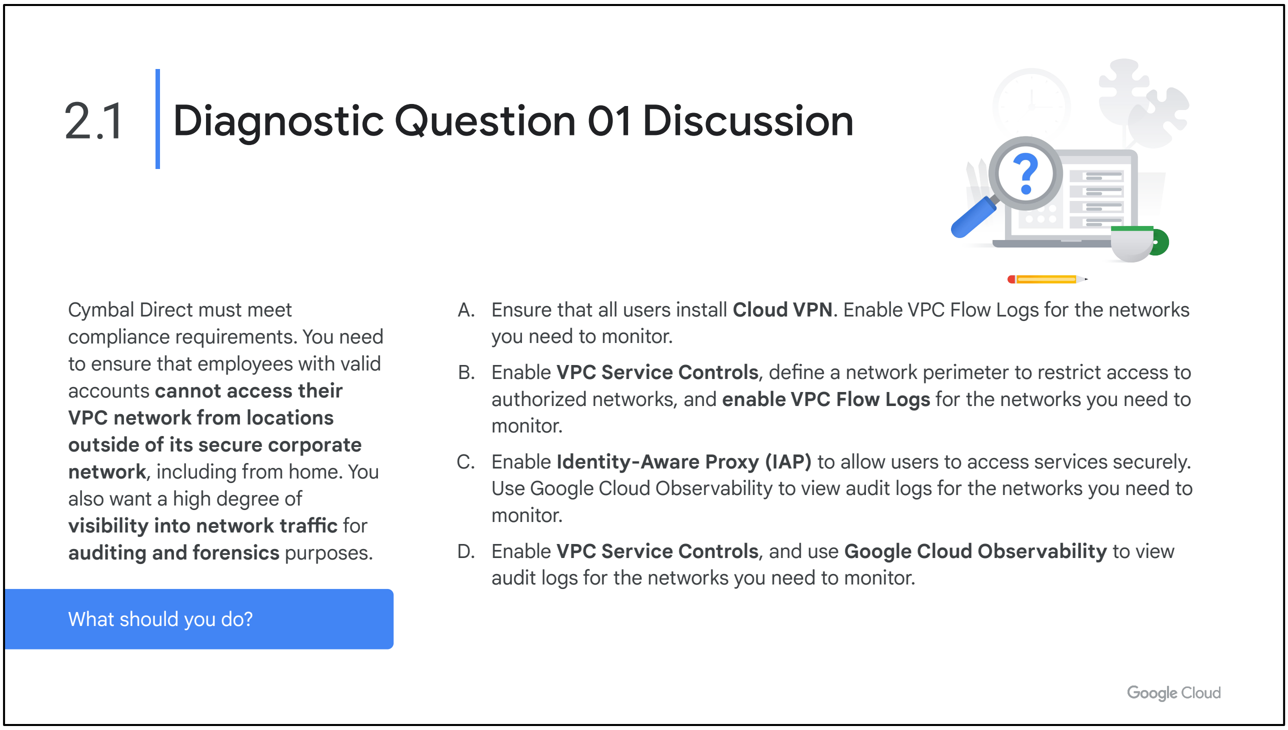
2.1 Configuring network topologies

Considerations include:

- Extending to on-premises environments (hybrid networking)
- Extending to a multicloud environment that may include Google Cloud to Google Cloud communication
- Security protection (e.g. intrusion protection, access control, firewalls)

As a Professional Cloud Architect, you need to make networking decisions that enable secure connections with Google Cloud. Many solutions will need to connect with an on-premises environment in a hybrid approach. Other solutions will extend into a multicloud environment with other networks using Google Cloud or other cloud providers. A Professional Cloud Architect needs a thorough understanding of the various the networking options available with Google Cloud.

Question 1 tested your ability to define compliance requirements for networks using VPC service controls. Question 2 asked you to differentiate between options for load balancers to scale and distribute traffic.



2.1 Diagnostic Question 01 Discussion

Cymbal Direct must meet compliance requirements. You need to ensure that employees with valid accounts **cannot access their VPC network from locations outside of its secure corporate network**, including from home. You also want a high degree of **visibility into network traffic** for **auditing and forensics** purposes.

What should you do?

A. Ensure that all users install **Cloud VPN**. Enable VPC Flow Logs for the networks you need to monitor.

B. Enable **VPC Service Controls**, define a network perimeter to restrict access to authorized networks, and **enable VPC Flow Logs** for the networks you need to monitor.

C. Enable **Identity-Aware Proxy (IAP)** to allow users to access services securely. Use Google Cloud Observability to view audit logs for the networks you need to monitor.

D. Enable **VPC Service Controls**, and use **Google Cloud Observability** to view audit logs for the networks you need to monitor.

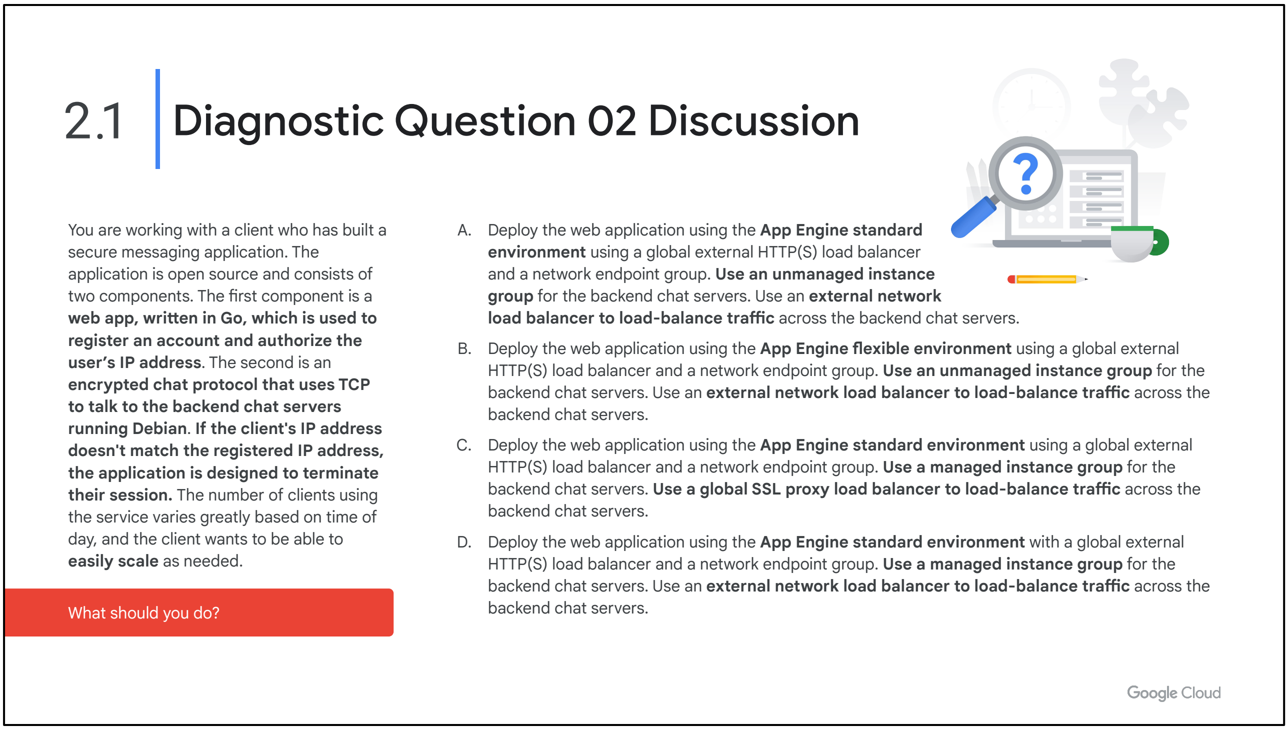
Feedback:
A. Incorrect. Cloud VPN lets a VPN appliance establish a tunnel, but it is not the type of VPN users run directly on their systems.
B. Correct! Enabling VPC Service Controls lets you define a network perimeter. VPC Flow Logs lets you log network-level communication to Compute Engine instances.
C. Incorrect. IAP secures an application by restricting access to valid, authorized accounts. In this scenario, the intention is to restrict access based on where the request is coming from.
D. Incorrect. Enabling VPC Service Controls lets you define a network perimeter. You also need to enable VPC Flow Logs. If you do not enable it, the network traffic flows will not be logged.

Where to look:
https://cloud.google.com/vpc/docs/flow-logs
https://cloud.google.com/vpc-service-controls/

Content mapping:
NA

Summary:
Using VPC Service Controls to enable a network perimeter lets you restrict access to services behind a private endpoint. You can restrict access to specific network ranges. Although Identity-Aware Proxy (IAP) provides secure access for valid accounts, the network perimeter determines where they can access from. Google

Cloud Observability is useful for viewing logs. To enable logging of network traffic, you must first enable VPC Flow Logs.



2.1 Diagnostic Question 02 Discussion

You are working with a client who has built a secure messaging application. The application is open source and consists of two components. The first component is a **web app, written in Go, which is used to register an account and authorize the user's IP address**. The second is an **encrypted chat protocol that uses TCP to talk to the backend chat servers running Debian. If the client's IP address doesn't match the registered IP address, the application is designed to terminate their session.** The number of clients using the service varies greatly based on time of day, and the client wants to be able to **easily scale** as needed.

What should you do?

A. Deploy the web application using the **App Engine standard environment** using a global external HTTP(S) load balancer and a network endpoint group. **Use an unmanaged instance group** for the backend chat servers. Use an **external network load balancer to load-balance traffic** across the backend chat servers.

B. Deploy the web application using the **App Engine flexible environment** using a global external HTTP(S) load balancer and a network endpoint group. **Use an unmanaged instance group** for the backend chat servers. Use an **external network load balancer to load-balance traffic** across the backend chat servers.

C. Deploy the web application using the **App Engine standard environment** using a global external HTTP(S) load balancer and a network endpoint group. **Use a managed instance group** for the backend chat servers. **Use a global SSL proxy load balancer to load-balance traffic** across the backend chat servers.

D. Deploy the web application using the **App Engine standard environment** with a global external HTTP(S) load balancer and a network endpoint group. **Use a managed instance group** for the backend chat servers. Use an **external network load balancer to load-balance traffic** across the backend chat servers.

Google Cloud

Feedback:
A. Incorrect. You should use a managed instance group to scale based on demand.
B. Incorrect. Go is supported in the App Engine standard environment, so there is no need to use the App Engine flexible environment. You should use a managed instance group to scale based on demand.
C. Incorrect. The traffic is already encrypted, so there's no need to offload SSL to the proxy. Additionally, SSL Proxy Load Balancing does not preserve the client's IP address.
D. Correct! Using App Engine allows for dynamic scaling based on demand, as does a managed instance group. Using an external network load balancer preserves the client's IP address.

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

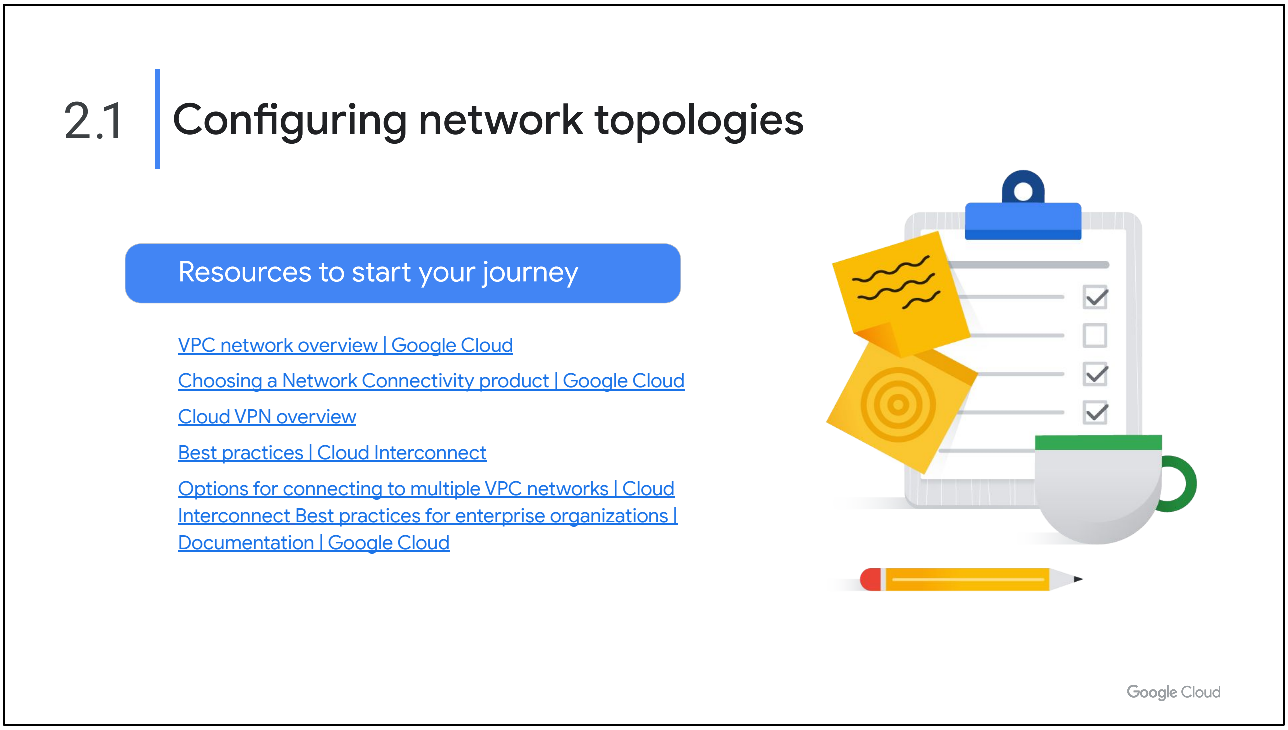
Content mapping:

- Architecting with Google Compute Engine (ILT)
    - M9 Load Balancing and Autoscaling

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

Summary:
Networking and load balancing are key topics for a Professional Cloud Architect. The

services you use in one environment can and will differ, but there will always be networking. To take advantage of working in a cloud environment, you need to be able to distribute traffic across multiple resources. You need to understand what options are available for load balancing and how to choose between them. Whenever you're distributing traffic across multiple resources, you're scaling horizontally. You will probably want to be able to horizontally scale dynamically, so understanding managed instance groups is also critical. There are several serverless options in Google Cloud; you should be familiar with all of them.



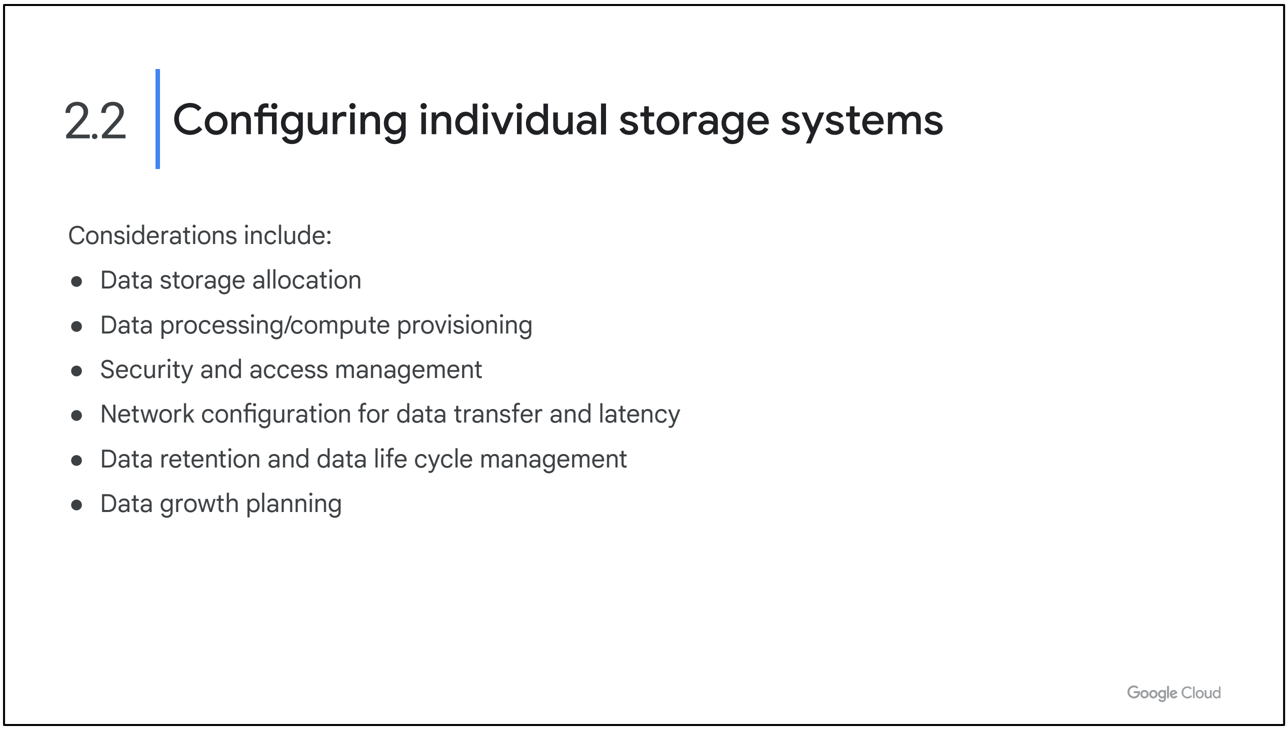
2.1 Configuring network topologies

Resources to start your journey

- VPC network overview | Google Cloud
- Choosing a Network Connectivity product | Google Cloud
- Cloud VPN overview
- Best practices | Cloud Interconnect
- Options for connecting to multiple VPC networks | Cloud Interconnect Best practices for enterprise organizations | Documentation | Google Cloud

The diagnostic questions you reviewed for this section objective asked about a few sample areas related to networking. You should be familiar with best practices for configuring network topologies in Google Cloud and hybrid and multi-cloud environments. These links provide a starting point to learn more. You'll find this list in your workbook.

https://cloud.google.com/vpc/docs/vpc
https://cloud.google.com/network-connectivity/docs/how-to/choose-product
https://cloud.google.com/network-connectivity/docs/vpn/concepts/overview
https://cloud.google.com/network-connectivity/docs/interconnect/concepts/best-practices
https://cloud.google.com/network-connectivity/docs/interconnect/how-to/enabling-multiple-networks-access-same-attachment
https://cloud.google.com/docs/enterprise/best-practices-for-enterprise-organizations



2.2 Configuring individual storage systems

Considerations include:

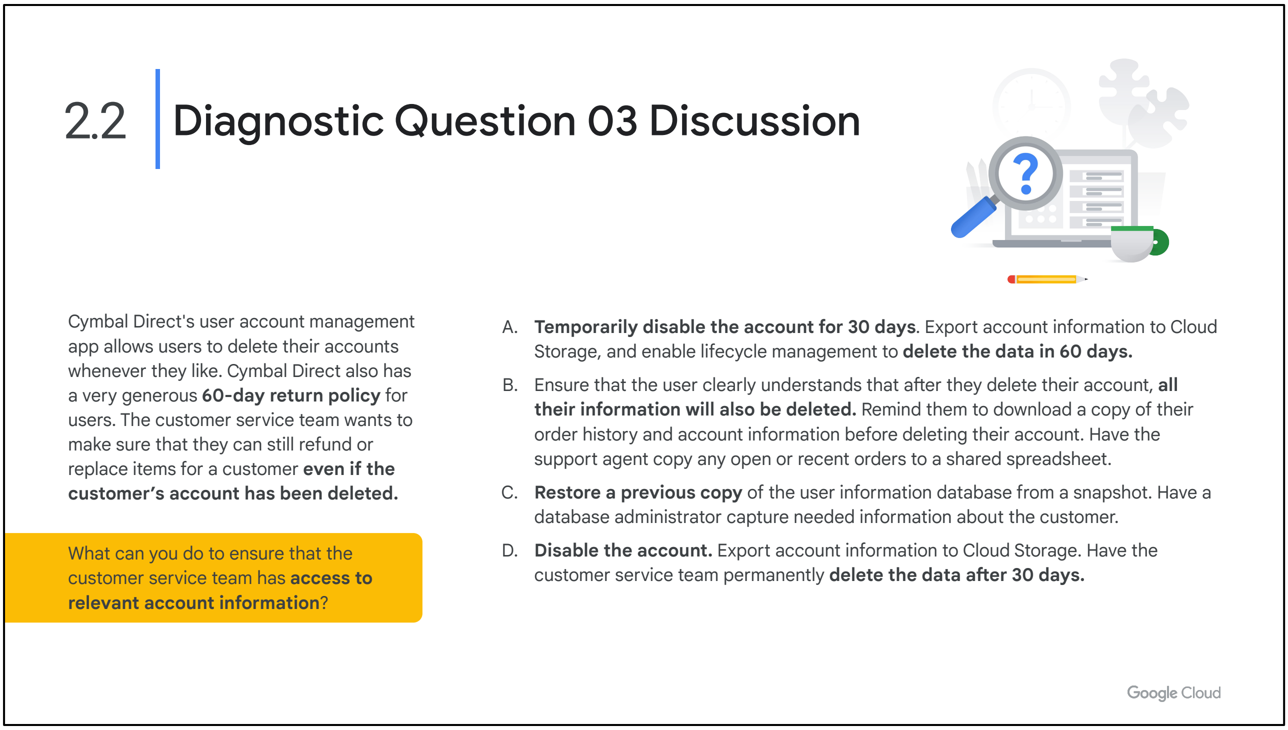
- Data storage allocation
- Data processing/compute provisioning
- Security and access management
- Network configuration for data transfer and latency
- Data retention and data life cycle management
- Data growth planning

Google Cloud

As a Professional Cloud Architect, you need to understand how to ingest, process, and secure data using the options available in Google Cloud.

Data, especially big data, has also historically been one of the most difficult parts of infrastructure to build, design, and maintain. Data is often an area of focus for companies that have strict compliance requirements, such as those that handle sensitive financial or medical data. Leveraging services like Cloud Storage, BigQuery, Spanner, or Bigtable lets you scale infrastructure to monumental sizes without monumental costs.

Question 3 tested your knowledge of implementing a lazy deletion approach to enable resilient data storage.



2.2 Diagnostic Question 03 Discussion

Cymbal Direct's user account management app allows users to delete their accounts whenever they like. Cymbal Direct also has a very generous **60-day return policy** for users. The customer service team wants to make sure that they can still refund or replace items for a customer **even if the customer's account has been deleted.**

What can you do to ensure that the customer service team has **access to relevant account information**?

A. **Temporarily disable the account for 30 days**. Export account information to Cloud Storage, and enable lifecycle management to **delete the data in 60 days.**

B. Ensure that the user clearly understands that after they delete their account, **all their information will also be deleted.** Remind them to download a copy of their order history and account information before deleting their account. Have the support agent copy any open or recent orders to a shared spreadsheet.

C. **Restore a previous copy** of the user information database from a snapshot. Have a database administrator capture needed information about the customer.

D. **Disable the account.** Export account information to Cloud Storage. Have the customer service team permanently **delete the data after 30 days.**

Google Cloud

Feedback:
A. Correct! This takes a lazy deletion approach and allows support or administrators to restore data later if necessary.
B. Incorrect. This doesn't achieve the goal of ensuring that the customer service team has access to the account information.
C. Incorrect. Support agents wouldn't be able to complete this solution, and it would require excessive work by administrators.
D. Incorrect. This will probably introduce human error and would require excessive work.

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

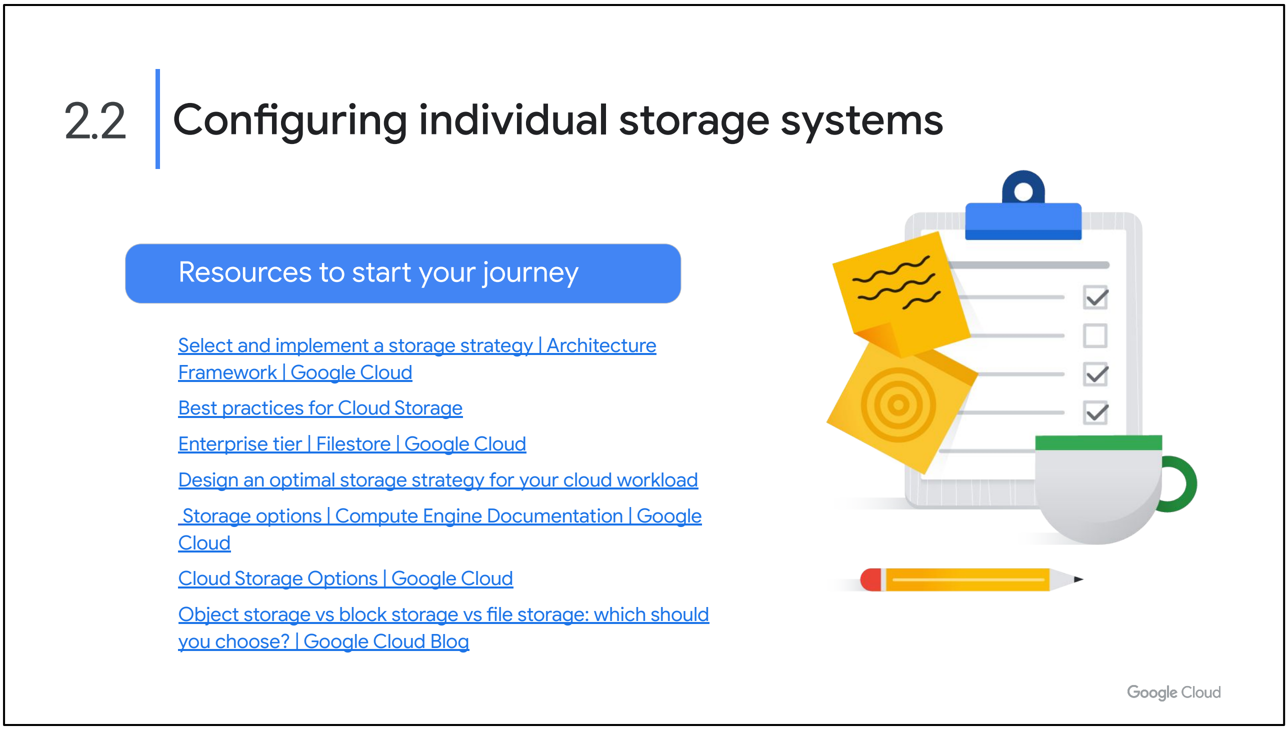
Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
    - M7 Designing Reliable Systems

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
    - M7 Designing Reliable Systems

Summary:
If information might be needed in the future, or if a user might have deleted it by mistake, it's a good idea not to immediately delete it. Instead, take a "lazy deletion" approach and allow for the user to restore the data or for support/administrators to do

it if necessary. How you implement lazy deletion will depend on what kind of storage solution you are using and the information lifecycle related to the data. No matter which method you choose, think ahead to what will happen when and if you need to restore data as you architect a solution. Lazy deletion can be especially useful when you are dealing with compliance or regulatory environments where data must be retained for specific periods of time.



2.2 Configuring individual storage systems

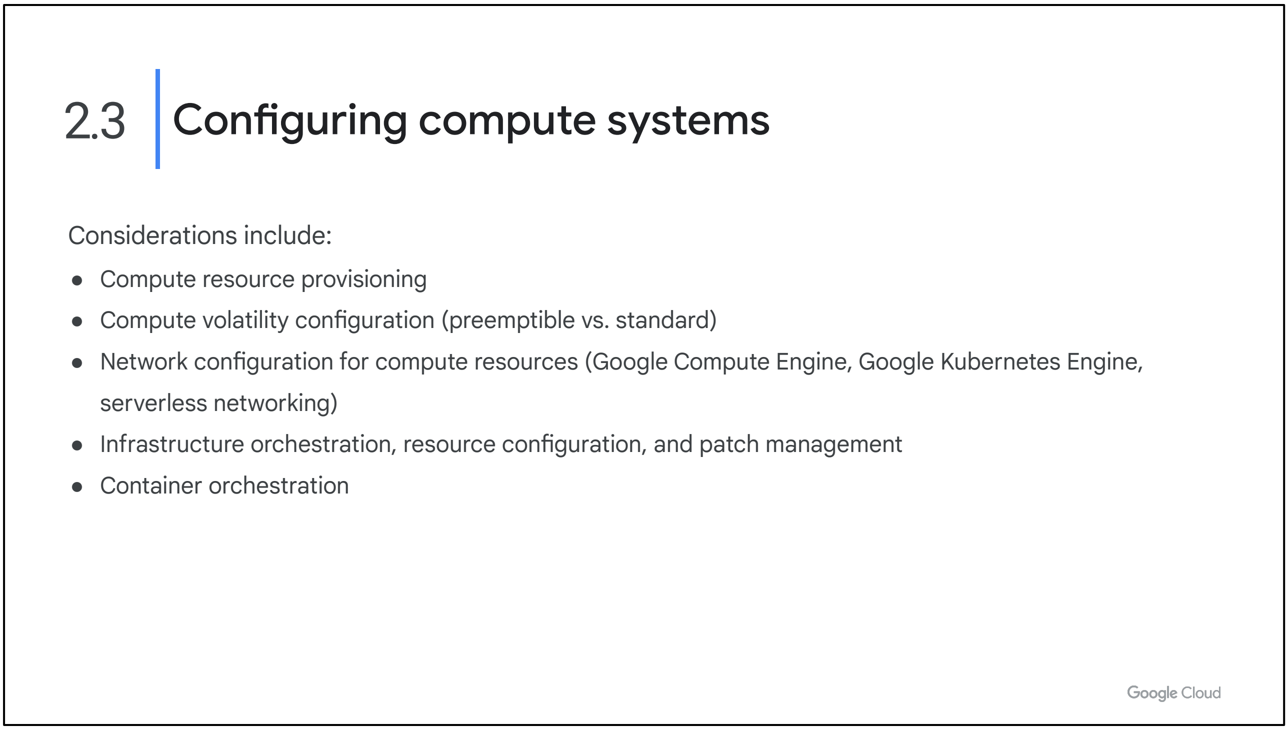
Resources to start your journey

- Select and implement a storage strategy | Architecture Framework | Google Cloud
- Best practices for Cloud Storage
- Enterprise tier | Filestore | Google Cloud
- Design an optimal storage strategy for your cloud workload
- Storage options | Compute Engine Documentation | Google Cloud
- Cloud Storage Options | Google Cloud
- Object storage vs block storage vs file storage: which should you choose? | Google Cloud Blog

Google Cloud

The diagnostic question you reviewed for this section objective addressed just one aspect of configuring individual storage systems - you should be familiar with how to use the many services available to ingest, process, manage, and secure data in Google Cloud. These links provide a starting point. You'll find this list in your workbook.

https://cloud.google.com/architecture/framework/system-design/storage
https://cloud.google.com/storage/docs/best-practices
https://cloud.google.com/filestore/docs/enterprise
https://cloud.google.com/architecture/storage-advisor
https://cloud.google.com/compute/docs/disks
https://cloud.google.com/products/storage
https://cloud.google.com/blog/topics/developers-practitioners/map-storage-options-google-cloud



2.3 Configuring compute systems

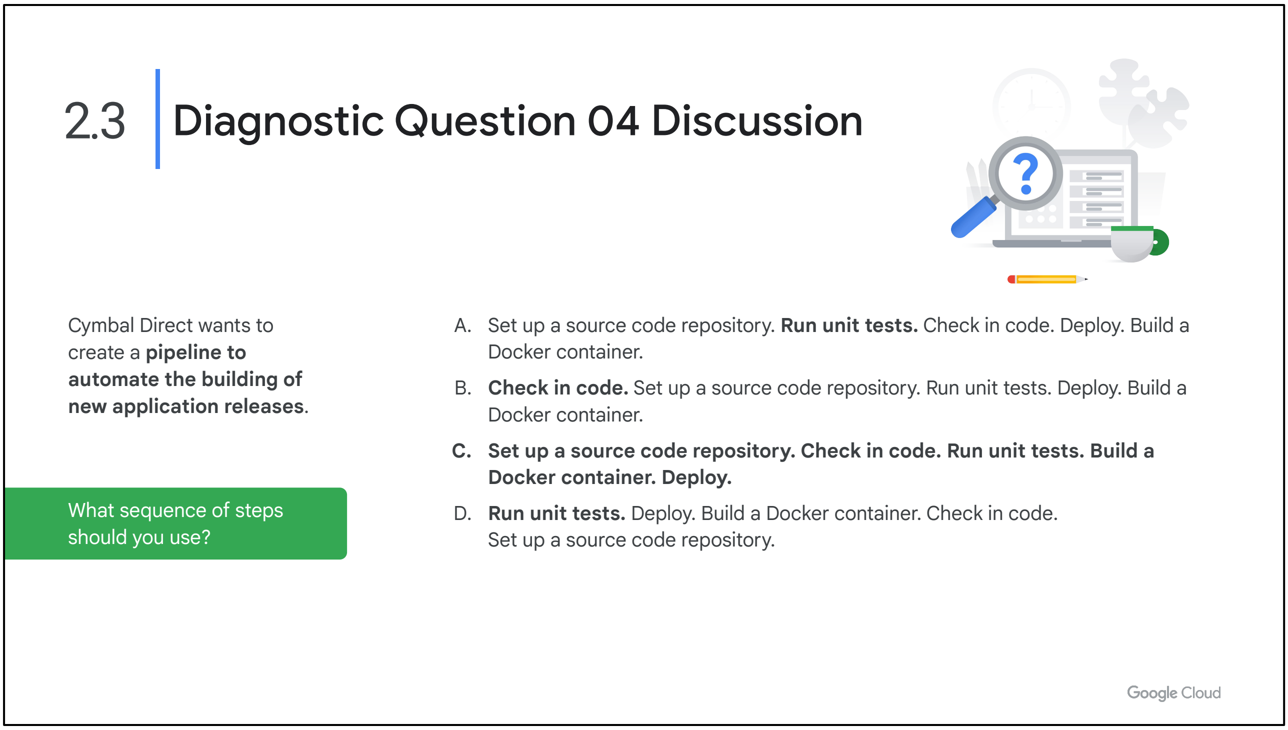
Considerations include:

- Compute resource provisioning
- Compute volatility configuration (preemptible vs. standard)
- Network configuration for compute resources (Google Compute Engine, Google Kubernetes Engine, serverless networking)
- Infrastructure orchestration, resource configuration, and patch management
- Container orchestration

Google Cloud

A Professional Cloud Architect needs to make choices about compute resources and how to configure them for a cloud solution. You should be familiar with the most common tools used in Google Cloud to provision resources, such as Terraform, gcloud, the Google Cloud console, and kubectl. You also need to understand how to properly ensure your compute resources function as intended by designing a network that exposes them securely and restricts access where appropriate.

Question 4 tested your knowledge of the steps to automate builds with Cloud Build and build triggers. Question 5 asked you to automate the deployment of Google Cloud services using Terraform. Question 6 tested your knowledge of creating infrastructure as code using Terraform. Question 7 tested your knowledge of building loosely coupled services and protocol for REST APIs. Question 8 asked you to select options to create and customize secure VM instances in Compute Engine. Question 9 tested your knowledge of configuring your environment for GKE deployments. Question 10 tested your knowledge of the options to configure load balancing.



2.3 Diagnostic Question 04 Discussion

Cymbal Direct wants to create a **pipeline to automate the building of new application releases**.

What sequence of steps should you use?

A. Set up a source code repository. **Run unit tests.** Check in code. Deploy. Build a Docker container.

B. **Check in code.** Set up a source code repository. Run unit tests. Deploy. Build a Docker container.

C. Set up a source code repository. Check in code. Run unit tests. Build a Docker container. Deploy.

D. **Run unit tests.** Deploy. Build a Docker container. Check in code. Set up a source code repository.

Google Cloud

Feedback:
A. Incorrect. Unit tests can't be run unless the code has been checked in for testing.
B. Incorrect. The source code repository must exist to check code in and do any subsequent steps.
C. Correct! Each step is dependent on the previous step. These are in the right order.
D. Incorrect. The source code repository must exist to check code in and do any subsequent steps.

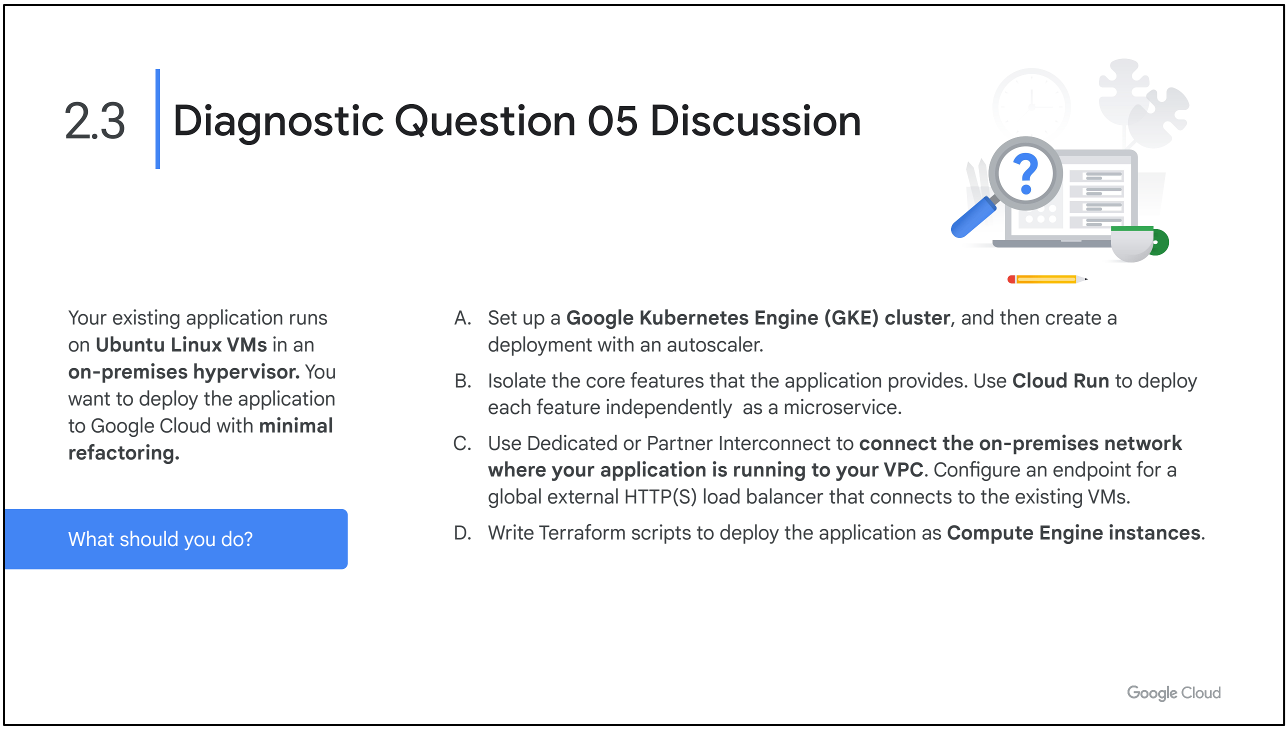
Where to look:
https://cloud.google.com/build/docs/

Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
  - M3 DevOps Automation

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
  - M3 DevOps Automation

Summary:
This sequence of steps represents a simple pipeline and could be substantially more complex, depending on the required tasks. To check in code, you must have a source code repository. Next, developers check in the code. Unit tests can be run to determine whether the build should execute. If all tests pass, the Docker image is then built and finally deployed.



2.3 Diagnostic Question 05 Discussion

Your existing application runs on **Ubuntu Linux VMs** in an **on-premises hypervisor.** You want to deploy the application to Google Cloud with **minimal refactoring.**

What should you do?

A. Set up a **Google Kubernetes Engine (GKE) cluster**, and then create a deployment with an autoscaler.

B. Isolate the core features that the application provides. Use **Cloud Run** to deploy each feature independently as a microservice.

C. Use Dedicated or Partner Interconnect to **connect the on-premises network where your application is running to your VPC**. Configure an endpoint for a global external HTTP(S) load balancer that connects to the existing VMs.

D. Write Terraform scripts to deploy the application as **Compute Engine instances**.

Feedback:
A. Incorrect. Changing from a virtual machine–based application deployment to a container-based deployment will probably likely require refactoring.
B. Incorrect. Changing from a virtual machine–based application deployment to Cloud Run will probably require refactoring.
C. Incorrect. This approach would allow you to leverage Google Cloud's load balancers, but would not be deploying to Google Cloud.
D. Correct! Terraform lets you manage how you deploy and manage a variety of services in Google Cloud, such as Compute Engine.

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

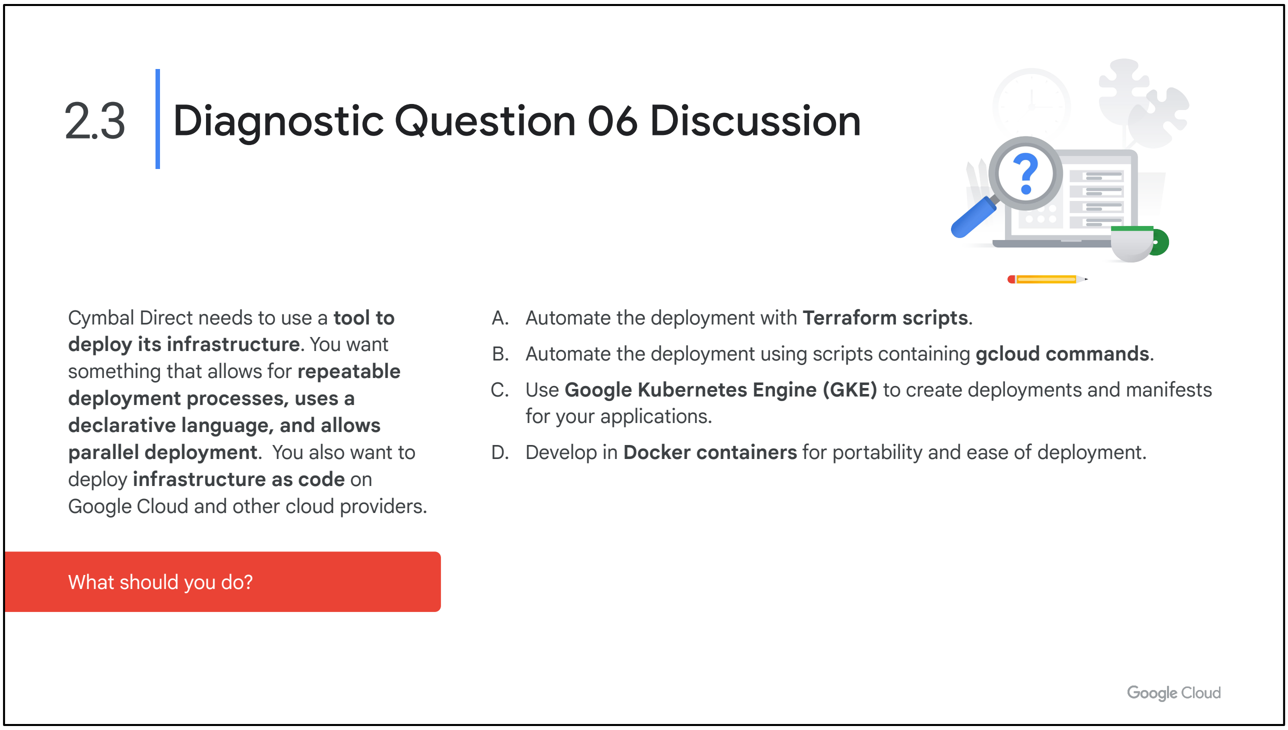
Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
  - M6 Deploying Applications to Google Cloud

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
  - M6 Deploying Applications to Google Cloud

Summary:
Although all of these are good ways to deploy, or expose, an application to Google Cloud, Cloud Run, and GKE will probably require some refactoring to the application. The hybrid network approach will make the application available via the load

balancer, but will not deploy it to Google Cloud. Because the application is already a virtual machine, migrating to Compute Engine with Terraform will use a lift-and-shift approach.



2.3 Diagnostic Question 06 Discussion

Cymbal Direct needs to use a **tool to deploy its infrastructure**. You want something that allows for **repeatable deployment processes, uses a declarative language, and allows parallel deployment**. You also want to deploy **infrastructure as code** on Google Cloud and other cloud providers.

What should you do?

A. Automate the deployment with **Terraform scripts**.

B. Automate the deployment using scripts containing **gcloud commands**.

C. Use **Google Kubernetes Engine (GKE)** to create deployments and manifests for your applications.

D. Develop in **Docker containers** for portability and ease of deployment.

Google Cloud

Feedback:
A. Correct! Terraform lets you automate and manage resources in multiple clouds.
B. Incorrect. Automation using scripts adds unnecessary complexity and does not have the same benefits of modern infrastructure automation tooling.
C. Incorrect. GKE is Google's managed Kubernetes service. Deployments accomplish many of these goals, but only for within Kubernetes. GKE is only available in Google Cloud, not other clouds.
D. Incorrect. Docker (or Docker-compatible) containers make deploying code much easier, but do not manage or orchestrate the process themselves. This is what a tool like Kubernetes is for.

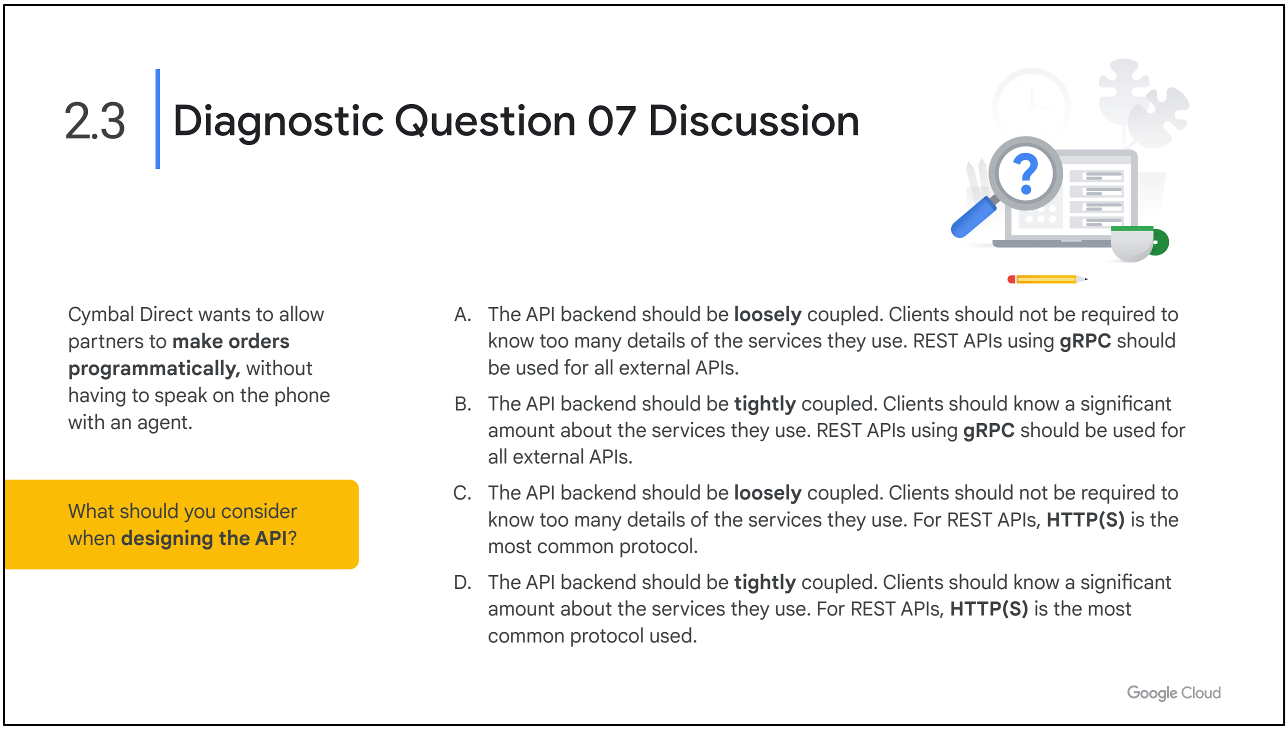
Where to look:
https://cloud.google.com/docs/terraform

Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
    - M3 DevOps Automation

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
    - M3 DevOps Automation

Summary:
Terraform is one of the most used infrastructure automation tools and has good support for multiple cloud providers.



2.3 Diagnostic Question 07 Discussion

Cymbal Direct wants to allow partners to **make orders programmatically,** without having to speak on the phone with an agent.

What should you consider when **designing the API**?

A. The API backend should be **loosely** coupled. Clients should not be required to know too many details of the services they use. REST APIs using **gRPC** should be used for all external APIs.

B. The API backend should be **tightly** coupled. Clients should know a significant amount about the services they use. REST APIs using **gRPC** should be used for all external APIs.

C. The API backend should be **loosely** coupled. Clients should not be required to know too many details of the services they use. For REST APIs, **HTTP(S)** is the most common protocol.

D. The API backend should be **tightly** coupled. Clients should know a significant amount about the services they use. For REST APIs, **HTTP(S)** is the most common protocol used.

Feedback:

A. Incorrect. If clients know extensive information about backend services, backend systems would be difficult to change or replace. REST APIs are protocol-agnostic, and HTTP(S) is the most common protocol for external APIs.

B. Incorrect. If an API is not loosely coupled, it can become an issue for maintenance, with large, complicated monolithic applications. REST APIs are protocol-agnostic, and HTTP(S) is the most common protocol for external APIs.

C. Correct! Loose coupling has several benefits, including maintainability, versioning, and reduced complexity. Clients not knowing the backend systems means that these systems can be more easily replaced or modified, and HTTP(S) is the most common protocol used for external REST APIs.

D. Incorrect. If an API is not loosely coupled, it can become an issue for maintenance, with large, complicated monolithic applications. REST APIs are protocol-agnostic, and HTTP(S) is the most common protocol for external APIs.

Where to look:
https://cloud.google.com/apis/design/

Content mapping:

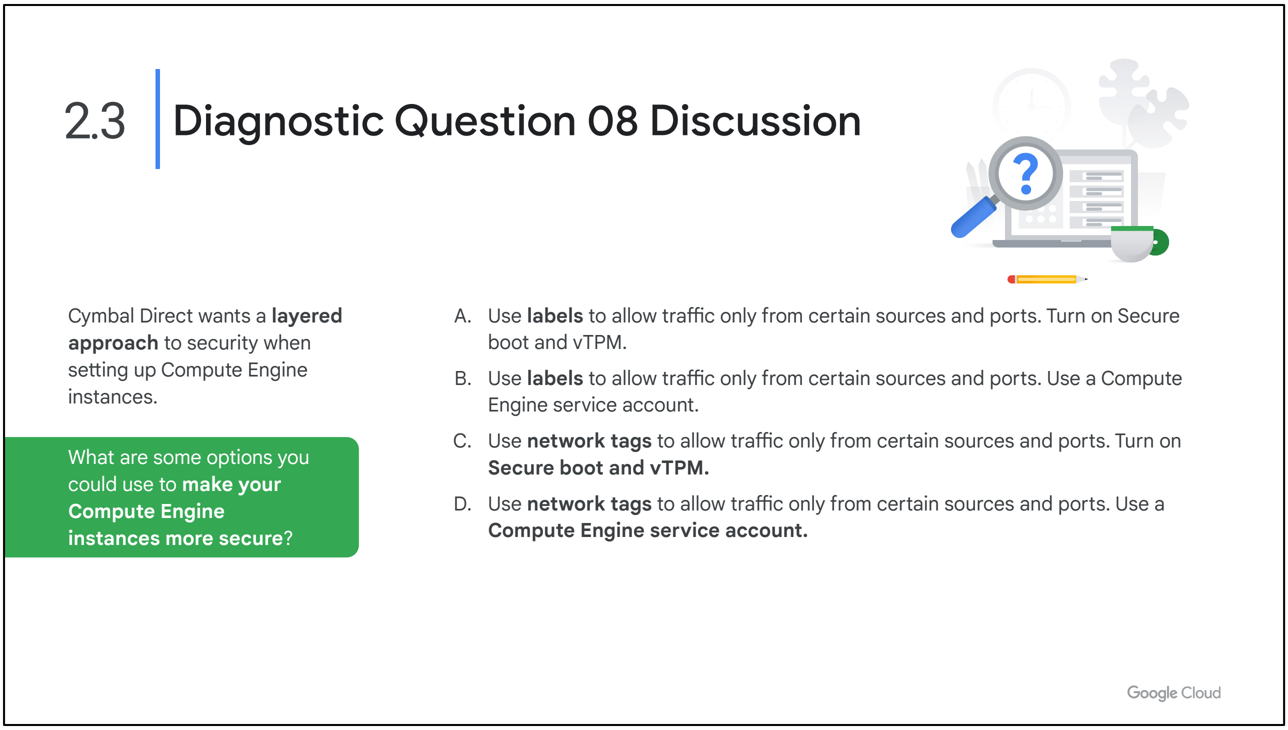
- Architecting with Google Cloud: Design and Process (ILT)
  - M2 Microservice Design and Architecture

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

Summary:
An API is effectively a contract between the API provider and the clients using it. As long as the client makes a request that is valid according to the API's specification, the request will be fulfilled. This is referred to as a loose coupling or a black box approach. A microservice-based architecture means that many independent parts (the microservices) can change. The client shouldn't need to know about the different parts.

When making updates, you need to make sure that your changes don't break older versions of the API specification in use by a client. One way is through a concept called a versioned contract. This common approach specifies which version the client wants to access as part of your API.

Follow the OpenAPI standard to help ensure loose coupling and versioned contracts. Atlassian and Pact offer tools to test API contracts. Although most modern APIs, especially those designed for external use, use HTTP(S) as the transport protocol, that's not a requirement. Many internal APIs at Google use gRPC, but that isn't a requirement. Most modern APIs decouple the transport protocol from the API.



2.3 Diagnostic Question 08 Discussion

Cymbal Direct wants a **layered approach** to security when setting up Compute Engine instances.

What are some options you could use to **make your Compute Engine instances more secure**?

A. Use **labels** to allow traffic only from certain sources and ports. Turn on Secure boot and vTPM.

B. Use **labels** to allow traffic only from certain sources and ports. Use a Compute Engine service account.

C. Use **network tags** to allow traffic only from certain sources and ports. Turn on **Secure boot and vTPM.**

D. Use **network tags** to allow traffic only from certain sources and ports. Use a **Compute Engine service account.**

Google Cloud

Feedback:
A. Incorrect. Labels are often confused with network tags. Tags are used with firewall rules, and labels are used for billing. Secure boot and vTPM protect the OS from being compromised.
B. Incorrect. Labels are often confused with network tags. Tags are used with firewall rules, and labels are used for billing. All Compute Engine instances have an associated service account. Creating an account specifically for an instance or type of instance with limited abilities instead of the default account could be a good approach to the principle of least privilege.
C. Correct! You can use network tags with firewall rules to automatically associate instances when they are created. Secure boot and vTPM protect the OS from being compromised.
D. Incorrect. All Compute Engine instances have an associated service account. Creating an account specifically for an instance or type of instance with limited abilities instead of the default account could be a good approach to the principle of least privilege.

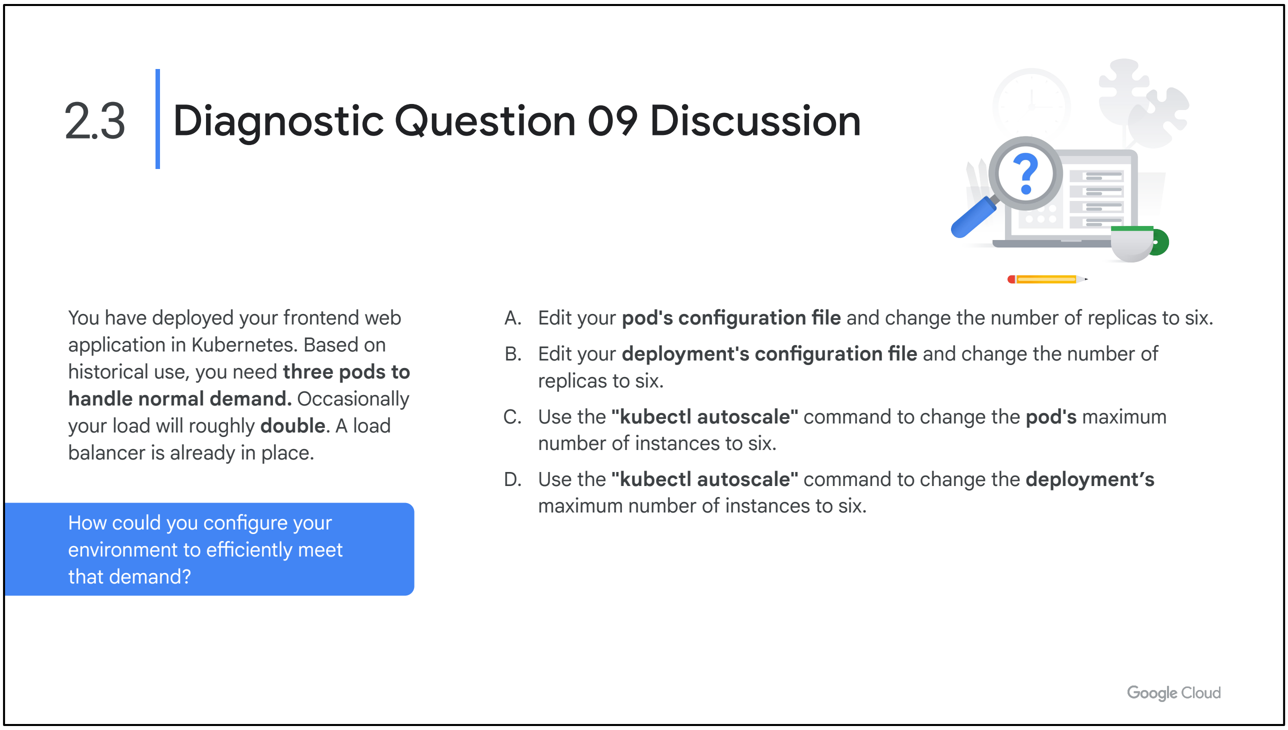
Where to look:
https://cloud.google.com/compute/docs/instances/create-start-instance

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:
You can do many things to make a Compute Engine instance more secure; the options mentioned are just a few of them. Remember that network tags are used for determining firewall rules, and labels are used for categorization and insight (such as tracking spending). Secure boot and vTPM allow for validating the operating system at boot time and are supported by several operating systems, but not all.



2.3 Diagnostic Question 09 Discussion

You have deployed your frontend web application in Kubernetes. Based on historical use, you need **three pods to handle normal demand.** Occasionally your load will roughly **double**. A load balancer is already in place.

How could you configure your environment to efficiently meet that demand?

A. Edit your **pod's configuration file** and change the number of replicas to six.

B. Edit your **deployment's configuration file** and change the number of replicas to six.

C. Use the **"kubectl autoscale"** command to change the **pod's** maximum number of instances to six.

D. Use the **"kubectl autoscale"** command to change the **deployment's** maximum number of instances to six.

Google Cloud

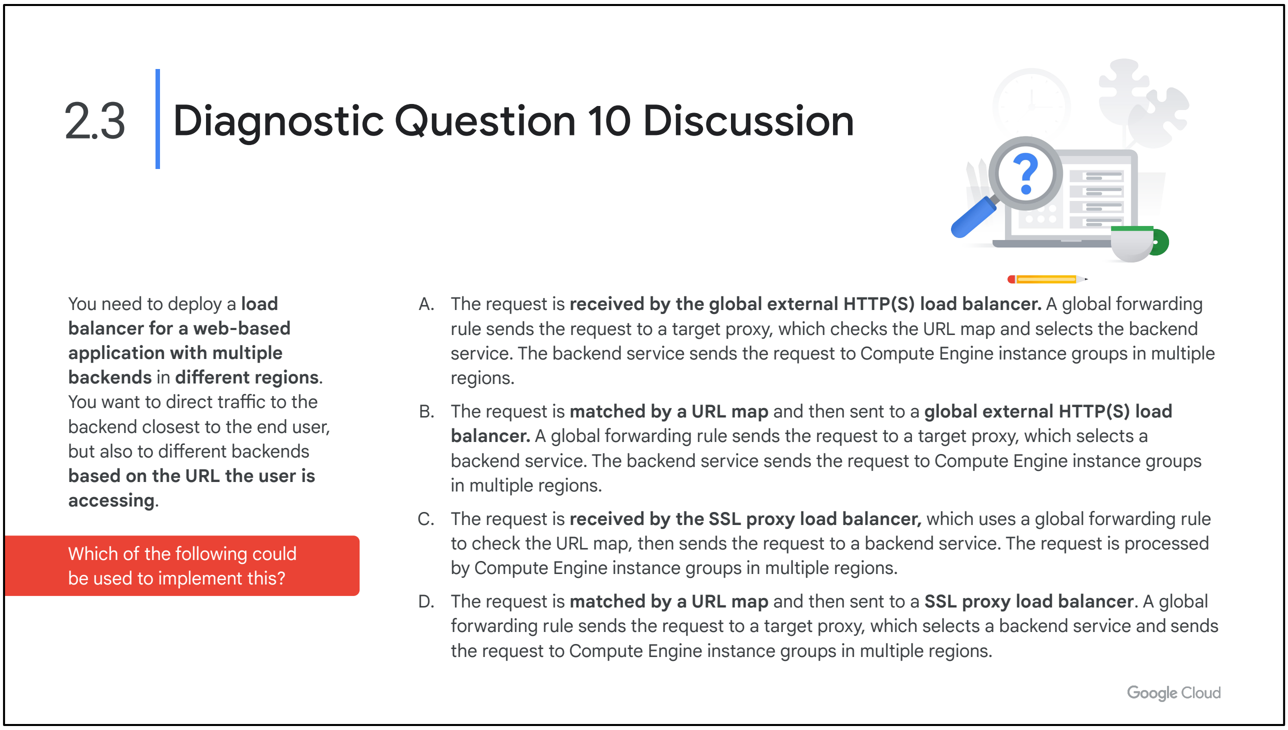
Feedback:
A. Incorrect. A deployment specifies the number of pods, not a pod itself, and setting the number to six means running additional instances when you don't need them.
B. Incorrect. Managing your deployments as code has a lot of benefits, but setting the number to six means running additional instances when you don't need them.
C. Incorrect. A deployment specifies the number of pods, not a pod itself.
D. Correct! This will allow Kubernetes to scale the number of pods automatically, based on a condition like CPU load or requests per second.

Where to look:
https://cloud.google.com/kubernetes-engine/docs/how-to/scaling-apps#autoscaling-deployments

Content mapping:

- Getting Started with Google Kubernetes Engine (ILT and On-demand)
  - M4 Kubernetes Operations

Summary:
As with a Compute Engine managed instance group, you can either specify a fixed number of instances or have GKE autoscale them for you. Autoscaling is generally going to be more efficient because unnecessary pods will not be running when they're not needed.



2.3 Diagnostic Question 10 Discussion

You need to deploy a **load balancer for a web-based application with multiple backends** in **different regions**. You want to direct traffic to the backend closest to the end user, but also to different backends **based on the URL the user is accessing**.

Which of the following could be used to implement this?

A. The request is **received by the global external HTTP(S) load balancer.** A global forwarding rule sends the request to a target proxy, which checks the URL map and selects the backend service. The backend service sends the request to Compute Engine instance groups in multiple regions.

B. The request is **matched by a URL map** and then sent to a **global external HTTP(S) load balancer.** A global forwarding rule sends the request to a target proxy, which selects a backend service. The backend service sends the request to Compute Engine instance groups in multiple regions.

C. The request is **received by the SSL proxy load balancer,** which uses a global forwarding rule to check the URL map, then sends the request to a backend service. The request is processed by Compute Engine instance groups in multiple regions.

D. The request is **matched by a URL map** and then sent to a **SSL proxy load balancer**. A global forwarding rule sends the request to a target proxy, which selects a backend service and sends the request to Compute Engine instance groups in multiple regions.

Google Cloud

Feedback:
A. Correct! This is the right order of operations.
B. Incorrect. The external global HTTP(S) load balancer must exist to provide the multicast IP address, and then route the request through the target proxy.
C. Incorrect. The SSL Proxy is not for HTTP(S) traffic. The question specifically states a web-based application.
D. Incorrect. The SSL Proxy is not for HTTP(S) traffic. The question specifically states a web-based application.

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

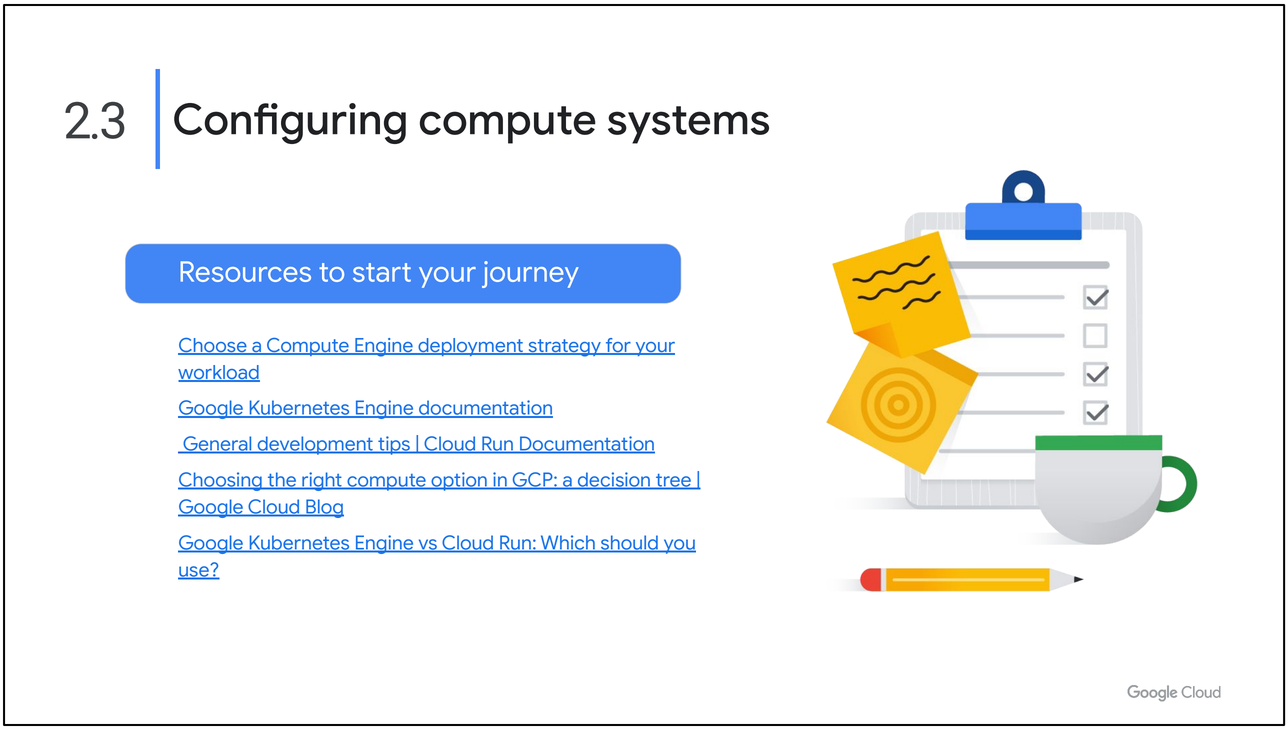
Content mapping:

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

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

Summary:
A request coming from the internet is processed by the HTTP(S) proxy. A URL map is then compared to the request to route it to the appropriate backend service. For example, you could have two backend services: one for serving video and one for service audio. The URL ending in “/audio” could be mapped to the backend serving

audio, and the URL ending in “/video” could be mapped to the backend serving video. Each backend service could have multiple backends, such as instance groups in different regions. The backend the traffic is sent to is determined by health, capacity, and geographic location.



2.3 Configuring compute systems

Resources to start your journey

- Choose a Compute Engine deployment strategy for your workload
- Google Kubernetes Engine documentation
- General development tips | Cloud Run Documentation
- Choosing the right compute option in GCP: a decision tree | Google Cloud Blog
- Google Kubernetes Engine vs Cloud Run: Which should you use?

Google Cloud

You can learn more about configuring compute systems in the courses on the Professional Cloud Architect learning path. You should be familiar with all of the compute options in Google Cloud. Here are some links to get started. You'll find this list in your workbook.

https://cloud.google.com/compute/docs/choose-compute-deployment-option
https://cloud.google.com/kubernetes-engine/docs
https://cloud.google.com/run/docs/tips/general
https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree
https://cloud.google.com/blog/products/containers-kubernetes/when-to-use-google-kubernetes-engine-vs-cloud-run-for-containers