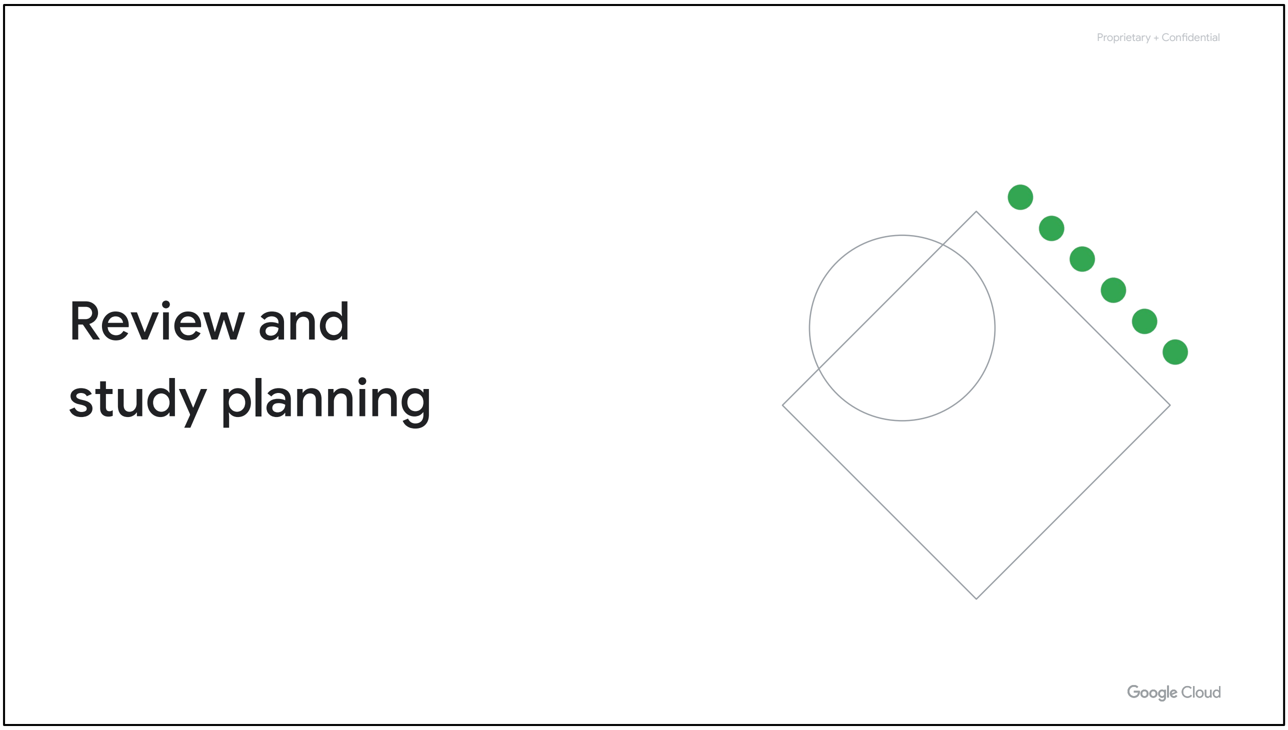
Google Cloud

Preparing for your Professional Cloud Architect Journey

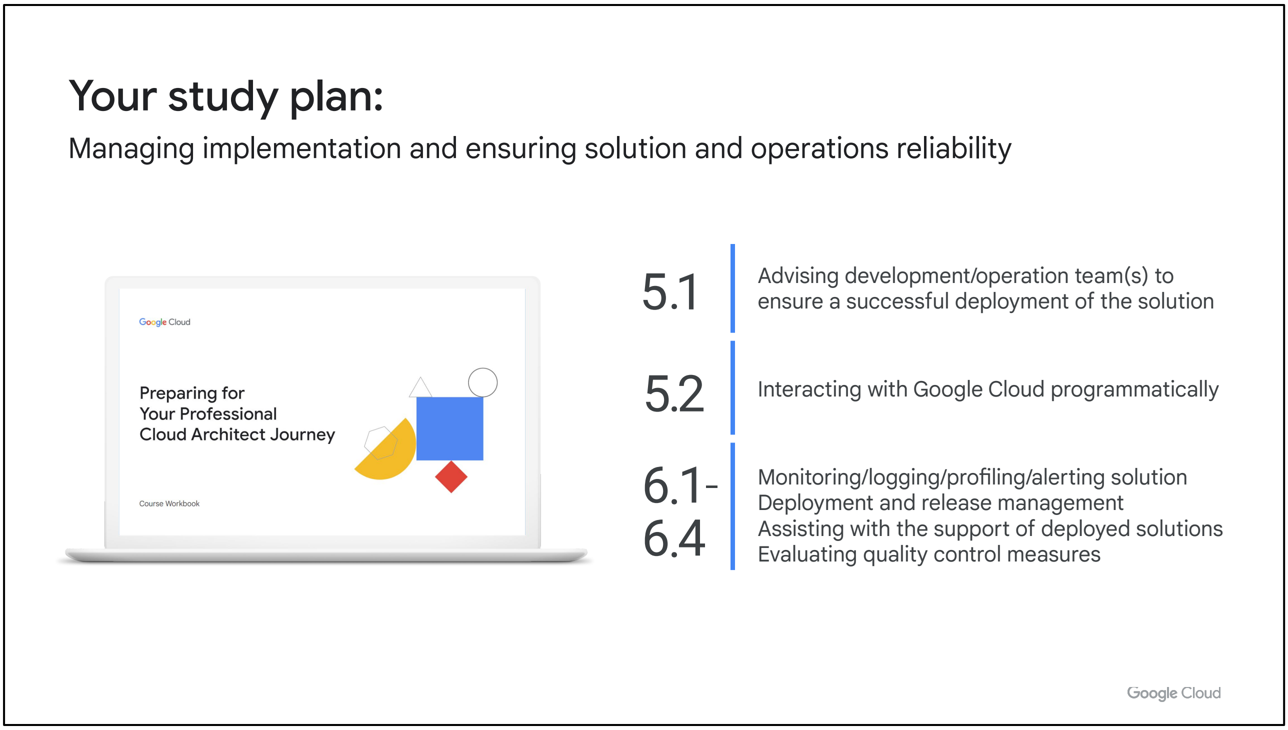
Module 5: Managing Implementation and Ensuring Solution and Operations Reliability

Welcome to Module 5: Managing Implementation and Ensuring Solution and Operations Reliability.



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.

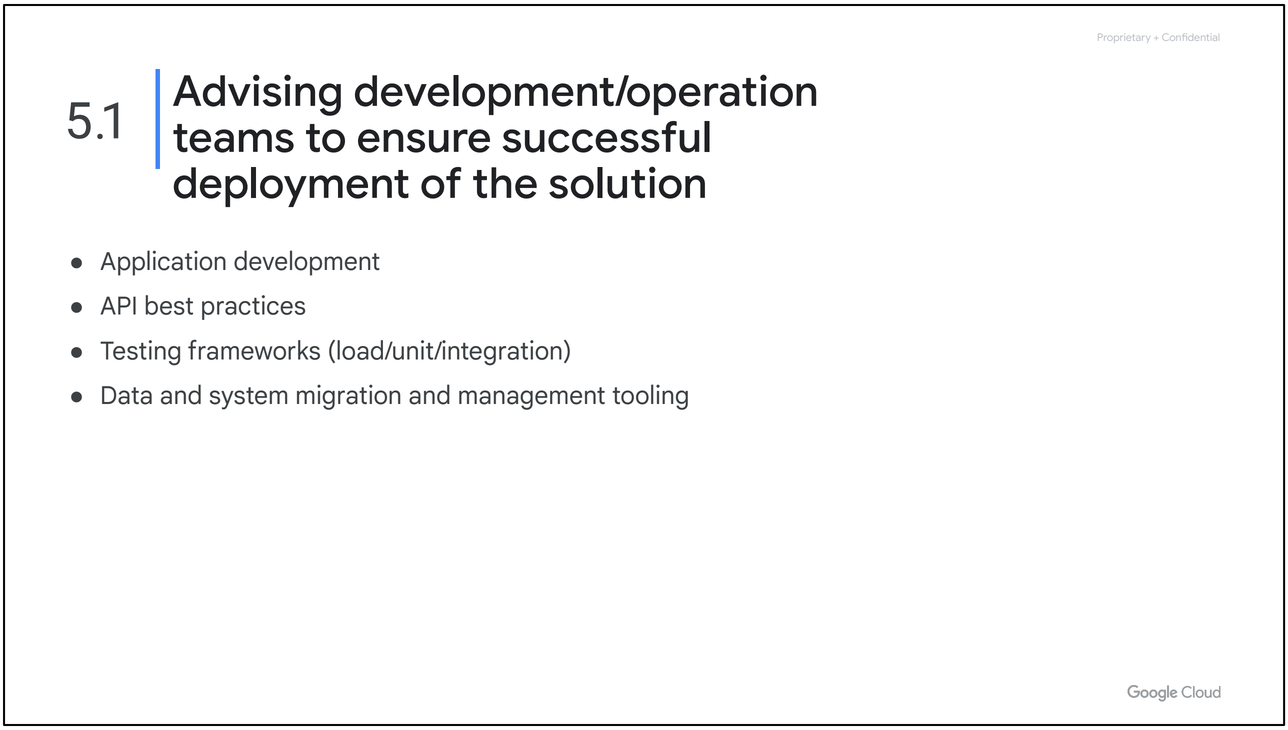

Google Cloud
Preparing for
Your Professional
Cloud Architect Journey
Course Workbook

Your study plan:

Managing implementation and ensuring solution and operations reliability

5.1	Advising development/operation team(s) to ensure a successful deployment of the solution
5.2	Interacting with Google Cloud programmatically
6.1-6.4	Monitoring/logging/profiling/alerting solution Deployment and release management Assisting with the support of deployed solutions Evaluating quality control measures

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.

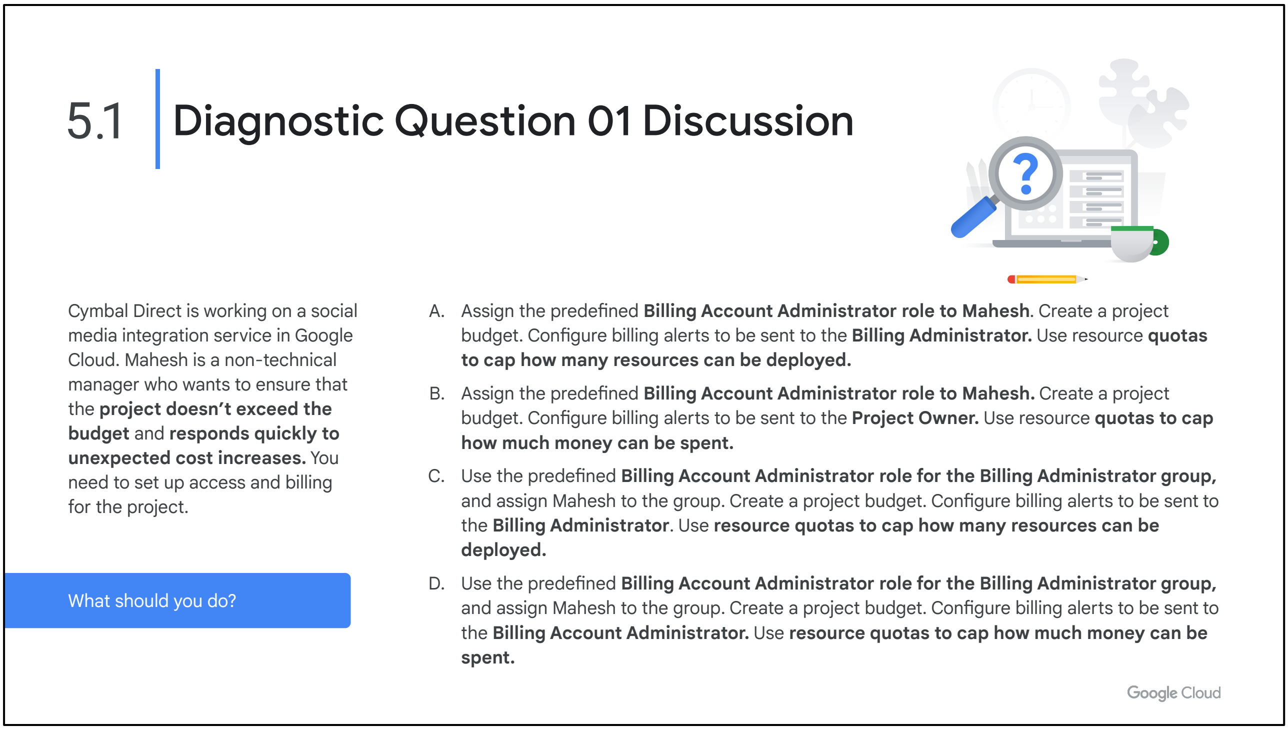


5.1 Advising development/operation teams to ensure successful deployment of the solution

- Application development
- API best practices
- Testing frameworks (load/unit/integration)
- Data and system migration and management tooling

As Professional Cloud Architect, your role doesn’t end with designing and configuring a solution. You are expected to be familiar with how to advise development and operations teams to ensure successful deployment of a solution.

Question 1 tested your knowledge of using quotas, billing, and budget alerts in Google Cloud. Question 2 examined KPIs for a Disaster Recovery strategy.



5.1 Diagnostic Question 01 Discussion

Cymbal Direct is working on a social media integration service in Google Cloud. Mahesh is a non-technical manager who wants to ensure that the **project doesn't exceed the budget** and **responds quickly to unexpected cost increases.** You need to set up access and billing for the project.

What should you do?

A. Assign the predefined **Billing Account Administrator role to Mahesh**. Create a project budget. Configure billing alerts to be sent to the **Billing Administrator.** Use resource **quotas to cap how many resources can be deployed.**

B. Assign the predefined **Billing Account Administrator role to Mahesh.** Create a project budget. Configure billing alerts to be sent to the **Project Owner.** Use resource **quotas to cap how much money can be spent.**

C. Use the predefined **Billing Account Administrator role for the Billing Administrator group,** and assign Mahesh to the group. Create a project budget. Configure billing alerts to be sent to the **Billing Administrator**. Use **resource quotas to cap how many resources can be deployed.**

D. Use the predefined **Billing Account Administrator role for the Billing Administrator group,** and assign Mahesh to the group. Create a project budget. Configure billing alerts to be sent to the **Billing Account Administrator.** Use **resource quotas to cap how much money can be spent.**

Google Cloud

Feedback:

A. Incorrect. Use groups with Identity and Access Management (IAM) to simplify management. Quotas are based on the number of resources, such as instances or CPU, not budget.

B. Incorrect. Use groups with IAM to simplify management. Billing Alerts should be sent to the Billing Administrator.

C. Correct! Use groups with IAM to simplify management. Billing Alerts should be sent to the Billing Administrator. Quotas are based on the number of resources, such as instances or CPU, not budget.

D. Incorrect. Quotas are based on the number of resources, such as instances or CPU, not budget.

Where to look:
https://cloud.google.com/billing/docs/how-to/budgets

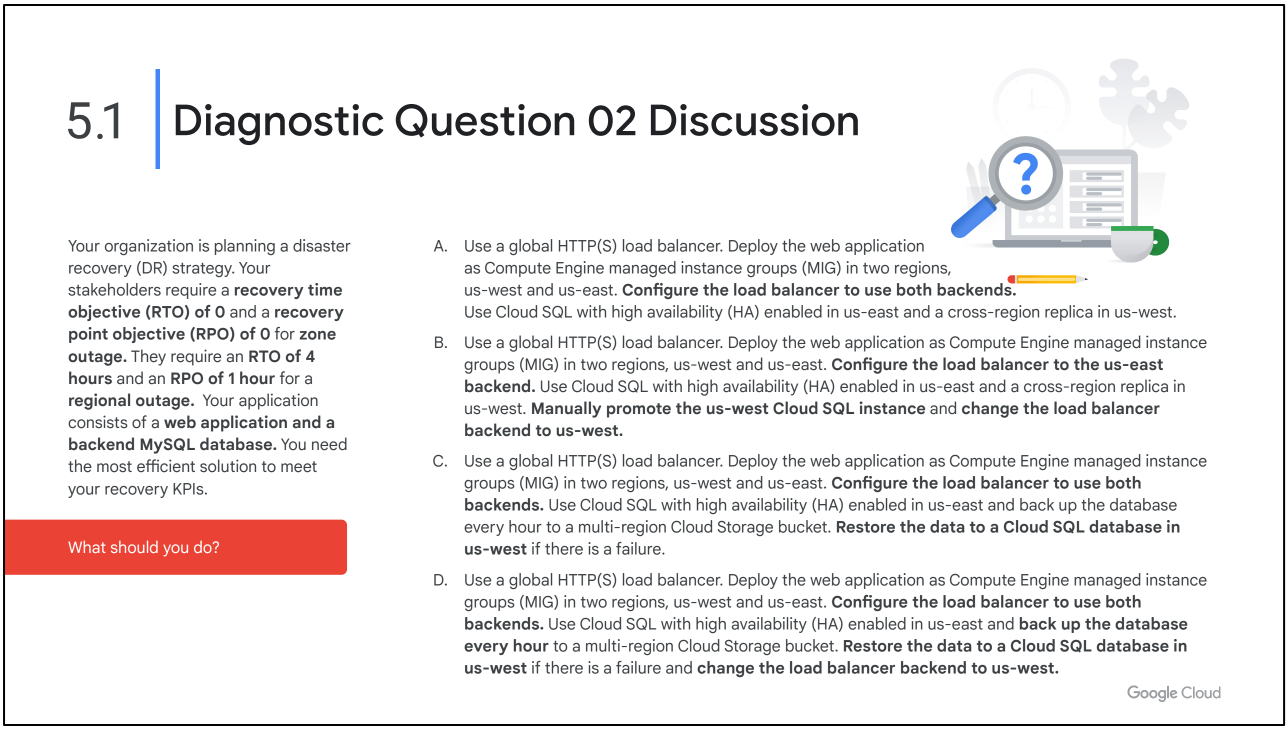
Content mapping:

- Architecting with Google Compute Engine (ILT)
  - M4 Identity and Access Management

- Essential Google Cloud Infrastructure: Core Services (On-demand)
  - M1 Identity and Access Management

Summary:

Budgets are useful for visibility into the amount of money spent and can even alert you when the budget is exceeded. You can use labels to organize your resources and define the limits you alert on. Budgets don't enforce your spending. Enforcing spending limits is your responsibility, and for good reason. Your budget could be high because your site or app is extremely successful. Cloud computing is a shared responsibility model. Google's responsibility is to ensure visibility into your spending, but you decide how much you spend.



5.1 Diagnostic Question 02 Discussion

Your organization is planning a disaster recovery (DR) strategy. Your stakeholders require a **recovery time objective (RTO) of 0** and a **recovery point objective (RPO) of 0** for **zone outage.** They require an **RTO of 4 hours** and an **RPO of 1 hour** for a **regional outage.** Your application consists of a **web application and a backend MySQL database.** You need the most efficient solution to meet your recovery KPIs.

What should you do?

A. Use a global HTTP(S) load balancer. Deploy the web application as Compute Engine managed instance groups (MIG) in two regions, us-west and us-east. **Configure the load balancer to use both backends.** Use Cloud SQL with high availability (HA) enabled in us-east and a cross-region replica in us-west.

B. Use a global HTTP(S) load balancer. Deploy the web application as Compute Engine managed instance groups (MIG) in two regions, us-west and us-east. **Configure the load balancer to the us-east backend.** Use Cloud SQL with high availability (HA) enabled in us-east and a cross-region replica in us-west. **Manually promote the us-west Cloud SQL instance** and **change the load balancer backend to us-west.**

C. Use a global HTTP(S) load balancer. Deploy the web application as Compute Engine managed instance groups (MIG) in two regions, us-west and us-east. **Configure the load balancer to use both backends.** Use Cloud SQL with high availability (HA) enabled in us-east and back up the database every hour to a multi-region Cloud Storage bucket. **Restore the data to a Cloud SQL database in us-west** if there is a failure.

D. Use a global HTTP(S) load balancer. Deploy the web application as Compute Engine managed instance groups (MIG) in two regions, us-west and us-east. **Configure the load balancer to use both backends.** Use Cloud SQL with high availability (HA) enabled in us-east and **back up the database every hour** to a multi-region Cloud Storage bucket. **Restore the data to a Cloud SQL database in us-west** if there is a failure and **change the load balancer backend to us-west.**

Google Cloud

Feedback:

A. Incorrect. This solution would send traffic to both regions, even though the Cloud SQL replica is read-only. Although sending traffic to both regions is not impossible, additional changes to the application architecture would be required, which will create additional complexity.

B. Correct! This solution ensures you meet RTO and RPO for both a zonal and regional outage. By adding the additional steps to manually change the load balancer and promote the Cloud SQL, you ensure the us-west region only accepts traffic after the database is ready to receive it.

C. Incorrect. This solution would send traffic to both regions, even though the Cloud SQL database either has not been created or contains old data. Your RPO is 1 hour for a regional disaster, and although you back up the database to Cloud Storage every hour, the backup itself takes time. Additionally, backing up the database could be disruptive and require locking of the tables, which would prevent writes.

D. Incorrect. Your RPO is 1 hour for a regional disaster, and although you back up the database to Cloud Storage every hour, the backup itself takes time. Additionally, backing up the database could be disruptive and require locking of the tables, which would prevent writes.

Where to look:
https://cloud.google.com/architecture/disaster-recovery

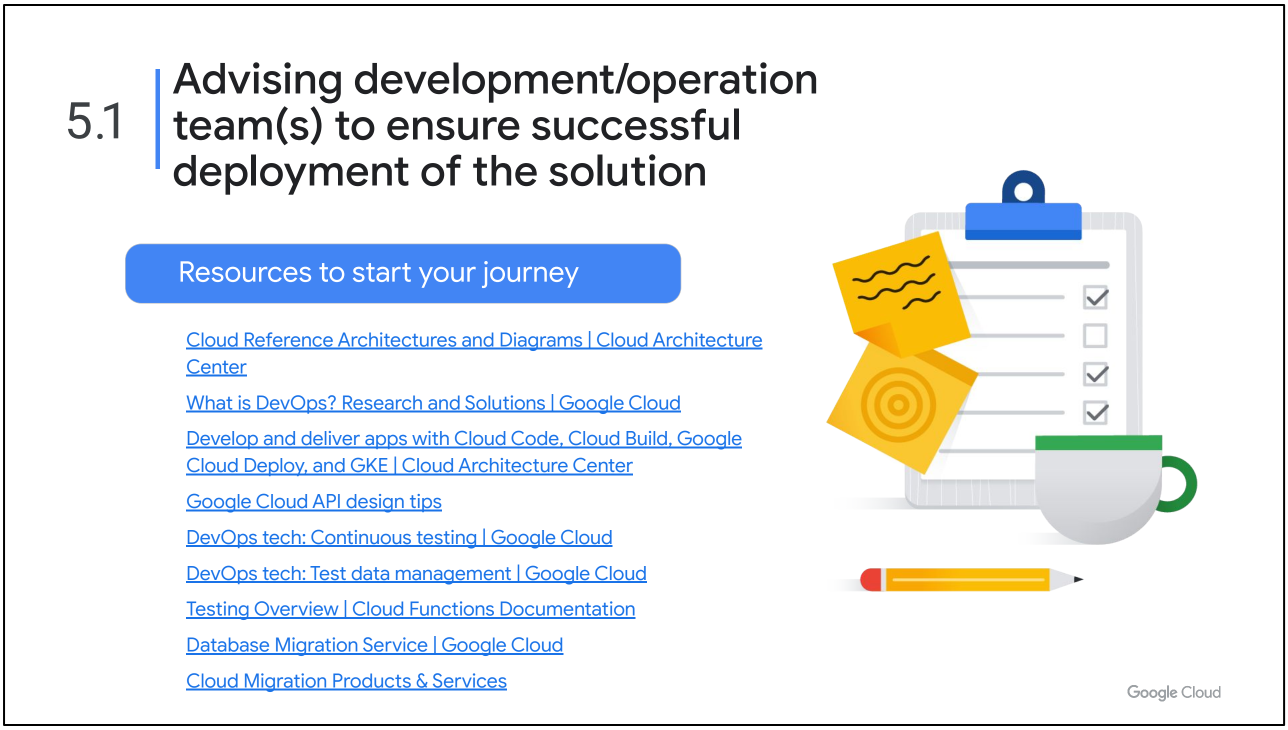
Content mapping:

- 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:
Establishing KPIs for RTO and RPO for an application is easier with Google Cloud compared to a more traditional environment, where you are responsible for all the layers of infrastructure. You can simplify a DR plan by thinking about the availability and durability of different services if a zonal or regional outage occurs. This example uses a traditional architecture with managed instance groups, but the same approach can be used for an implementation using GKE or serverless options. If you change the application slightly and use Spanner as the database backend, this solution could be even more reliable. This would give you a fully managed, multi-regional database backend with 99.999% uptime.



5.1 Advising development/operation team(s) to ensure successful deployment of the solution

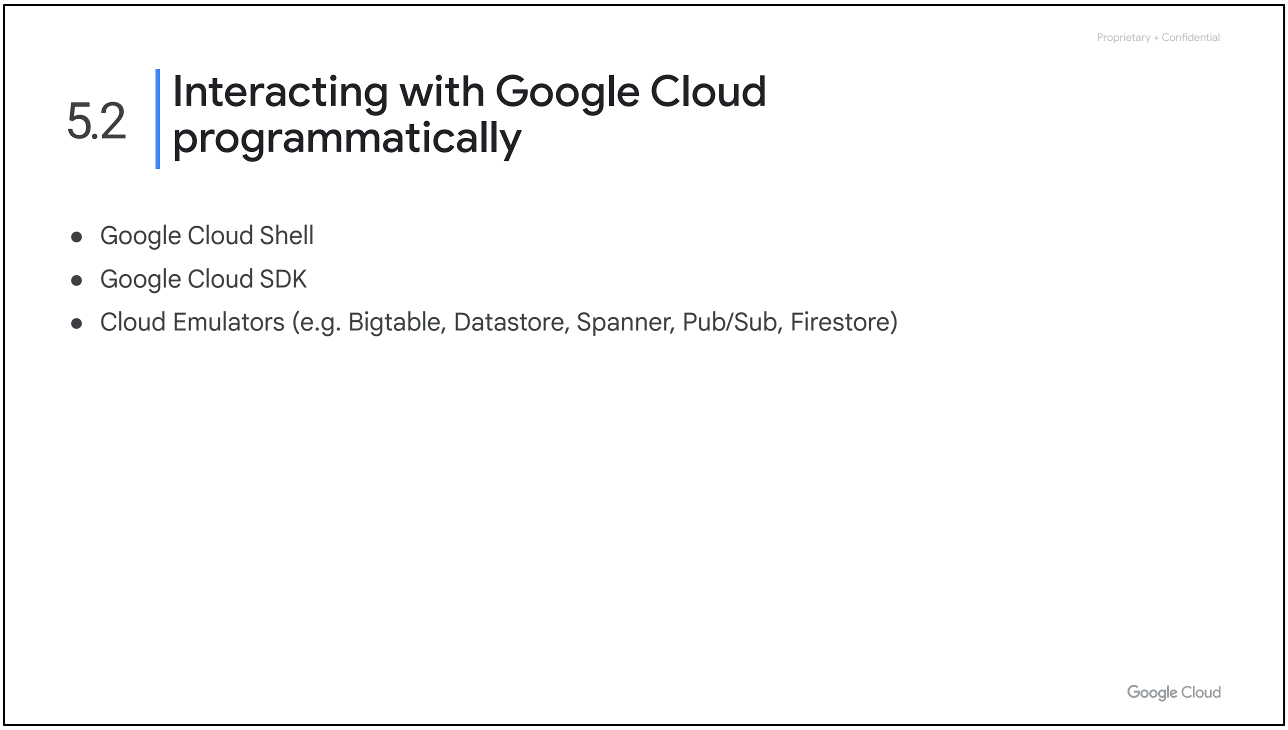
Resources to start your journey

- Cloud Reference Architectures and Diagrams | Cloud Architecture Center
- What is DevOps? Research and Solutions | Google Cloud
- Develop and deliver apps with Cloud Code, Cloud Build, Google Cloud Deploy, and GKE | Cloud Architecture Center
- Google Cloud API design tips
- DevOps tech: Continuous testing | Google Cloud
- DevOps tech: Test data management | Google Cloud
- Testing Overview | Cloud Functions Documentation
- Database Migration Service | Google Cloud
- Cloud Migration Products & Services

Google Cloud

You need to be familiar with Google-recommended practices to help ensure success as a cloud solution moves into deployment. You reviewed diagnostic questions that address a few ways you could be asked to serve in an advisory role as a Professional Cloud Architect. A great place to get started with learning best practices is the Cloud Architecture Center. You'll find this link in your workbook.

https://cloud.google.com/architecture
https://cloud.google.com/devops
https://cloud.google.com/architecture/app-development-and-delivery-with-cloud-code-gcb-cd-and-gke
https://cloud.google.com/blog/products/api-management/google-cloud-api-design-tips
https://cloud.google.com/apis/design
https://cloud.google.com/architecture/devops/devops-tech-test-automation
https://cloud.google.com/architecture/devops/devops-tech-test-data-management
https://cloud.google.com/functions/docs/testing/test-overview
https://cloud.google.com/database-migration
https://cloud.google.com/products/cloud-migration

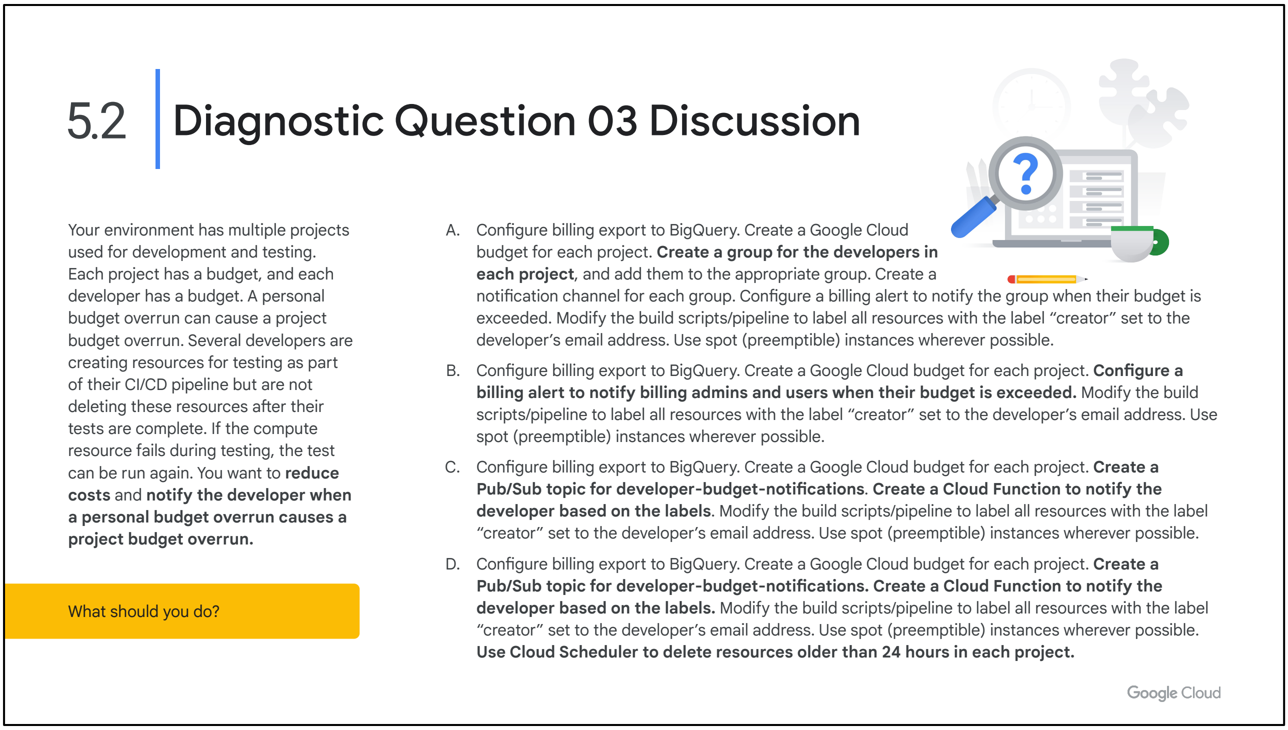


5.2 Interacting with Google Cloud programmatically

- Google Cloud Shell
- Google Cloud SDK
- Cloud Emulators (e.g. Bigtable, Datastore, Spanner, Pub/Sub, Firestore)

As Professional Cloud Architect you should understand how to interact with Google Cloud programmatically using Google Cloud Shell, Google Cloud SDK and Cloud Emulators.

Question 3 tested your ability to use Google Cloud budgets to monitor and optimize service cost.



5.2 Diagnostic Question 03 Discussion

Your environment has multiple projects used for development and testing. Each project has a budget, and each developer has a budget. A personal budget overrun can cause a project budget overrun. Several developers are creating resources for testing as part of their CI/CD pipeline but are not deleting these resources after their tests are complete. If the compute resource fails during testing, the test can be run again. You want to **reduce costs** and **notify the developer when a personal budget overrun causes a project budget overrun.**

What should you do?

A. Configure billing export to BigQuery. Create a Google Cloud budget for each project. **Create a group for the developers in each project**, and add them to the appropriate group. Create a notification channel for each group. Configure a billing alert to notify the group when their budget is exceeded. Modify the build scripts/pipeline to label all resources with the label "creator" set to the developer's email address. Use spot (preemptible) instances wherever possible.

B. Configure billing export to BigQuery. Create a Google Cloud budget for each project. **Configure a billing alert to notify billing admins and users when their budget is exceeded.** Modify the build scripts/pipeline to label all resources with the label "creator" set to the developer's email address. Use spot (preemptible) instances wherever possible.

C. Configure billing export to BigQuery. Create a Google Cloud budget for each project. **Create a Pub/Sub topic for developer-budget-notifications. Create a Cloud Function to notify the developer based on the labels**. Modify the build scripts/pipeline to label all resources with the label "creator" set to the developer's email address. Use spot (preemptible) instances wherever possible.

D. Configure billing export to BigQuery. Create a Google Cloud budget for each project. **Create a Pub/Sub topic for developer-budget-notifications. Create a Cloud Function to notify the developer based on the labels.** Modify the build scripts/pipeline to label all resources with the label "creator" set to the developer's email address. Use spot (preemptible) instances wherever possible. **Use Cloud Scheduler to delete resources older than 24 hours in each project.**

Google Cloud

Feedback:

A. Incorrect. This will notify the entire group of developers regardless of who exceeded their budget.

B. Incorrect. This will notify the Billing Account Administrator and all other users of the project, regardless of who exceeded their budget.

C. Correct! You can have billing notifications sent to a Pub/Sub topic that triggers a Cloud Function. The function can then notify the appropriate developer.

D. Incorrect. Tests can be rerun if an infrastructure failure occurs, such as a machine being preempted, but this doesn't mean that everything in the project is safe to delete. Additional steps would also be required to perform the deletion because Cloud Scheduler can only schedule deletion, not delete things.

Where to look:
https://cloud.google.com/billing/docs/how-to/budgets-programmatic-notifications

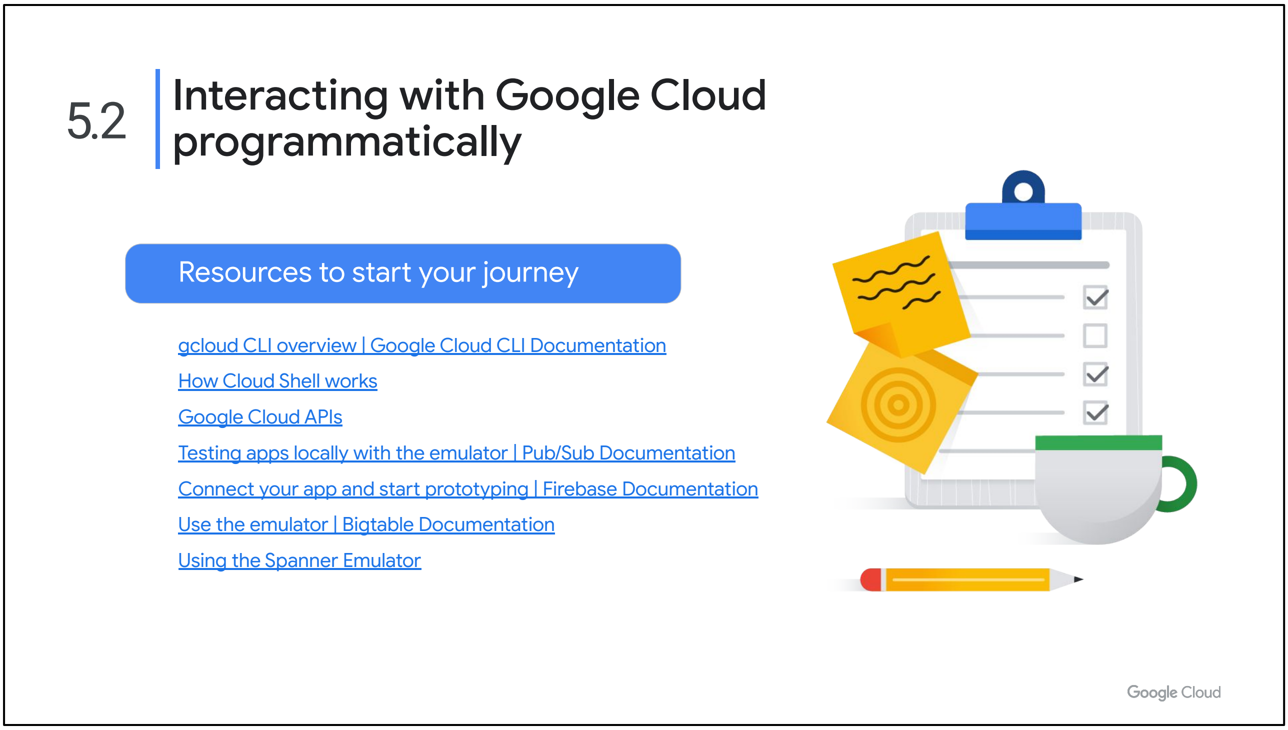
Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
  - M9 Maintenance and Monitoring

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
  - M9 Designing Reliable Systems

Summary:

Google Cloud budgets help with identifying the source of costs. You can configure notifications to be sent several different ways when a budget is exceeded and triggers an alert. Budgets don’t act as a cap: they do not prevent further spending or delete the resources that created the expense. If you send the notification to Pub/Sub, you then have near-unlimited flexibility with how you deal with the alerts because they can be handled programmatically. This isn’t limited to billing alerts; you can apply the same methodology to any kind of event that triggers a notification.



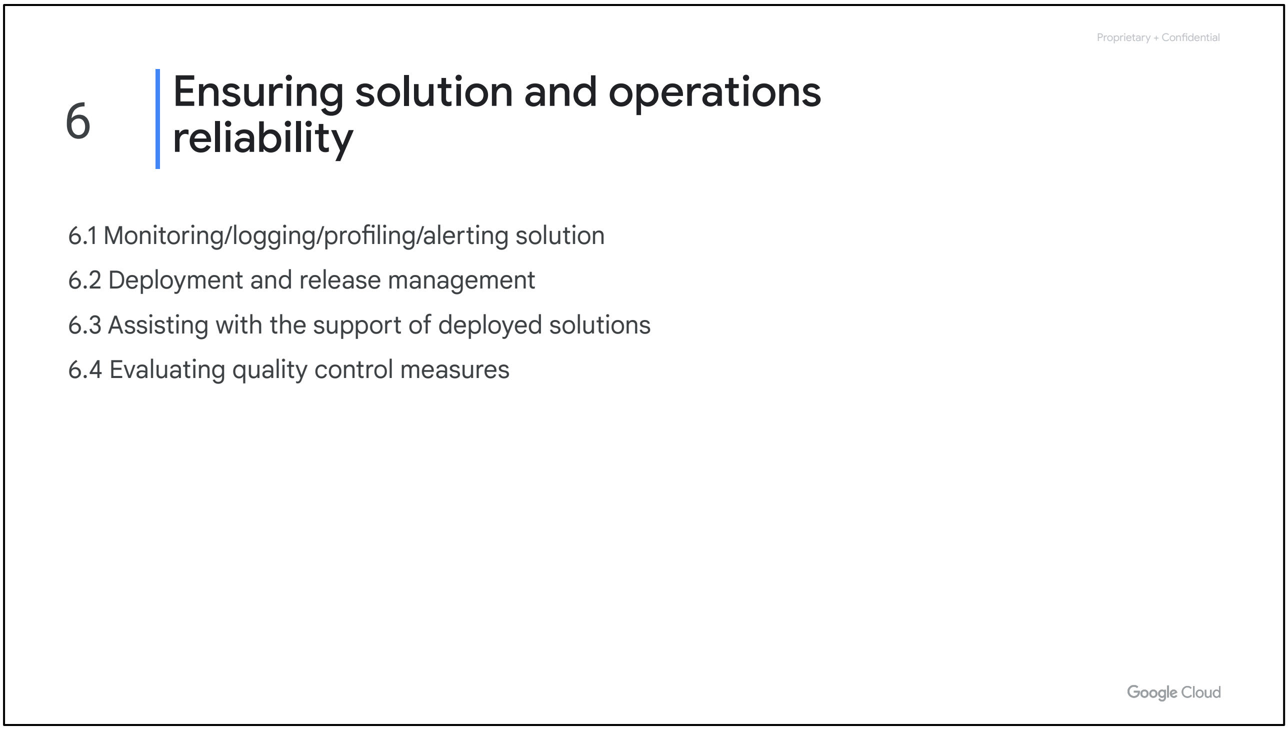
5.2 Interacting with Google Cloud programmatically

Resources to start your journey

- gcloud CLI overview | Google Cloud CLI Documentation
- How Cloud Shell works
- Google Cloud APIs
- Testing apps locally with the emulator | Pub/Sub Documentation
- Connect your app and start prototyping | Firebase Documentation
- Use the emulator | Bigtable Documentation
- Using the Spanner Emulator

Our diagnostic question addressed only one example of interacting with Google Cloud programmatically. These are some resources you can use to learn more about connecting programmatically to Google Cloud. You'll find this list in your workbook.

https://cloud.google.com/sdk/gcloud
https://cloud.google.com/shell/docs/how-cloud-shell-works
https://cloud.google.com/apis/docs/overview
https://cloud.google.com/pubsub/docs/emulator
https://firebase.google.com/docs/emulator-suite/connect_and_prototype?database=Firestore
https://cloud.google.com/bigtable/docs/emulator
https://cloud.google.com/spanner/docs/emulator



6 Ensuring solution and operations reliability

6.1 Monitoring/logging/profiling/alerting solution

6.2 Deployment and release management

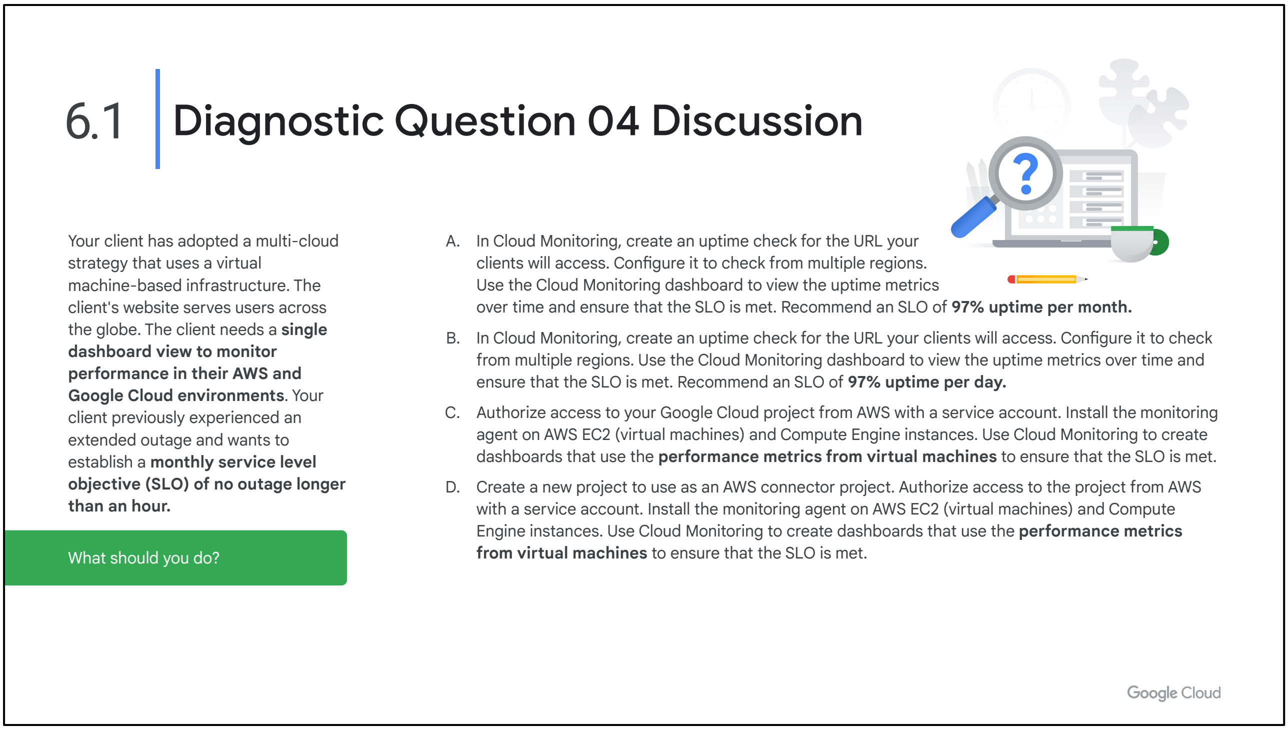
6.3 Assisting with the support of deployed solutions

6.4 Evaluating quality control measures

Because the 4 objectives for section 6 do not have elaborating details in the exam guide, let’s consider them together.

As Professional Cloud Architect you role can also include ensuring solution and operations reliability. Depending on the organization and teams you work with, this could cover a wide range of responsibilities. You should be familiar with monitoring and logging in Google Cloud’s operations suite, deployment and release management, assisting with the support of deployed solutions, and evaluating quality control measures.

Question 4 tested your ability to analyze whether your services meet their service level objectives. Question 5 explored how to use alerts and uptime checks with Cloud Monitoring. Question 6 tested your knowledge of analyzing and optimizing technical and business processes. Question 7 asked you to identify services to use for monitoring, logging, error reporting, tracing, and debugging. Question 8 tested your ability to respond to service outages. Question 9 tested your knowledge of integrated monitoring, alerting, and debugging.



6.1 Diagnostic Question 04 Discussion

Your client has adopted a multi-cloud strategy that uses a virtual machine-based infrastructure. The client's website serves users across the globe. The client needs a **single dashboard view to monitor performance in their AWS and Google Cloud environments**. Your client previously experienced an extended outage and wants to establish a **monthly service level objective (SLO) of no outage longer than an hour.**

What should you do?

A. In Cloud Monitoring, create an uptime check for the URL your clients will access. Configure it to check from multiple regions. Use the Cloud Monitoring dashboard to view the uptime metrics over time and ensure that the SLO is met. Recommend an SLO of **97% uptime per month.**

B. In Cloud Monitoring, create an uptime check for the URL your clients will access. Configure it to check from multiple regions. Use the Cloud Monitoring dashboard to view the uptime metrics over time and ensure that the SLO is met. Recommend an SLO of **97% uptime per day.**

C. Authorize access to your Google Cloud project from AWS with a service account. Install the monitoring agent on AWS EC2 (virtual machines) and Compute Engine instances. Use Cloud Monitoring to create dashboards that use the **performance metrics from virtual machines** to ensure that the SLO is met.

D. Create a new project to use as an AWS connector project. Authorize access to the project from AWS with a service account. Install the monitoring agent on AWS EC2 (virtual machines) and Compute Engine instances. Use Cloud Monitoring to create dashboards that use the **performance metrics from virtual machines** to ensure that the SLO is met.

Google Cloud

Feedback:

A. Incorrect. An SLO of no more than 3% downtime over the course of a month would mean that a downtime of 21 hours was acceptable.

B. Correct! An SLO of no more than 3% downtime over the course of a day would mean that a downtime of more than 43 minutes would exceed it.

C. Incorrect. Having visibility into both AWS and Google Cloud is an advantage of Google Cloud Observability. An SLO should be evaluated from the user's perspective (uptime), not the internals of your environment.

D. Incorrect. Having visibility into both AWS and Google Cloud is an advantage of Google Cloud Observability. An SLO should be evaluated from the user's perspective (uptime), not the internals of your environment.

Where to look:
https://cloud.google.com/architecture/adopting-slos

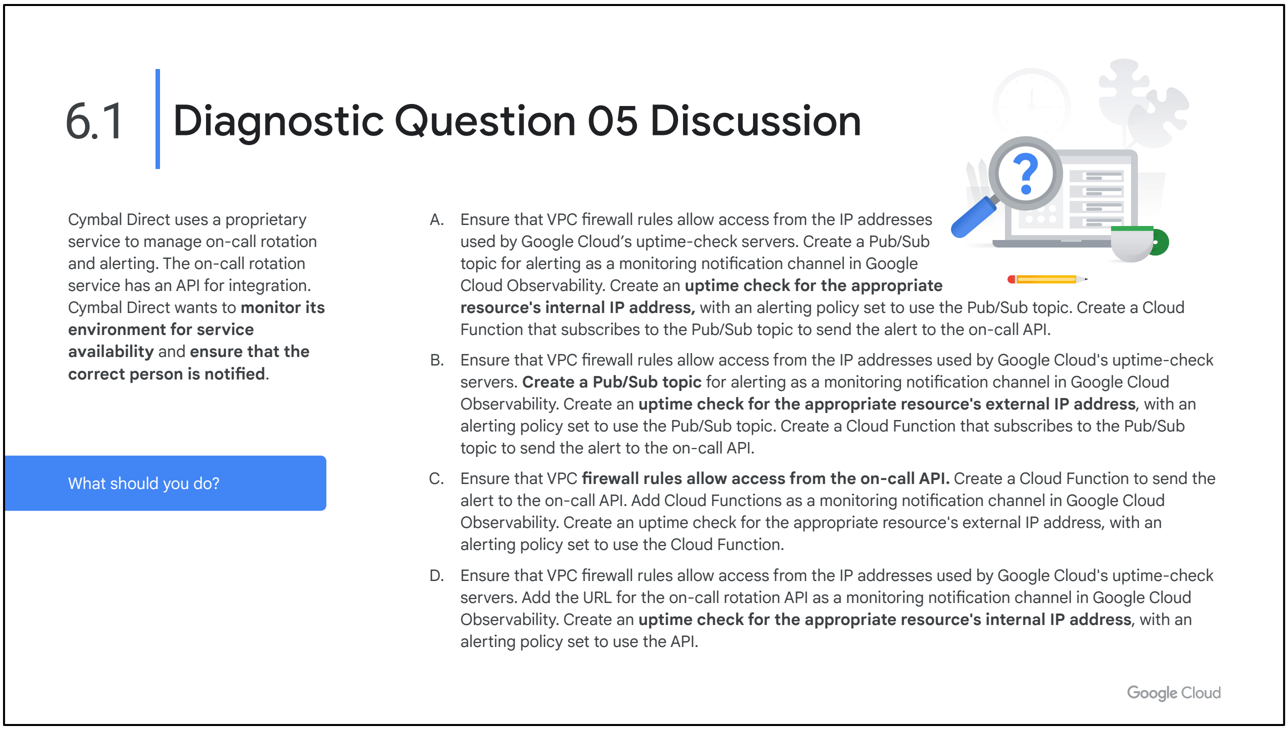
Content mapping:

- Architecting with Google Cloud: Design and Process (ILT)
    - M9 Maintenance and Monitoring

- Reliable Google Cloud Infrastructure: Design and Process (On-demand)
    - M9 Designing Reliable Systems

Summary:
Google Cloud Observability is a powerful tool for creating dashboards, metrics, health

checks, reports, alerts, and more. You can use Google Cloud Observability for visibility into both AWS and Google Cloud. Instead of trying to map all the metrics to SLOs, adopt your user's perspective to define a SLO.



6.1 Diagnostic Question 05 Discussion

Cymbal Direct uses a proprietary service to manage on-call rotation and alerting. The on-call rotation service has an API for integration. Cymbal Direct wants to **monitor its environment for service availability** and **ensure that the correct person is notified**.

What should you do?

A. Ensure that VPC firewall rules allow access from the IP addresses used by Google Cloud’s uptime-check servers. Create a Pub/Sub topic for alerting as a monitoring notification channel in Google Cloud Observability. Create an **uptime check for the appropriate resource's internal IP address,** with an alerting policy set to use the Pub/Sub topic. Create a Cloud Function that subscribes to the Pub/Sub topic to send the alert to the on-call API.

B. Ensure that VPC firewall rules allow access from the IP addresses used by Google Cloud's uptime-check servers. **Create a Pub/Sub topic** for alerting as a monitoring notification channel in Google Cloud Observability. Create an **uptime check for the appropriate resource's external IP address**, with an alerting policy set to use the Pub/Sub topic. Create a Cloud Function that subscribes to the Pub/Sub topic to send the alert to the on-call API.

C. Ensure that VPC **firewall rules allow access from the on-call API.** Create a Cloud Function to send the alert to the on-call API. Add Cloud Functions as a monitoring notification channel in Google Cloud Observability. Create an uptime check for the appropriate resource's external IP address, with an alerting policy set to use the Cloud Function.

D. Ensure that VPC firewall rules allow access from the IP addresses used by Google Cloud's uptime-check servers. Add the URL for the on-call rotation API as a monitoring notification channel in Google Cloud Observability. Create an **uptime check for the appropriate resource's internal IP address**, with an alerting policy set to use the API.

Feedback:

A. Incorrect. An external uptime check is what is required.

B. Correct! Using Pub/Sub as a notification channel gives you flexibility to adapt how notifications are sent.

C. Incorrect. The IP addresses that Google uses to connect need to be allowed in the firewall, not the on-call API. A Cloud Function can subscribe to a Pub/Sub topic to send alerts, but can't be used as a notification channel directly.

D. Incorrect. You cannot send notifications directly to an API. You need to translate the alert programmatically so that the API can receive the notification.

Where to look:
https://cloud.google.com/monitoring/uptime-checks

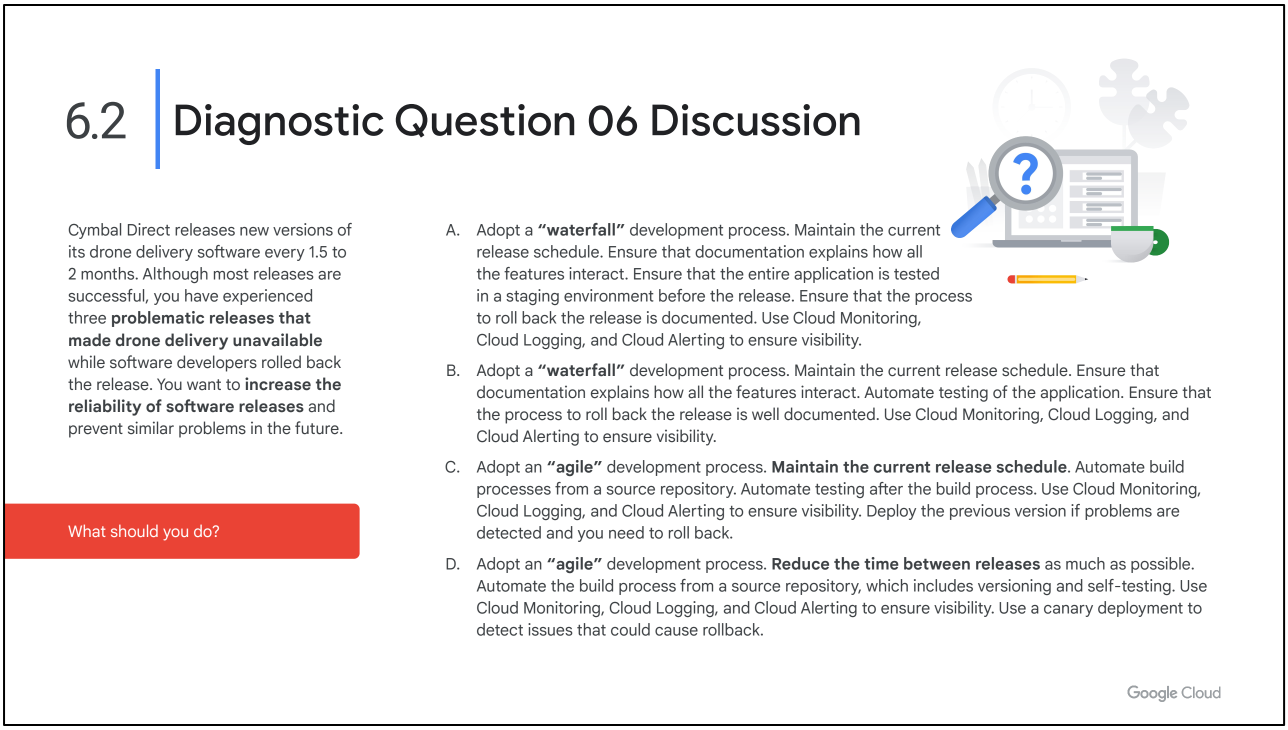
Content mapping:

- Architecting with Google Compute Engine (ILT)
    - M7 Resource Monitoring

- Essential Google Cloud Infrastructure: Core Services (On-demand)
    - M4 Resource Monitoring

Summary:

Using Pub/Sub is only one example of how to integrate a third-party tool to handle sending notifications to the person on-call. You can use Cloud Functions, App Engine, or scripts on a server, or the service used to manage on-call might be able to subscribe to the Pub/Sub topic directly. Pub/Sub is a good option for a notification channel if a standard one, like email, isn’t suitable. Uptime checks are from a user's perspective, and only check external IP addresses. Uptime checks are only one of many metrics you could use.



6.2 Diagnostic Question 06 Discussion

Cymbal Direct releases new versions of its drone delivery software every 1.5 to 2 months. Although most releases are successful, you have experienced three **problematic releases that made drone delivery unavailable** while software developers rolled back the release. You want to **increase the reliability of software releases** and prevent similar problems in the future.

What should you do?

A. Adopt a **"waterfall"** development process. Maintain the current release schedule. Ensure that documentation explains how all the features interact. Ensure that the entire application is tested in a staging environment before the release. Ensure that the process to roll back the release is documented. Use Cloud Monitoring, Cloud Logging, and Cloud Alerting to ensure visibility.

B. Adopt a **"waterfall"** development process. Maintain the current release schedule. Ensure that documentation explains how all the features interact. Automate testing of the application. Ensure that the process to roll back the release is well documented. Use Cloud Monitoring, Cloud Logging, and Cloud Alerting to ensure visibility.

C. Adopt an **"agile"** development process. **Maintain the current release schedule**. Automate build processes from a source repository. Automate testing after the build process. Use Cloud Monitoring, Cloud Logging, and Cloud Alerting to ensure visibility. Deploy the previous version if problems are detected and you need to roll back.

D. Adopt an **"agile"** development process. **Reduce the time between releases** as much as possible. Automate the build process from a source repository, which includes versioning and self-testing. Use Cloud Monitoring, Cloud Logging, and Cloud Alerting to ensure visibility. Use a canary deployment to detect issues that could cause rollback.

Google Cloud

Feedback:

A. Incorrect. A waterfall process means that you are generally targeting large full releases. This approach was appropriate for boxed software that incurred significant costs in terms of time, manufacturing resources, infrastructure, and expense for a release. Larger releases are more complex and more likely to break, and they are difficult to troubleshoot because many changes are made at the same time. Smaller, frequent releases with an automated build process that includes integrated testing with Test Driven Development (TDD) are less likely to require rollbacks, and rollbacks are simpler.

B. Incorrect. A waterfall process means that you are generally targeting large full releases. This approach was appropriate for boxed software that incurred significant costs in terms of time, manufacturing resources, infrastructure, and expense for a release. Larger releases are more complex and more likely to break, and they are difficult to troubleshoot because many changes are made at the same time. Smaller, frequent releases with an automated build process that includes integrated testing with Test Driven Development (TDD) are less likely to require rollbacks, and rollbacks are simpler.

C. Incorrect. An agile development process should generally reduce the time between releases as much as possible. Testing should be integrated into the build. Using a canary deployment can let you detect issues before you deploy a new version at scale.

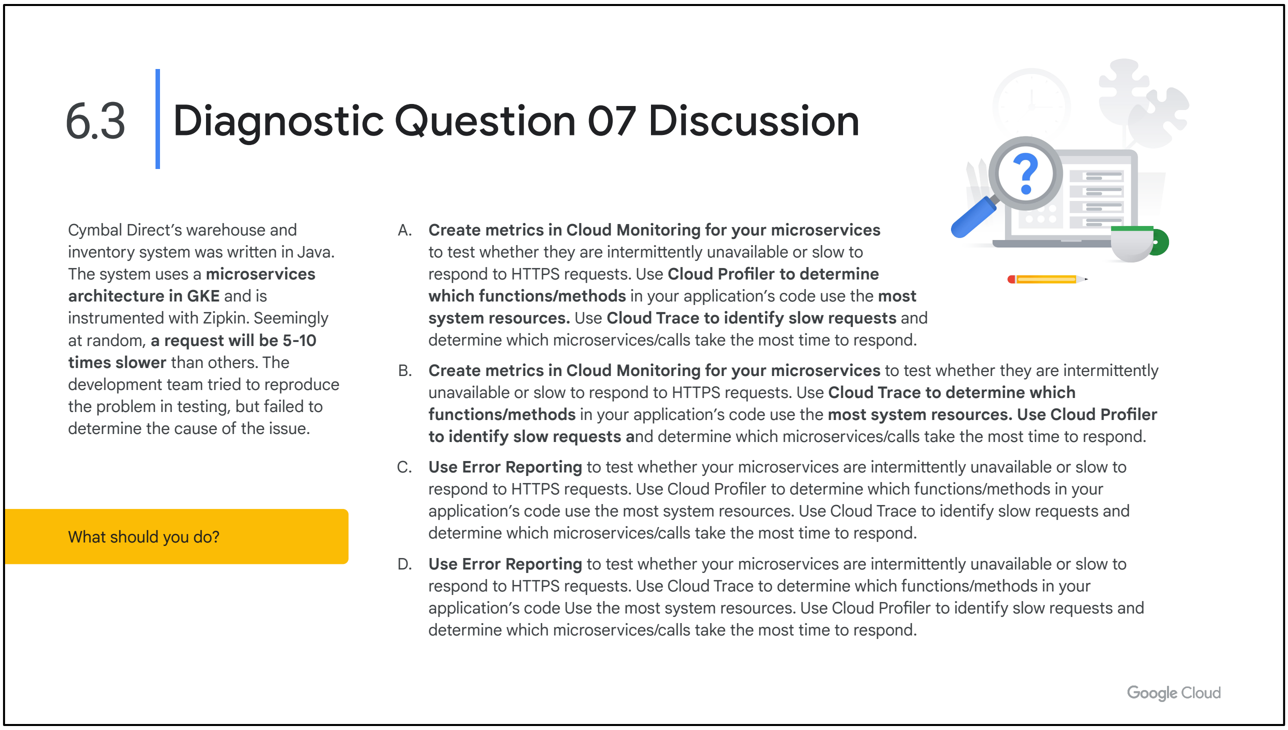
D. Correct! A modern CI/CD pipeline lets you release smaller changes more frequently and includes integrated testing. Using a canary deployment can let you detect issues before you deploy your new version at scale.

Where to look:
https://cloud.google.com/architecture/continuous-delivery-jenkins-kubernetes-engine

Content mapping:
N/A

Summary:
Modern software development has improved significantly over the last decade. Using continuous integration/continuous deployment (CI/CD) pipelines has become standard practice.

Keep your source code in a source code repository such as Cloud Source Repository, and a tool such as Jenkins or Spinnaker. Use test-driven development to integrate testing into your pipeline. Benefit from the features in GKE, which you can use to easily release new versions with a canary or blue/green deployment. Leverage these capabilities to deploy production often, and if something breaks, address it immediately by aborting the release or rolling it back.



6.3 Diagnostic Question 07 Discussion

Cymbal Direct's warehouse and inventory system was written in Java. The system uses a **microservices architecture in GKE** and is instrumented with Zipkin. Seemingly at random, **a request will be 5-10 times slower** than others. The development team tried to reproduce the problem in testing, but failed to determine the cause of the issue.

What should you do?

A. **Create metrics in Cloud Monitoring for your microservices** to test whether they are intermittently unavailable or slow to respond to HTTPS requests. Use **Cloud Profiler to determine which functions/methods** in your application's code use the **most system resources.** Use **Cloud Trace to identify slow requests** and determine which microservices/calls take the most time to respond.

B. **Create metrics in Cloud Monitoring for your microservices** to test whether they are intermittently unavailable or slow to respond to HTTPS requests. Use **Cloud Trace to determine which functions/methods** in your application's code use the **most system resources. Use Cloud Profiler to identify slow requests a**nd determine which microservices/calls take the most time to respond.

C. **Use Error Reporting** to test whether your microservices are intermittently unavailable or slow to respond to HTTPS requests. Use Cloud Profiler to determine which functions/methods in your application's code use the most system resources. Use Cloud Trace to identify slow requests and determine which microservices/calls take the most time to respond.

D. **Use Error Reporting** to test whether your microservices are intermittently unavailable or slow to respond to HTTPS requests. Use Cloud Trace to determine which functions/methods in your application's code Use the most system resources. Use Cloud Profiler to identify slow requests and determine which microservices/calls take the most time to respond.

Google Cloud

Feedback:

A. Correct! Capturing metrics about the health of your microservices could identify an issue. Cloud Profiler can help find the functions or methods in your code that use unusual amounts of CPU, memory, or other system resources. This might indicate where to look for performance problems. Cloud Trace identifies which requests have the highest latency and narrows the scope to the microservices that cause the problem.

B. Incorrect. Cloud Profiler can help find functions or methods in your code that use unusual amounts of CPU, memory, or other system resources. Cloud Trace identifies which requests have the highest latency and narrows the scope to the microservices that cause the problem.

C. Incorrect. Error Reporting captures application errors/exceptions in your code and lets you view the errors in a central place.

D. Incorrect. Error Reporting captures application errors/exceptions in your code and lets you view the errors in a central place. Cloud Profiler can help find functions or methods in your code that use unusual amounts of CPU, memory, or other system resources. Cloud Trace identifies which requests have the highest latency and narrows the scope to the microservices that cause the problem.

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

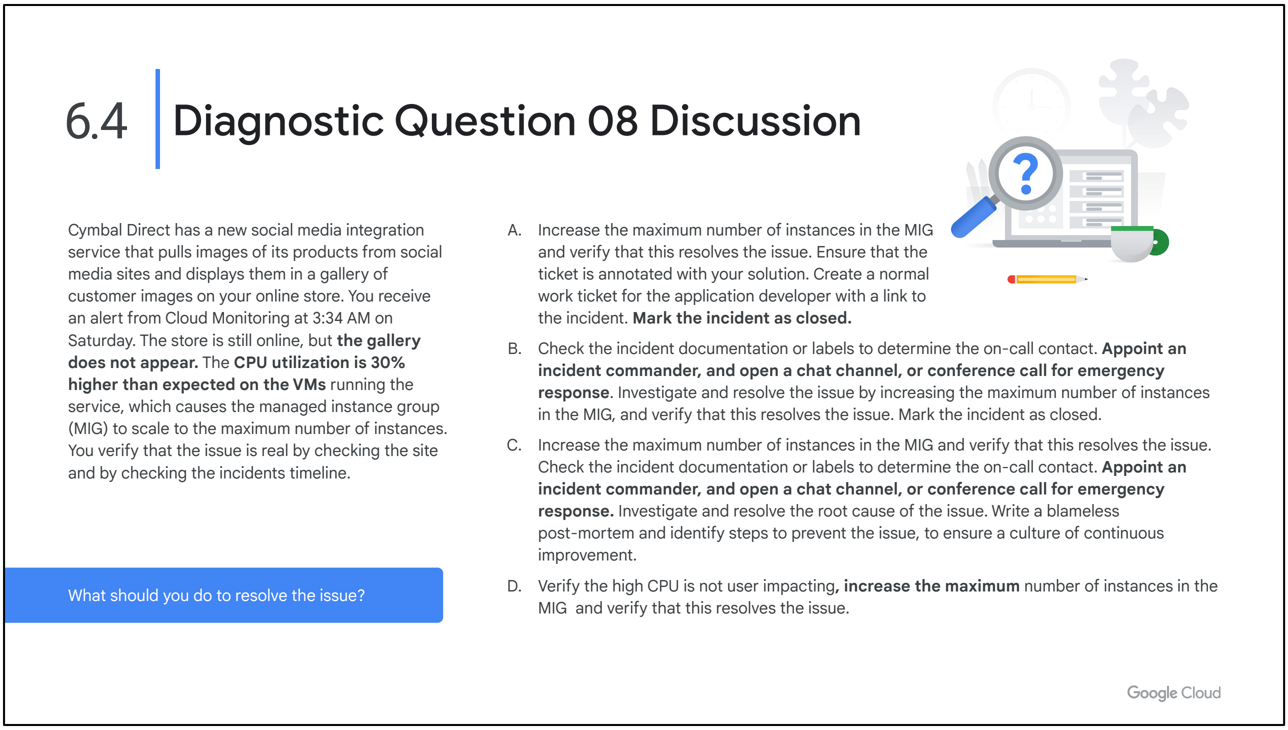
Content mapping:

- Architecting with Google Compute Engine (ILT)

    - M7 Resource Monitoring

- Essential Google Cloud Infrastructure: Core Services (On-demand)
    - M4 Resource Monitoring

Summary:
Google Cloud’s operations suite provides tools that can help you discover and diagnose issues in your environment. Microservices add an extra level of complexity during troubleshooting because you need to account for the communication between the microservices. Pinpointing the root of an issue can be difficult, especially if it is intermittent. Using the integrated tools in the operations suite lets you seamlessly switch between tools to find your issue. Cloud Trace can be especially useful when you determine which microservice causes a bottleneck. However, you need to use the Cloud Tracing Agent, OpenTelemetry (previously OpenCensus), or Zipkin to ensure that instrumentation is enabled.



6.4 Diagnostic Question 08 Discussion

Cymbal Direct has a new social media integration service that pulls images of its products from social media sites and displays them in a gallery of customer images on your online store. You receive an alert from Cloud Monitoring at 3:34 AM on Saturday. The store is still online, but **the gallery does not appear.** The **CPU utilization is 30% higher than expected on the VMs** running the service, which causes the managed instance group (MIG) to scale to the maximum number of instances. You verify that the issue is real by checking the site and by checking the incidents timeline.

What should you do to resolve the issue?

A. Increase the maximum number of instances in the MIG and verify that this resolves the issue. Ensure that the ticket is annotated with your solution. Create a normal work ticket for the application developer with a link to the incident. **Mark the incident as closed.**

B. Check the incident documentation or labels to determine the on-call contact. **Appoint an incident commander, and open a chat channel, or conference call for emergency response**. Investigate and resolve the issue by increasing the maximum number of instances in the MIG, and verify that this resolves the issue. Mark the incident as closed.

C. Increase the maximum number of instances in the MIG and verify that this resolves the issue. Check the incident documentation or labels to determine the on-call contact. **Appoint an incident commander, and open a chat channel, or conference call for emergency response.** Investigate and resolve the root cause of the issue. Write a blameless post-mortem and identify steps to prevent the issue, to ensure a culture of continuous improvement.

D. Verify the high CPU is not user impacting**, increase the maximum** number of instances in the MIG and verify that this resolves the issue.

Google Cloud

Feedback:
A. Incorrect. Google Cloud Observability will close an incident if the alerting condition is no longer being met.
B. Incorrect. This answer is more appropriate for a critical incident. This response doesn't consider the severity of the issue and the impact on the developer and response team. Managing a service should not require a "heroic effort." Take basic mitigation steps, such as increasing the number of instances, and the developer can fix the issue on Monday. Google Cloud Observability will close an incident if the alerting condition is no longer being met.
C. Incorrect. This answer is more appropriate for a critical incident. This response doesn't consider the severity of the issue and the impact on the developer and response team. Managing a service should not require a "heroic effort." Take basic mitigation steps such as increasing the number of instances, and the developer can fix the issue on Monday.
D. Correct! This appropriately responds to the issue by increasing the number of instances and doesn't require a "heroic effort" by having the developer or response team resolve the issue in the middle of the night.

Where to look:
https://cloud.google.com/monitoring/alerts/incidents-events
https://sre.google/workbook/incident-response/

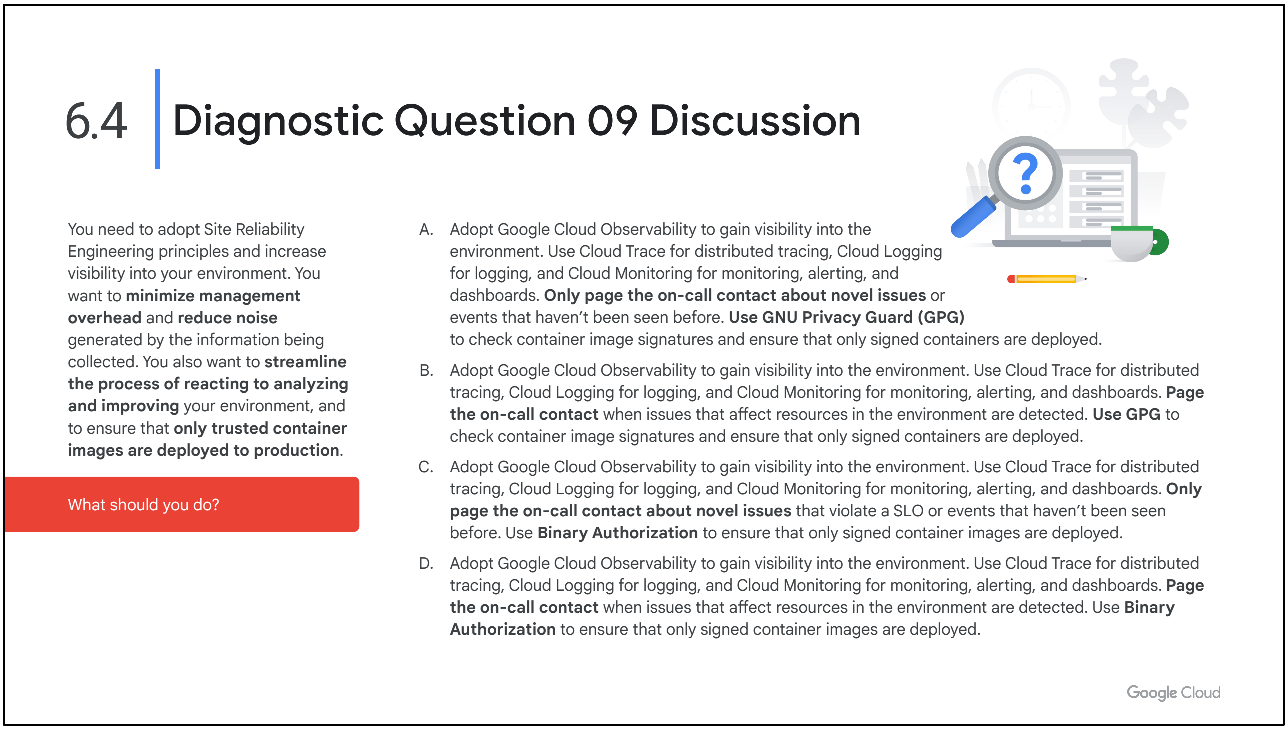
Content mapping:

- Architecting with Google Compute Engine (ILT)

  - M7 Resource Monitoring

- Essential Google Cloud Infrastructure: Core Services (On-demand)
  - M4 Resource Monitoring

Summary:
Google Cloud Observability was designed in the context of Google's understanding of how an incident should be responded to. To use Cloud Monitoring or any other tool that includes monitoring and alerting, you should consider how you would respond to an actual alert. It could be by formally appointing someone to be the incident commander and going through the steps of identification, coordination, resolution, and closure for a critical issue. You could also respond by patching the issue until it is more formally dealt with during business hours, thus creating a better work culture. You should investigate both the capabilities of Google Cloud Observability and the processes and values outlined in Google's Site Reliability Engineering (SRE) books.



6.4 Diagnostic Question 09 Discussion

You need to adopt Site Reliability Engineering principles and increase visibility into your environment. You want to **minimize management overhead** and **reduce noise** generated by the information being collected. You also want to **streamline the process of reacting to analyzing and improving** your environment, and to ensure that **only trusted container images are deployed to production**.

What should you do?

A. Adopt Google Cloud Observability to gain visibility into the environment. Use Cloud Trace for distributed tracing, Cloud Logging for logging, and Cloud Monitoring for monitoring, alerting, and dashboards. **Only page the on-call contact about novel issues** or events that haven't been seen before. **Use GNU Privacy Guard (GPG)** to check container image signatures and ensure that only signed containers are deployed.

B. Adopt Google Cloud Observability to gain visibility into the environment. Use Cloud Trace for distributed tracing, Cloud Logging for logging, and Cloud Monitoring for monitoring, alerting, and dashboards. **Page the on-call contact** when issues that affect resources in the environment are detected. **Use GPG** to check container image signatures and ensure that only signed containers are deployed.

C. Adopt Google Cloud Observability to gain visibility into the environment. Use Cloud Trace for distributed tracing, Cloud Logging for logging, and Cloud Monitoring for monitoring, alerting, and dashboards. **Only page the on-call contact about novel issues** that violate a SLO or events that haven't been seen before. Use **Binary Authorization** to ensure that only signed container images are deployed.

D. Adopt Google Cloud Observability to gain visibility into the environment. Use Cloud Trace for distributed tracing, Cloud Logging for logging, and Cloud Monitoring for monitoring, alerting, and dashboards. **Page the on-call contact** when issues that affect resources in the environment are detected. Use **Binary Authorization** to ensure that only signed container images are deployed.

Google Cloud

Feedback:

A. Incorrect. Use Binary Authorization to ensure that only signed container images are deployed.

B. Incorrect. On-call contacts are expected to respond with urgency to every page. Frequent paging will lead to fatigue in the short term and burnout in the long term, which eventually reduces the reliability of the service. Use Binary Authorization to ensure that only signed container images are deployed.

C. Correct! Google Cloud Observability is tightly integrated with different components in the suite and other open source tools. It allows for streamlined analysis of issues without requiring additional management overhead to set up and maintain the tools.

D. Incorrect. On-call contacts are expected to respond with urgency to every page. Frequent paging will lead to fatigue in the short term and burnout in the long term, which eventually reduces the reliability of the service.

Where to look:

https://sre.google/sre-book/monitoring-distributed-systems/
https://cloud.google.com/products/operations
https://cloud.google.com/binary-authorization

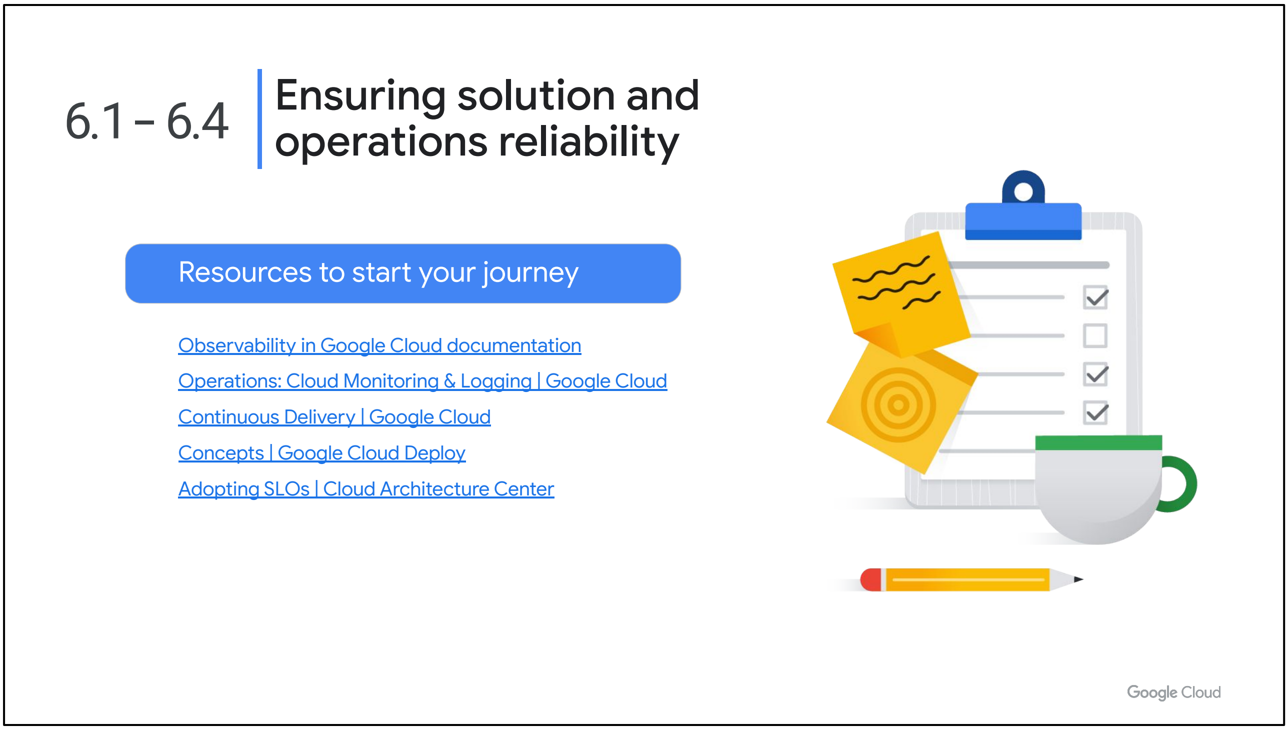
Content mapping:

- Architecting with Google Compute Engine (ILT)
  - M7 Resource Monitoring

- Essential Google Cloud Infrastructure: Core Services (On-demand)

- M4 Resource Monitoring

Summary:
Google Cloud Observability is an integrated monitoring, logging, tracing, alerting, and visualization tool for applications and services that run in Google Cloud and elsewhere. You can monitor AWS instances or deliver metrics from your on-premises Anthos Kubernetes cluster. It's built on many open source tools, such as Fluentd, and is continuously extended. All of this is a managed service, and you don't need to spend additional time and resources setting up a collection of different tools. Troubleshooting an analysis is faster because you can move between different tools in the suite easily based on what part of an issue you are troubleshooting. The best monitoring tools in the world aren't effective if you don't have good policies around what you should monitor and how you should respond to issues. Google has a set of best practices defined in the freely available Site Reliability Engineering (SRE) book. Binary Authorization is one of several tools you can use to ensure the quality of container images that are deployed to production by requiring only images from trusted sources. You can integrate it into CI/CD pipelines and consider enabling Artifact Analysis to detect security issues.



6.1 – 6.4 Ensuring solution and operations reliability

Resources to start your journey

- [Observability in Google Cloud documentation]
- [Operations: Cloud Monitoring & Logging | Google Cloud]
- [Continuous Delivery | Google Cloud]
- [Concepts | Google Cloud Deploy]
- [Adopting SLOs | Cloud Architecture Center]

Your ability to ensure solution and operations reliability will grow with experience working in Google Cloud. These are some resources you can use to learn more about best practices. You'll find this list in your workbook.

https://cloud.google.com/stackdriver/docs
https://cloud.google.com/products/operations
https://cloud.google.com/solutions/continuous-delivery
https://cloud.google.com/deploy/docs/concepts/
https://cloud.google.com/architecture/adopting-slos