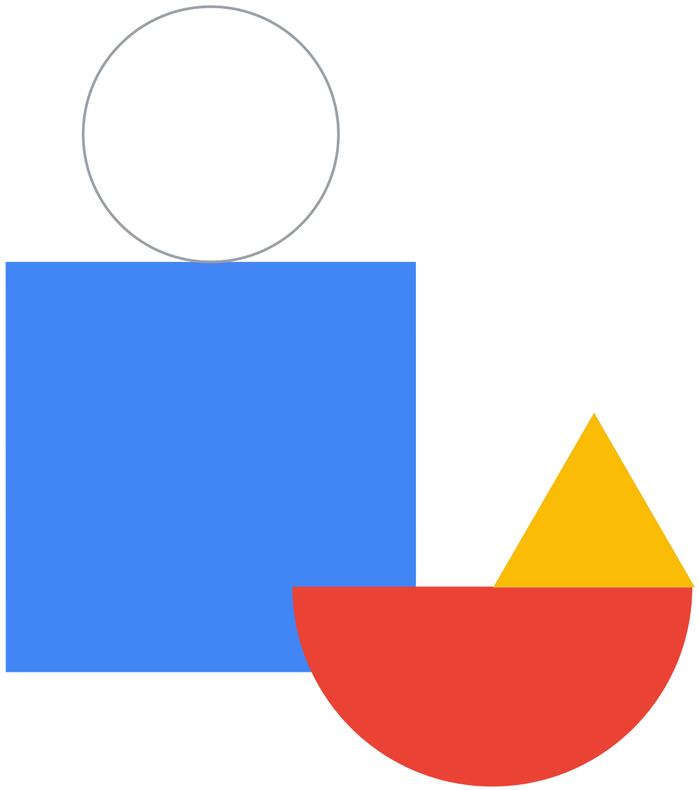
Google Cloud



Preparing for your Professional Cloud Architect Journey

Module 5: Managing Implementation and Ensuring Solution and Operations Reliability

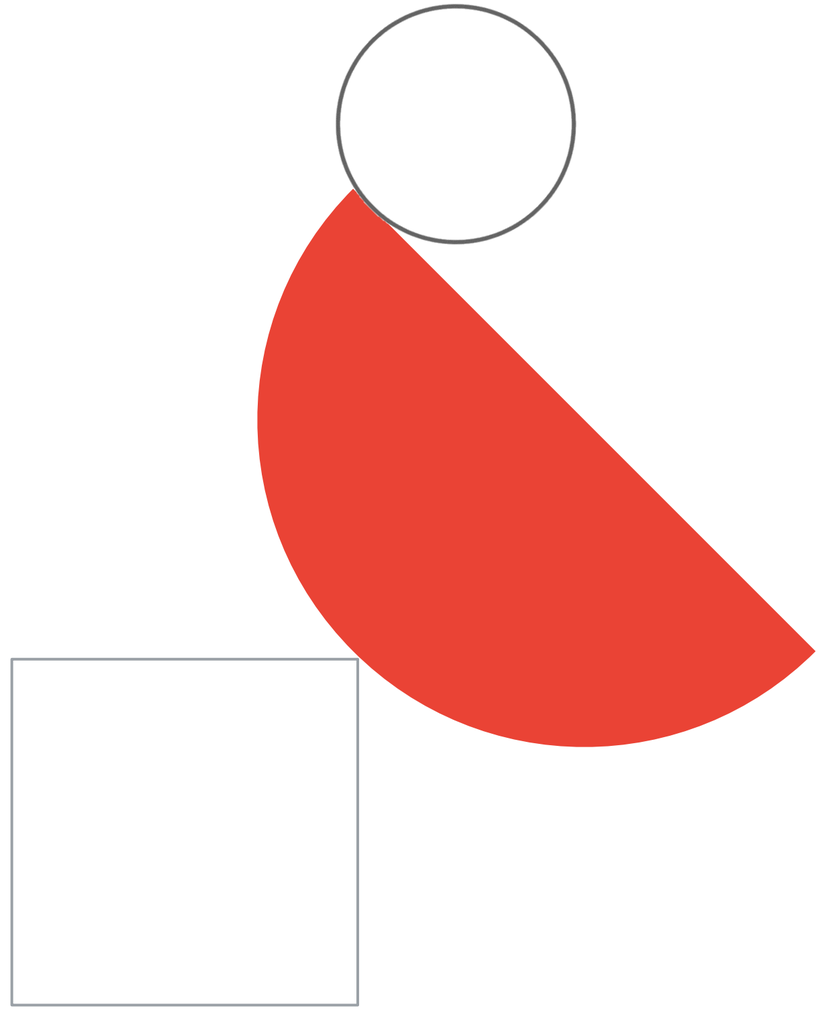
Module agenda

01 Implementation, operations, and reliability at Cymbal Direct

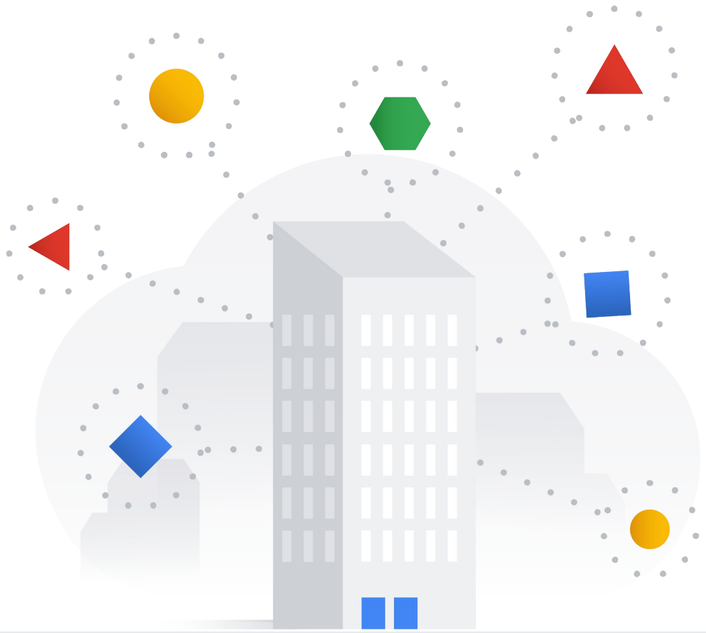
02 Diagnostic questions

03 Review and study planning

Implementation, operations, and reliability at Cymbal Direct



Your role in implementation, operations, and reliability

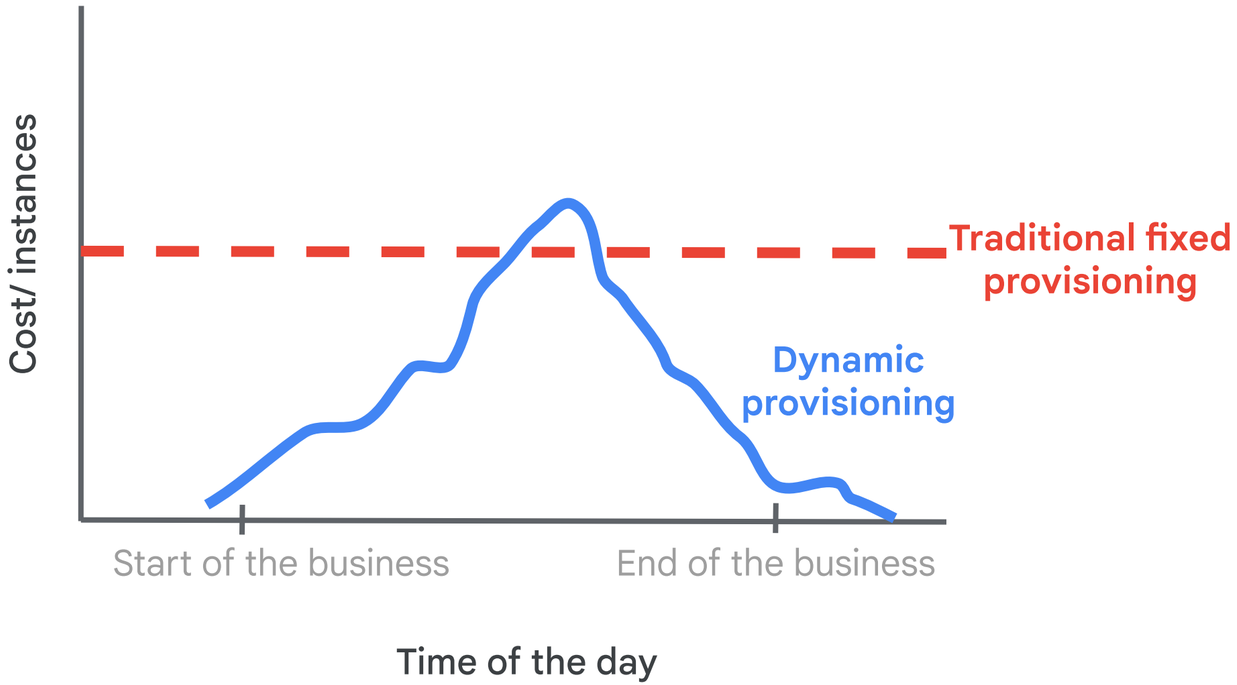


- Describe best practices for development and operations teams to ensure successful solution deployment.
- Explain methods to interact with Google Cloud programmatically.
- Explain methodologies for managing configuration and code updates and tools available for monitoring and analyzing KPIs.

Cymbal Direct

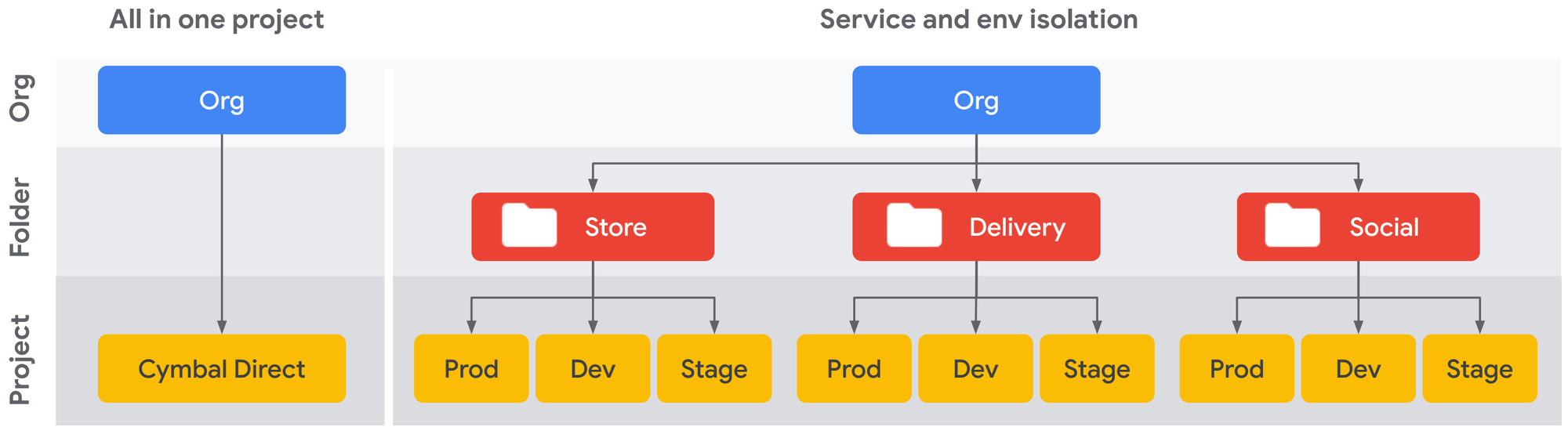
Best practices

Describe best practices for development and operations teams to ensure successful solution deployment.

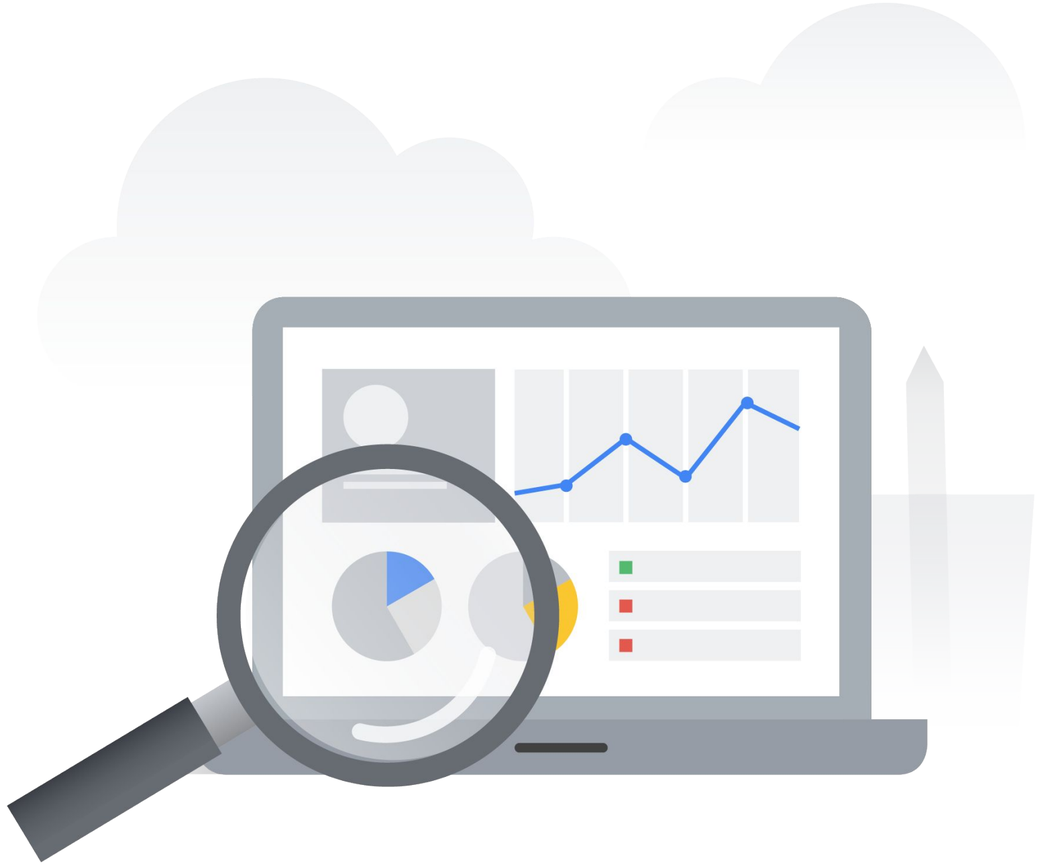


Programmatic access

Explain methods to interact with Google Cloud programmatically



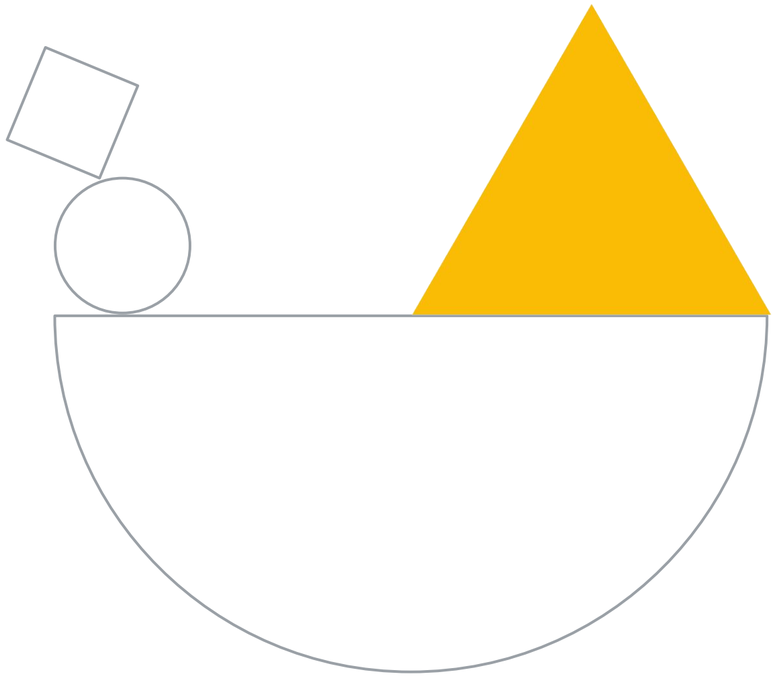
Configuration and code updates



Monitoring and KPIs

Tools available for monitoring and analyzing KPIs.

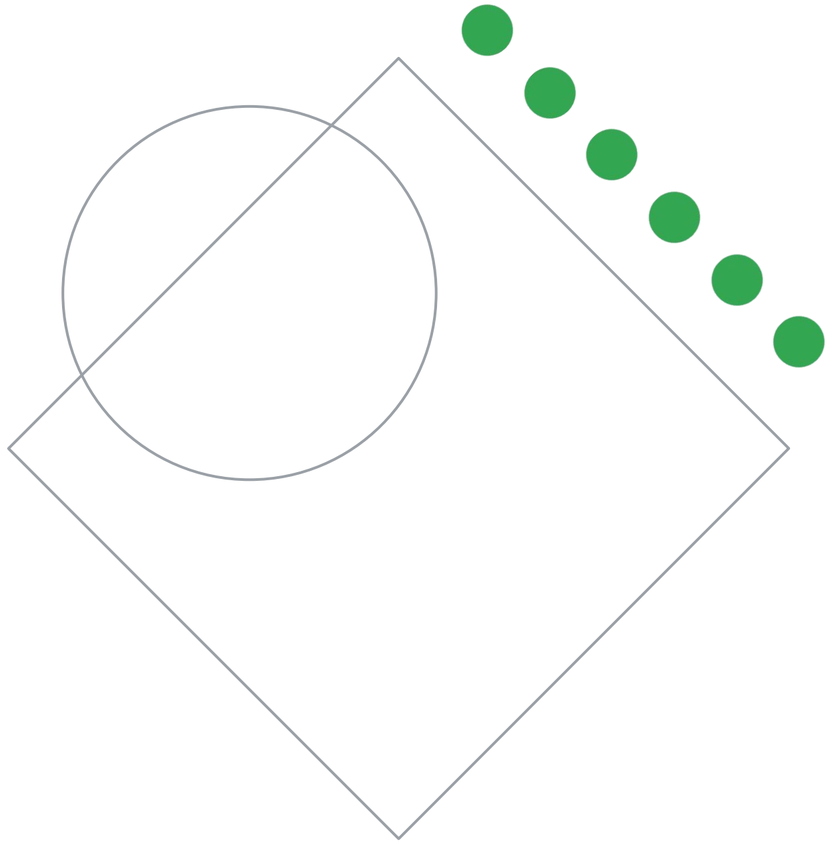
Diagnostic questions



Quiz Forms

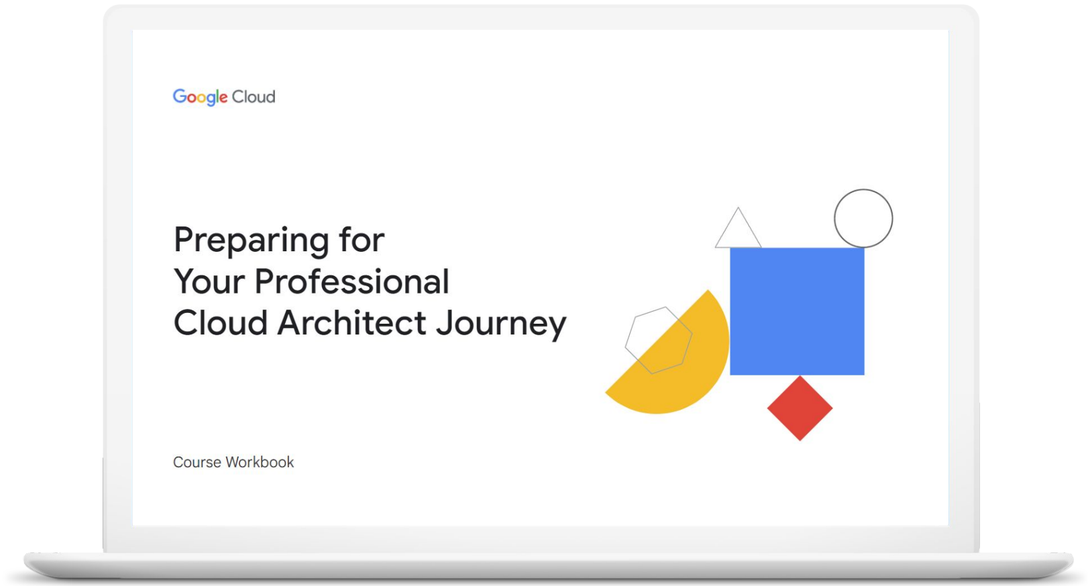
- Quiz forms are provided for you to answer the diagnostic questions
- The instructor will provide you a link
- The quiz is also available in the workbook

Review and study planning



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

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

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.**

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.**

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

5.2 Interacting with Google Cloud programmatically

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

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.**

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.**

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 | Cloud Pub/Sub Documentation

Connect your app and start prototyping | Firebase Documentation

Use the emulator | Cloud Bigtable Documentation

Using the Cloud Spanner 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

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.

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.

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.

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.

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.

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.

6.4 Diagnostic Question 10 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's operations suite 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's operations suite 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's operations suite 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's operations suite 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.

6.4 Diagnostic Question 10 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's operations suite 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's operations suite 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's operations suite 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's operations suite 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.

6.1 – 6.4 Ensuring solution and operations reliability

Resources to start your journey

- Google Cloud operations suite documentation
- Operations: Cloud Monitoring & Logging | Google Cloud
- Cloud operations grows with monitoring, logging, more | Google Cloud Blog
- Continuous Delivery | Google Cloud
- Concepts | Google Cloud Deploy
- Adopting SLOs | Cloud Architecture Center

Knowledge Check 1

How could Cymbal Direct design their code to maximize their savings by running spot (preemptable) instances?

A. Use an interpreted language like Python.

B. Create an API for their software.

C. Externalize state.

D. Use Cloud Monitoring to get performance info.

Knowledge Check 1

How could Cymbal Direct design their code to maximize their savings by running spot (preemptable) instances?

A. Use an interpreted language like Python.

B. Create an API for their software.

C. Externalize state.

D. Use Cloud Monitoring to get performance info.

Knowledge Check 2

How could Cymbal Direct save money for testing/development resources?

A. Have all developers sign up for the free $300 credit.

B. Provision resources only when needed using Terraform.

C. Give all developers the ability to provision resources for themselves.

D. Only use the smallest size compute engine instances.

Knowledge Check 2

How could Cymbal Direct save money for testing/development resources?

A. Have all developers sign up for the free $300 credit.

B. Provision resources only when needed using Terraform.

C. Give all developers the ability to provision resources for themselves.

D. Only use the smallest size compute engine instances.

Appendix

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.**

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's operations suite. 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's operations suite. 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's operations suite. 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's operations suite. Create an **uptime check for the appropriate resource's internal IP address**, with an alerting policy set to use the API.

6.3 Diagnostic Question 08 Discussion

You are using Cloud Run to deploy a **Flask web application named app.py written in Python**. In your testing and staging environments, the application performed as expected. When the application was deployed to production, product search results displayed products that should have been filtered out based on the user's preferences. The developer believes **this performance issue would result from the 'user.productFilter' variable either not being set or not being evaluated correctly**. You want **visibility into what is happening, but also want to minimize user impact**, because this is not a critical bug.

What should you do?

A. Use ssh to connect to the **Compute Engine instance** where Cloud Run is running. Run the command **'python3 -m pdb app.py'** to debug the application.

B. Use ssh to connect to the **Compute Engine instance** where Cloud Run is running. Use the command **'pip install google-python-cloud-debugger' to install Cloud Debugger**. Use the 'gcloud debug' command to debug the application.

C. Modify the Dockerfile for the Cloud Run application. Change the RUN command to **'python3 -m pdb /app.py'.** Modify the script to import pdb. **Deploy to Cloud Run** as a canary build.

D. Modify the Dockerfile for the Cloud Run application. Add 'RUN '**pip install google-python-cloud-debugger**' to the Dockerfile. Modify the script to import googleclouddebugger. Use '**gcloud debug**' to debug the application.

6.4 Diagnostic Question 09 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.