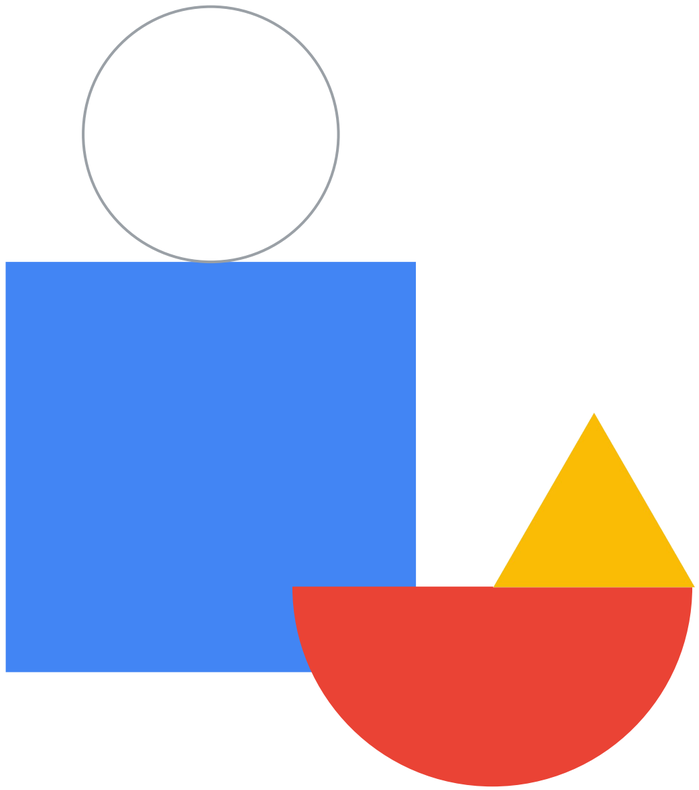
Google Cloud

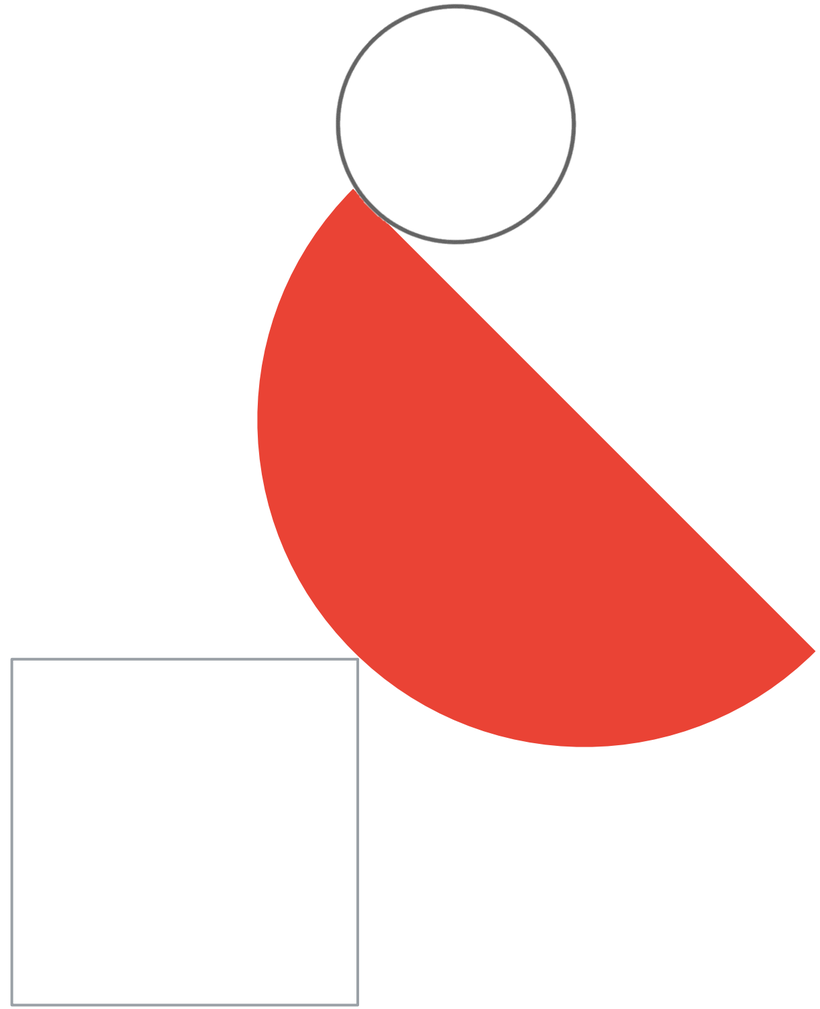


Preparing for your Professional Cloud Architect Journey

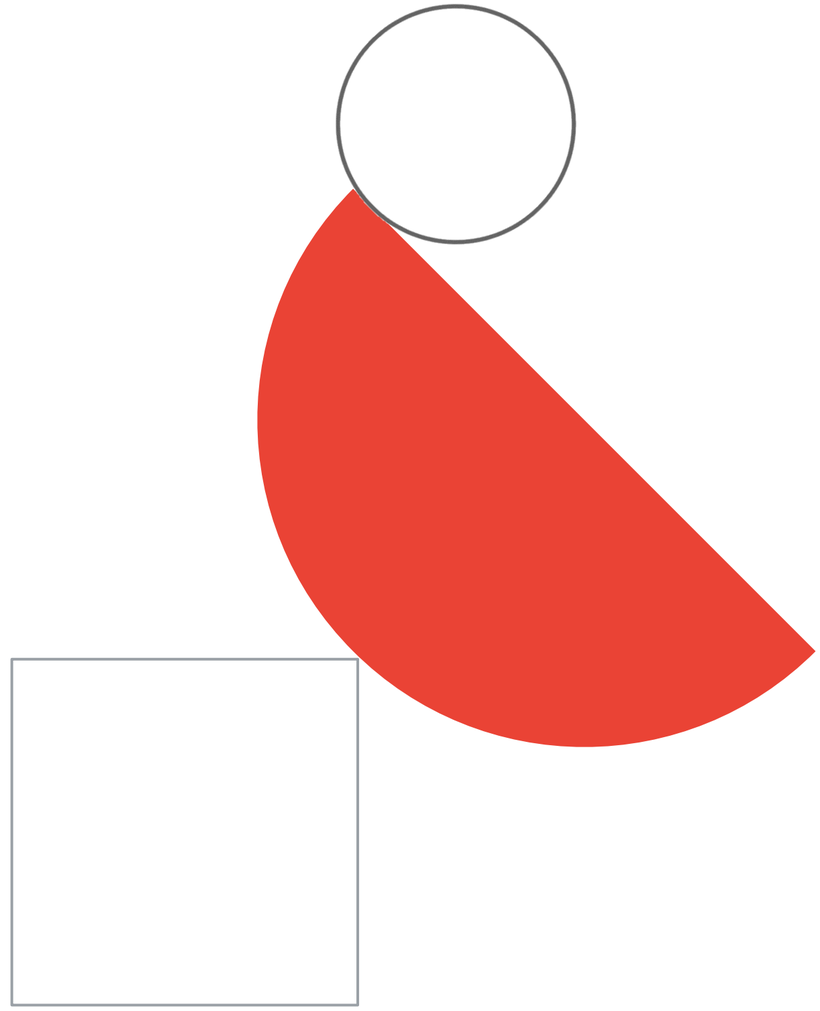
Module 5: Managing Implementation and Ensuring Solution and Operations Reliability

Week 6 agenda

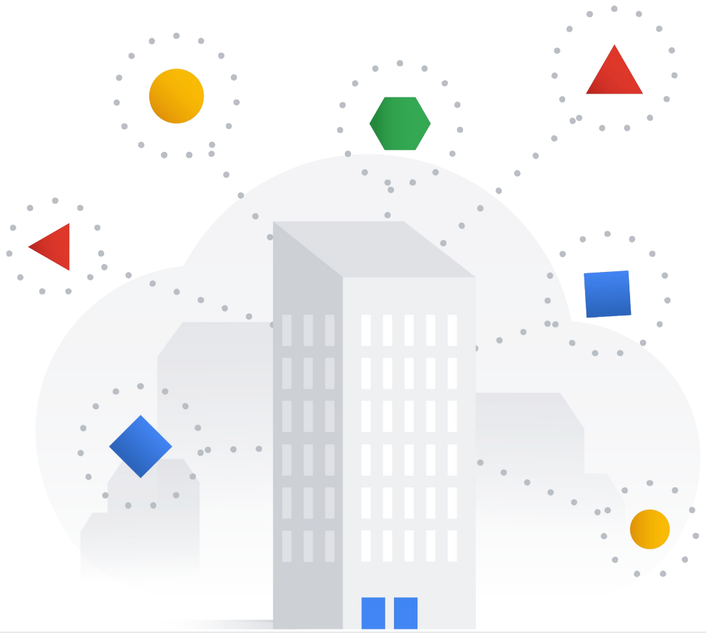
QUIZ time!



Implementation, operations, and reliability at Cymbal Direct



Your role in implementation, operations, and reliability

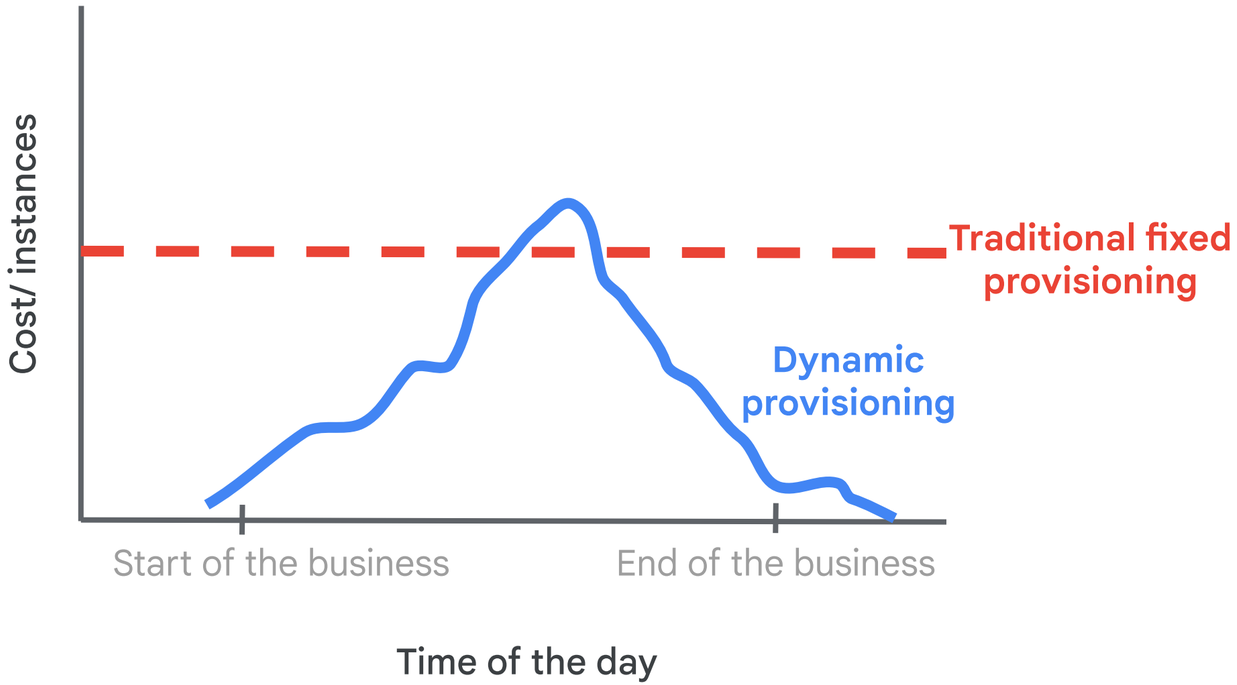


- Describe best practices for development and operations teams to ensure successful solution deployment.
- Explain methods to interact with Google Cloud programmatically.
- Explain methodologies for managing configuration and code updates and tools available for monitoring and analyzing KPIs.

Cymbal Direct

Best practices

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



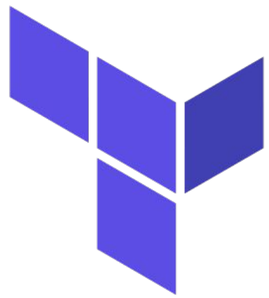
Configuration and code updates

Deployment Manager vs Terraform

- DM is in *maintenance mode* and customers are **discouraged from using it**
- The recommended way to define infrastructure is Terraform. Google actively maintains the Google Terraform module
- Customers using Deployment Manager will be able to migrate using dm-convert
- For customers heavily invested in Kubernetes, **Config Connector** is a Terraform alternative. It enables them to manage GCP resources with Kubernetes YAMLs. However it doesn't have feature parity with Terraform yet.

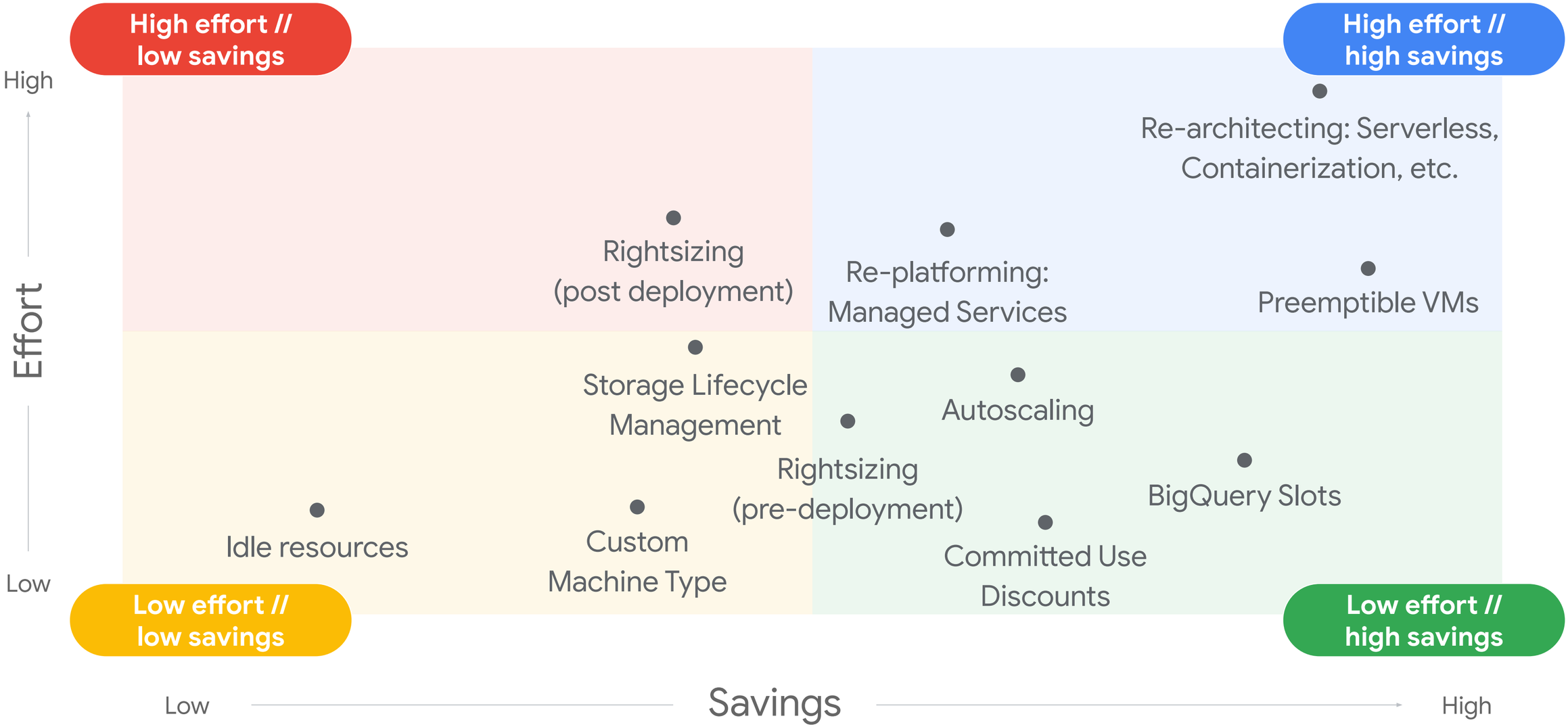
Cloud Deployment Manager

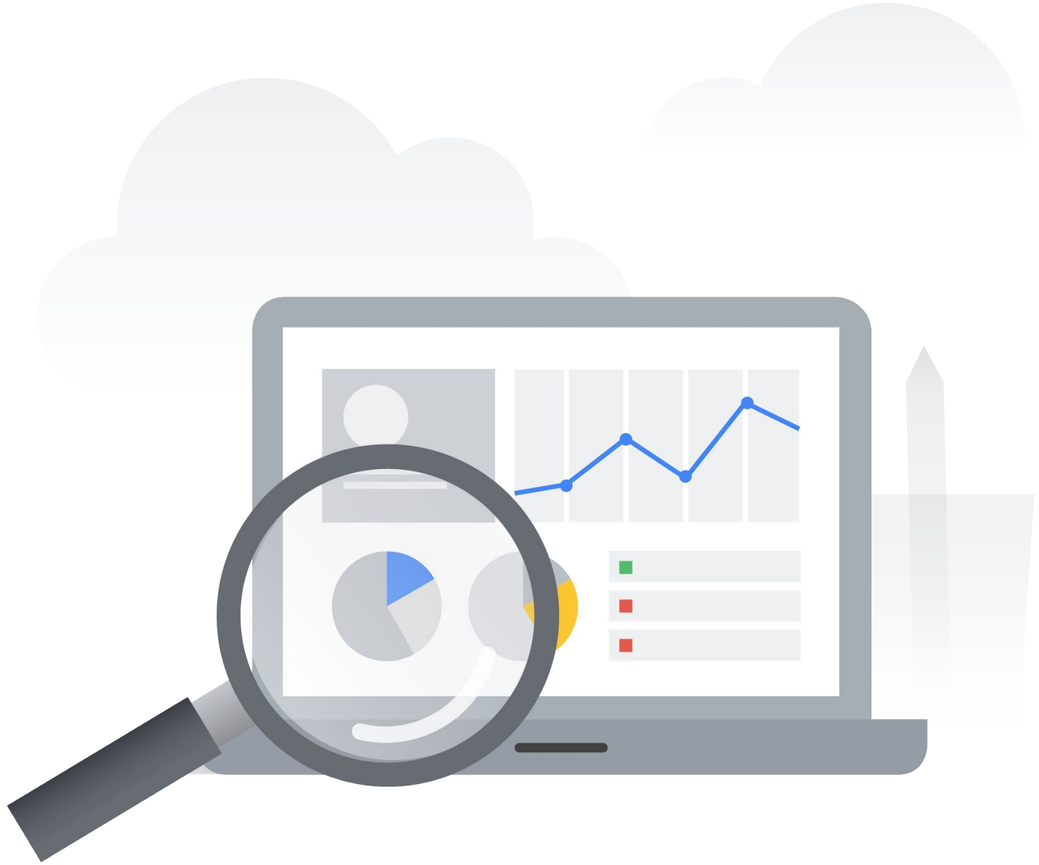
vs



Terraform

Cost Optimization Matrix

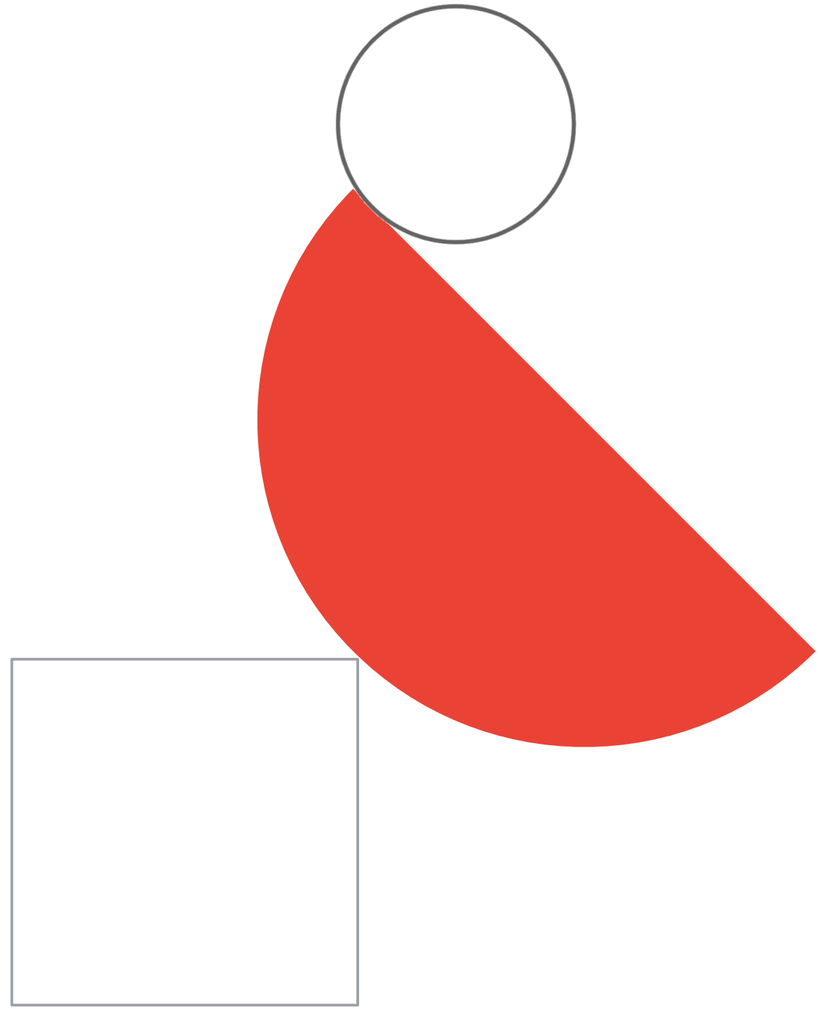




Monitoring and KPIs

Tools available for monitoring and analyzing KPIs.

Business continuity in GCP



Business Continuity = Planning for failure

A well-designed system can answer the question: "What happens when a zone or region has a 1, 5, 10, or 30 minute outage?" This should be considered at many layers, including:

- What will my customers experience during an outage?
- How will I detect that an outage is happening?
- What happens to my application during an outage?
- What happens to my data during an outage?
- What happens to my other applications due to an outage (due to cross-dependencies)?
- What do I need to do in order to recover after an outage is resolved? Who does it?
- Who do I need to notify about an outage, within what time period?

Resource	Examples	Availability design goal	Implied downtime
Zonal	Compute Engine, Persistent Disk	99.9%	8.75 hours / year
Regional	Regional Cloud Storage, Replicated Persistent Disk, Regional Google Kubernetes Engine	99.99%	52 minutes / year

High Availability for...

- Compute Engine ⇒ ?
- GKE ⇒ ?
- App Engine ⇒ ?
- Cloud SQL ⇒ ?
- Cloud Spanner ⇒ ?
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ ?
- App Engine ⇒ ?
- Cloud SQL ⇒ ?
- Cloud Spanner ⇒ ?
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ Regional clusters, Load Balancers
- App Engine ⇒ ?
- Cloud SQL ⇒ ?
- Cloud Spanner ⇒ ?
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ Regional clusters, Load Balancers
- App Engine ⇒ automatic scaling, regional resource
- Cloud SQL ⇒ ?
- Cloud Spanner ⇒ ?
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ Regional clusters, Load Balancers
- App Engine ⇒ automatic scaling, regional resource
- Cloud SQL ⇒ HA “checkbox” for multi-zone
- Cloud Spanner ⇒ ?
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ Regional clusters, Load Balancers
- App Engine ⇒ automatic scaling, regional resource
- Cloud SQL ⇒ HA “checkbox” for multi-zone
- Cloud Spanner ⇒ Multi-instance deployment with automatic failover
- Cloud Storage => ?

High Availability for...

- Compute Engine ⇒ regional MIGs, Load Balancers
- GKE ⇒ Regional clusters, Load Balancers
- App Engine ⇒ automatic scaling, regional resource
- Cloud SQL ⇒ HA “checkbox” for multi-zone
- Cloud Spanner ⇒ Multi-instance deployment with automatic failover
- Cloud Storage => regional / dual/ multi-region bucket, optional replication

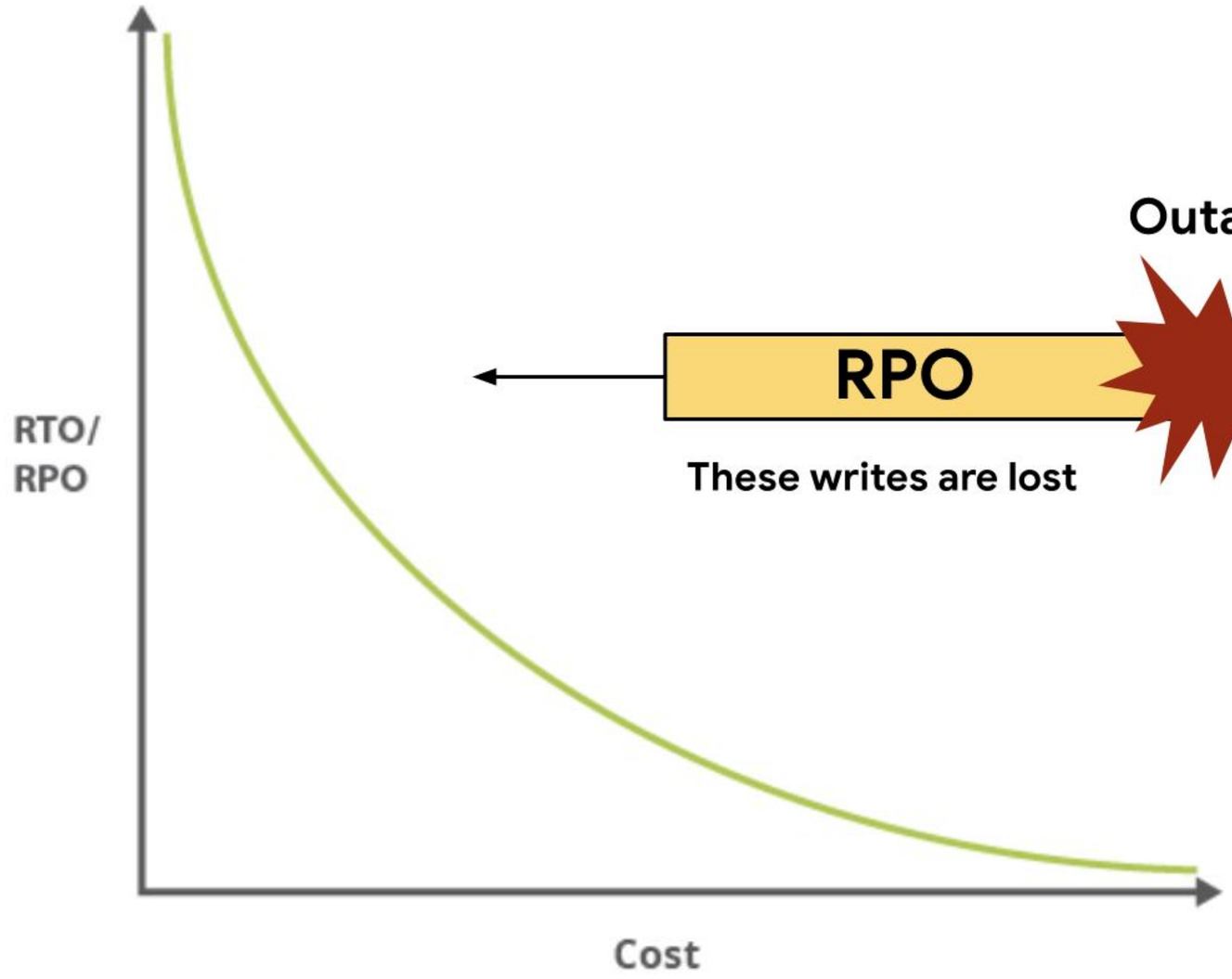
TIP

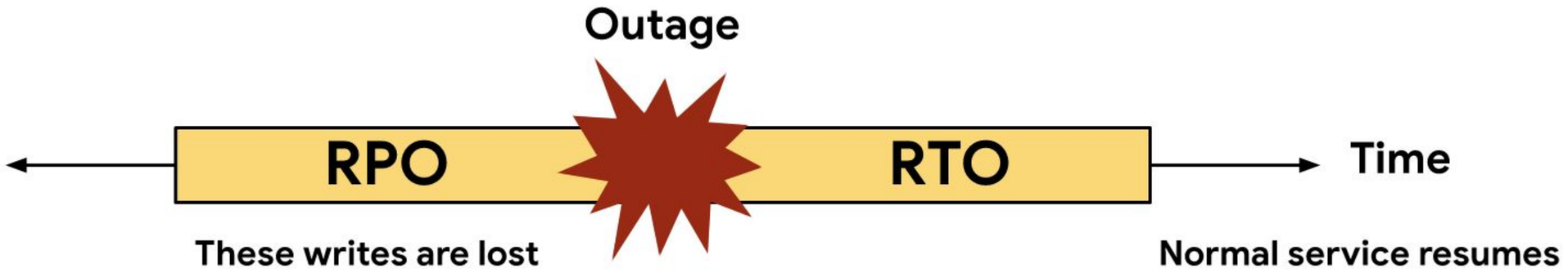
Product Reference for Business Continuity

TIP

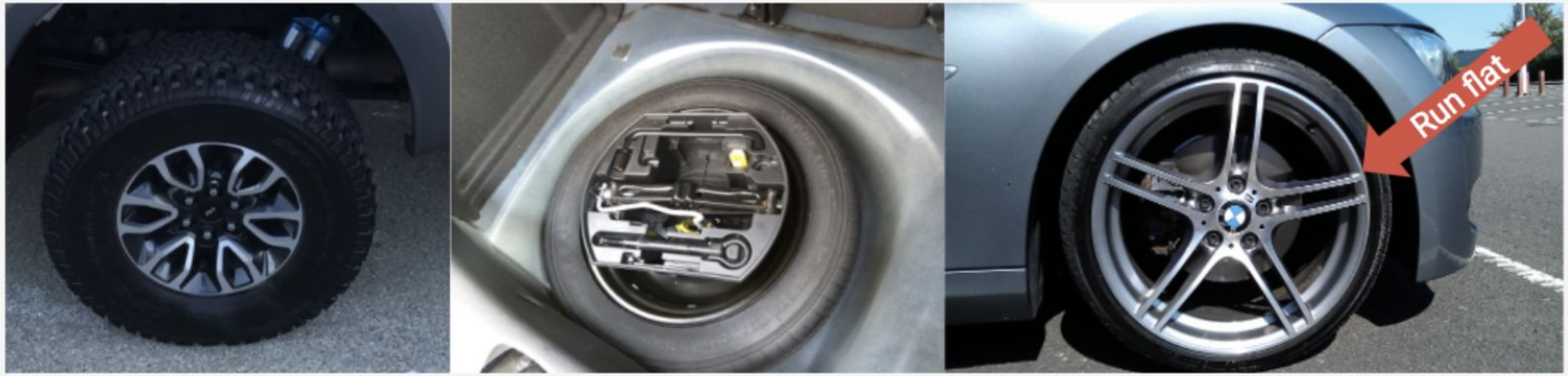
The more managed a service is, the easier it is to ensure at least 99,99% SLA

Disaster Recovery architecture is driven by RPO and RTO





Disaster Recovery architecture is driven by RPO and RTO



Application criticality	% of Apps	Example apps	Zone outage	Region outage
Tier 1 (most important)	5%	Typically global or external customer-facing applications such as real-time payments and eCommerce storefronts.	RTO Zero RPO Zero	RTO Zero RPO Zero
Tier 2	35%	Typically regional applications or important internal applications such as CRM or ERP.	RTO 15mins RPO 15mins	RTO 1hr RPO 1hr
Tier 3 (least important)	60%	Typically team or departmental applications, such as back office, leave booking, internal travel, accounting, and HR.	RTO 1hr RPO 1hr	RTO 12hrs RPO 12hrs

EXAMPLE: Disaster Recovery for Cloud SQL instance...

- Cold:
- Warm:
- Hot:

EXAMPLE: Disaster Recovery for Cloud SQL instance...

- Cold: backups offloaded to another region
- Warm:
- Hot:

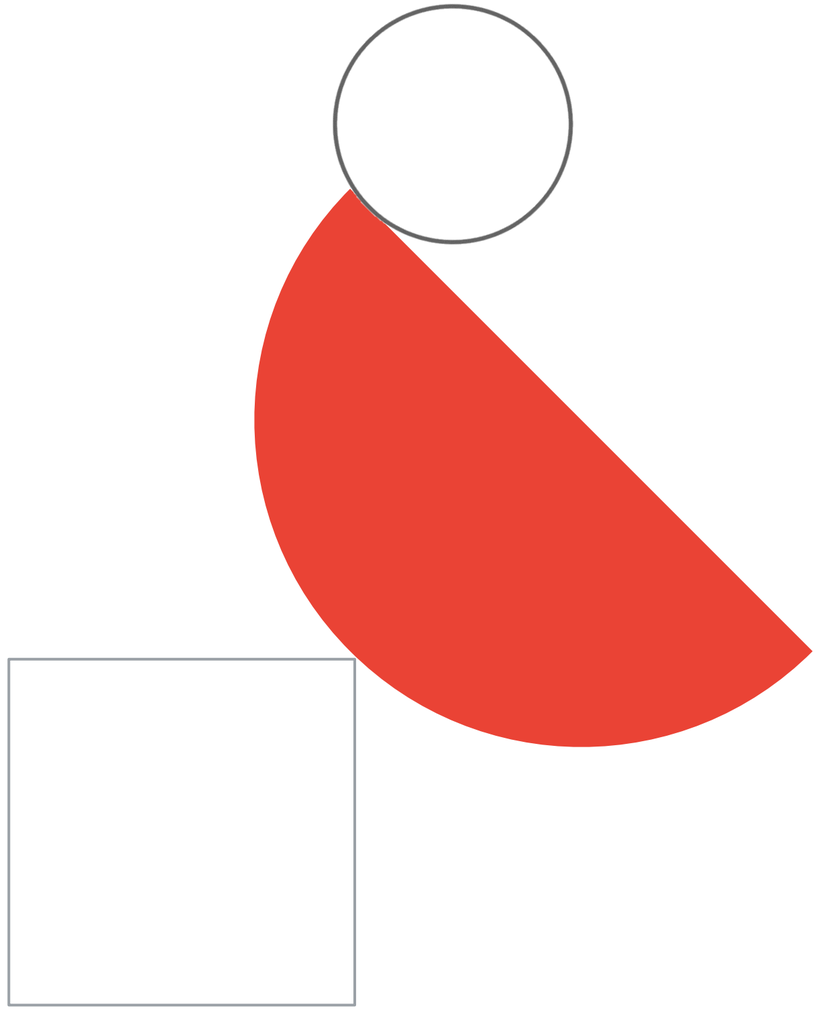
EXAMPLE: Disaster Recovery for Cloud SQL instance...

- Cold: backups offloaded to another region
- Warm: read-only replicas in a different region with asynchronous replication
- Hot:

EXAMPLE: Disaster Recovery for Cloud SQL instance...

- Cold: backups offloaded to another region
- Warm: read-only replicas in a different region with asynchronous replication
- Hot: ... none available out of the box. Alternatives:
  - Migrate to Cloud Spanner ?
  - MySQL on 2 GCE VMs (NOT Cloud SQL) with DRBD, load balancer in front and automatic failover. Details here.
  - Other Do-It-Yourself options

“Where should I run my stuff?” game



Where should I run my stuff?

- Containers =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources => GCE, GKE

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources => GCE, GKE
- Prefer managed / serverless / focus on code =>

Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources => GCE, GKE
- Prefer managed / serverless / focus on code => Cloud Functions, Cloud Run, App Engine

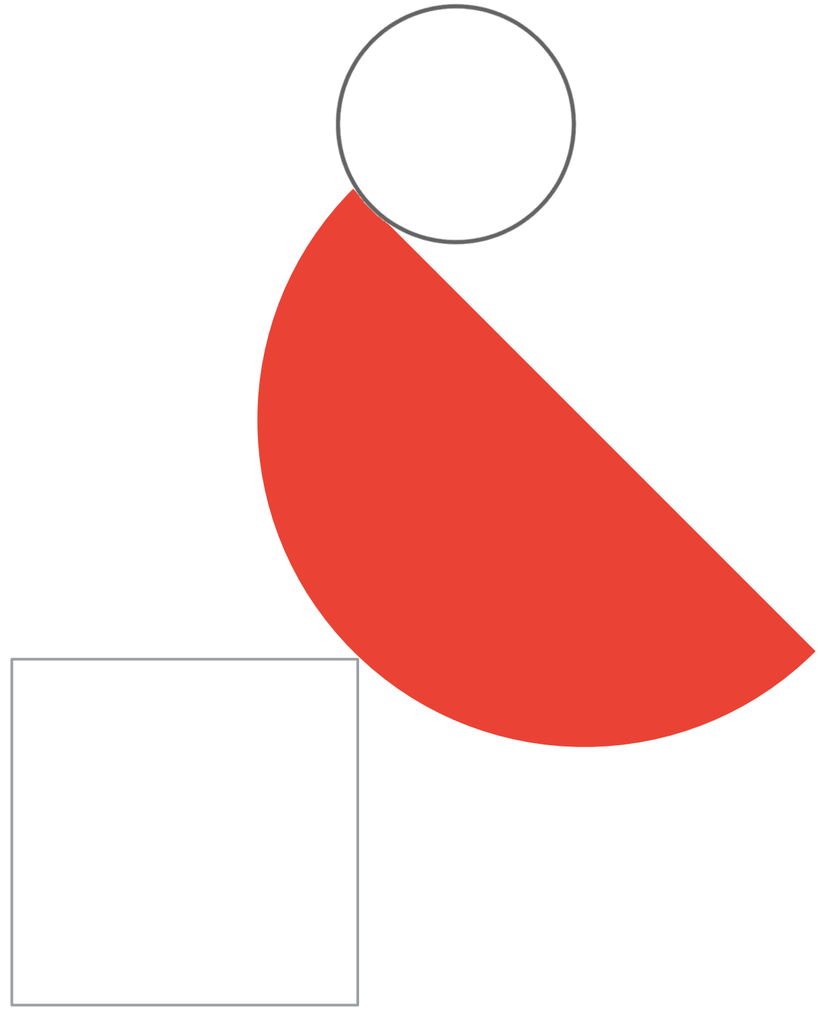
Where should I run my stuff?

- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources => GCE, GKE
- Prefer managed / serverless / focus on code => Cloud Functions, Cloud Run, App Engine
- Billing based on resources =>

Where should I run my stuff?

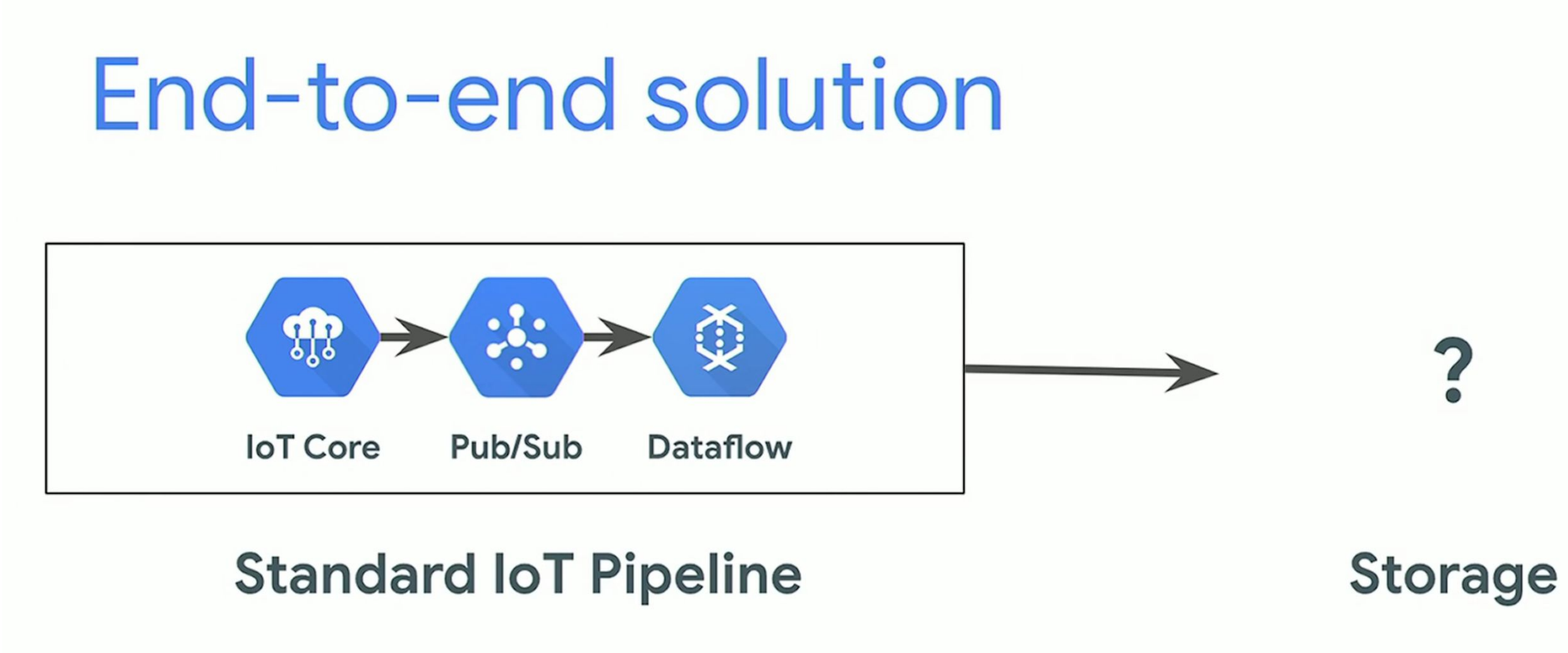
- Containers => GKE, Cloud Run, (App Engine)
- Specific OS, licensing, backup => GCE
- Hybrid, multi-cloud, portability => GKE (Anthos), Cloud Run
- Web applications => Cloud Run (App Engine)
- Event-based processing => Cloud Functions
- Want to squeeze every drop from provisioned resources => GCE, GKE
- Prefer managed / serverless / focus on code => Cloud Functions, Cloud Run, App Engine
- Billing based on resources => GCE, GKE

TerramEarth case study analysis



TerramEarth

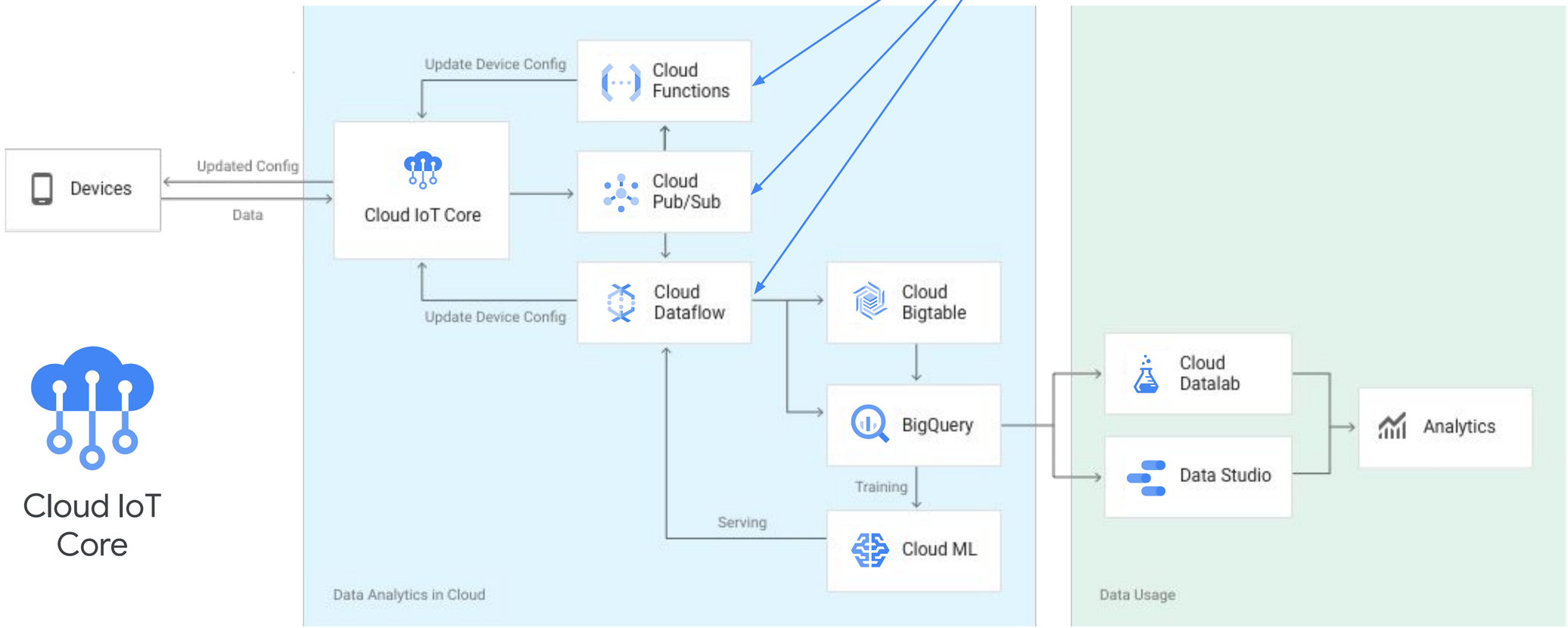
IoT pipeline



Internet of things (IoT)

TIP

Do you know how each of these services contribute to the IoT solution?



Cloud IoT Core

Proposed Technical Solutions

TerramEarth

- Similar to this architecture -> it's good to have a look at this solution.
- Telemetry streaming from vehicles: IoT Core service for managing devices and securing traffic.
- Critical IoT pipeline (near real-time checks of streaming telemetry data):
  - IoT Core -> Pub/Sub -> Dataflow (stream) -> BigQuery / BigTable (data storage) -> Application / BigQuery (query engine, since BigQuery can read from different sources, including BigTable)
- Analytics pipeline (daily batch uploads of telemetry from vehicles):
  - IoT Core -> Pub/Sub OR GCS -> Dataflow (batch) -> BigQuery (for analyzing this data) + optionally BigQuery ML (ML for predicting failures of vehicles based on data stored in BigQuery).
- Compute env: no concrete option. Possibilities that make most sense are: App Engine (portal for partners), GKE and Cloud Run.
- Interconnect: Possibly with 99.99% availability via multiple connections in different regions, using Cloud Router
- API creation, deployment and management: Apigee
- CI/CD:
  - Code repo: Cloud Source Repositories, Artifacts repo: Artifact Registry (which replaced old service: Container Registry)
  - Pipeline: GCP-native: Cloud Build + Cloud Deploy (new service, probably not yet covered in the PCA exam), 3rd party: Jenkins, Spinnaker
- Security-focused services: Binary Authorization, Web Security Scanner, Container Threat Detection, Data Loss Prevention, Organization Policy Service
- Secret management: Secret Manager
- Encryption key management: Key Management Service (KMS), possibly with HSM / EKM. What is envelope encryption?
- Network Monitoring:
  - VPC Flow logs -> Cloud Logging
  - Firewall rule logs -> Firewall Insights

[Terramearth case study] Diagnostic Question #1

For this question, refer to the TerramEarth case study. TerramEarth has a legacy web application that you cannot migrate to cloud. However, you still want to build a cloud-native way to monitor the application. If the application goes down, you want the URL to point to a "Site is unavailable" page as soon as possible. You also want your Ops team to receive a notification for the issue. You need to build a reliable solution for minimum cost.

What should you do?

A. Create a scheduled job in Cloud Run to invoke a container every minute. The container will check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

B. Create a cron job on a Compute Engine VM that runs every minute. The cron job invokes a Python program to check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

C. Create a Cloud Monitoring uptime check to validate the application URL. If it fails, put a message in a Pub/Sub queue that triggers a Cloud Function to switch the URL to the "Site is unavailable" page, and notify the Ops team.

D. Use Cloud Error Reporting to check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

[Terramearth case study] Diagnostic Question #1

For this question, refer to the TerramEarth case study. TerramEarth has a legacy web application that you cannot migrate to cloud. However, you still want to build a cloud-native way to monitor the application. If the application goes down, you want the URL to point to a "Site is unavailable" page as soon as possible. You also want your Ops team to receive a notification for the issue. You need to build a reliable solution for minimum cost.

What should you do?

A. Create a scheduled job in Cloud Run to invoke a container every minute. The container will check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

B. Create a cron job on a Compute Engine VM that runs every minute. The cron job invokes a Python program to check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

C. Create a Cloud Monitoring uptime check to validate the application URL. If it fails, put a message in a Pub/Sub queue that triggers a Cloud Function to switch the URL to the "Site is unavailable" page, and notify the Ops team.

D. Use Cloud Error Reporting to check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.

[Terramearth case study] Diagnostic Question #2

For this question, refer to the TerramEarth case study. TerramEarth has about 1 petabyte (PB) of vehicle testing data in a private data center. You want to move the data to Cloud Storage for your machine learning team. Currently, a 1-Gbps interconnect link is available for you. The machine learning team wants to start using the data in a month.

What should you do?

A. Request Transfer Appliances from Google Cloud, export the data to appliances, and return the appliances to Google Cloud.

B. Configure the Storage Transfer service from Google Cloud to send the data from your data center to Cloud Storage.

C. Make sure there are no other users consuming the 1Gbps link, and use multi-thread transfer to upload the data to Cloud Storage.

D. Export files to an encrypted USB device, send the device to Google Cloud, and request an import of the data to Cloud Storage.

[Terramearth case study] Diagnostic Question #2

For this question, refer to the TerramEarth case study. TerramEarth has about 1 petabyte (PB) of vehicle testing data in a private data center. You want to move the data to Cloud Storage for your machine learning team. Currently, a 1-Gbps interconnect link is available for you. The machine learning team wants to start using the data in a month.

What should you do?

A. Request Transfer Appliances from Google Cloud, export the data to appliances, and return the appliances to Google Cloud.

B. Configure the Storage Transfer service from Google Cloud to send the data from your data center to Cloud Storage.

C. Make sure there are no other users consuming the 1Gbps link, and use multi-thread transfer to upload the data to Cloud Storage.

D. Export files to an encrypted USB device, send the device to Google Cloud, and request an import of the data to Cloud Storage.

[Terramearth case study] Diagnostic Question #3

For this question, refer to the TerramEarth case study. You have broken down a legacy monolithic application into a few containerized RESTful microservices.
You want to run those microservices on Cloud Run. You also want to make sure the services are highly available with low latency to your customers.

What should you do?

A. Deploy Cloud Run services to multiple zones. Create Cloud Endpoints that point to the services. Create a global HTTP(S) Load Balancing instance and attach the Cloud Endpoints to its backend.

B. Deploy Cloud Run services to multiple regions. Create serverless network endpoint groups pointing to the services. Add the serverless NEGs to a backend service that is used by a global HTTP(S) Load Balancing instance.

C. Deploy Cloud Run services to multiple regions. In Cloud DNS, create a latency-based DNS name that points to the services.

D. Deploy Cloud Run services to multiple zones. Create a TCP/IP global load balancer. Add the Cloud Run Endpoints to its backend service.

[Terramearth case study] Diagnostic Question #3

For this question, refer to the TerramEarth case study. You have broken down a legacy monolithic application into a few containerized RESTful microservices.
You want to run those microservices on Cloud Run. You also want to make sure the services are highly available with low latency to your customers.

What should you do?

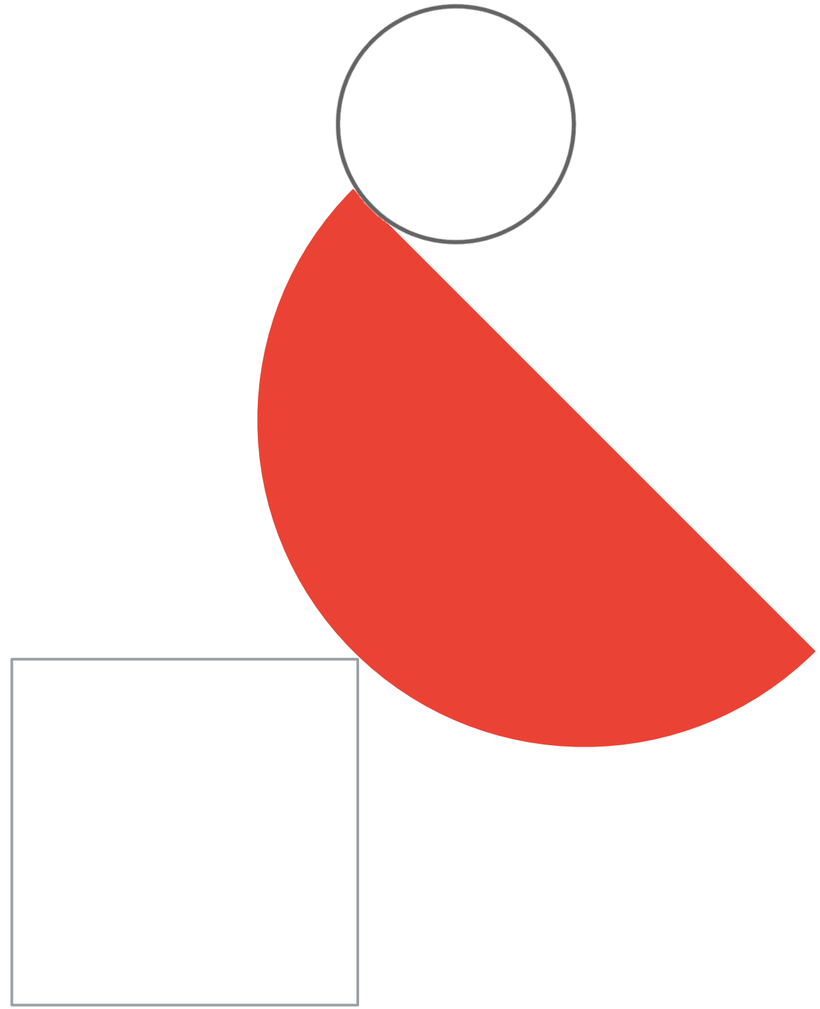
A. Deploy Cloud Run services to multiple zones. Create Cloud Endpoints that point to the services. Create a global HTTP(S) Load Balancing instance and attach the Cloud Endpoints to its backend.

B. Deploy Cloud Run services to multiple regions. Create serverless network endpoint groups pointing to the services. Add the serverless NEGs to a backend service that is used by a global HTTP(S) Load Balancing instance.

C. Deploy Cloud Run services to multiple regions. In Cloud DNS, create a latency-based DNS name that points to the services.

D. Deploy Cloud Run services to multiple zones. Create a TCP/IP global load balancer. Add the Cloud Run Endpoints to its backend service.

[optional] Links to useful materials



Optional materials 1

[READING]

- go through below sketchnotes:
  a. GCS
  b. Cloud SQL
  c. Cloud Spanner
  d. BigQuery
  e. BigTable
  f. Choosing a Database option
  g. Data Transfer Options
  h. [IMPORTANT] Cloud Operations Suite (Monitoring, Logging, Trace, Profiler, Debugger)
  i. Network Intelligence Center
  j. Dataproc
  k. Dataflow
  l. Data Loss Prevention
  m. Cloud compliance
  n. Data security
  o. Cloud security foundations

Optional materials 2

- Read about private access options for GCP services.
- Go through this interactive Developer cheat sheet and make sure you are familiar with most of the services from different areas, mainly: Compute; Storage; Database; Networking; DevOps CI/CD; Identity and Security; Migration to Google Cloud.
- Play around with this online Architecture Diagramming Tool, which can:
  a. Help you sooo much in your future work as a cloud architect. No more 3rd party tools, no more searching for the appropriate GCP icons
  b. Visualize and memorize some of the common architectural diagrams. You can create different standard solutions like the one below with just one click of a mouse using the "Diagrams" area. Since many PCA questions require you to choose the right service setup, knowing those universal patterns will surely be helpful.
- Useful DLP blog post.
- Read about Cloud NAT service
- Secret Manager Best practices

Optional materials 3

[VIDEOS]

- [HIGHLY RECOMMENDED] Example of how to define architecture for a serverless finance system: Designing a serverless finance system on Google Cloud
- Why you shouldn't aim at 100% uptime and what is an error budget: Why you shouldn't aim for 100% uptime
- SLIs, SLOs, SLAs in 8 mins: SLIs, SLOs, SLAs, oh my! (class SRE implements DevOps)
- DevOps vs SRE: What's the Difference Between DevOps and SRE? (class SRE implements DevOps)
- Cloud Operations Suite services: Cloud operations spotlight
- Private Service Connect: What is Private Service Connect?
- How to secure your cloud environment: How to secure your cloud environment
- Securing customer data: Securing customer data
- Network Connectivity Test: Get started with Connectivity Test in Network Intelligence Center
- Apigee: Intro to Apigee API management
- Apigee X: Introduction to Apigee X
- Securing hardware in GCP: Securing your hardware for your software
- Firewall Insights: Get Started with Firewall Insights in Network Intelligence Center
- Best Practices for Cloud Monitoring: Best Practices for Cloud Monitoring

Optional materials 4

- Automating Cloud Monitoring dashboards: Automating Cloud Monitoring dashboards
- Top use-cases for serverless: Top use cases to start your serverless journey
- How to build APIs for serverless workloads with Google Cloud

[PODCASTS]

- Cloud Audit Logging
- The art of SLOs
- SRE vs DevOps
- Resiliency at Shopify

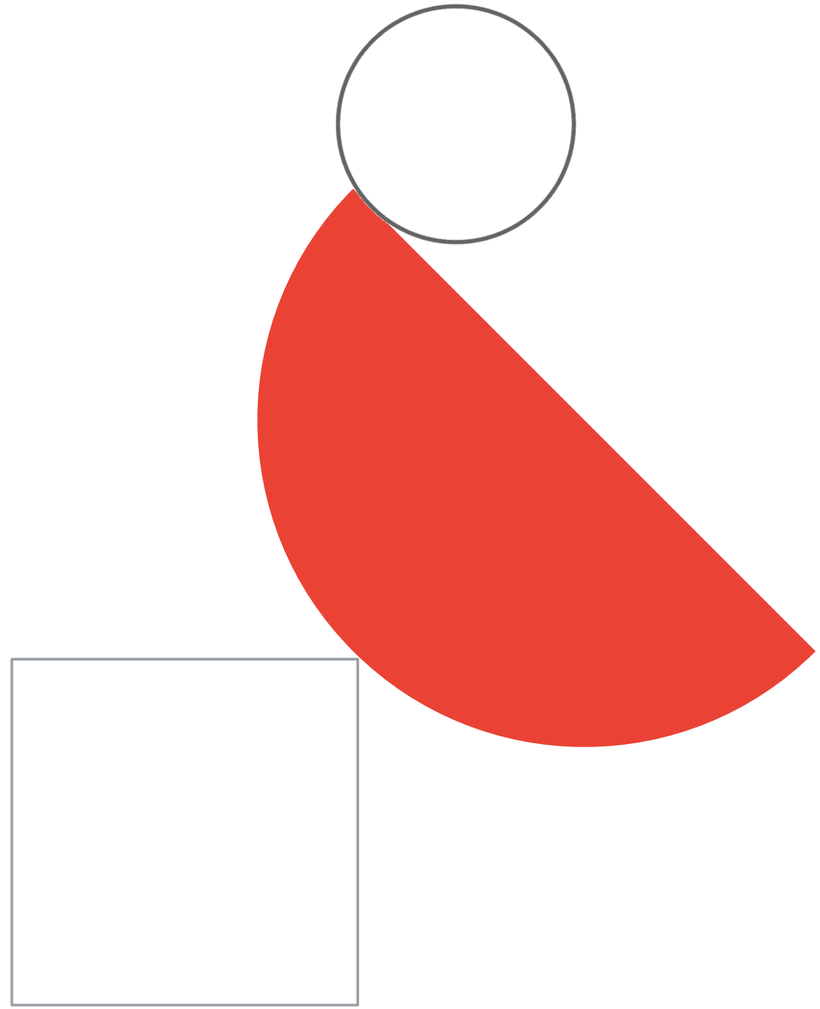
[DEEP DIVES]

- Web Security Scanner overview.
- Technical overview of Internet of Things.
- [video] Managing IoT Storage with Google's Cloud Platform (Google I/O'19).
- Chapters 2, 6, 8 and 17 (1st half) from Google SRE book.

Survey time!

We value your feedback on this course and ask that you take a few minutes to fill out the survey for this course. You will find the link in your classroom, and can ask your instructor if you have any questions.

BONUS CONTENT



Very subjective way to evaluate if you're ready...

		Professional Cloud Architect (PCA)	
0: none	not covered on the exam at all		
1: basics	high-level functionality and use-cases		
2: medium	basics + prerequisites, limitations, common IAM roles, ability to integrate with other services, most common architectures		
3: advanced	medium + being able to deploy, troubleshoot and manage		
4: expert	advanced + know every detail about the service in complex configurations - huge scale, HA, DR etc		
		Recommended minimum knowldege level for PCA	**My knowledge level (self-assesment)**
	Compute Environment		
	Compute Engine (GCE)	3: advanced	0: none
	Managing access to VMs (OS Login etc)	2: medium	0: none
	Persistent Disks	3: advanced	0: none
	GCE Instance Groups	2: medium	0: none
	Multi-NIC VMs	1: basics	0: none
	App Engine flexible environment	2: medium	0: none
	App Engine standard environment	2: medium	0: none
	Cloud GPUs	1: basics	0: none
	Migrate for Compute Engine	1: basics	0: none
	VMware Engine	1: basics	0: none
	Cloud Run	2: medium	0: none
	Cloud Functions	2: medium	0: none
	Bare Metal Solution	1: basics	0: none
	GCVE	1: basics	0: none
	Preemptible VMs	2: medium	0: none
	Sole-tenant Nodes	2: medium	0: none

Google Cloud

LINK - switch to "PCA" tab

Bonus quiz

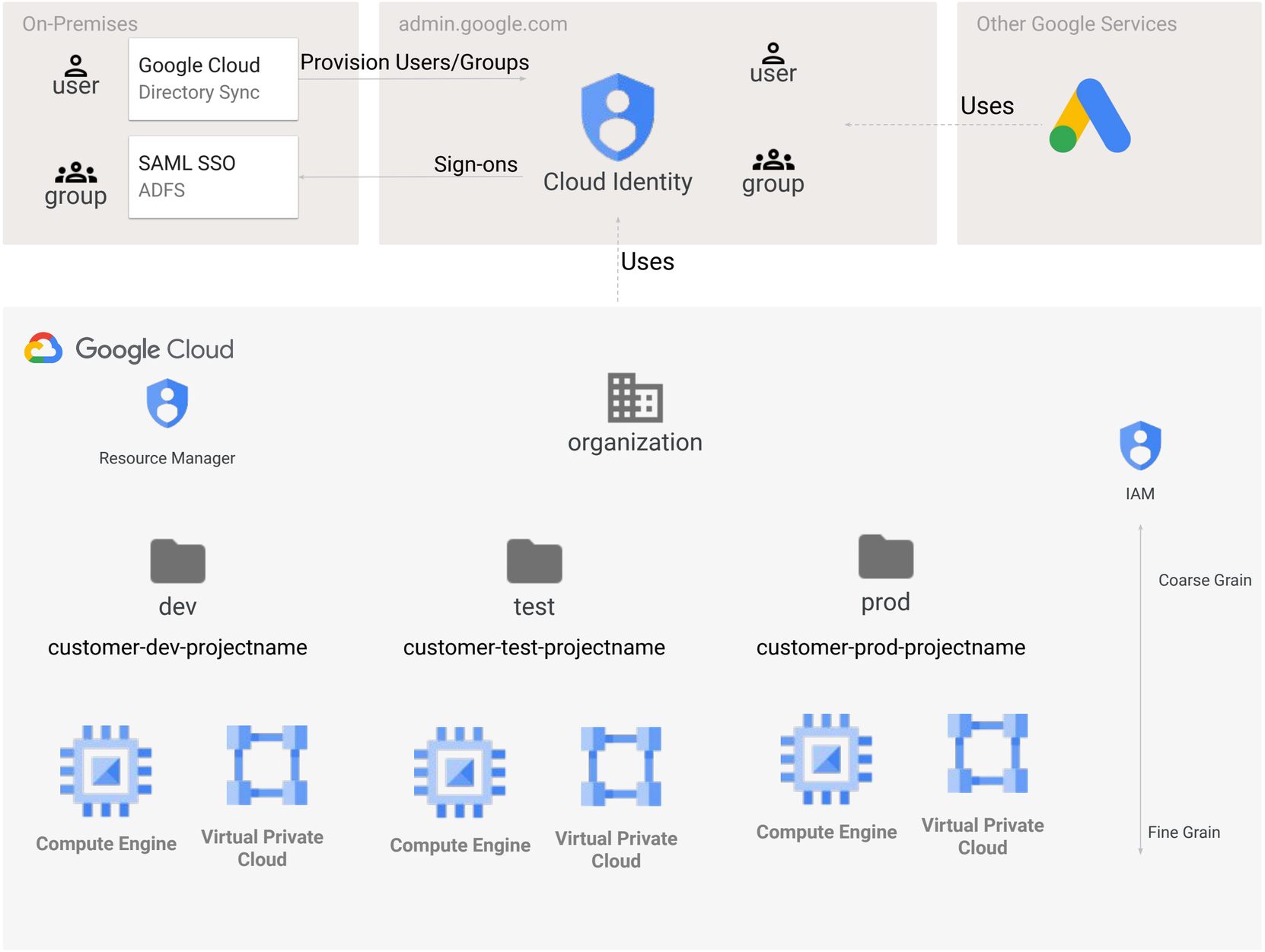
Pre-exam quiz 1

Pre-exam quiz 2

~30 exam-like questions which should help you evaluate your exam-readiness.

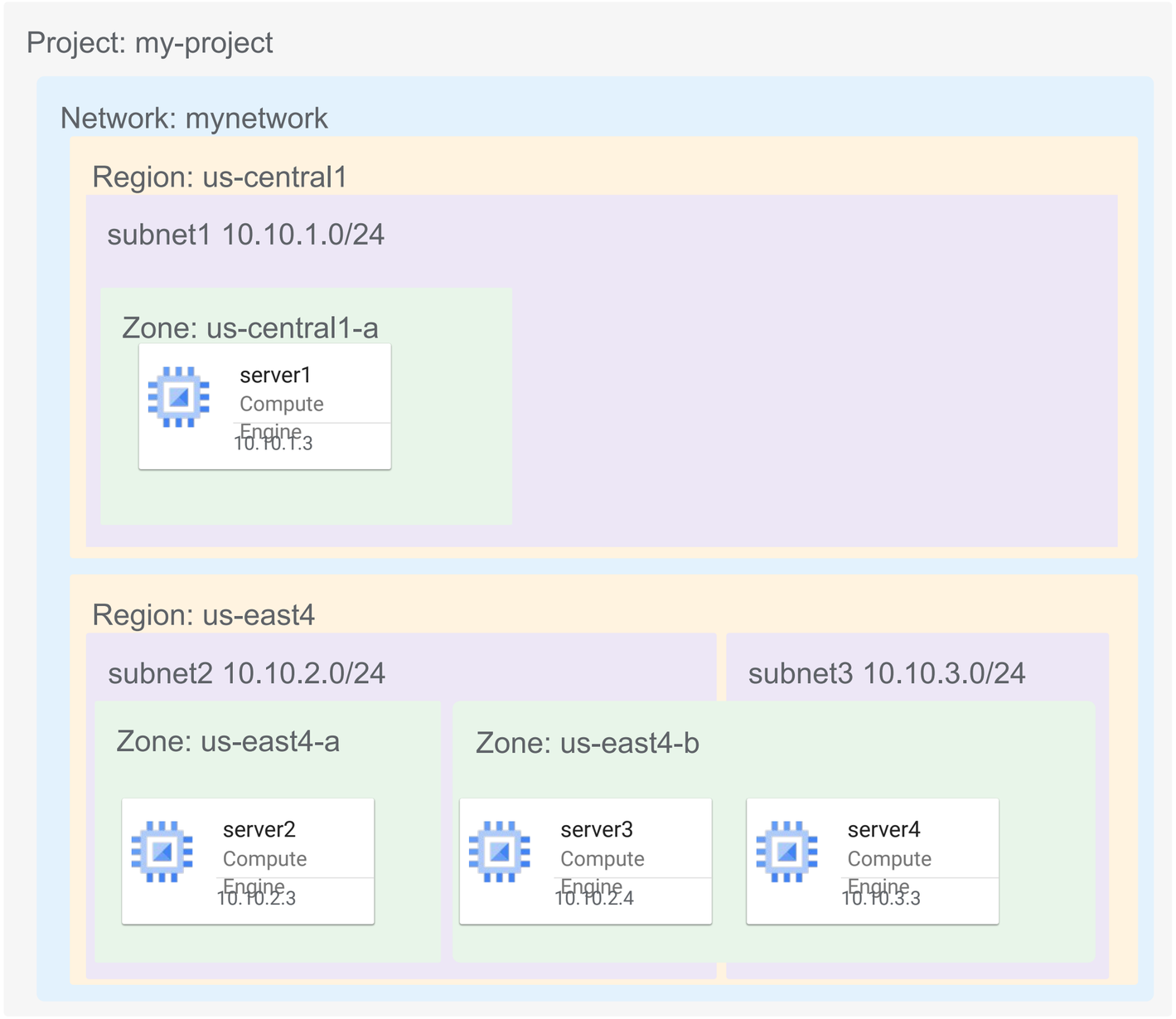
Study Cards

PCA Study Cards - IAM & Cloud Identity



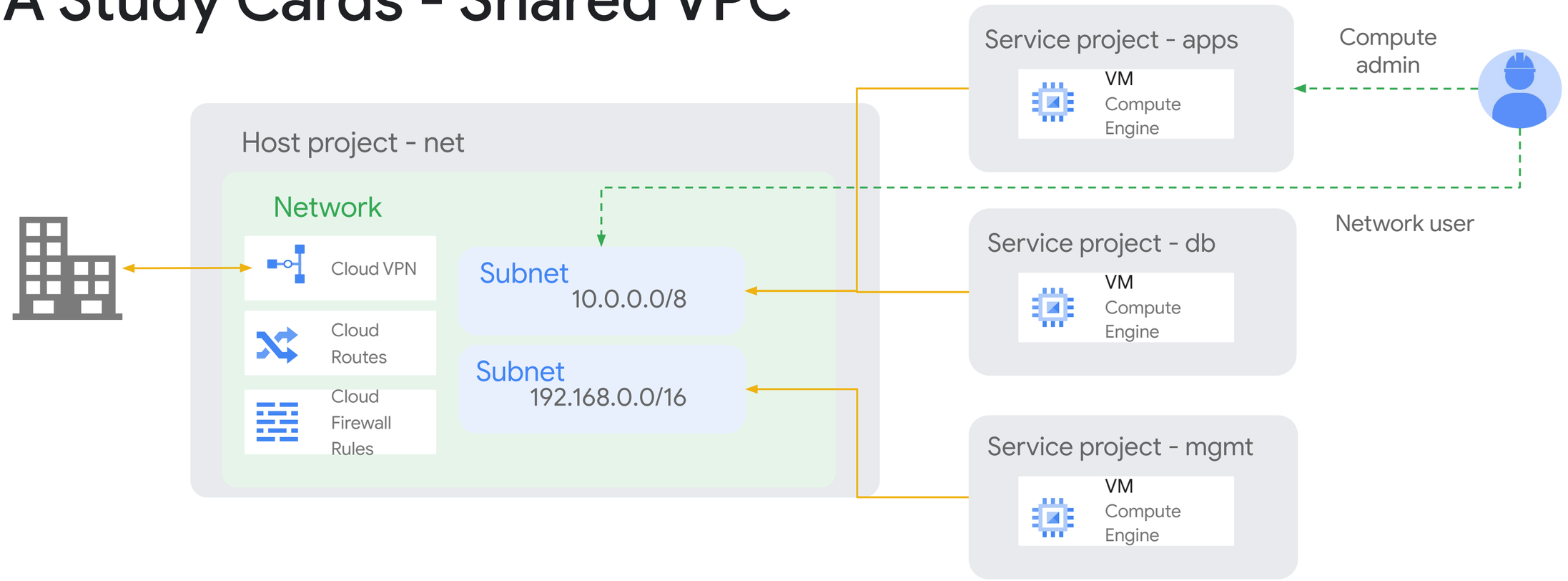
- SSO and GCDS are mutually exclusive (although often used together)
- Resource Manager is where your hierarchy is defined
  - An organization is technically optional
  - Folders are optional as well
    - Can be nested
- IAM Permissions can be assigned at any level (org, folder, project, resource)
  - Lower generally is least privilege
- Three types of roles: Basic (primitive), Pre-defined, Custom
  - Basic (owner, editor, viewer) are generally limited to non-production or special cases
  - Pre-defined are most common
  - Custom have some limitations (not all permissions, limited number)
- Best practices
  - Assign to groups rather than user accounts
  - Assign lowest level practical
  - Assign fewest permissions possible to “get the job done”
  - Assigning at a higher level effects all current and future resources

PCA Study Cards - networking basics



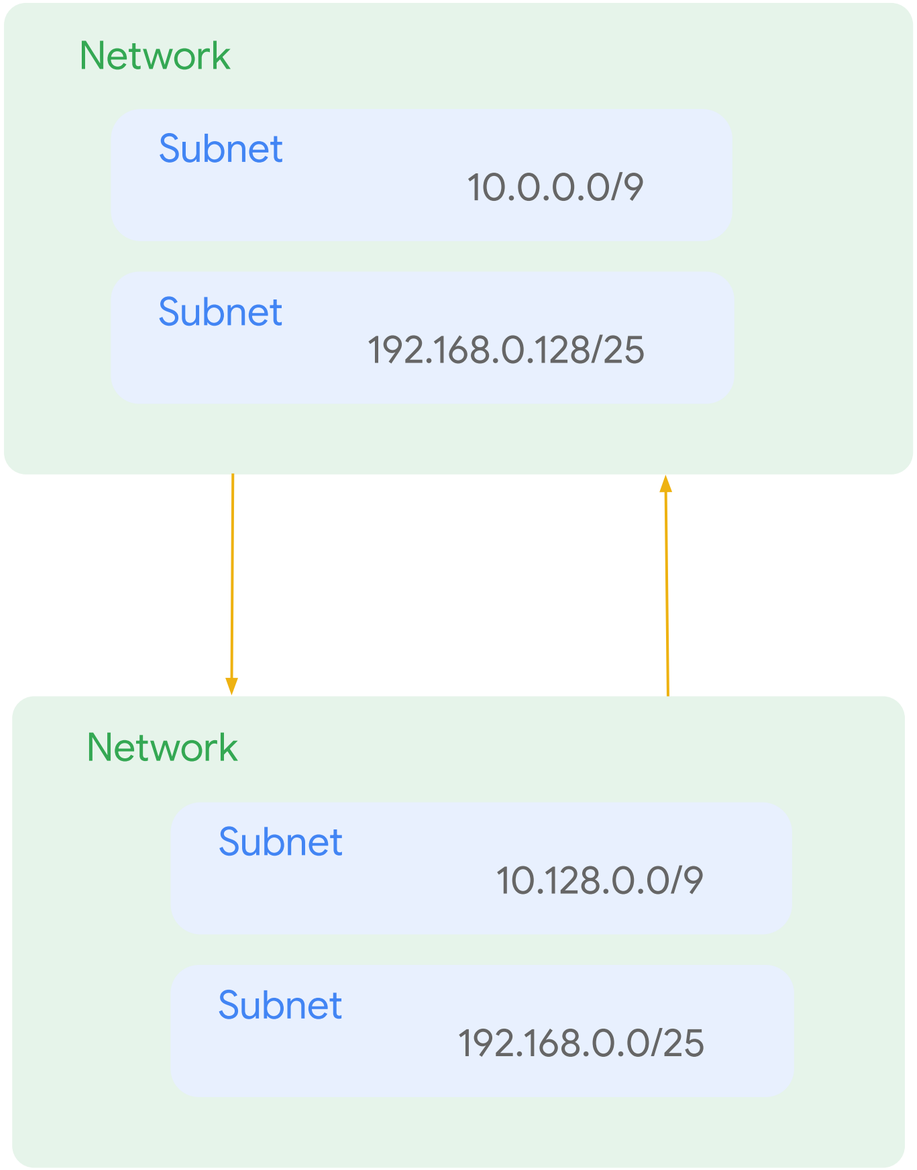
- A VPC belongs to 1 project
- A VPC can be present in every region across GCP (and is in the default configuration)
- No additional configuration is required for servers to communicate globally (VPNs or routers)
- A subnet crosses zones within a region, but cannot cross regional boundaries
- Implied Firewall Rules (65535):
  - Allow all egress traffic
  - Deny all ingress traffic
- Default rules
  - Allow SSH, ICMP, RDP
  - Block SMTP Traffic
- Lower the number of firewall rule the higher the priority (1 > 10)
- Components of a firewall rule:
  - Direction (ingress / egress)
  - Priority (0 to 65535)
  - Action (Allow / Deny)
  - Enforcement Status
  - Target
  - Source
  - Protocol
  - Log (1 or 0)

PCA Study Cards - Shared VPC



- ➔ Shared VPC is the most common way to share networks. Allows you the flexibility of having many projects (good for security / billings / etc) without the overhead of managing a lot of VPCs.
- ➔ Allows you to setup a robust network in the host project and share subnet(s) with service projects.
- ➔ Allows good security segmentation as admins on compute nodes don't need to admin network functions (only need user permissions).
- ➔ Connectivity to other networks (VPN and interconnects) and firewall rules can be centrally managed in the host project.
- ➔ Host and service projects ***must*** belong to the same GCP organization

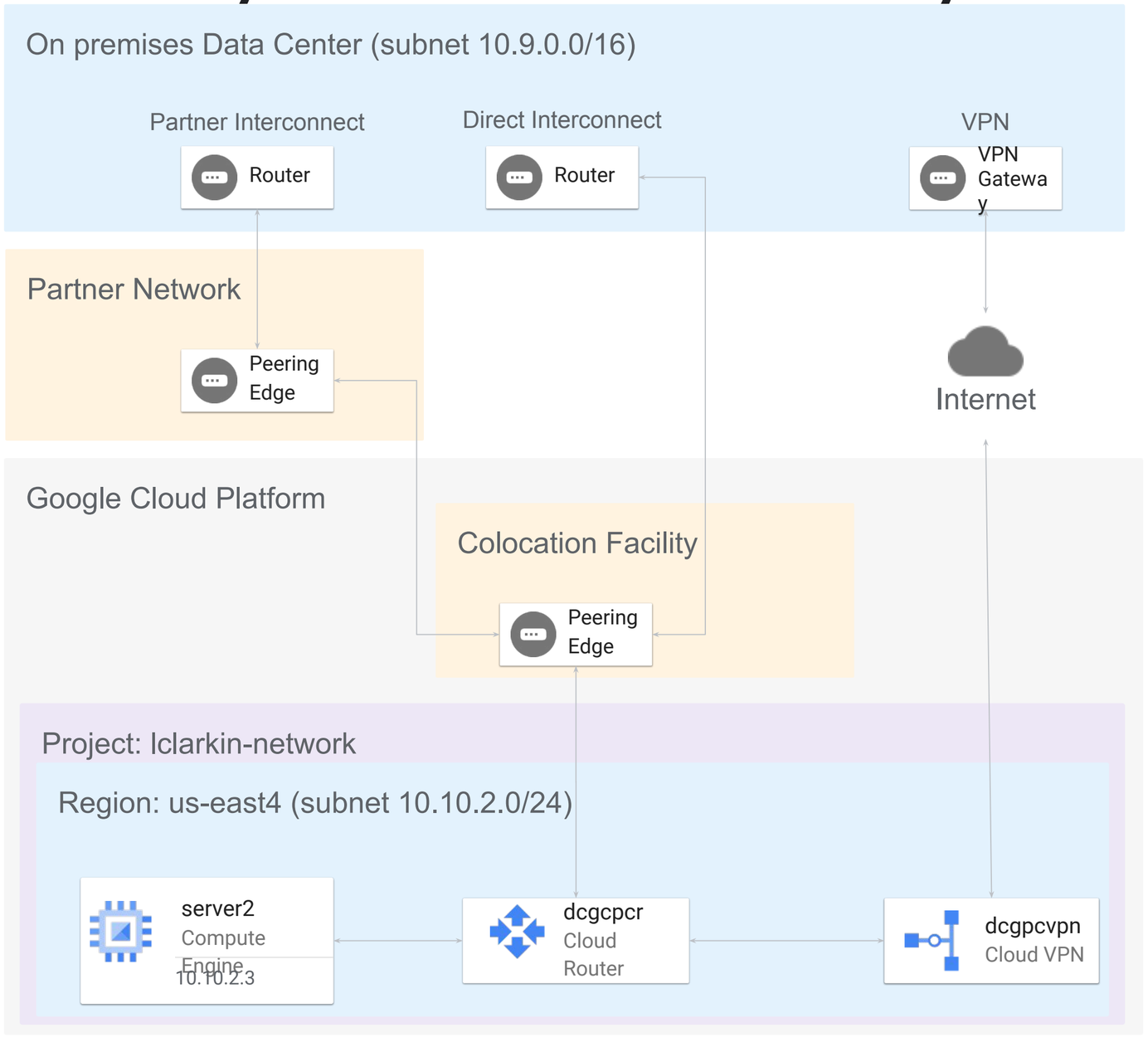
PCA Study Cards - VPC Peering



- ➔ Peering works both within and between GCP organizations
- ➔ When setting up the peering you determine which subnet(s) to publish routes to
- ➔ Administrators on both sides must configure the peering in order for it to work
- ➔ The peering between the networks is **not** transitive, so traffic will not route to any other networks peered
- ➔ Links between the networks are high throughput and very low latency (unlike connecting via a VPN)
- ➔ IP Networks ***cannot*** overlap

Note: Starting to see peering as part of the solution for GCP Products: Apigee X and Datastream configurations both require peering as part of the setup

PCA Study Cards - Connectivity

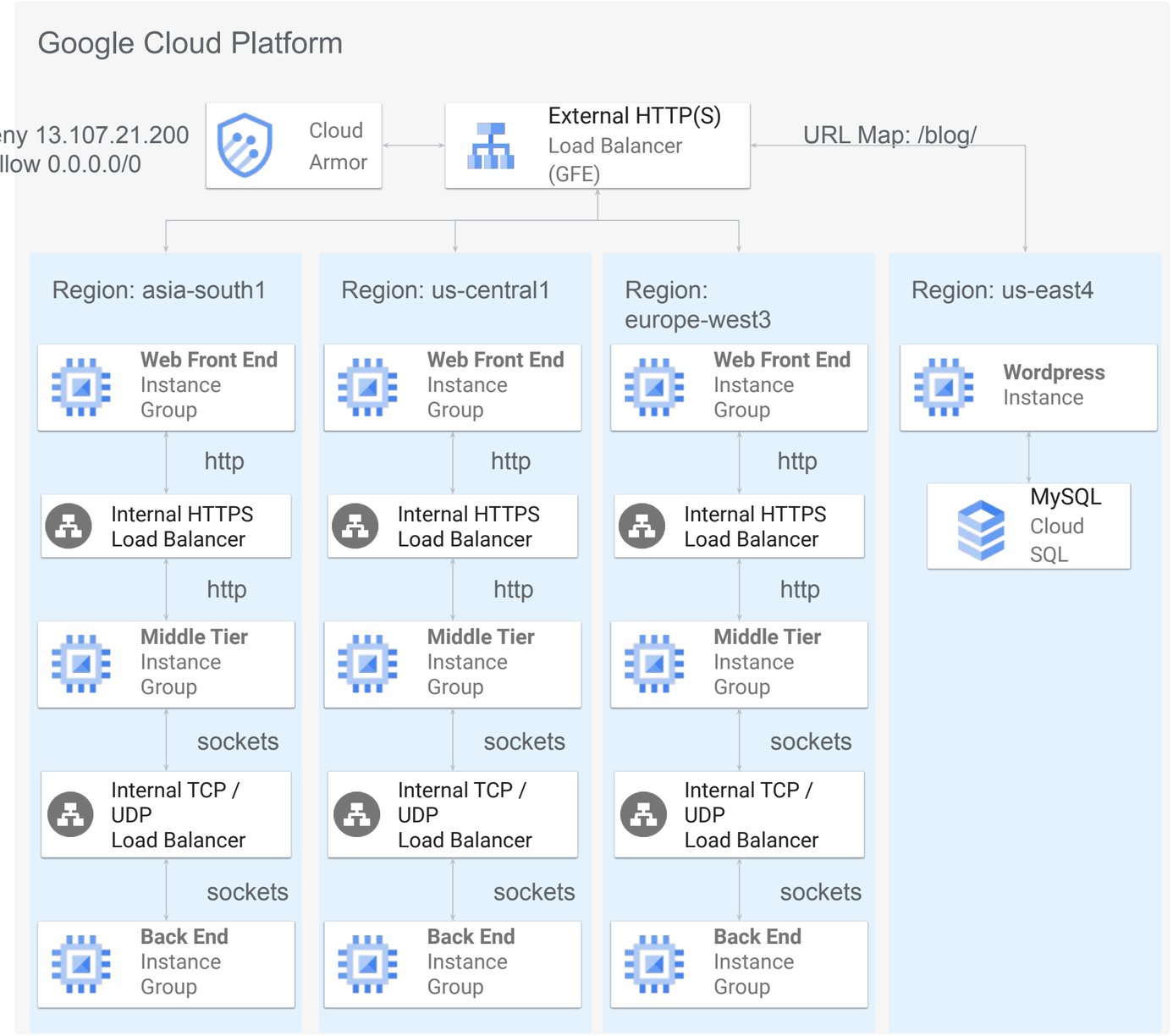


- Speeds (*)
  - VPN up to 3 gbps
  - Partner up to 50 gbps
  - Dedicated up to 100 gbps
- VPN Connections always go over the internet
  - The connection is encrypted using IPSec
  - There are pre-shared keys exchanged to facilitate
  - Must have a public IP address
- Interconnects (both direct and partner) are always to GCP, not Google
  - Consumer services / Workspace still go over the internet
- You should ***never*** have only 1 connection into GCP
  - Connections should be in two separate regions (not zones)
  - You can use a different solution to backup the primary (primary interconnect, vpn as backup)
- IP Address ranges cannot overlap in any of the architectures

* Can stack some of these solution for higher speeds

Google Cloud

PCA Study Cards - Load Balancing



- External HTTP(S) Load Balancer
  - Global Service (*)
  - Traffic to “closest” endpoint
  - Single Anycast IP Address
  - Can be used for workloads on-premises or other clouds
- URL Map apply to both Internal and External HTTP(s) Load balancers
  - Directs to different backends
  - Based on a fragment of the url or host names
- Cloud Armor
  - rules to protect vulnerable backend services from OWASP Top 10 attacks like SQL Injection and cross site scripting
  - Allow / Deny lists for IP addresses and regions
  - Like Firewall rules, lower the number higher the priority (1>10)
  - Named IP list are 3rd party maintained list for malicious IP addresses
- Additional items to remember:
  - Health Checks on backends
  - Firewall rules
  - SSL Proxy (not shown) is for non-http traffic

* requires premium network tier

Google Cloud

Most common Organization Policies

Policy Constraint	Description
compute.vmExternalIpAccess	A list of project/zone/instance names that are allowed to have external IP addresses and deny all others. Attempts to create any other VMs with an external IP address will fail.
compute.trustedImageProjects	A list of projects that contain trusted images that can be used as the basis for a VM and deny all others. Attempting to instantiate a VM with an image from another project is denied.
compute.skipDefaultNetworkCreation	Disables the creation of default VPC when creating a project. The default VPC uses auto mode subnetworks and includes default firewall rules which are often incompatible with production deployments.
iam.disableServiceAccountKeyCreation	This boolean constraint disables the creation of service account external keys where this constraint is set to `True`.
compute.restrictVpcPeering	This list constraint defines the set of VPC networks that are allowed to be peered with the VPC networks belonging to this project, folder, or organization.
serviceuser.services	This list constraint defines the set of services and their APIs that can be enabled on this resource and below. By default, all services are allowed.
gcp.resourceLocations	BETA: This list constraint defines the set of locations where location-based GCP resources can be created. Policies for this constraint can specify multi-regions such as asia and europe, regions such as us-east1 or europe-west1, or individual zones such as europe-west1-b as allowed or denied locations.
sql.restrictPublicIp	This boolean constraint restricts configuring Public IP on Cloud SQL instances where this constraint is set to True. This constraint is not retroactive, Cloud SQL instances with existing Public IP access will still work even after this constraint is enforced. By default, Public IP access is allowed to Cloud SQL instances.
sql.disableDefaultEncryptionCreation	BETA: Restrict default Google-managed encryption on Cloud SQL instances
compute.requireShieldedVm	This boolean constraint, when set to True, requires that all new Compute Engine VM instances use Shielded disk images with Secure Boot, vTPM, and Integrity Monitoring options enabled. Secure Boot can be disabled after creation, if desired. Shielded VM features add verifiable integrity and exfiltration resistance to your VMs.
compute.restrictSharedVpcHostProjects	Restrict Shared VPC Host Projects This list constraint defines the set of Shared VPC host projects that projects at or below this resource can attach to. By default, a project can attach to any host project in the same organization, thereby becoming a service project.
iam.allowedPolicyMemberDomains	This list constraint defines the set of members that can be added to Cloud IAM policies. By default, all user identities are allowed to be added to Cloud IAM policies. The allowed/denied list must specify one or more Cloud Identity or G Suite customer IDs. If this constraint is active, only identities in the allowed list will be eligible to be added to Cloud IAM policies.

DNS options

An internal metadata server acts as DNS resolver, and is automatically set as such as part of DHCP leases.

Internal DNS

Records are automatically created for VMs primary and internal IP's with the following FQDN:

- [INSTANCE_NAME].[ZONE].c.[PROJECT_ID].internal

Used for resolution within the same project and VPC

Cloud DNS

Scalable, reliable (**100% SLA**), and managed authoritative DNS service for public and private records offering

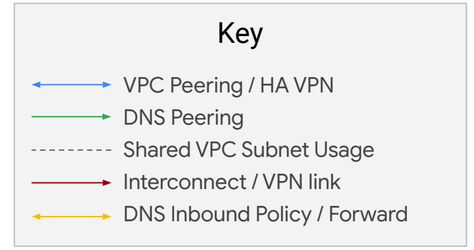
Private: Used for providing a namespace that is only visible inside the VPC

Public: Used for providing authoritative DNS resolution to clients on the public internet.

Google Cloud

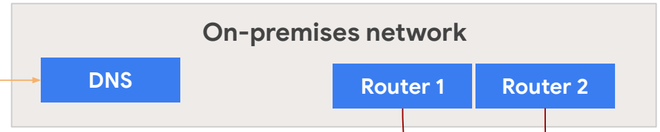
Reference architecture (final version)

Hub-and-spoke with VPC peering - Segmentation based on environments



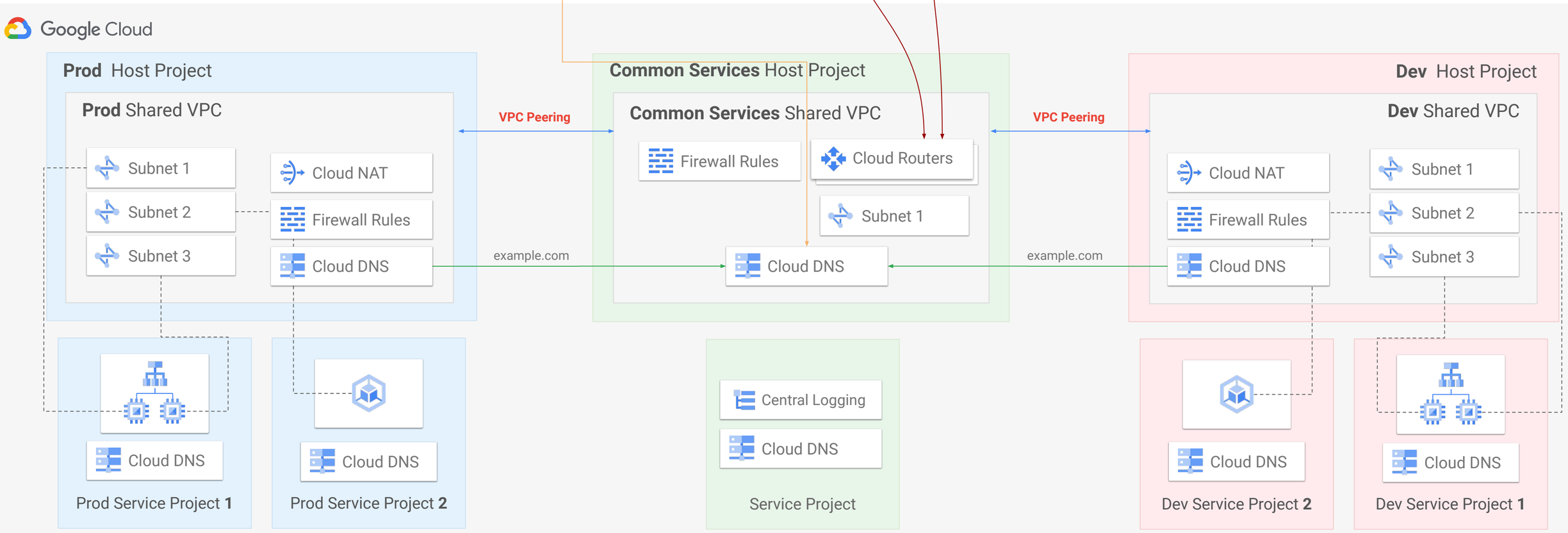
Network security control
- Centralized network security administration
- Central services (NAT, DNS, etc.) deployed in Shared VPC

Scalability
- **Up to 25 spokes**, per VPC peering limitations
- Each spoke can have a high number of service projects

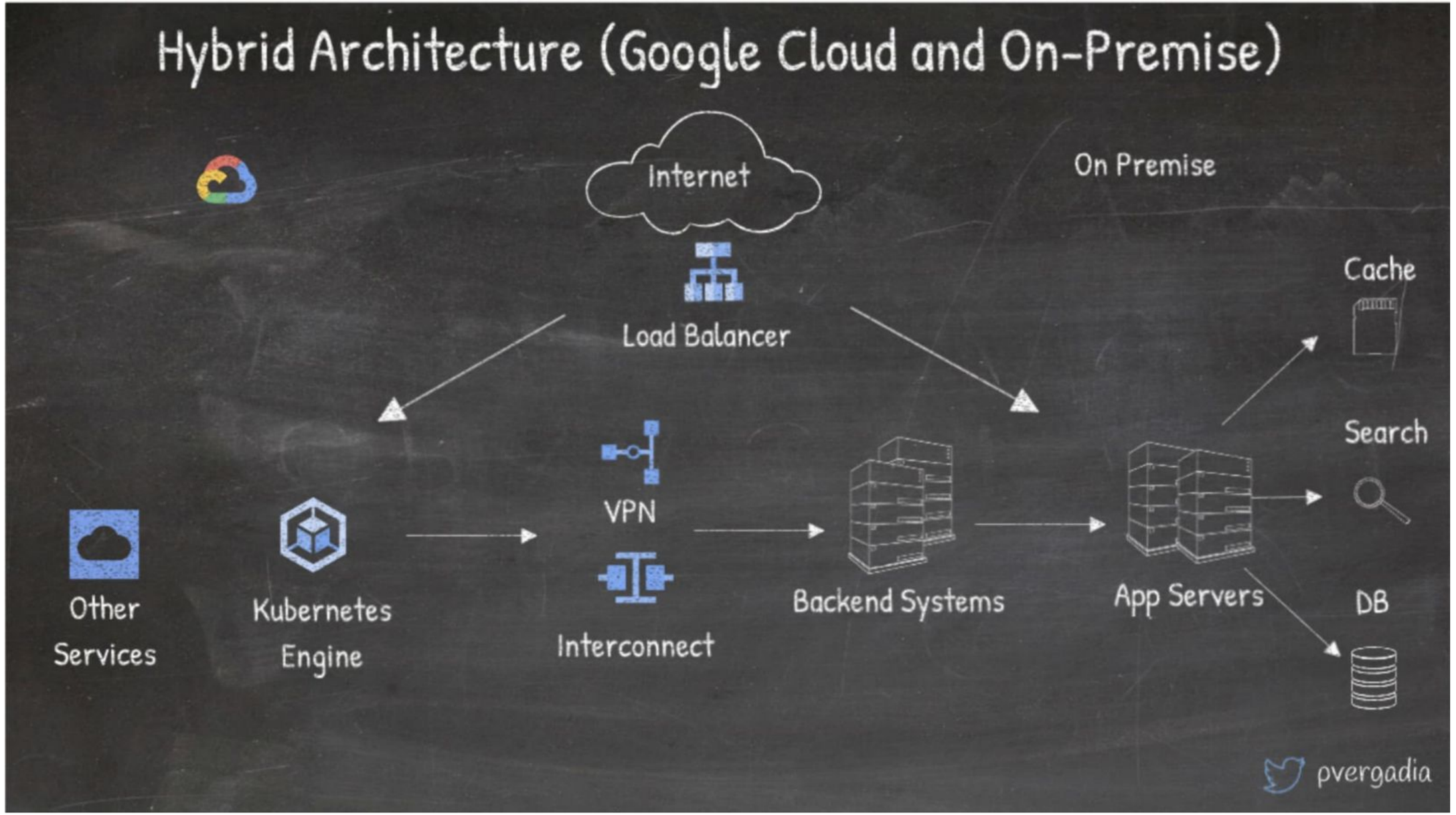


Spokes isolation
- Spokes are isolated as VPC peering is **non-transitive**

Central control versus autonomy
- Full networking autonomy for spokes, based on a separate shared VPC network



Exam notes & tips



[13 sample GCP architectures]

Exam tips&tricks

- READ questions (at least) TWICE, look for key statements, details matter!!!
- The exam does not have a fixed, public pass score, although people say it should be around 70%.
  - Do not think about the math too much. Focus on the exam, don't get disturbed by a few questions in a row you found difficult.
- How to boot a docker container faster? Use alpine slim linux distro
- GCS and Spanner are multi regional for data sync
- Cloud VPN is a regional service.
- Know most popular IAM roles for each service (with special focus on GCE, GCS, GKE and BigQuery)
- Datastore/Firestore is great backend for App Engine, Storing Game state
- Analytics and SQL = BigQuery
  - Make sure to understand BigQuery table partitioning and clustering.
- Make sure to have as much skills about GKE and Kubernetes as possible (you can expect questions about deployments, GKE-related gcloud commands, container use-cases etc).
- Examples of simple Back out / Roll out Plan:
  - Enable object versioning on the website's static data files stored in GCS
  - Use managed instance groups with the “update-instances” command when starting a rolling update
- GCP Projects can be billed separately (different billing accounts).
- Each resource can be zonal, regional, multi-regional or global. And it has to be assigned to a SINGLE project.
- Custom roles can only be applied at orgs or projects, not folders!
- Service Account User role allows a person to use the specified role on a vm, if they have access to it.
  - That means a user can have elevated privileges (for example, to delete other VMs) if the Service Account of a VM has adequate privileges.

Exam tips&tricks

- IAP's short definition: API gateway that authenticates and authorizes HTTPS requests using IAM.
  - IAP is often related to questions, where users need to log in to applications deployed to GCP (instead of accessing GCP resources like GCE, App Engine etc directly)
- You can download/upload files to/from your local machine using Cloud Shell or browser-based ssh connection to a VM.
- In order to use Global Load Balancing and CDN, premium network tier must be chosen.
- Auto mode network can be converted to custom, but custom mode network cannot be converted to auto.
- Subnet CIDR range: First IP address reserved for network (0), second address reserved for gw (1), Second to last reserved, last reserved for broadcast.
  - It adds up to 4 unusable IP addresses in each subnet -> might play a role when sizing small subnets.
- Firewall Priority: first matching rule applied, no further rules are evaluated.
- Standard FQDN of a GCP VM: hostname.zone.c.project-id.internal.
- Cloud VPN: max 3 Gbps, can be aggregated.
  - Supports both ikev1 and ikev2
  - Max MTU 1460 (max and recommended value at the same time)
- Partner Interconnect: up to 10 Gbit; Dedicated Interconnect: From 10G to 100G (or multipliers)
- Global Load Balancers support IPv6 (Proxy; 2nd session inside GCP uses IPv4 anyway)
  - Https load balancer natively supports websockets
- Audit logs are kept for 30 days. If needed, they can be exported to cloud storage, bigquery and cloud pub/sub.
- PD encryption: HDDs uses AES128, SSDs uses AES256
- Know how to share PD snapshots with a different project (snapshot created directly from destination project).

Exam tips&tricks

- Common PD snapshot use cases:
  - Upgrade/downgrade disk type
  - Migrate machines to other zones
  - Reduce disk size
  - Can be converted to images, which can then be used to create VMs also in other projects / regions
- Once a MIG is deployed, it’s possible to update instances using the “rolling-action start-update” feature. Support also canary updates.
  - New template is needed for instance update -> it's not possible to update existing VM template.
- Querying metadata from inside the VM:
  curl -H "Metadata-Flavor: Google"
  http://169.254.169.254/computeMetadata/v1/instance/network-interfaces/0/access-configs/0/external-ip && echo
- How to move a VM to another zone:
  - Use gcloud compute instances move
    OR:
  - Snapshot disk, restore snapshotted disk in other zone, create new vm and reattach persistent disk restored, re-assign static IP if needed, delete old backups and vms if needed
- GCS public ACL scopes:
  - allUsers: anyone, even without Google account
  - allAuthenticatedUsers: anyone authenticated with a Google account
- Pub/Sub:
  - Messages can be up to 10MB
  - Messages not delivered are stored up to 7 days
  - Ensures "at least once" message delivery. If "exactly once" is needed, messages need to be streamed through Dataflow

Exam tips&tricks

- Firestore:
  - Schema-less
  - A document is a collection of key, value pairs
  - Documents are grouped in collections
  - For mobile, web, IoT apps at global scale
  - Supports ACID transactions
  - Backwards compatible with cloud datastore (firestore is the new generation of Datastore)
- Feel free to sign-up for free 300$ worth of GCP credits to deploy your own project and play around without time pressure
  - Details here
  - You still need to provide credit card details
  - Credits valid for 90 days
  - You can use some free GCP resources regardless of the 300$ / 90 day offer.

Useful gcloud commands

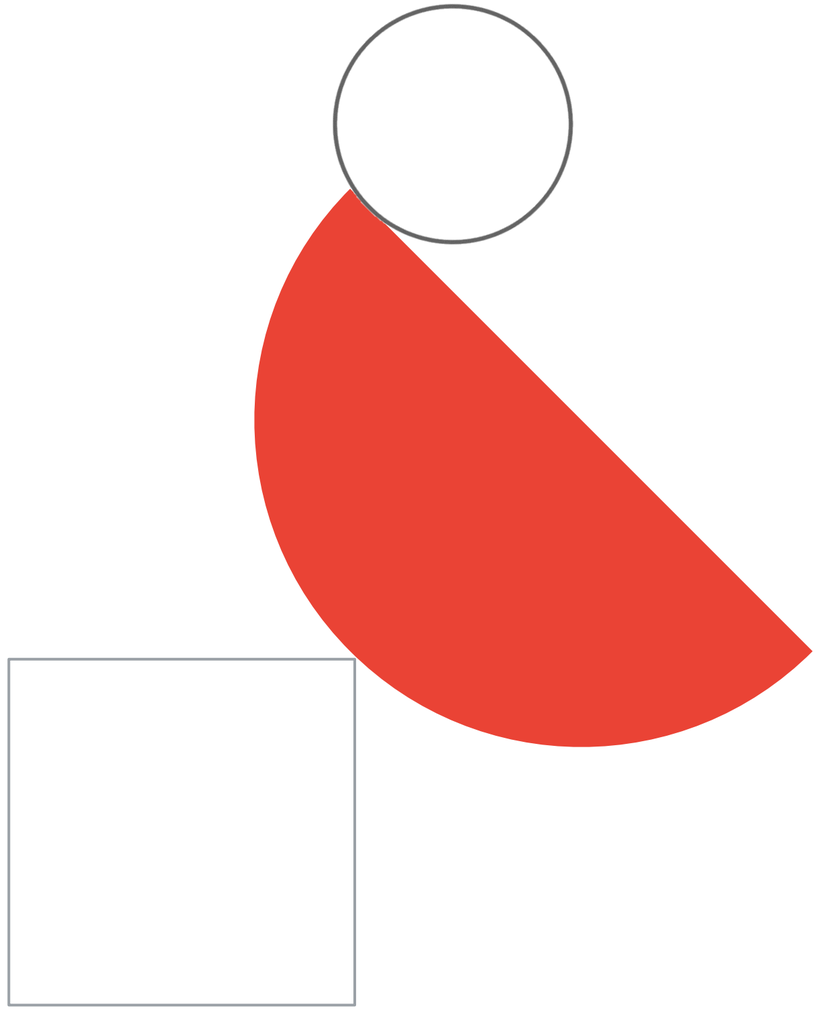
- GCP Command Cheat Sheet from medium.com
- List vnet: *gcloud compute networks list*
- List subnets *gcloud compute networks subnets list --sort-by=NETWORK*
- List firewall rules: *gcloud compute firewall-rules list --sort-by=NETWORK*
- Create vnet *gcloud compute networks create privatenet --subnet-mode=custom*
- Create subnet *gcloud compute networks subnets create privatesubnet-us --network=privatenet --region=us-central1 --range=172.16.0.0/24*
- Create firewall rules *gcloud compute firewall-rules create <FIREWALL_NAME> --network privatenet --allow tcp,udp,icmp --source-ranges <IP_RANGE>*
- Create project: *gcloud projects create [--folder=FOLDER_ID] [--labels=[KEY=VALUE,...]] [--name=NAME] [--organization=ORGANIZATION_ID] [--set-as-default] [--enable-cloud-apis x,y,z]*
- Set project in console: *gcloud config set project [PROJECT_ID]*
- Create role: *gcloud iam roles create*
- Bind role to user/SA/group: *gcloud projects add-iam-policy-binding PROJECT_ID --member=EMAIL --role=ROLE*
- Create deployment: *gcloud deployment-manager deployments create my-first-depl --config mydeploy.yaml*
- Update deployment: *gcloud deployment-manager deployments update my-first-depl --config mydeploy.yaml*
- Create GKE cluster: *gcloud container clusters create MYCLUSTER --zone MY_ZONE --num-nodes 2*
- Resize GKE cluster: *gcloud container clusters resize NAME --size=SIZE [--async] [--node-pool=NODE_POOL] [--region=REGION | --zone=ZONE, -z ZONE] [GCLOUD_WIDE_FLAG ...]*
- Get credentials to GKE cluster: *gcloud container clusters get-credentials echo-cluster --zone=us-central1-a*
- Create App Engine app: *gcloud app create --project=$DEVSHELL_PROJECT_ID*
- Execute app locally *dev_appserver.py $(pwd)*

Useful gcloud commands

- Deploy App Engine app: *gcloud app deploy YOUR_APP_MANIFEST.yaml*
- Create BigQuery table with partition that expires: *bq mk --time_partitioning_type=DAY --time_partitioning_expiration=259200 DATASET.TABLE*
- Query BigQuery table: *bq query "select * as name_of_my_query from mydataset.mytable"*
- List BigQuery jobs: *bq ls -j -a PROJECT (-j = jobs, -a = all users)*
- Cloud Build: trigger build and store image: *gcloud builds submit --tag gcr.io/$DEVSHELL_PROJECT_ID/devops-image:v0.1 .*
- Cloud Build: trigger build using a config file: *gcloud builds submit --config cloudbuild.yaml .*

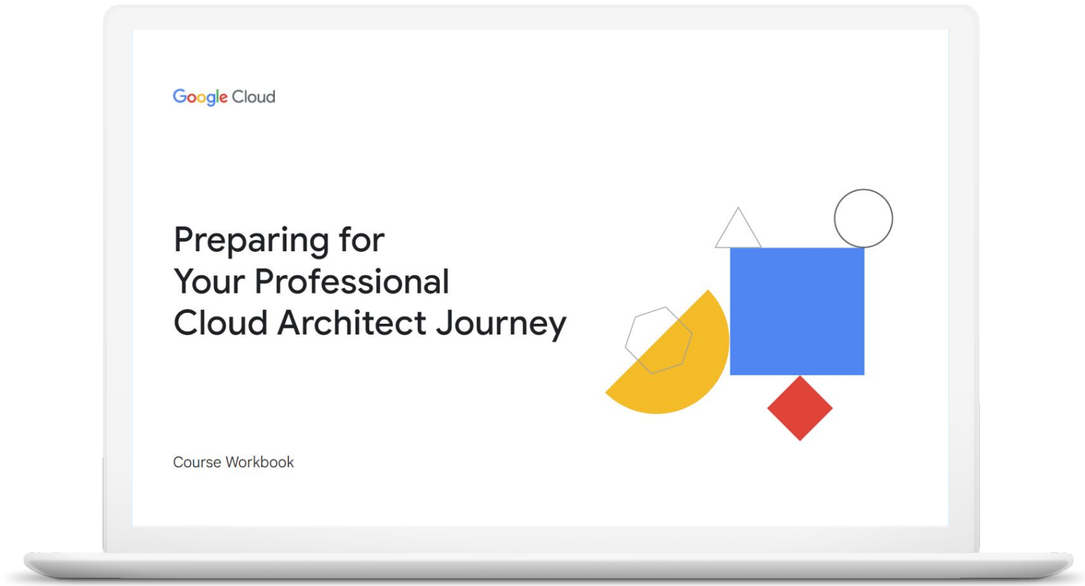
Diagnostic Questions

for Exam Guide Section 5: Managing implementation and Section 6: Ensuring solution and operations reliability



PCA Exam Guide Sections 5&6:

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 Account 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 Account 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 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.**

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.**

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.

6.1 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.

6.1 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.

6.2 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.

6.3 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.

6.3 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.

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.

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.

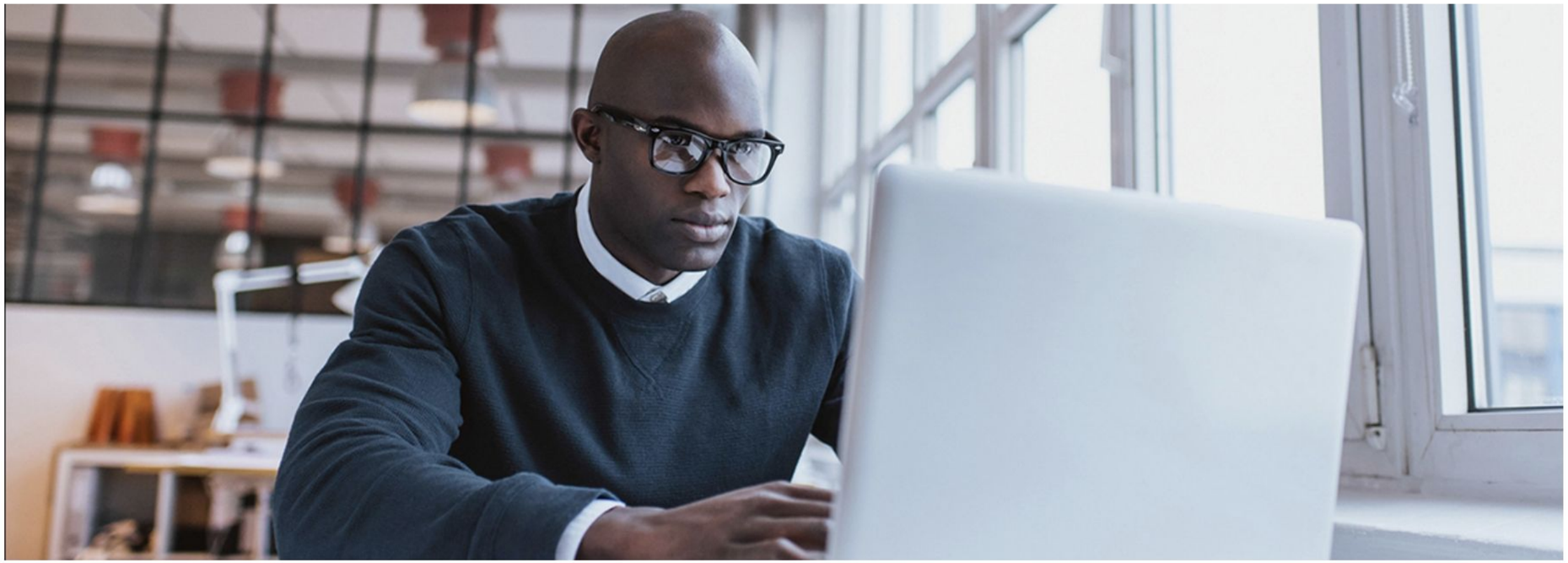
6.1 – 6.4 Ensuring solution and operations reliability

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

How to register for the exa

Webassessor by KRYTERION



webassessor.com

Google Cloud

Q & A

Make sure to...

Enjoy the journey as much as the destination!