# Professional Cloud Architect on Google Cloud Platform

## Google Professional-Cloud-Architect

## Version Demo

## Total Demo Questions: 15

## Total Premium Questions: 259

## Buy Premium PDF

https://dumpsboss.com

support@dumpsboss.com

# Topic Break Down

| Topic | No. of Questions |
|---|---|
| Topic 1, Case Study 1 | 6 |
| Topic 2, Case Study 2 | 54 |
| Topic 3, Case Study 3 | 6 |
| Topic 4, Case Study 4 | 6 |
| Topic 5, Case Study 5 | 7 |
| Topic 6, Mixed Questions | 180 |
| Total | 259 |

## QUESTION NO: 1

For this question, refer to the EHR Healthcare case study. You need to define the technical architecture for securely deploying workloads to Google Cloud. You also need to ensure that only verified containers are deployed using Google Cloud services. What should you do? (Choose two.)

**A.** Enable Binary Authorization on GKE, and sign containers as part of a CI/CD pipeline.

**B.** Configure Jenkins to utilize Kritis to cryptographically sign a container as part of a CI/CD pipeline.

**C.** Configure Container Registry to only allow trusted service accounts to create and deploy containers from the registry.

**D.** Configure Container Registry to use vulnerability scanning to confirm that there are no vulnerabilities before deploying the workload.

**ANSWER: A B**

## QUESTION NO: 2

You want to create a private connection between your instances on Compute Engine and your on-premises data center. You require a connection of at least 20 Gbps. You want to follow Googlerecommended practices. How should you set up the connection?

**A.** Create a VPC and connect it to your on-premises data center using Dedicated Interconnect.

**B.** Create a VPC and connect it to your on-premises data center using a single Cloud VPN.

**C.** Create a Cloud Content Delivery Network (Cloud CDN) and connect it to your on-premises data center using Dedicated Interconnect.

**D.** Create a Cloud Content Delivery Network (Cloud CDN) and connect it to your on-premises datacenter using a single Cloud VPN.

**ANSWER: A**

## QUESTION NO: 3

Your company creates rendering software which users can download from the company website. Your company has customers all over the world. You want to minimize latency for all your customers. You want to follow Google-recommended practices.

How should you store the files?

**A.** Save the files in a Multi-Regional Cloud Storage bucket.

**B.** Save the files in a Regional Cloud Storage bucket, one bucket per zone of the region.

**C.** Save the files in multiple Regional Cloud Storage buckets, one bucket per zone per region.

**D.** Save the files in multiple Multi-Regional Cloud Storage buckets, one bucket per multi-region.

**ANSWER: A**

## QUESTION NO: 4

Your company wants you to build a highly reliable web application with a few public APIs as the backend. You don't expect a lot of user traffic, but traffic could spike occasionally. You want to leverage Cloud Load Balancing, and the solution must be cost-effective for users. What should you do?

**A.** Store static content such as HTML and images in Cloud CDN. Host the APIs on App Engine and store the user data in Cloud SQL.

**B.** Store static content such as HTML and images in a Cloud Storage bucket. Host the APIs on a zonal Google Kubernetes Engine cluster with worker nodes in multiple zones, and save the user data in Cloud Spanner.

**C.** Store static content such as HTML and images in Cloud CDN. Use Cloud Run to host the APIs and save the user data in Cloud SQL.

**D.** Store static content such as HTML and images in a Cloud Storage bucket. Use Cloud Functions to host the APIs and save the user data in Firestore.

**ANSWER: B**

## QUESTION NO: 5

You need to reduce the number of unplanned rollbacks of erroneous production deployments in your company's web hosting platform. Improvement to the QA/Test processes accomplished an 80% reduction.

Which additional two approaches can you take to further reduce the rollbacks? (Choose two.)

**A.** Introduce a green-blue deployment model

**B.** Replace the QA environment with canary releases

**C.** Fragment the monolithic platform into microservices

**D.** Reduce the platform's dependency on relational database systems

**E.** Replace the platform's relational database systems with a NoSQL database

**ANSWER: A C**

## QUESTION NO: 6

You are helping the QA team to roll out a new load-testing tool to test the scalability of your primary cloud services that run on Google Compute Engine with Cloud Bigtable.

Which three requirements should they include? (Choose three.)

**A.** Ensure that the load tests validate the performance of Cloud Bigtable

**B.** Create a separate Google Cloud project to use for the load-testing environment

**C.** Schedule the load-testing tool to regularly run against the production environment

**D.** Ensure all third-party systems your services use is capable of handling high load

**E.** Instrument the production services to record every transaction for replay by the load-testing tool

**F.** Instrument the load-testing tool and the target services with detailed logging and metrics collection

**ANSWER: A B F**

## QUESTION NO: 7

A lead engineer wrote a custom tool that deploys virtual machines in the legacy data center. He wants to migrate the custom tool to the new cloud environment. You want to advocate for the adoption of Google Cloud Deployment Manager.

What are two business risks of migrating to Cloud Deployment Manager? (Choose two.)

**A.** Cloud Deployment Manager uses Python

**B.** Cloud Deployment Manager APIs could be deprecated in the future

**C.** Cloud Deployment Manager is unfamiliar to the company's engineers

**D.** Cloud Deployment Manager requires a Google APIs service account to run

**E.** Cloud Deployment Manager can be used to permanently delete cloud resources

**F.** Cloud Deployment Manager only supports automation of Google Cloud resources

**ANSWER: B F**

## QUESTION NO: 8

For this question, refer to the Mountkirk Games case study. You are in charge of the new Game Backend Platform architecture. The game communicates with the backend over a REST API.

You want to follow Google-recommended practices. How should you design the backend?

**A.** Create an instance template for the backend. For every region, deploy it on a multi-zone managed instance group. Use an L4 load balancer.

**B.** Create an instance template for the backend. For every region, deploy it on a single-zone managed instance group. Use an L4 load balancer.

**C.** Create an instance template for the backend. For every region, deploy it on a multi-zone managed instance group. Use an L7 load balancer.

**D.** Create an instance template for the backend. For every region, deploy it on a single-zone managed instance group. Use an L7 load balancer.

**ANSWER: A**

**Explanation:**

Company overview

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They have recently started expanding to other platforms after successfully migrating their on-premises environments to Google Cloud.

Their most recent endeavor is to create a retro-style first-person shooter (FPS) game that allows hundreds of simultaneous players to join a geo-specific digital arena from multiple platforms and locations. A real-time digital banner will display a global leaderboard of all the top players across every active arena.

Solution concept

Mountkirk Games is building a new multiplayer game that they expect to be very popular. They plan to deploy the game's backend on Google Kubernetes Engine so they can scale rapidly and use Google's global load balancer to route players to the closest regional game arenas. In order to keep the global leader board in sync, they plan to use a multi-region Spanner cluster.

Existing technical environment

The existing environment was recently migrated to Google Cloud, and five games came across using lift-and-shift virtual machine migrations, with a few minor exceptions. Each new game exists in an isolated Google Cloud project nested below a folder that maintains most of the permissions and network policies. Legacy games with low traffic have been consolidated into a single project. There are also separate environments for development and testing.

Business requirements

- Support multiple gaming platforms.
- Support multiple regions.
- Support rapid iteration of game features.
- Minimize latency.
- Optimize for dynamic scaling.
- Use managed services and pooled resources. ▪ Minimize costs.

Technical requirements

- Dynamically scale based on game activity.

- Publish scoring data on a near real-time global leaderboard.
- Store game activity logs in structured files for future analysis.
- Use GPU processing to render graphics server-side for multi-platform support. ▪ Support eventual migration of legacy games to this new platform.

Executive statement

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles. Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

## QUESTION NO: 9

You need to develop procedures to verify resilience of disaster recovery for remote recovery using GCP. Your production environment is hosted on-premises. You need to establish a secure, redundant connection between your on-premises network and the GCP network.

What should you do?

**A.** Verify that Dedicated Interconnect can replicate files to GCP. Verify that direct peering can establish a secure connection between your networks if Dedicated Interconnect fails.

**B.** Verify that Dedicated Interconnect can replicate files to GCP. Verify that Cloud VPN can establish a secure connection between your networks if Dedicated Interconnect fails.

**C.** Verify that the Transfer Appliance can replicate files to GCP. Verify that direct peering can establish a secure connection between your networks if the Transfer Appliance fails.

**D.** Verify that the Transfer Appliance can replicate files to GCP. Verify that Cloud VPN can establish a secure connection between your networks if the Transfer Appliance fails.

**ANSWER: B**

## QUESTION NO: 10

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 availability 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 availability zones. Create a TCP/IP global load balancer. Add the Cloud Run Endpoints to its backend service.

**ANSWER: C**

## QUESTION NO: 11

One of the developers on your team deployed their application in Google Container Engine with the Dockerfile below. They report that their application deployments are taking too long.

```
FROM ubuntu:16.04

COPY . /src

RUN apt-get update && apt-get install -y python python-pip

RUN pip install -r requirements.txt
```

You want to optimize this Dockerfile for faster deployment times without adversely affecting the app's functionality.

Which two actions should you take? (Choose two.)

**A.** Remove Python after running pip

**B.** Remove dependencies from requirements.txt

**C.** Use a slimmed-down base image like Alpine Linux

**D.** Use larger machine types for your Google Container Engine node pools

**E.** Copy the source after he package dependencies (Python and pip) are installed

**ANSWER: C E**

**Explanation:**

The speed of deployment can be changed by limiting the size of the uploaded app, limiting the complexity of the build necessary in the Dockerfile, if present, and by ensuring a fast and reliable internet connection.

Note: Alpine Linux is built around musl libc and busybox. This makes it smaller and more resource efficient than traditional GNU/Linux distributions. A container requires no more than 8 MB and a minimal installation to disk requires around 130 MB of storage. Not only do you get a fully-fledged Linux environment but a large selection of packages from the repository.

Reference: https://groups.google.com/forum/#!topic/google-appengine/hZMEkmmObDU https://www.alpinelinux.org/about/

## QUESTION NO: 12

You are developing an application using different microservices that should remain internal to the cluster. You want to be able to configure each microservice with a specific number of replicas. You also want to be able to address a specific

microservice from any other microservice in a uniform way, regardless of the number of replicas the microservice scales to. You need to implement this solution on Google Kubernetes Engine. What should you do?

**A.** Deploy each microservice as a Deployment. Expose the Deployment in the cluster using a Service, and use the Service DNS name to address it from other microservices within the cluster.

**B.** Deploy each microservice as a Deployment. Expose the Deployment in the cluster using an Ingress, and use the Ingress IP address to address the Deployment from other microservices within the cluster.

**C.** Deploy each microservice as a Pod. Expose the Pod in the cluster using a Service, and use the Service DNS name to address the microservice from other microservices within the cluster.

**D.** Deploy each microservice as a Pod. Expose the Pod in the cluster using an Ingress, and use the Ingress IP address name to address the Pod from other microservices within the cluster.

**ANSWER: A**

## QUESTION NO: 13

Your marketing department wants to send out a promotional email campaign. The development team wants to minimize direct operation management. They project a wide range of possible customer responses, from 100 to 500,000 click-through per day. The link leads to a simple website that explains the promotion and collects user information and preferences. Which infrastructure should you recommend? (Choose two.)

**A.** Use Google App Engine to serve the website and Google Cloud Datastore to store user data.

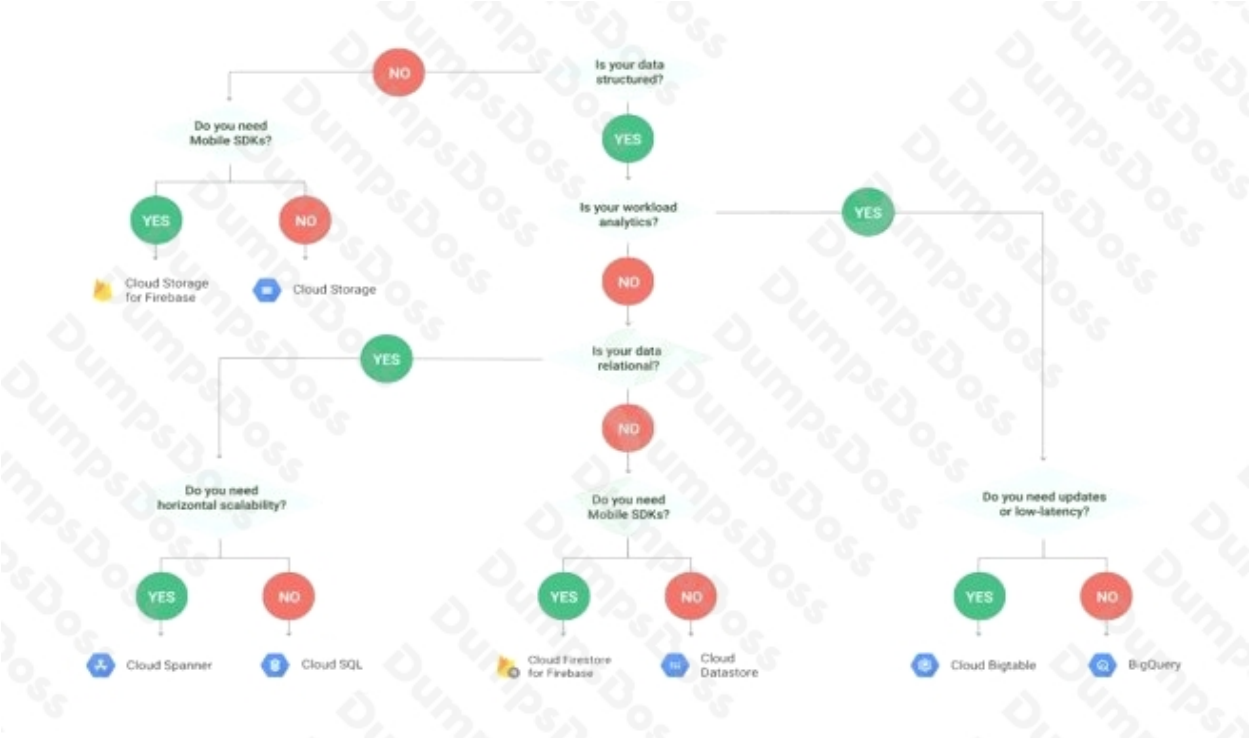
**B.** Use a Google Container Engine cluster to serve the website and store data to persistent disk.

**C.** Use a managed instance group to serve the website and Google Cloud Bigtable to store user data.

**D.** Use a single Compute Engine virtual machine (VM) to host a web server, backend by Google Cloud SQL.

**ANSWER: A C**

**Explanation:**




Reference: https://cloud.google.com/storage-options/

## QUESTION NO: 14

You want to establish a Compute Engine application in a single VPC across two regions. The application must communicate over VPN to an on-premises network. How should you deploy the VPN?

**A.** Use VPC Network Peering between the VPC and the on-premises network.

**B.** Expose the VPC to the on-premises network using IAM and VPC Sharing.

**C.** Create a global Cloud VPN Gateway with VPN tunnels from each region to the on-premises peer gateway.

**D.** Deploy Cloud VPN Gateway in each region. Ensure that each region has at least one VPN tunnel to the on-premises peer gateway.

**ANSWER: D**

## QUESTION NO: 15

Your company places a high value on being responsive and meeting customer needs quickly. Their primary business objectives are release speed and agility. You want to reduce the chance of security errors being accidentally introduced.

Which two actions can you take? (Choose two.)

**A.** Ensure every code check-in is peer reviewed by a security SME

**B.** Use source code security analyzers as part of the CI/CD pipeline

**C.** Ensure you have stubs to unit test all interfaces between components

**D.** Enable code signing and a trusted binary repository integrated with your CI/CD pipeline

**E.** Run a vulnerability security scanner as part of your continuous-integration /continuous-delivery (CI/CD) pipeline

**ANSWER: B E**