Free Questions for Associate-Cloud-Engineer by actualtestdumps

Shared by Slater on 18-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Your company is running a three-tier web application on virtual machines that use a MySQL database. You need to create an estimated total cost of cloud infrastructure to run this application on Google Cloud instances and Cloud SQL. What should you do?

Options:

A) Use the Google Cloud Pricing Calculator to determine the cost of every Google Cloud resource you expect to use.
Use similar size instances for the web server, and use your current on-premises machines as a comparison for Cloud SQL.

B) Create a Google spreadsheet with multiple Google Cloud resource combinations. On a separate sheet, import the current Google Cloud prices and use these prices for the calculations within formulas.

C) Implement a similar architecture on Google Cloud, and run a reasonable load test on a smaller scale. Check the billing information, and calculate the estimated costs based on the real load your system usually handles.

D) Use the Google Cloud Pricing Calculator and select the Cloud Operations template to define your web application with as much detail as possible.

Answer:

A

Explanation:

The Google Cloud Pricing Calculator is a tool that allows you to estimate the cost of using Google Cloud products and services based on your specific needs and preferences. You can add various Google Cloud resources to your estimate, such as Compute Engine instances, Cloud SQL instances, network services, storage options, and more. You can also adjust the parameters of each resource, such as the region, zone, machine type, disk size, network egress, and usage hours. The Google Cloud Pricing Calculator will show you the monthly and hourly cost of each resource, as well as the total cost of your estimate. You can also save, share, or export your estimate for future reference.

To create an estimated total cost of cloud infrastructure to run your three-tier web application on Google Cloud instances and Cloud SQL, you should use the Google Cloud Pricing Calculator to determine the cost of every Google Cloud resource you expect to use. You should use similar size instances for the web server, such as n1-standard-1 or n1-standard-2, depending on your application's CPU and memory requirements. You should also use your current on-premises machines as a comparison for Cloud SQL, such as MySQL 5.7 or MySQL 8.0, depending on your database version. You should also consider the network and storage costs associated with your application, such as the network egress, persistent disk, and Cloud Storage.

The other options are not correct because they either do not use the Google Cloud Pricing Calculator, or they do not provide a reliable or accurate estimate of the cloud infrastructure cost. Option B is not correct because creating a Google spreadsheet with multiple Google Cloud resource combinations is a manual and tedious process that may not reflect the current Google Cloud prices or discounts. Option C is not correct because implementing a similar architecture on Google Cloud, and running a reasonable load test on a smaller scale, is a time-consuming and costly process that may not represent the actual load or performance of your application. Option D is not correct because using the Google Cloud Pricing Calculator and selecting the Cloud Operations template will not define your web application with as much detail as possible, as the Cloud Operations template is designed for estimating the cost of monitoring and logging services, not the compute, database, network, and storage services.

Question 2

Question Type: MultipleChoice

You want to enable your development team to deploy new features to an existing Cloud Run service in production. To minimize the risk associated with a new revision, you want to reduce the number of customers who might be affected by an outage without introducing any development or operational costs to your customers. You want to follow Google-recommended practices for managing revisions to a service. What should you do?

Options:

A) Gradually roll out the new revision and split customer traffic between the revisions to allow rollback in case a problem occurs.

B) Deploy your application to a second Cloud Run service, and ask your customers to use the second Cloud Run service.

C) Ask your customers to retry access to your service with exponential backoff to mitigate any potential problems after the new revision is deployed.

D) Send all customer traffic to the new revision, and roll back to a previous revision if you witness any problems in production.

Answer:

A

Explanation:

Cloud Run is a fully managed platform that allows you to deploy and run containerized applications without worrying about the underlying infrastructure. Cloud Run automatically scales your application based on the incoming requests, and you only pay for the resources you use. Cloud Run also supports deploying multiple revisions of your service, and splitting the customer traffic between them. This allows you to gradually roll out new features, test them in production, and monitor their performance and reliability. If you encounter any problems with the new revision, you can easily roll back to a previous revision by adjusting the traffic split. This is a Google-recommended practice for managing revisions to a service, as it minimizes the risk associated with a new revision, reduces the number of customers who might be affected by an outage, and does not introduce any development or operational costs to your customers.

The other options are not correct because they either do not follow the Google-recommended practice, or they do not reduce the risk or cost associated with a new revision. Option B is not correct because deploying your application to a second Cloud Run service, and asking your customers to use the second Cloud Run service, is not a gradual roll out, but a complete switch. This may cause confusion and inconvenience for your customers, and it does not allow you to monitor the performance and reliability of the new revision in comparison with the old one. Option C is not correct because asking your customers to retry access to your service with exponential backoff to mitigate any potential problems after the new revision is deployed, is not a good user experience, and it does not address the root cause of the problems. Option D is not correct because sending all customer traffic to the new revision, and rolling back to a previous revision if you witness any problems in production, is a risky and costly strategy, as it exposes all your customers to the

potential problems, and it does not allow you to test the new revision before deploying it to all customers.

Question 3

Question Type: MultipleChoice

You are planning to migrate the following on-premises data management solutions to Google Cloud:

* One MySQL cluster for your main database

* Apache Kafka for your event streaming platform

* One Cloud SQL for PostgreSQL database for your analytical and reporting needs

You want to implement Google-recommended solutions for the migration. You need to ensure that the new solutions provide global scalability and require minimal operational and infrastructure management. What should you do?

Options:

A) Migrate from MySQL to Cloud SQL, from Kafka to Pub/Sub, and from Cloud SQL for PostgreSQL to BigQuery.

B) Migrate from MySQL to Cloud Spanner, from Kafka to Pub/Sub, and from Cloud SQL for PostgreSQL to BigQuery.

C) Migrate from MySQL to Cloud Spanner, from Kafka to Memorystore, and from Cloud SQL for PostgreSQL to Cloud SQL.

D) Migrate from MySQL to Cloud SQL, from Kafka to Memorystore, and from Cloud SQL for PostgreSQL to Cloud SQL.

Answer:

B

Explanation:

Cloud Spanner, Pub/Sub, and BigQuery are Google-recommended solutions for migrating your on-premises data management solutions to Google Cloud. They provide global scalability and require minimal operational and infrastructure management. Here is why:

Cloud Spanner is a fully managed, relational database service that offers strong consistency, high availability, and horizontal scalability across regions and continents. It is compatible with MySQL and can handle high-throughput OLTP workloads. It is a good choice for migrating your main database from MySQL to Google Cloud, as it can support your application's performance, availability, and growth requirements.

Pub/Sub is a fully managed, real-time messaging service that allows you to send and receive messages between independent applications. It is compatible with Apache Kafka and can handle high-volume event streaming and data ingestion. It is a good choice for migrating your event streaming platform from Kafka to Google Cloud, as it can support your application's reliability, scalability, and

flexibility requirements.

BigQuery is a fully managed, serverless data warehouse that offers fast SQL queries, high scalability, and low cost. It is compatible with Cloud SQL for PostgreSQL and can handle complex analytical and reporting needs. It is a good choice for migrating your analytical and reporting database from Cloud SQL for PostgreSQL to Google Cloud, as it can support your application's speed, efficiency, and innovation requirements.

The other options are not correct because they either do not provide the required scalability, performance, or functionality. Option A is not correct because Cloud SQL is a regional, managed database service that does not offer the same level of scalability, availability, and consistency as Cloud Spanner. Option C is not correct because Memorystore is a regional, managed in-memory data store that does not offer the same level of functionality, durability, and integration as Pub/Sub. Option D is not correct because Cloud SQL is a regional, managed database service that does not offer the same level of scalability, performance, and cost-effectiveness as BigQuery.

Migrating from MySQL to Cloud Spanner

Migrating from Apache Kafka to Cloud Pub/Sub

Migrating from Cloud SQL for PostgreSQL to BigQuery

Question 4

Question Type: **MultipleChoice**

Your application stores files on Cloud Storage by using the Standard Storage class. The application only requires access to files created in the last 30 days. You want to automatically save costs on files that are no longer accessed by the application. What should you do?

Options:

A) Enable object versioning on the storage bucket and add lifecycle rules to expire non-current versions after 30 days.

B) Create an object lifecycle on the storage bucket to change the storage class to Archive Storage for objects with an age over 30 days.

C) Create a retention policy on the storage bucket of 30 days, and lock the bucket by using a retention policy lock.

D) Create a cron job in Cloud Scheduler to call a Cloud Functions instance every day to delete files older than 30 days.

Answer:

B

Explanation:

Object lifecycle management is a feature of Cloud Storage that allows you to automatically manage the storage class and retention of your objects. You can use object lifecycle management to reduce the cost of storing your data by transitioning objects to lower-cost storage classes, such as Archive Storage, which is designed for long-term storage of data that is rarely accessed. You can also use object lifecycle management to delete objects that are no longer needed after a certain period of time. To use object lifecycle management, you need to create a lifecycle configuration, which is a set of rules that specify the actions to take on objects that match certain conditions. You can apply a lifecycle configuration to a storage bucket, and it will affect all objects in that bucket.

In this scenario, you want to automatically save costs on files that are no longer accessed by the application after 30 days. Therefore, you should create an object lifecycle on the storage bucket to change the storage class to Archive Storage for objects with an age over 30 days. This way, you can keep the files in the bucket, but pay less for storing them. Archive Storage has the lowest cost per GB among all storage classes, but also has the highest retrieval and early deletion fees. Therefore, it is suitable for data that is rarely accessed and has a long retention period.

The other options are not correct because they either do not reduce the cost of storage, or they delete the files that you may still need. Option A is not correct because enabling object versioning on the storage bucket and adding lifecycle rules to expire non-current versions after 30 days will only affect the old versions of the files, not the current ones. Therefore, you will still pay the same amount for storing the current files in the Standard Storage class. Option C is not correct because creating a retention policy on the storage bucket of 30 days, and locking the bucket by using a retention policy lock will prevent you from deleting or modifying any object in the bucket for 30 days. This will not reduce the cost of storage, and it will also limit your flexibility to manage your data. Option D is not correct because creating a cron job in Cloud Scheduler to call a Cloud Functions instance every day to delete files older than 30 days will permanently remove the files from the bucket. This may not be what you want, as you may still need to access the files for backup, audit, or compliance purposes.

Object Lifecycle Management

Storage Classes

Archive Storage

Question 5

You are creating an application that will run on Google Kubernetes Engine. You have identified MongoDB as the most suitable database system for your application and want to deploy a managed MongoDB environment that provides a support SLA. What should you do?

Options:

A) Create a Cloud Bigtable cluster and use the HBase API

B) Deploy MongoDB Alias from the Google Cloud Marketplace

C) Download a MongoDB installation package and run it on Compute Engine instances

D) Download a MongoDB installation package, and run it on a Managed Instance Group

Answer:

B

Explanation:

https://console.cloud.google.com/marketplace/details/gc-launcher-for-mongodb-atlas/mongodb-atlas

Question 6

Your company's security vulnc'ability managen->enl policy wonts 3 member of the security tearr ;o have visibility into vulnerabi ities and other OS metadata for a specific Compute Engine instance This Compute Engine instance hosts a critical application in your Gongle Cloud p'oject. You need to irrplement your company's security vulnerability management policy. What should you dc?

Options:

A) * Ensuie that the Ops Agent Is Installed on the Compute Engine instance.

* Create a custom metric in the Ctoud Monitoring dashboard.

* Provide the security team menber with access to this dashboard.

B) * Ensure :hat the Ops Agent is installed on tie Compute Engine instance.

* Provide the security team member roles/configure.inventoryViewer permission.

C) * Ensure that the OS Config agent Is Installed on the Compute Engine instance.

* Provide the security team member roles/configure.vulnerabilityViewer permission.

D) * Ensure that the OS Config agent is installed on the Compute Engine instance

* Create a log sink Co a BigQuery dataset.

* Provide the security team member with access to this dataset.

Answer:

C

Question 7

30. You are running multiple microservices in a Kubernetes Engine cluster. One microservice is rendering images. The microservice responsible for the image rendering requires a large amount of CPU time compared to the memory it requires. The other microservices are workloads that are optimized for n1-standard machine types. You need to optimize your cluster so that all workloads are using resources as efficiently as possible. What should you do?

Options:

A) Assign the pods of the image rendering microservice a higher pod priority than the older microservices

B) Create a node pool with compute-optimized machine type nodes for the image rendering microservice Use the node pool with general-purpose
machine type nodes for the other microservices

C) Use the node pool with general-purpose machine type nodes for lite mage rendering microservice Create a nodepool with compute-optimized machine type nodes for the other microservices

D) Configure the required amount of CPU and memory in the resource requests specification of the image rendering microservice deployment Keep the resource requests for the other microservices at the default

Answer:

B

Question 8

Question Type: **MultipleChoice**

&. You are working for a hospital that stores Its medical images in an on-premises data room. The hospital wants to use Cloud Storage for archival storage of these images. The hospital wants an automated process to upload any new medical images to Cloud Storage. You need to design and implement a solution. What should you do?

Options:

A) Deploy a Dataflow job from the batch template 'Datastore lo Cloud Storage' Schedule the batch job on the desired interval

B) In the Cloud Console, go to Cloud Storage Upload the relevant images to the appropriate bucket

C) Create a script that uses the gsutil command line interface to synchronize the on-premises storage with Cloud Storage Schedule the script as a cron job

D) Create a Pub/Sub topic, and enable a Cloud Storage trigger for the Pub/Sub topic. Create an application that sends all medical images to the Pub/Sub lope

Answer:

A

To Get Premium Files for Associate-Cloud-Engineer Visit

https://www.p2pexams.com/products/associate-cloud-engineer

For More Free Questions Visit

https://www.p2pexams.com/google/pdf/associate-cloud-engineer