Google Cloud platform – PDE questions

Questions:

You need to migrate a Redis database from an on-premises data center to a Memorystore for Redis instance. You want to follow Google-recommended practices and perform the migration for minimal cost, time and effort. What should you do?

A. Make an RDB backup of the Redis database, use the gsutil utility to copy the RDB file into a Cloud Storage bucket, and then import the RDB file into the Memorystore for Redis instance.

B. Make a secondary instance of the Redis database on a Compute Engine instance and then perform a live cutover.

C. Create a Dataflow job to read the Redis database from the on-premises data center and write the data to a Memorystore for Redis instance.

D. Write a shell script to migrate the Redis data and create a new Memorystore for Redis instance.

Your platform on your on-premises environment generates 100 GB of data daily, composed of millions of structured JSON text files. Your on-premises environment cannot be accessed from the public internet. You want to use Google Cloud products to query and explore the platform data. What should you do?

A. Use Cloud Scheduler to copy data daily from your on-premises environment to Cloud Storage. Use the BigQuery Data Transfer Service to import data into BigQuery.

B. Use a Transfer Appliance to copy data from your on-premises environment to Cloud Storage. Use the BigQuery Data Transfer Service to import data into BigQuery.

C. Use Transfer Service for on-premises data to copy data from your on-premises environment to Cloud Storage. Use the BigQuery Data Transfer Service to import data into BigQuery.

D. Use the BigQuery Data Transfer Service dataset copy to transfer all data into BigQuery.

A TensorFlow machine learning model on Compute Engine virtual machines (n2-standard-32) takes two days to complete training. The model has custom TensorFlow operations that must run partially on a CPU. You want to reduce the training time in a cost-effective manner. What should you do?

A. Change the VM type to n2-highmem-32.

B. Change the VM type to e2-standard-32.

C. Train the model using a VM with a GPU hardware accelerator.

D. Train the model using a VM with a TPU hardware accelerator.

You want to create a machine learning model using BigQuery ML and create an endpoint for hosting the model using Vertex AI. This will enable the processing of continuous streaming data in near-real time from multiple vendors. The data may contain invalid values. What should you do?

A. Create a new BigQuery dataset and use streaming inserts to land the data from multiple vendors. Configure your BigQuery ML model to use the "ingestion" dataset as the framing data.

B. Use BigQuery streaming inserts to land the data from multiple vendors where your BigQuery dataset ML model is deployed.

C. Create a Pub/Sub topic and send all vendor data to it. Connect a Cloud Function to the topic to process the data and store it in BigQuery.

D. Create a Pub/Sub topic and send all vendor data to it. Use Dataflow to process and sanitize the Pub/Sub data and stream it to BigQuery

You have a data processing application that runs on Google Kubernetes Engine (GKE). Containers need to be launched with their latest available configurations from a container registry. Your GKE nodes need to have GPUs, local SSDs, and 8 Gbps bandwidth. You want to efficiently provision the data processing infrastructure and manage the deployment process. What should you do?

A. Use Compute Engine startup scripts to pull container images and use gcloud commands to provision the infrastructure.

B. Use Cloud Build to schedule a job using Terraform build to provision the infrastructure and launch with the most current container images.

C. Use GKE to autoscale containers and use gcloud commands to provision the infrastructure.

D. Use Dataflow to provision the data pipeline and use Cloud Scheduler to run the job.

You need ads data to serve AI models and historical data for analytics. Longtail and outlier data points need to be identified. You want to cleanse the data in near-real time before running it through AI models. What should you do?

A. Use Cloud Storage as a data warehouse, shell scripts for processing, and BigQuery to create views for desired datasets.

B. Use Dataflow to identify longtail and outlier data points programmatically, with BigQuery as a sink.

C. Use BigQuery to ingest, prepare, and then analyze the data, and then run queries to create views.

D. Use Cloud Composer to identify longtail and outlier data points, and then output a usable dataset to BigQuery

You are collecting IoT sensor data from millions of devices across the world and storing the data in BigQuery. Your access pattern is based on recent data, filtered by location_id and device_version with the following query:

```
SELECT
    MAX(temperature)
FROM
    acme_iot_data.sensors
WHERE
    create_date > DATE_SUB(CURRENT_DATE(), INTERVAL 7 day)
    AND location_id = "SW1W9TQ"
    AND device_version = "202007r3"
```

You want to optimize your queries for cost and performance. How should you structure your data?

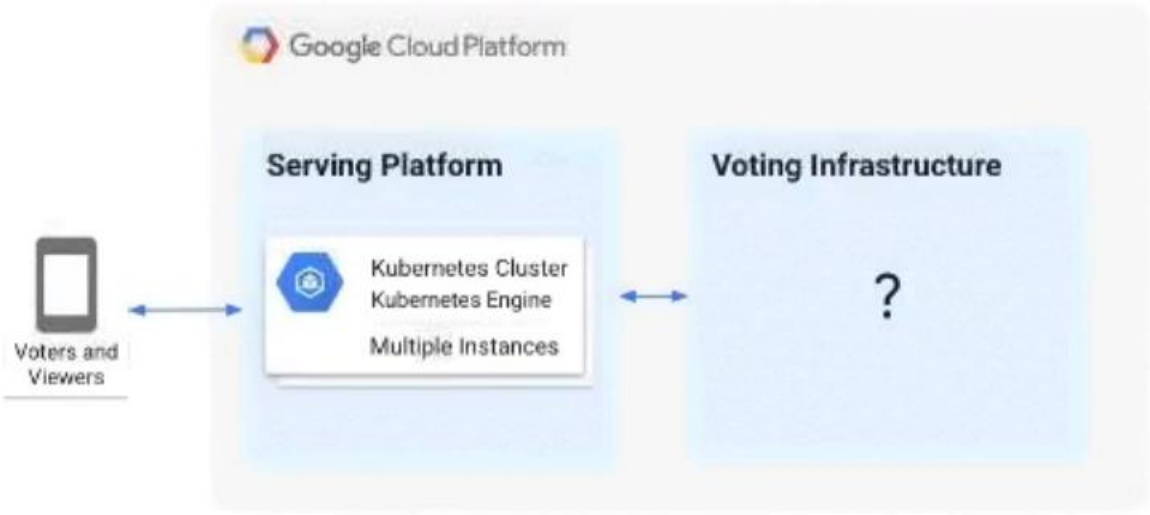
A. Partition table data by create_date, location_id, and device_version.

B. Partition table data by create_date, cluster table data by location_id, and device_version.

C. Cluster table data by create_date, location_id, and device_version.

D. Cluster table data by create_date, partition by location_id, and device_version.

A live TV show asks viewers to cast votes using their mobile phones. The event generates a large volume of data during a 3-minute period. You are in charge of the "Voting infrastructure" and must ensure that the platform can handle the load and that all votes are processed. You must display partial results while voting is open. After voting closes, you need to count the votes exactly once while optimizing cost. What should you do?



A. Create a Memorystore instance with a high availability (HA) configuration.

B. Create a Cloud SQL for PostgreSQL database with high availability (HA) configuration and multiple read replicas.

C. Write votes to a Pub/Sub topic and have Cloud Functions subscribe to it and write votes to BigQuery.

D. Write votes to a Pub/Sub topic and load into both Bigtable and BigQuery via a Dataflow pipeline. Query Bigtable for real-time results and BigQuery for later analysis. Shut down the Bigtable instance when voting concludes.

A shipping company has live package-tracking data that is sent to an Apache Kafka stream in real time. This is then loaded into BigQuery. Analysts in your company want to query the tracking data in BigQuery to analyze geospatial trends in the lifecycle of a package. The table was originally created with ingest-date partitioning. Over time, the query processing time has increased. You need to copy all the data to a new clustered table. What should you do?

A. Re-create the table using data partitioning on the package delivery date.

B. Implement clustering in BigQuery on the package-tracking ID column.

C. Implement clustering in BigQuery on the ingest date column.

D. Tier older data onto Cloud Storage files and create a BigQuery table using Cloud Storage as an external data source.

You are designing a data mesh on Google Cloud with multiple distinct data engineering teams building data products. The typical data curation design pattern consists of landing files in Cloud Storage, transforming raw data in Cloud Storage and BigQuery datasets, and storing the final curated data product in BigQuery datasets. You need to configure Dataplex to ensure that each team can access only the assets needed to build their data products. You also need to ensure that teams can easily share the curated data product. What should you do?

A.

1. Create a single Dataplex virtual lake and create a single zone to contain landing, raw, and curated data.
2. Provide each data engineering team access to the virtual lake.

B.

1. Create a single Dataplex virtual lake and create a single zone to contain landing, raw, and curated data.
2. Build separate assets for each data product within the zone.
3. Assign permissions to the data engineering teams at the zone level.

C.

1. Create a Dataplex virtual lake for each data product, and create a single zone to contain landing, raw, and curated data.
2. Provide the data engineering teams with full access to the virtual lake assigned to their data product.

D.

1. Create a Dataplex virtual lake for each data product, and create multiple zones for landing, raw, and curated data.
2. Provide the data engineering teams with full access to the virtual lake assigned to their data product.

You are using BigQuery with a multi-region dataset that includes a table with the daily sales volumes. This table is updated multiple times per day. You need to protect your sales table in case of regional failures with a recovery point objective (RPO) of less than 24 hours, while keeping costs to a minimum. What should you do?

A. Schedule a daily export of the table to a Cloud Storage dual or multi-region bucket.

B. Schedule a daily copy of the dataset to a backup region.

C. Schedule a daily BigQuery snapshot of the table.

D. Modify ETL job to load the data into both the current and another backup region.

You are troubleshooting your Dataflow pipeline that processes data from Cloud Storage to BigQuery. You have discovered that the Dataflow worker nodes cannot communicate with one another. Your networking team relies on Google Cloud network tags to define firewall rules. You need to identify the issue while following Google-recommended networking security practices. What should you do?

A. Determine whether your Dataflow pipeline has a custom network tag set.

B. Determine whether there is a firewall rule set to allow traffic on TCP ports 12345 and 12346 for the Dataflow network tag.

C. Determine whether there is a firewall rule set to allow traffic on TCP ports 12345 and 12346 on the subnet used by Dataflow workers.

D. Determine whether your Dataflow pipeline is deployed with the external IP address option enabled.

Your company's customer_order table in BigQuery stores the order history for 10 million customers, with a table size of 10 PB. You need to create a dashboard for the support team to view the order history. The dashboard has two filters, country_name and username. Both are string data types in the BigQuery table. When a filter is applied, the dashboard fetches the order

history from the table and displays the query results. However, the dashboard is slow to show the results when applying the filters to the following query:

SELECT date, order, status, FROM customer_order

Where country = 'country_name' AND username = '<username>'

How should you redesign the BigQuery table to support faster access?

A. Cluster the table by country and username fields.

B. Cluster the table by country field, and partition by username field.

C. Partition the table by country and username fields.

D. Partition the table by _PARTITIONTIME.

You have a Standard Tier Memorystore for Redis instance deployed in a production environment. You need to simulate a Redis instance failover in the most accurate disaster recovery situation and ensure that the failover has no impact on production data. What should you do?

A. Create a Standard Tier Memorystore for Redis instance in the development environment. Initiate a manual failover by using the limited-data-loss data protection mode.

B. Create a Standard Tier Memorystore for Redis instance in a development environment. Initiate a manual failover by using the force-data-loss data protection mode.

C. Increase one replica to Redis instance in production environment. Initiate a manual failover by using the force-data-loss data protection mode.

D. Initiate a manual failover by using the limited-data-loss data protection mode to the Memorystore for Redis instance in the production environment.

You are administering a BigQuery dataset that uses a customer-managed encryption key (CMEK). You need to share the dataset with a partner organization that does not have access to your CMEK. What should you do?

A. Provide the partner organization a copy of your CMEKs to decrypt the data.

B. Export the tables to parquet files to a Cloud Storage bucket and grant the storageinsights.viewer role on the bucket to the partner organization.

C. Copy the tables you need to share to a dataset without CMEKs. Create an Analytics Hub listing for this dataset.

D. Create an authorized view that contains the CMEK to decrypt the data when accessed.

You are developing an Apache Beam pipeline to extract data from a Cloud SQL instance by using JdbcIO. You have two projects running in Google Cloud. The pipeline will be deployed and

executed on Dataflow in Project A. The Cloud SQL. instance is running in Project B and does not have a public IP address. After deploying the pipeline, you noticed that the pipeline failed to extract data from the Cloud SQL instance due to connection failure. You verified that VPC Service Controls and shared VPC are not in use in these projects. You want to resolve this error while ensuring that the data does not go through the public internet. What should you do?

A. Set up VPC Network Peering between Project A and Project B. Add a firewall rule to allow the peered subnet range to access all instances on the network.

B. Turn off the external IP addresses on the Dataflow worker. Enable Cloud NAT in Project A.

C. Add the external IP addresses of the Dataflow worker as authorized networks in the Cloud SQL instance.

D. Set up VPC Network Peering between Project A and Project B. Create a Compute Engine instance without external IP address in Project B on the peered subnet to serve as a proxy server to the Cloud SQL database.

You have a BigQuery table that contains customer data, including sensitive information such as names and addresses. You need to share the customer data with your data analytics and consumer support teams securely. The data analytics team needs to access the data of all the customers but must not be able to access the sensitive data. The consumer support team needs access to all data columns but must not be able to access customers that no longer have active contracts. You enforced these requirements by using an authorized dataset and policy tags. After implementing these steps, the data analytics team reports that they still have access to the sensitive columns. You need to ensure that the data analytics team does not have access to restricted data. What should you do? (Choose two.)

A. Create two separate authorized datasets; one for the data analytics team and another for the consumer support team.

B. Ensure that the data analytics team members do not have the Data Catalog Fine-Grained Reader role for the policy tags.

C. Replace the authorized dataset with an authorized view. Use row-level security and apply filter_expression to limit data access.

D. Remove the bigquery.dataViewer role from the data analytics team on the authorized datasets.

E. Enforce access control in the policy tag taxonomy.

You have a Cloud SQL for PostgreSQL instance in Region' with one read replica in Region2 and another read replica in Region3. An unexpected event in Region' requires that you perform disaster recovery by promoting a read replica in Region2. You need to ensure that your application has the same database capacity available before you switch over the connections. What should you do?

A. Enable zonal high availability on the primary instance. Create a new read replica in a new region.

B. Create a cascading read replica from the existing read replica in Region3.

C. Create two new read replicas from the new primary instance, one in Region3 and one in a new region.

D. Create a new read replica in Region1, promote the new read replica to be the primary instance, and enable zonal high availability.

You orchestrate ETL pipelines by using Cloud Composer. One of the tasks in the Apache Airflow directed acyclic graph (DAG) relies on a third-party service. You want to be notified when the task fails. What should you do?

A. Assign a function with notification logic to the on_retry_callback parameter for the operator responsible for the task at risk.

B. Configure a Cloud Monitoring alert on the sla_missed metric associated with the task at risk to trigger a notification.

C. Assign a function with notification logic to the on_failure_callback parameter tor the operator responsible for the task at risk.

D. Assign a function with notification logic to the sla_miss_callback parameter for the operator responsible for the task at risk.

You are migrating your on-premises data warehouse to BigQuery. One of the upstream data sources resides on a MySQL. database that runs in your on-premises data center with no public IP addresses. You want to ensure that the data ingestion into BigQuery is done securely and does not go through the public internet. What should you do?

A. Update your existing on-premises ETL tool to write to BigQuery by using the BigQuery Open Database Connectivity (ODBC) driver. Set up the proxy parameter in the simba.googlebigqueryodbc.ini file to point to your data center's NAT gateway.

B. Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Set up Cloud Interconnect between your on-premises data center and Google Cloud. Use Private connectivity as the connectivity method and allocate an IP address range within your VPC network to the Datastream connectivity configuration. Use Server-only as the encryption type when setting up the connection profile in Datastream.

C. Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Use Forward-SSH tunnel as the connectivity method to establish a secure tunnel between Datastream and your on-premises MySQL database through a tunnel server in your on-premises data center. Use None as the encryption type when setting up the connection profile in Datastream.

D. Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Gather Datastream public IP addresses of the Google Cloud region that will be used to set up the stream. Add

those IP addresses to the firewall allowlist of your on-premises data center. Use IP Allowlisting as the connectivity method and Server-only as the encryption type when setting up the connection profile in Datastream.

You store and analyze your relational data in BigQuery on Google Cloud with all data that resides in US regions. You also have a variety of object stores across Microsoft Azure and Amazon Web Services (AWS), also in US regions. You want to query all your data in BigQuery daily with as little movement of data as possible. What should you do?

A. Use BigQuery Data Transfer Service to load files from Azure and AWS into BigQuery.

B. Create a Dataflow pipeline to ingest files from Azure and AWS to BigQuery.

C. Load files from AWS and Azure to Cloud Storage with Cloud Shell gsutil rsync arguments.

D. Use the BigQuery Omni functionality and BigLake tables to query files in Azure and AWS.

You have a variety of files in Cloud Storage that your data science team wants to use in their models. Currently, users do not have a method to explore, cleanse, and validate the data in Cloud Storage. You are looking for a low code solution that can be used by your data science team to quickly cleanse and explore data within Cloud Storage. What should you do?

A. Provide the data science team access to Dataflow to create a pipeline to prepare and validate the raw data and load data into BigQuery for data exploration.

B. Create an external table in BigQuery and use SQL to transform the data as necessary. Provide the data science team access to the external tables to explore the raw data.

C. Load the data into BigQuery and use SQL to transform the data as necessary. Provide the data science team access to staging tables to explore the raw data.

D. Provide the data science team access to Dataprep to prepare, validate, and explore the data within Cloud Storage.

You are building an ELT solution in BigQuery by using Dataform. You need to perform uniqueness and null value checks on your final tables. What should you do to efficiently integrate these checks into your pipeline?

A. Build BigQuery user-defined functions (UDFs).

B. Create Dataplex data quality tasks.

C. Build Dataform assertions into your code.

D. Write a Spark-based stored procedure.

A web server sends click events to a Pub/Sub topic as messages. The web server includes an eventTimestamp attribute in the messages, which is the time when the click occurred. You have a Dataflow streaming job that reads from this Pub/Sub topic through a subscription, applies some transformations, and writes the result to another Pub/Sub topic for use by the advertising department. The advertising department needs to receive each message within 30 seconds of the corresponding click occurrence, but they report receiving the messages late. Your Dataflow job's system lag is about 5 seconds, and the data freshness is about 40 seconds. Inspecting a few messages show no more than 1 second lag between their eventTimestamp and publishTime. What is the problem and what should you do?

A. The advertising department is causing delays when consuming the messages. Work with the advertising department to fix this.

B. Messages in your Dataflow job are taking more than 30 seconds to process. Optimize your job or increase the number of workers to fix this.

C. Messages in your Dataflow job are processed in less than 30 seconds, but your job cannot keep up with the backlog in the Pub/Sub subscription. Optimize your job or increase the number of workers to fix this.

D. The web server is not pushing messages fast enough to Pub/Sub. Work with the web server team to fix this.

Your organization stores customer data in an on-premises Apache Hadoop cluster in Apache Parquet format. Data is processed on a daily basis by Apache Spark jobs that run on the cluster. You are migrating the Spark jobs and Parquet data to Google Cloud. BigQuery will be used on future transformation pipelines so you need to ensure that your data is available in BigQuery. You want to use managed services, while minimizing ETL data processing changes and overhead costs. What should you do?

A. Migrate your data to Cloud Storage and migrate the metadata to Dataproc Metastore (DPMS). Refactor Spark pipelines to write and read data on Cloud Storage, and run them on Dataproc Serverless.

B. Migrate your data to Cloud Storage and register the bucket as a Dataplex asset. Refactor Spark pipelines to write and read data on Cloud Storage, and run them on Dataproc Serverless.

C. Migrate your data to BigQuery. Refactor Spark pipelines to write and read data on BigQuery, and run them on Dataproc Serverless.

D. Migrate your data to BigLake. Refactor Spark pipelines to write and read data on Cloud Storage, and run them on Dataproc on Compute Engine.

Your organization has two Google Cloud projects, project A and project B. In project A, you have a Pub/Sub topic that receives data from confidential sources. Only the resources in project A

should be able to access the data in that topic. You want to ensure that project B and any future project cannot access data in the project A topic. What should you do?

A. Add firewall rules in project A so only traffic from the VPC in project A is permitted.

B. Configure VPC Service Controls in the organization with a perimeter around project A.

C. Use Identity and Access Management conditions to ensure that only users and service accounts in project A. can access resources in project A.

D. Configure VPC Service Controls in the organization with a perimeter around the VPC of project A.

You stream order data by using a Dataflow pipeline and write the aggregated result to Memorystore. You provisioned a Memorystore for Redis instance with Basic Tier, 4 GB capacity, which is used by 40 clients for read-only access. You are expecting the number of read-only clients to increase significantly to a few hundred and you need to be able to support the demand. You want to ensure that read and write access availability is not impacted, and any changes you make can be deployed quickly. What should you do?

A. Create a new Memorystore for Redis instance with Standard Tier. Set capacity to 4 GB and read replica to No read replicas (high availability only). Delete the old instance.

B. Create a new Memorystore for Redis instance with Standard Tier. Set capacity to 5 GB and create multiple read replicas. Delete the old instance.

C. Create a new Memorystore for Memcached instance. Set a minimum of three nodes, and memory per node to 4 GB. Modify the Dataflow pipeline and all clients to use the Memcached instance. Delete the old instance.

D. Create multiple new Memorystore for Redis instances with Basic Tier (4 GB capacity). Modify the Dataflow pipeline and new clients to use all instances.

You have a streaming pipeline that ingests data from Pub/Sub in production. You need to update this streaming pipeline with improved business logic. You need to ensure that the updated pipeline reprocesses the previous two days of delivered Pub/Sub messages. What should you do? (Choose two.)

A. Use the Pub/Sub subscription clear-retry-policy flag

B. Use Pub/Sub Snapshot capture two days before the deployment.

C. Create a new Pub/Sub subscription two days before the deployment.

D. Use the Pub/Sub subscription retain-acked-messages flag.

E. Use Pub/Sub Seek with a timestamp.

You currently use a SQL-based tool to visualize your data stored in BigQuery. The data visualizations require the use of outer joins and analytic functions. Visualizations must be based on data that is no less than 4 hours old. Business users are complaining that the visualizations are too slow to generate. You want to improve the performance of the visualization queries while minimizing the maintenance overhead of the data preparation pipeline. What should you do?

A. Create materialized views with the allow_non_incremental_definition option set to true for the visualization queries. Specify the max_staleness parameter to 4 hours and the enable_refresh parameter to true. Reference the materialized views in the data visualization tool.

B. Create views for the visualization queries. Reference the views in the data visualization tool.

C. Create a Cloud Function instance to export the visualization query results as parquet files to a Cloud Storage bucket. Use Cloud Scheduler to trigger the Cloud Function every 4 hours. Reference the parquet files in the data visualization tool.

D. Create materialized views for the visualization queries. Use the incremental updates capability of BigQuery materialized views to handle changed data automatically. Reference the materialized views in the data visualization tool.

You need to modernize your existing on-premises data strategy. Your organization currently uses:

- **Apache Hadoop clusters for processing multiple large data sets, including on-premises Hadoop Distributed File System (HDFS) for data replication.**
- **Apache Airflow to orchestrate hundreds of ETL pipelines with thousands of job steps.**

You need to set up a new architecture in Google Cloud that can handle your Hadoop workloads and requires minimal changes to your existing orchestration processes. What should you do?

A. Use Bigtable for your large workloads, with connections to Cloud Storage to handle any HDFS use cases. Orchestrate your pipelines with Cloud Composer.

B. Use Dataproc to migrate Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Orchestrate your pipelines with Cloud Composer.

C. Use Dataproc to migrate Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Convert your ETL pipelines to Dataflow.

D. Use Dataproc to migrate your Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Use Cloud Data Fusion to visually design and deploy your ETL pipelines.

You recently deployed several data processing jobs into your Cloud Composer 2 environment. You notice that some tasks are failing in Apache Airflow. On the monitoring dashboard, you see an increase in the total workers memory usage, and there were worker pod evictions. You need to resolve these errors. What should you do? (Choose two.)

A. Increase the directed acyclic graph (DAG) file parsing interval.

B. Increase the Cloud Composer 2 environment size from medium to large.

C. Increase the maximum number of workers and reduce worker concurrency.

D. Increase the memory available to the Airflow workers.

E. Increase the memory available to the Airflow triggerer.

You are on the data governance team and are implementing security requirements to deploy resources. You need to ensure that resources are limited to only the europe-west3 region. You want to follow Google-recommended practices. What should you do?

A. Set the constraints/gcp.resourceLocations organization policy constraint to in:europe-west3-locations.

B. Deploy resources with Terraform and implement a variable validation rule to ensure that the region is set to the europe-west3 region for all resources.

C. Set the constraints/gcp.resourceLocations organization policy constraint to in:eu-locations.

D. Create a Cloud Function to monitor all resources created and automatically destroy the ones created outside the europe-west3 region.

You are a BigQuery admin supporting a team of data consumers who run ad hoc queries and downstream reporting in tools such as Looker. All data and users are combined under a single organizational project. You recently noticed some slowness in query results and want to troubleshoot where the slowdowns are occurring. You think that there might be some job queuing or slot contention occurring as users run jobs, which slows down access to results. You need to investigate the query job information and determine where performance is being affected. What should you do?

A. Use slot reservations for your project to ensure that you have enough query processing capacity and are able to allocate available slots to the slower queries.

B. Use Cloud Monitoring to view BigQuery metrics and set up alerts that let you know when a certain percentage of slots were used.

C. Use available administrative resource charts to determine how slots are being used and how jobs are performing over time. Run a query on the INFORMATION_SCHEMA to review query performance.

D. Use Cloud Logging to determine if any users or downstream consumers are changing or deleting access grants on tagged resources.

You migrated a data backend for an application that serves 10 PB of historical product data for analytics. Only the last known state for a product, which is about 10 GB of data, needs to be served through an API to the other applications. You need to choose a cost-effective persistent

storage solution that can accommodate the analytics requirements and the API performance of up to 1000 queries per second (QPS) with less than 1 second latency. What should you do?

A.

1. Store the historical data in BigQuery for analytics.
2. Use a materialized view to precompute the last state of a product.
3. Serve the last state data directly from BigQuery to the API.

B.

1. Store the products as a collection in Firestore with each product having a set of historical changes.
2. Use simple and compound queries for analytics.
3. Serve the last state data directly from Firestore to the API.

C.

1. Store the historical data in Cloud SQL for analytics.
2. In a separate table, store the last state of the product after every product change.
3. Serve the last state data directly from Cloud SQL to the API.

D.

1. Store the historical data in BigQuery for analytics.
2. In a Cloud SQL table, store the last state of the product after every product change.
3. Serve the last state data directly from Cloud SQL to the API.

You want to schedule a number of sequential load and transformation jobs. Data files will be added to a Cloud Storage bucket by an upstream process. There is no fixed schedule for when the new data arrives. Next, a Dataproc job is triggered to perform some transformations and write the data to BigQuery. You then need to run additional transformation jobs in BigQuery. The transformation jobs are different for every table. These jobs might take hours to complete. You need to determine the most efficient and maintainable workflow to process hundreds of tables and provide the freshest data to your end users. What should you do?

A.

1. Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Cloud Storage, Dataproc, and BigQuery operators.
2. Use a single shared DAG for all tables that need to go through the pipeline.
3. Schedule the DAG to run hourly.

B.

1. Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Cloud Storage, Dataproc, and BigQuery operators.
2. Create a separate DAG for each table that needs to go through the pipeline.
3. Schedule the DAGs to run hourly.

C.

1. Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Dataproc and BigQuery operators.
2. Use a single shared DAG for all tables that need to go through the pipeline.
3. Use a Cloud Storage object trigger to launch a Cloud Function that triggers the DAG.

D.

1. Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Dataproc and BigQuery operators.
2. Create a separate DAG for each table that needs to go through the pipeline.
3. Use a Cloud Storage object trigger to launch a Cloud Function that triggers the DAG.

You are deploying a MySQL database workload onto Cloud SQL. The database must be able to scale up to support several readers from various geographic regions. The database must be highly available and meet low RTO and RPO requirements, even in the event of a regional outage. You need to ensure that interruptions to the readers are minimal during a database failover. What should you do?

A. Create a highly available Cloud SQL instance in region A. Create a highly available read replica in region B. Scale up read workloads by creating cascading read replicas in multiple regions. Backup the Cloud SQL instances to a multi-regional Cloud Storage bucket. Restore the Cloud SQL backup to a new instance in another region when Region A is down.

B. Create a highly available Cloud SQL instance in region A. Scale up read workloads by creating read replicas in multiple regions. Promote one of the read replicas when region A is down.

C. Create a highly available Cloud SQL instance in region A. Create a highly available read replica in region B. Scale up read workloads by creating cascading read replicas in multiple regions. Promote the read replica in region B when region A is down.

D. Create a highly available Cloud SQL instance in region A. Scale up read workloads by creating read replicas in the same region. Failover to the standby Cloud SQL instance when the primary instance fails.

You are planning to load some of your existing on-premises data into BigQuery on Google Cloud. You want to either stream or batch-load data, depending on your use case. Additionally, you want to mask some sensitive data before loading into BigQuery. You need to do this in a programmatic way while keeping costs to a minimum. What should you do?

A. Use Cloud Data Fusion to design your pipeline, use the Cloud DLP plug-in to de-identify data within your pipeline, and then move the data into BigQuery.

B. Use the BigQuery Data Transfer Service to schedule your migration. After the data is populated in BigQuery, use the connection to the Cloud Data Loss Prevention (Cloud DLP) API to de-identify the necessary data.

C. Create your pipeline with Dataflow through the Apache Beam SDK for Python, customizing separate options within your code for streaming, batch processing, and Cloud DLP. Select BigQuery as your data sink.

D. Set up Datastream to replicate your on-premise data on BigQuery.

You want to encrypt the customer data stored in BigQuery. You need to implement per-user crypto-deletion on data stored in your tables. You want to adopt native features in Google Cloud to avoid custom solutions. What should you do?

A. Implement Authenticated Encryption with Associated Data (AEAD) BigQuery functions while storing your data in BigQuery.

B. Create a customer-managed encryption key (CMEK) in Cloud KMS. Associate the key to the table while creating the table.

C. Create a customer-managed encryption key (CMEK) in Cloud KMS. Use the key to encrypt data before storing in BigQuery.

D. Encrypt your data during ingestion by using a cryptographic library supported by your ETL pipeline.

The data analyst team at your company uses BigQuery for ad-hoc queries and scheduled SQL pipelines in a Google Cloud project with a slot reservation of 2000 slots. However, with the recent introduction of hundreds of new non time-sensitive SQL pipelines, the team is encountering frequent quota errors. You examine the logs and notice that approximately 1500 queries are being triggered concurrently during peak time. You need to resolve the concurrency issue. What should you do?

A. Increase the slot capacity of the project with baseline as 0 and maximum reservation size as 3000.

B. Update SQL pipelines to run as a batch query, and run ad-hoc queries as interactive query jobs.

C. Increase the slot capacity of the project with baseline as 2000 and maximum reservation size as 3000.

D. Update SQL pipelines and ad-hoc queries to run as interactive query jobs.

You are designing a data mesh on Google Cloud by using Dataplex to manage data in BigQuery and Cloud Storage. You want to simplify data asset permissions. You are creating a customer virtual lake with two user groups:

- **Data engineers, which require full data lake access**
- **Analytic users, which require access to curated data.**

You need to assign access rights to these two groups. What should you do?

A.

1. Grant the dataplex.dataOwner role to the data engineer group on the customer data lake.
2. Grant the dataplex.dataReader role to the analytic user group on the customer curated zone.

B.

1. Grant the dataplex.dataReader role to the data engineer group on the customer data lake.
2. Grant the dataplex.dataOwner to the analytic user group on the customer curated zone.

C.

1. Grant the bigquery.dataOwner role on BigQuery datasets and the storage.objectCreator role on Cloud Storage buckets to data engineers.
2. Grant the bigquery.dataViewer role on BigQuery datasets and the storage.objectViewer role on Cloud Storage buckets to analytic users.

D.

1. Grant the bigquery.dataViewer role on BigQuery datasets and the storage.objectViewer role on Cloud Storage buckets to data engineers.
2. Grant the bigquery.dataOwner role on BigQuery datasets and the storage.objectEditor role on Cloud Storage buckets to analytic users.

You are designing the architecture of your application to store data in Cloud Storage. Your application consists of pipelines that read data from a Cloud Storage bucket that contains raw data and write the data to a second bucket after processing. You want to design an architecture with Cloud Storage resources that are capable of being resilient if a Google Cloud regional failure occurs. You want to minimize the recovery point objective (RPO) if a failure occurs, with no impact on applications that use the stored data. What should you do?

A. Adopt multi-regional Cloud Storage buckets in your architecture.

B. Adopt two regional Cloud Storage buckets, and update your application to write the output on both buckets.

C. Adopt a dual-region Cloud Storage bucket, and enable turbo replication in your architecture.

D. Adopt two regional Cloud Storage buckets, and create a daily task to copy from one bucket to the other.

You have designed an Apache Beam processing pipeline that reads from a Pub/Sub topic. The topic has a message retention duration of one day, and writes to a Cloud Storage bucket. You need to select a bucket location and processing strategy to prevent data loss in case of a regional outage with an RPO of 15 minutes. What should you do?

A.

1. Use a dual-region Cloud Storage bucket.
2. Monitor Dataflow metrics with Cloud Monitoring to determine when an outage occurs.
3. Seek the subscription back in time by 15 minutes to recover the acknowledged messages.
4. Start the Dataflow job in a secondary region.

B.

1. Use a multi-regional Cloud Storage bucket.
2. Monitor Dataflow metrics with Cloud Monitoring to determine when an outage occurs.
3. Seek the subscription back in time by 60 minutes to recover the acknowledged messages.
4. Start the Dataflow job in a secondary region.

C.

1. Use a regional Cloud Storage bucket.
2. Monitor Dataflow metrics with Cloud Monitoring to determine when an outage occurs.
3. Seek the subscription back in time by one day to recover the acknowledged messages
4. Start the Dataflow job in a secondary region and write in a bucket in the same region.

D.

1. Use a dual-region Cloud Storage bucket with turbo replication enabled.
2. Monitor Dataflow metrics with Cloud Monitoring to determine when an outage occurs.
3. Seek the subscription back in time by 60 minutes to recover the acknowledged messages.
4. Start the Dataflow job in a secondary region.

You are preparing data that your machine learning team will use to train a model using BigQueryML. They want to predict the price per square foot of real estate. The training data has a column for the price and a column for the number of square feet. Another feature column called 'feature1' contains null values due to missing data. You want to replace the nulls with zeros to keep more data points. Which query should you use?

1.
```
SELECT * EXCEPT(feature1),
IFNULL(feature1,0)  AS feature1_cleaned
FROM training_data;
```
2.
```
SELECT * EXCEPT(price, square_feet),
price/square_feet AS price_per_sqft
FROM training_data
WHERE feature1 IS NOT NULL;
```
3.
```
SELECT * EXCEPT(price, square_feet feature1),
price/square_feet AS price_per_sqft,
IFNULL(feature1,0)  AS feature1_cleaned
FROM training_data;
```
4.
```
SELECT *
FROM training_data
WHERE feature1 IS NOT NULL;
```

Different teams in your organization store customer and performance data in BigQuery. Each team needs to keep full control of their collected data, be able to query data within their projects, and be able to exchange their data with other teams. You need to implement an organization-wide solution, while minimizing operational tasks and costs. What should you do?

A. Ask each team to create authorized views of their data. Grant the biquery.jobUser role to each team.

B. Create a BigQuery scheduled query to replicate all customer data into team projects.

C. Ask each team to publish their data in Analytics Hub. Direct the other teams to subscribe to them.

D. Enable each team to create materialized views of the data they need to access in their projects.

You are developing a model to identify the factors that lead to sales conversions for your customers. You have completed processing your data. You want to continue through the model development lifecycle. What should you do next?

A. Use your model to run predictions on fresh customer input data.

B. Monitor your model performance, and make any adjustments needed.

C. Delineate what data will be used for testing and what will be used for training the model.

D. Test and evaluate your model on your curated data to determine how well the model performs.

You have one BigQuery dataset which includes customers' street addresses. You want to retrieve all occurrences of street addresses from the dataset. What should you do?

A. Write a SQL query in BigQuery by using REGEXP_CONTAINS on all tables in your dataset to find rows where the word "street" appears.

B. Create a deep inspection job on each table in your dataset with Cloud Data Loss Prevention and create an inspection template that includes the STREET_ADDRESS infoType.

C. Create a discovery scan configuration on your organization with Cloud Data Loss Prevention and create an inspection template that includes the STREET_ADDRESS infoType.

D. Create a de-identification job in Cloud Data Loss Prevention and use the masking transformation.

Your company operates in three domains: airlines, hotels, and ride-hailing services. Each domain has two teams: analytics and data science, which create data assets in BigQuery with the help of a central data platform team. However, as each domain is evolving rapidly, the central data platform team is becoming a bottleneck. This is causing delays in deriving insights from data, and resulting in stale data when pipelines are not kept up to date. You need to design a data mesh architecture by using Dataplex to eliminate the bottleneck. What should you do?

A.

1. Create one lake for each team. Inside each lake, create one zone for each domain.
2. Attach each of the BigQuery datasets created by the individual teams as assets to the respective zone.
3. Have the central data platform team manage all zones' data assets.

B.

1. Create one lake for each team. Inside each lake, create one zone for each domain.
2. Attach each of the BigQuery datasets created by the individual teams as assets to the respective zone.
3. Direct each domain to manage their own zone's data assets.

C.

1. Create one lake for each domain. Inside each lake, create one zone for each team.
2. Attach each of the BigQuery datasets created by the individual teams as assets to the respective zone.
3. Direct each domain to manage their own lake's data assets.

D.

1. Create one lake for each domain. Inside each lake, create one zone for each team.
2. Attach each of the BigQuery datasets created by the individual teams as assets to the respective zone.
3. Have the central data platform team manage all lakes' data assets.

dataset.inventory_vm sample records:

Row	id	name	components.**name**	components.**qty**
1	vm02781	d-jp-kfk-02-02	vcpu	2
			memory	8
			boot_disk	10
			disk_1	50
2	vm11490	i-jp-kfk-02-07	vcpu	16
			memory	64
			boot_disk	10
			disk_1	200
3	vm18130	i-jp-kfk-02-08	vcpu	8
			memory	8
			boot_disk	10

You have an inventory of VM data stored in the BigQuery table. You want to prepare the data for regular reporting in the most cost-effective way. You need to exclude VM rows with fewer than 8 vCPU in your report. What should you do?

A. Create a view with a filter to drop rows with fewer than 8 vCPU, and use the UNNEST operator.

B. Create a materialized view with a filter to drop rows with fewer than 8 vCPU, and use the WITH common table expression.

C. Create a view with a filter to drop rows with fewer than 8 vCPU, and use the WITH common table expression.

D. Use Dataflow to batch process and write the result to another BigQuery table.

Your team is building a data lake platform on Google Cloud. As a part of the data foundation design, you are planning to store all the raw data in Cloud Storage. You are expecting to ingest approximately 25 GB of data a day and your billing department is worried about the increasing cost of storing old data. The current business requirements are:

- **The old data can be deleted anytime.**
- **There is no predefined access pattern of the old data.**
- **The old data should be available instantly when accessed.**
- **There should not be any charges for data retrieval.**

What should you do to optimize for cost?

A. Create the bucket with the Autoclass storage class feature.

B. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to nearline, 90 days to coldline, and 365 days to archive storage class. Delete old data as needed.

C. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to coldline, 90 days to nearline, and 365 days to archive storage class. Delete old data as needed.

D. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to nearline, 45 days to coldline, and 60 days to archive storage class. Delete old data as needed.

Your company's data platform ingests CSV file dumps of booking and user profile data from upstream sources into Cloud Storage. The data analyst team wants to join these datasets on the email field available in both the datasets to perform analysis. However, personally identifiable information (PII) should not be accessible to the analysts. You need to de-identify the email field in both the datasets before loading them into BigQuery for analysts. What should you do?

A.

1. Create a pipeline to de-identify the email field by using recordTransformations in Cloud Data Loss Prevention (Cloud DLP) with masking as the de-identification transformations type.
2. Load the booking and user profile data into a BigQuery table.

B.

1. Create a pipeline to de-identify the email field by using recordTransformations in Cloud DLP with format-preserving encryption with FFX as the de-identification transformation type.
2. Load the booking and user profile data into a BigQuery table.

C.

1. Load the CSV files from Cloud Storage into a BigQuery table and enable dynamic data masking.
2. Create a policy tag with the email mask as the data masking rule.
3. Assign the policy to the email field in both tables.
4. Assign the Identity and Access Management bigquerydatapolicy.maskedReader role for the BigQuery tables to the analysts.

D.

1. Load the CSV files from Cloud Storage into a BigQuery table, and enable dynamic data masking
2. Create a policy tag with the default masking value as the data masking rule
3. Assign the policy to the email field in both tables.
4. Assign the Identity and Access Management bigquerydatapolicy.maskedReader role for the BigQuery tables to the analysts.

You have important legal hold documents in a Cloud Storage bucket. You need to ensure that these documents are not deleted or modified. What should you do?

A. Set a retention policy. Lock the retention policy.

B. Set a retention policy. Set the default storage class to Archive for long-term digital preservation.

C. Enable the Object Versioning feature. Add a lifecycle rule.

D. Enable the Object Versioning feature. Create a copy in a bucket in a different region.

You are designing a data warehouse in BigQuery to analyze sales data for a telecommunication service provider. You need to create a data model for customers, products, and subscriptions. All customers, products, and subscriptions can be updated monthly, but you must maintain a historical record of all data. You plan to use the visualization layer for current and historical reporting. You need to ensure that the data model is simple, easy-to-use, and cost-effective. What should you do?

A. Create a normalized model with tables for each entity. Use snapshots before updates to track historical data.

B. Create a normalized model with tables for each entity. Keep all input files in a Cloud Storage bucket to track historical data.

C. Create a denormalized model with nested and repeated fields. Update the table and use snapshots to track historical data.

D. Create a denormalized, append-only model with nested and repeated fields. Use the ingestion timestamp to track historical data.

You are deploying a batch pipeline in Dataflow. This pipeline reads data from Cloud Storage, transforms the data, and then writes the data into BigQuery. The security team has enabled an organizational constraint in Google Cloud, requiring all Compute Engine instances to use only internal IP addresses and no external IP addresses. What should you do?

A. Ensure that your workers have network tags to access Cloud Storage and BigQuery. Use Dataflow with only internal IP addresses.

B. Ensure that the firewall rules allow access to Cloud Storage and BigQuery. Use Dataflow with only internal IPs.

C. Create a VPC Service Controls perimeter that contains the VPC network and add Dataflow, Cloud Storage, and BigQuery as allowed services in the perimeter. Use Dataflow with only internal IP addresses.

D. Ensure that Private Google Access is enabled in the subnetwork. Use Dataflow with only internal IP addresses.

You are running a Dataflow streaming pipeline, with Streaming Engine and Horizontal Autoscaling enabled. You have set the maximum number of workers to 1000. The input of your pipeline is Pub/Sub messages with notifications from Cloud Storage. One of the pipeline transforms reads CSV files and emits an element for every CSV line. The job performance is low, the pipeline is using only 10 workers, and you notice that the autoscaler is not spinning up additional workers. What should you do to improve performance?

A. Enable Vertical Autoscaling to let the pipeline use larger workers.

B. Change the pipeline code and introduce a Reshuffle step to prevent fusion.

C. Update the job to increase the maximum number of workers.

D. Use Dataflow Prime and enable Right Fitting to increase the worker resources.

You have an Oracle database deployed in a VM as part of a Virtual Private Cloud (VPC) network. You want to replicate and continuously synchronize 50 tables to BigQuery. You want to minimize the need to manage infrastructure. What should you do?

A. Deploy Apache Kafka in the same VPC network, use Kafka Connect Oracle Change Data Capture (CDC), and Dataflow to stream the Kafka topic to BigQuery.

B. Create a Pub/Sub subscription to write to BigQuery directly. Deploy the Debezium Oracle connector to capture changes in the Oracle database, and sink to the Pub/Sub topic.

C. Deploy Apache Kafka in the same VPC network, use Kafka Connect Oracle change data capture (CDC), and the Kafka Connect Google BigQuery Sink Connector.

D. Create a Datastream service from Oracle to BigQuery, use a private connectivity configuration to the same VPC network, and a connection profile to BigQuery.

You are deploying an Apache Airflow directed acyclic graph (DAG) in a Cloud Composer 2 instance. You have incoming files in a Cloud Storage bucket that the DAG processes, one file at a time. The Cloud Composer instance is deployed in a subnetwork with no Internet access. Instead of running the DAG based on a schedule, you want to run the DAG in a reactive way every time a new file is received. What should you do?

A.

1. Enable Private Google Access in the subnetwork, and set up Cloud Storage notifications to a Pub/Sub topic.
2. Create a push subscription that points to the web server URL.

B.

1. Enable the Cloud Composer API, and set up Cloud Storage notifications to trigger a Cloud Function.
2. Write a Cloud Function instance to call the DAG by using the Cloud Composer API and the web server URL.
3. Use VPC Serverless Access to reach the web server URL.

C.

1. Enable the Airflow REST API, and set up Cloud Storage notifications to trigger a Cloud Function instance.
2. Create a Private Service Connect (PSC) endpoint.
3. Write a Cloud Function that connects to the Cloud Composer cluster through the PSC endpoint.

D.

1. Enable the Airflow REST API, and set up Cloud Storage notifications to trigger a Cloud Function instance.
2. Write a Cloud Function instance to call the DAG by using the Airflow REST API and the web server URL.
3. Use VPC Serverless Access to reach the web server URL.

You are planning to use Cloud Storage as part of your data lake solution. The Cloud Storage bucket will contain objects ingested from external systems. Each object will be ingested once,

and the access patterns of individual objects will be random. You want to minimize the cost of storing and retrieving these objects. You want to ensure that any cost optimization efforts are transparent to the users and applications. What should you do?

A. Create a Cloud Storage bucket with Autoclass enabled.

B. Create a Cloud Storage bucket with an Object Lifecycle Management policy to transition objects from Standard to Coldline storage class if an object age reaches 30 days.

C. Create a Cloud Storage bucket with an Object Lifecycle Management policy to transition objects from Standard to Coldline storage class if an object is not live.

D. Create two Cloud Storage buckets. Use the Standard storage class for the first bucket, and use the Coldline storage class for the second bucket. Migrate objects from the first bucket to the second bucket after 30 days.

You have several different file type data sources, such as Apache Parquet and CSV. You want to store the data in Cloud Storage. You need to set up an object sink for your data that allows you to use your own encryption keys. You want to use a GUI-based solution. What should you do?

A. Use Storage Transfer Service to move files into Cloud Storage.

B. Use Cloud Data Fusion to move files into Cloud Storage.

C. Use Dataflow to move files into Cloud Storage.

D. Use BigQuery Data Transfer Service to move files into BigQuery.

Your business users need a way to clean and prepare data before using the data for analysis. Your business users are less technically savvy and prefer to work with graphical user interfaces to define their transformations. After the data has been transformed, the business users want to perform their analysis directly in a spreadsheet. You need to recommend a solution that they can use. What should you do?

A. Use Dataprep to clean the data, and write the results to BigQuery. Analyze the data by using Connected Sheets.

B. Use Dataprep to clean the data, and write the results to BigQuery. Analyze the data by using Looker Studio.

C. Use Dataflow to clean the data, and write the results to BigQuery. Analyze the data by using Connected Sheets.

D. Use Dataflow to clean the data, and write the results to BigQuery. Analyze the data by using Looker Studio.

You have two projects where you run BigQuery jobs:

- **One project runs production jobs that have strict completion time SLAs. These are high priority jobs that must have the required compute resources available when needed. These jobs generally never go below a 300 slot utilization, but occasionally spike up an additional 500 slots.**
- **The other project is for users to run ad-hoc analytical queries. This project generally never uses more than 200 slots at a time. You want these ad-hoc queries to be billed based on how much data users scan rather than by slot capacity. You need to ensure that both projects have the appropriate compute resources available. What should you do?**

A. Create a single Enterprise Edition reservation for both projects. Set a baseline of 300 slots. Enable autoscaling up to 700 slots.

B. Create two reservations, one for each of the projects. For the SLA project, use an Enterprise Edition with a baseline of 300 slots and enable autoscaling up to 500 slots. For the ad-hoc project, configure on-demand billing.

C. Create two Enterprise Edition reservations, one for each of the projects. For the SLA project, set a baseline of 300 slots and enable autoscaling up to 500 slots. For the ad-hoc project, set a reservation baseline of 0 slots and set the ignore idle slots flag to False.

D. Create two Enterprise Edition reservations, one for each of the projects. For the SLA project, set a baseline of 800 slots. For the ad-hoc project, enable autoscaling up to 200 slots.

You want to migrate your existing Teradata data warehouse to BigQuery. You want to move the historical data to BigQuery by using the most efficient method that requires the least amount of programming, but local storage space on your existing data warehouse is limited. What should you do?

A. Use BigQuery Data Transfer Service by using the Java Database Connectivity (JDBC) driver with FastExport connection.

B. Create a Teradata Parallel Transporter (TPT) export script to export the historical data, and import to BigQuery by using the bq command-line tool.

C. Use BigQuery Data Transfer Service with the Teradata Parallel Transporter (TPT) tbuild utility.

D. Create a script to export the historical data, and upload in batches to Cloud Storage. Set up a BigQuery Data Transfer Service instance from Cloud Storage to BigQuery.

You are on the data governance team and are implementing security requirements. You need to encrypt all your data in BigQuery by using an encryption key managed by your team. You must implement a mechanism to generate and store encryption material only on your on-premises hardware security module (HSM). You want to rely on Google managed solutions. What should you do?

A. Create the encryption key in the on-premises HSM, and import it into a Cloud Key Management Service (Cloud KMS) key. Associate the created Cloud KMS key while creating the BigQuery resources.

B. Create the encryption key in the on-premises HSM and link it to a Cloud External Key Manager (Cloud EKM) key. Associate the created Cloud KMS key while creating the BigQuery resources.

C. Create the encryption key in the on-premises HSM, and import it into Cloud Key Management Service (Cloud HSM) key. Associate the created Cloud HSM key while creating the BigQuery resources.

D. Create the encryption key in the on-premises HSM. Create BigQuery resources and encrypt data while ingesting them into BigQuery.

You maintain ETL pipelines. You notice that a streaming pipeline running on Dataflow is taking a long time to process incoming data, which causes output delays. You also noticed that the pipeline graph was automatically optimized by Dataflow and merged into one step. You want to identify where the potential bottleneck is occurring. What should you do?

A. Insert a Reshuffle operation after each processing step, and monitor the execution details in the Dataflow console.

B. Insert output sinks after each key processing step, and observe the writing throughput of each block.

C. Log debug information in each ParDo function, and analyze the logs at execution time.

D. Verify that the Dataflow service accounts have appropriate permissions to write the processed data to the output sinks.

You are running your BigQuery project in the on-demand billing model and are executing a change data capture (CDC) process that ingests data. The CDC process loads 1 GB of data every 10 minutes into a temporary table, and then performs a merge into a 10 TB target table. This process is very scan intensive and you want to explore options to enable a predictable cost model. You need to create a BigQuery reservation based on utilization information gathered from BigQuery Monitoring and apply the reservation to the CDC process. What should you do?

A. Create a BigQuery reservation for the dataset.

B. Create a BigQuery reservation for the job.

C. Create a BigQuery reservation for the service account running the job.

D. Create a BigQuery reservation for the project.

You are designing a fault-tolerant architecture to store data in a regional BigQuery dataset. You need to ensure that your application is able to recover from a corruption event in your tables that occurred within the past seven days. You want to adopt managed services with the lowest RPO and most cost-effective solution. What should you do?

A. Access historical data by using time travel in BigQuery.

B. Export the data from BigQuery into a new table that excludes the corrupted data

C. Create a BigQuery table snapshot on a daily basis.

D. Migrate your data to multi-region BigQuery buckets.

You are building a streaming Dataflow pipeline that ingests noise level data from hundreds of sensors placed near construction sites across a city. The sensors measure noise level every ten seconds and send that data to the pipeline when levels reach above 70 dBA. You need to detect the average noise level from a sensor when data is received for a duration of more than 30 minutes, but the window ends when no data has been received for 15 minutes. What should you do?

A. Use session windows with a 15-minute gap duration.

B. Use session windows with a 30-minute gap duration.

C. Use hopping windows with a 15-minute window, and a thirty-minute period.

D. Use tumbling windows with a 15-minute window and a fifteen-minute .withAllowedLateness operator.

You are creating a data model in BigQuery that will hold retail transaction data. Your two largest tables, sales_transaction_header and sales_transaction_line, have a tightly coupled immutable relationship. These tables are rarely modified after load and are frequently joined when queried. You need to model the sales_transaction_header and sales_transaction_line tables to improve the performance of data analytics queries. What should you do?

A. Create a sales_transaction table that holds the sales_transaction_header information as rows and the sales_transaction_line rows as nested and repeated fields.

B. Create a sales_transaction table that holds the sales_transaction_header and sales_transaction_line information as rows, duplicating the sales_transaction_header data for each line.

C. Create a sales_transaction table that stores the sales_transaction_header and sales_transaction_line data as a JSON data type.

D. Create separate sales_transaction_header and sales_transaction_line tables and, when querying, specify the sales_transaction_line first in the WHERE clause.

You created a new version of a Dataflow streaming data ingestion pipeline that reads from Pub/Sub and writes to BigQuery. The previous version of the pipeline that runs in production uses a 5-minute window for processing. You need to deploy the new version of the pipeline without losing any data, creating inconsistencies, or increasing the processing latency by more than 10 minutes. What should you do?

A. Update the old pipeline with the new pipeline code.

B. Snapshot the old pipeline, stop the old pipeline, and then start the new pipeline from the snapshot.

C. Drain the old pipeline, then start the new pipeline.

D. Cancel the old pipeline, then start the new pipeline.

Your organization's data assets are stored in BigQuery, Pub/Sub, and a PostgreSQL instance running on Compute Engine. Because there are multiple domains and diverse teams using the data, teams in your organization are unable to discover existing data assets. You need to design a solution to improve data discoverability while keeping development and configuration efforts to a minimum. What should you do?

A. Use Data Catalog to automatically catalog BigQuery datasets. Use Data Catalog APIs to manually catalog Pub/Sub topics and PostgreSQL tables.

B. Use Data Catalog to automatically catalog BigQuery datasets and Pub/Sub topics. Use Data Catalog APIs to manually catalog PostgreSQL tables.

C. Use Data Catalog to automatically catalog BigQuery datasets and Pub/Sub topics. Use custom connectors to manually catalog PostgreSQL tables.

D. Use customer connectors to manually catalog BigQuery datasets, Pub/Sub topics, and PostgreSQL tables.

You need to create a SQL pipeline. The pipeline runs an aggregate SQL transformation on a BigQuery table every two hours and appends the result to another existing BigQuery table. You need to configure the pipeline to retry if errors occur. You want the pipeline to send an email notification after three consecutive failures. What should you do?

A. Use the BigQueryUpsertTableOperator in Cloud Composer, set the retry parameter to three, and set the email_on_failure parameter to true.

B. Use the BigQueryInsertJobOperator in Cloud Composer, set the retry parameter to three, and set the email_on_failure parameter to true.

C. Create a BigQuery scheduled query to run the SQL transformation with schedule options that repeats every two hours, and enable email notifications.

D. Create a BigQuery scheduled query to run the SQL transformation with schedule options that repeats every two hours, and enable notification to Pub/Sub topic. Use Pub/Sub and Cloud Functions to send an email after three failed executions.

You are monitoring your organization’s data lake hosted on BigQuery. The ingestion pipelines read data from Pub/Sub and write the data into tables on BigQuery. After a new version of the ingestion pipelines is deployed, the daily stored data increased by 50%. The volumes of data in Pub/Sub

remained the same and only some tables had their daily partition data size doubled. You need to investigate and fix the cause of the data increase. What should you do?

A.

1. Check for duplicate rows in the BigQuery tables that have the daily partition data size doubled.
2. Schedule daily SQL jobs to deduplicate the affected tables.
3. Share the deduplication script with the other operational teams to reuse if this occurs to other tables.

B.

1. Check for code errors in the deployed pipelines.
2. Check for multiple writing to pipeline BigQuery sink.
3. Check for errors in Cloud Logging during the day of the release of the new pipelines.
4. If no errors, restore the BigQuery tables to their content before the last release by using time travel.

C.

1. Check for duplicate rows in the BigQuery tables that have the daily partition data size doubled.
2. Check the BigQuery Audit logs to find job IDs.
3. Use Cloud Monitoring to determine when the identified Dataflow jobs started and the pipeline code version.
4. When more than one pipeline ingests data into a table, stop all versions except the latest one.

D.

1. Roll back the last deployment.
2. Restore the BigQuery tables to their content before the last release by using time travel.
3. Restart the Dataflow jobs and replay the messages by seeking the subscription to the timestamp of the release.

You have a BigQuery dataset named "customers". All tables will be tagged by using a Data Catalog tag template named "gdpr". The template contains one mandatory field, "has_sensitive_data", with a boolean value. All employees must be able to do a simple search and find tables in the dataset that have either true or false in the "has_sensitive_data' field. However, only the Human Resources (HR) group should be able to see the data inside the tables for which "has_sensitive data" is true. You give the all employees group the bigquery.metadataViewer and bigquery.connectionUser roles on the dataset. You want to minimize configuration overhead. What should you do next?

A. Create the "gdpr" tag template with private visibility. Assign the bigquery.dataViewer role to the HR group on the tables that contain sensitive data.

B. Create the "gdpr" tag template with private visibility. Assign the datacatalog.tagTemplateViewer role on this tag to the all employees group, and assign the bigquery.dataViewer role to the HR group on the tables that contain sensitive data.

C. Create the “gdpr” tag template with public visibility. Assign the bigquery.dataViewer role to the HR group on the tables that contain sensitive data.

D. Create the “gdpr” tag template with public visibility. Assign the datacatalog.tagTemplateViewer role on this tag to the all employees group, and assign the bigquery.dataViewer role to the HR group on the tables that contain sensitive data.

You are creating the CI/CD cycle for the code of the directed acyclic graphs (DAGs) running in Cloud Composer. Your team has two Cloud Composer instances: one instance for development and another instance for production. Your team is using a Git repository to maintain and develop the code of the DAGs. You want to deploy the DAGs automatically to Cloud Composer when a certain tag is pushed to the Git repository. What should you do?

A.

1. Use Cloud Build to copy the code of the DAG to the Cloud Storage bucket of the development instance for DAG testing.
2. If the tests pass, use Cloud Build to copy the code to the bucket of the production instance.

B

1. Use Cloud Build to build a container with the code of the DAG and the KubernetesPodOperator to deploy the code to the Google Kubernetes Engine (GKE) cluster of the development instance for testing.
2. If the tests pass, use the KubernetesPodOperator to deploy the container to the GKE cluster of the production instance.

C

1. Use Cloud Build to build a container and the KubernetesPodOperator to deploy the code of the DAG to the Google Kubernetes Engine (GKE) cluster of the development instance for testing
2. If the tests pass, copy the code to the Cloud Storage bucket of the production instance.

D.

1. Use Cloud Build to copy the code of the DAG to the Cloud Storage bucket of the development instance for DAG testing
2. If the tests pass, use Cloud Build to build a container with the code of the DAG and the KubernetesPodOperator to deploy the container to the Google Kubernetes Engine (GKE) cluster of the production instance.

You have a BigQuery table that ingests data directly from a Pub/Sub subscription. The ingested data is encrypted with a Google-managed encryption key. You need to meet a new organization policy that requires you to use keys from a centralized Cloud Key Management Service (Cloud KMS) project to encrypt data at rest. What should you do?

A. Use Cloud KMS encryption key with Dataflow to ingest the existing Pub/Sub subscription to the existing BigQuery table.

B. Create a new BigQuery table by using customer-managed encryption keys (CMEK), and migrate the data from the old BigQuery table.

C. Create a new Pub/Sub topic with CMEK and use the existing BigQuery table by using Google-managed encryption key.

D. Create a new BigQuery table and Pub/Sub topic by using customer-managed encryption keys (CMEK), and migrate the data from the old BigQuery table.

You created an analytics environment on Google Cloud so that your data scientist team can explore data without impacting the on-premises Apache Hadoop solution. The data in the on-premises Hadoop Distributed File System (HDFS) cluster is in Optimized Row Columnar (ORC) formatted files with multiple columns of Hive partitioning. The data scientist team needs to be able to explore the data in a similar way as they used the on-premises HDFS cluster with SQL on the Hive query engine. You need to choose the most cost-effective storage and processing solution. What should you do?

A. Import the ORC files to Bigtable tables for the data scientist team.

B. Import the ORC files to BigQuery tables for the data scientist team.

C. Copy the ORC files on Cloud Storage, then deploy a Dataproc cluster for the data scientist team.

D. Copy the ORC files on Cloud Storage, then create external BigQuery tables for the data scientist team.

You are designing a Dataflow pipeline for a batch processing job. You want to mitigate multiple zonal failures at job submission time. What should you do?

A. Submit duplicate pipelines in two different zones by using the --zone flag.

B. Set the pipeline staging location as a regional Cloud Storage bucket.

C. Specify a worker region by using the --region flag.

D. Create an Eventarc trigger to resubmit the job in case of zonal failure when submitting the job.

You are designing a real-time system for a ride hailing app that identifies areas with high demand for rides to effectively reroute available drivers to meet the demand. The system ingests data from multiple sources to Pub/Sub, processes the data, and stores the results for visualization and analysis in real-time dashboards. The data sources include driver location updates every 5 seconds and app-based booking events from riders. The data processing involves real-time aggregation of supply and demand data for the last 30 seconds, every 2 seconds, and storing the results in a low-latency system for visualization. What should you do?

A. Group the data by using a tumbling window in a Dataflow pipeline, and write the aggregated data to Memorystore.

B. Group the data by using a hopping window in a Dataflow pipeline, and write the aggregated data to Memorystore.

C. Group the data by using a session window in a Dataflow pipeline, and write the aggregated data to BigQuery.

D. Group the data by using a hopping window in a Dataflow pipeline, and write the aggregated data to BigQuery.

Your car factory is pushing machine measurements as messages into a Pub/Sub topic in your Google Cloud project. A Dataflow streaming job, that you wrote with the Apache Beam SDK, reads these messages, sends acknowledgment to Pub/Sub, applies some custom business logic in a DoFn instance, and writes the result to BigQuery. You want to ensure that if your business logic fails on a message, the message will be sent to a Pub/Sub topic that you want to monitor for alerting purposes. What should you do?

A. Enable retaining of acknowledged messages in your Pub/Sub pull subscription. Use Cloud Monitoring to monitor the subscription/num_retained_acked_messages metric on this subscription.

B. Use an exception handling block in your Dataflow's DoFn code to push the messages that failed to be transformed through a side output and to a new Pub/Sub topic. Use Cloud Monitoring to monitor the topic/num_unacked_messages_by_region metric on this new topic.

C. Enable dead lettering in your Pub/Sub pull subscription, and specify a new Pub/Sub topic as the dead letter topic. Use Cloud Monitoring to monitor the subscription/dead_letter_message_count metric on your pull subscription.

D. Create a snapshot of your Pub/Sub pull subscription. Use Cloud Monitoring to monitor the snapshot/num_messages metric on this snapshot.

You want to store your team's shared tables in a single dataset to make data easily accessible to various analysts. You want to make this data readable but unmodifiable by analysts. At the same time, you want to provide the analysts with individual workspaces in the same project, where they can create and store tables for their own use, without the tables being accessible by other analysts. What should you do?

A. Give analysts the BigQuery Data Viewer role at the project level. Create one other dataset, and give the analysts the BigQuery Data Editor role on that dataset.

B. Give analysts the BigQuery Data Viewer role at the project level. Create a dataset for each analyst, and give each analyst the BigQuery Data Editor role at the project level.

C. Give analysts the BigQuery Data Viewer role on the shared dataset. Create a dataset for each analyst, and give each analyst the BigQuery Data Editor role at the dataset level for their assigned dataset.

D. Give analysts the BigQuery Data Viewer role on the shared dataset. Create one other dataset and give the analysts the BigQuery Data Editor role on that dataset.

You are running a streaming pipeline with Dataflow and are using hopping windows to group the data as the data arrives. You noticed that some data is arriving late but is not being marked as late data, which is resulting in inaccurate aggregations downstream. You need to find a solution that allows you to capture the late data in the appropriate window. What should you do?

A. Use watermarks to define the expected data arrival window. Allow late data as it arrives.

B. Change your windowing function to tumbling windows to avoid overlapping window periods.

C. Change your windowing function to session windows to define your windows based on certain activity.

D. Expand your hopping window so that the late data has more time to arrive within the grouping.

You work for a large ecommerce company. You store your customer's order data in Bigtable. You have a garbage collection policy set to delete the data after 30 days and the number of versions is set to 1. When the data analysts run a query to report total customer spending, the analysts sometimes see customer data that is older than 30 days. You need to ensure that the analysts do not see customer data older than 30 days while minimizing cost and overhead. What should you do?

A. Set the expiring values of the column families to 29 days and keep the number of versions to 1.

B. Use a timestamp range filter in the query to fetch the customer's data for a specific range.

C. Schedule a job daily to scan the data in the table and delete data older than 30 days.

D. Set the expiring values of the column families to 30 days and set the number of versions to 2.

You are using a Dataflow streaming job to read messages from a message bus that does not support exactly once delivery. Your job then applies some transformations and loads the result into BigQuery. You want to ensure that your data is being streamed into BigQuery with exactly once delivery semantics. You expect your ingestion throughput into BigQuery to be about 1.5 GB per second. What should you do?

A. Use the BigQuery Storage Write API and ensure that your target BigQuery table is regional.

B. Use the BigQuery Storage Write API and ensure that your target BigQuery table is multiregional.

C. Use the BigQuery Streaming API and ensure that your target BigQuery table is regional.

D. Use the BigQuery Streaming API and ensure that your target BigQuery table is multiregional.

You have created an external table for Apache Hive partitioned data that resides in a Cloud Storage bucket, which contains a large number of files. You notice that queries against this table are slow. You want to improve the performance of these queries. What should you do?

A. Change the storage class of the Hive partitioned data objects from Coldline to Standard.

B. Create an individual external table for each Hive partition by using a common table name prefix. Use wildcard table queries to reference the partitioned data.

C. Upgrade the external table to a BigLake table. Enable metadata caching for the table.

D. Migrate the Hive partitioned data objects to a multi-region Cloud Storage bucket.

You have a network of 1000 sensors. The sensors generate time series data: one metric per sensor per second, along with a timestamp. You already have 1 TB of data and expect the data to grow by 1 GB every day. You need to access this data in two ways. The first access pattern requires retrieving the metric from one specific sensor stored at a specific timestamp, with a median single-digit millisecond latency. The second access pattern requires running complex analytic queries on the data, including joins, once a day. How should you store this data?

A. Store your data in BigQuery. Concatenate the sensor ID and timestamp and use it as the primary key.

B. Store your data in Bigtable. Concatenate the sensor ID and timestamp and use it as the row key. Perform an export to BigQuery every day.

C. Store your data in Bigtable. Concatenate the sensor ID and metric and use it as the row key. Perform an export to BigQuery every day.

D. Store your data in BigQuery. Use the metric as a primary key.

You have 100 GB of data stored in a BigQuery table. This data is outdated and will only be accessed one or two times a year for analytics with SQL. For backup purposes, you want to store this data to be immutable for 3 years. You want to minimize storage costs. What should you do?

A.

1. Create a BigQuery table clone.
2. Query the clone when you need to perform analytics.

B.

1. Create a BigQuery table snapshot.
2. Restore the snapshot when you need to perform analytics.

C.

1. Perform a BigQuery export to a Cloud Storage bucket with archive storage class.
2. Enable versioning on the bucket.
3. Create a BigQuery external table on the exported files.

D.

1. Perform a BigQuery export to a Cloud Storage bucket with archive storage class.
2. Set a locked retention policy on the bucket.
3. Create a BigQuery external table on the exported files.

You have thousands of Apache Spark jobs running in your on-premises Apache Hadoop cluster. You want to migrate the jobs to Google Cloud. You want to use managed services to run your jobs instead of maintaining a long-lived Hadoop cluster yourself. You have a tight timeline and want to keep code changes to a minimum. What should you do?

A. Move your data to BigQuery. Convert your Spark scripts to a SQL-based processing approach.

B. Rewrite your jobs in Apache Beam. Run your jobs in Dataflow.

C. Copy your data to Compute Engine disks. Manage and run your jobs directly on those instances.

D. Move your data to Cloud Storage. Run your jobs on Dataproc.

You are administering shared BigQuery datasets that contain views used by multiple teams in your organization. The marketing team is concerned about the variability of their monthly BigQuery analytics spend using the on-demand billing model. You need to help the marketing team establish a consistent BigQuery analytics spend each month. What should you do?

A. Create a BigQuery Enterprise reservation with a baseline of 250 slots and autoscaling set to 500 for the marketing team and bill them back accordingly.

B. Establish a BigQuery quota for the marketing team and limit the maximum number of bytes scanned each day.

C. Create a BigQuery reservation with a baseline of 500 slots with no autoscaling for the marketing team and bill them back accordingly.

D. Create a BigQuery Standard pay-as-you go reservation with a baseline of 0 slots and autoscaling set to 500 for the marketing team and bill them back accordingly.

You are part of a healthcare organization where data is organized and managed by respective data owners in various storage services. As a result of this decentralized ecosystem, discovering and managing data has become difficult. You need to quickly identify and implement a cost optimized solution to assist your organization with the following:

- **Data management and discovery**
- **Data lineage tracking**

- **Data quality validation**

How should you build the solution?

A. Use BigLake to convert the current solution into a data lake architecture.

B. Build a new data discovery tool on Google Kubernetes Engine that helps with new source onboarding and data lineage tracking.

C. Use BigQuery to track data lineage, and use Dataprep to manage data and perform data quality validation.

D. Use Dataplex to manage data, track data lineage, and perform data quality validation.

You have data located in BigQuery that is used to generate reports for your company. You have noticed some weekly executive report fields do not correspond to format according to company standards. For example, report errors include different telephone formats and different country code identifiers. This is a frequent issue, so you need to create a recurring job to normalize the data. You want a quick solution that requires no coding.

What should you do?

A. Use Cloud Data Fusion and Wrangler to normalize the data and set up a recurring job.

B. Use Dataflow SQL to create a job that normalizes the data, and that after the first run of the job, schedule the pipeline to execute recurrently.

C. Create a Spark job and submit it to Dataproc Serverless.

D. Use BigQuery and GoogleSQL to normalize the data, and schedule recurring queries in BigQuery.

You are designing a messaging system by using Pub/Sub to process clickstream data with an event-driven consumer app that relies on a push subscription. You need to configure the messaging system that is reliable enough to handle temporary downtime of the consumer app. You also need the messaging system to store the input messages that cannot be consumed by the subscriber. The system needs to retry failed messages gradually, avoiding overloading the consumer app, and store the failed messages after a maximum of 10 retries in a topic. How should you configure the Pub/Sub subscription?

A. Increase the acknowledgement deadline to 10 minutes.

B. Use immediate redelivery as the subscription retry policy and configure dead lettering to a different topic with maximum delivery attempts set to 10.

C. Use exponential backoff as the subscription retry policy and configure dead lettering to the same source topic with maximum delivery attempts set to 10.

D. Use exponential backoff as the subscription retry policy and configure dead lettering to a different topic with maximum delivery attempts set to 10.

You designed a data warehouse in BigQuery to analyze sales data. You want a self-serving, low-maintenance, and cost- effective solution to share the sales dataset to other business units in your organization. What should you do?

A. Create an Analytics Hub private exchange and publish the sales dataset.

B. Enable the other business units' projects to access the authorized views of the sales dataset.

C. Create and share views with the users in the other business units.

D. Use the BigQuery Data Transfer Service to create a schedule that copies the sales dataset to the other business units' projects.

You have terabytes of customer behavioural data streaming from Google Analytics into BigQuery daily. Your customers' information, such as their preferences, is hosted on a Cloud SQL for MySQL database. Your CRM database is hosted on a Cloud SQL for PostgreSQL instance. The marketing team wants to use your customers' information from the two databases and the customer behavioural data to create marketing campaigns for yearly active customers. You need to ensure that the marketing team can run the campaigns over 100 times a day on typical days and up to 300 during sales. At the same time, you want to keep the load on the Cloud SQL databases to a minimum. What should you do?

A. Create BigQuery connections to both Cloud SQL databases. Use BigQuery federated queries on the two databases and the Google Analytics data on BigQuery to run these queries.

B. Create a job on Apache Spark with Dataproc Serverless to query both Cloud SQL databases and the Google Analytics data on BigQuery for these queries.

C. Create streams in Datastream to replicate the required tables from both Cloud SQL databases to BigQuery for these queries.

D. Create a Dataproc cluster with Trino to establish connections to both Cloud SQL databases and BigQuery, to execute the queries.

Your organization is modernizing their IT services and migrating to Google Cloud. You need to organize the data that will be stored in Cloud Storage and BigQuery. You need to enable a data mesh approach to share the data between sales, product design, and marketing departments.

What should you do?

A.

1. Create a project for storage of the data for each of your departments.
2. Enable each department to create Cloud Storage buckets and BigQuery datasets.
3. Create user groups for authorized readers for each bucket and dataset.

4. Enable the IT team to administer the user groups to add or remove users as the departments' request.

B.

1. Create multiple projects for storage of the data for each of your departments' applications.
2. Enable each department to create Cloud Storage buckets and BigQuery datasets.
3. Publish the data that each department shared in Analytics Hub.
4. Enable all departments to discover and subscribe to the data they need in Analytics Hub.

C.

1. Create a project for storage of the data for your organization.
2. Create a central Cloud Storage bucket with three folders to store the files for each department.
3. Create a central BigQuery dataset with tables prefixed with the department name.
4. Give viewer rights for the storage project for the users of your departments.

D.

1. Create multiple projects for storage of the data for each of your departments' applications.
2. Enable each department to create Cloud Storage buckets and BigQuery datasets.
3. In Dataplex, map each department to a data lake and the Cloud Storage buckets, and map the BigQuery datasets to zones.
4. Enable each department to own and share the data of their data lakes.

You work for a large ecommerce company. You are using Pub/Sub to ingest the clickstream data to Google Cloud for analytics. You observe that when a new subscriber connects to an existing topic to analyze data, they are unable to subscribe to older data. For an upcoming yearly sale event in two months, you need a solution that, once implemented, will enable any new subscriber to read the last 30 days of data. What should you do?

A. Create a new topic and publish the last 30 days of data each time a new subscriber connects to an existing topic.

B. Set the topic retention policy to 30 days.

C. Set the subscriber retention policy to 30 days.

D. Ask the source system to re-push the data to Pub/Sub and subscribe to it.

You are designing the architecture to process your data from Cloud Storage to BigQuery by using Dataflow. The network team provided you with the Shared VPC network and subnetwork to be used by your pipelines. You need to enable the deployment of the pipeline on the Shared VPC network. What should you do?

A. Assign the compute.networkUser role to the Dataflow service agent.

B. Assign the compute.networkUser role to the service account that executes the Dataflow pipeline.

C. Assign the dataflow.admin role to the Dataflow service agent.

D. Assign the dataflow.admin role to the service account that executes the Dataflow pipeline.

Your infrastructure team has set up an interconnect link between Google Cloud and the on-premises network. You are designing a high-throughput streaming pipeline to ingest data in streaming from an Apache Kafka cluster hosted on- premises. You want to store the data in BigQuery, with as minimal latency as possible. What should you do?

A. Setup a Kafka Connect bridge between Kafka and Pub/Sub. Use a Google-provided Dataflow template to read the data from Pub/Sub and write the data to BigQuery.

B. Use a proxy host in the VPC in Google Cloud connecting to Kafka. Write a Dataflow pipeline, read data from the proxy host, and write the data to BigQuery.

C. Use Dataflow, write a pipeline that reads the data from Kafka, and writes the data to BigQuery.

D. Setup a Kafka Connect bridge between Kafka and Pub/Sub. Write a Dataflow pipeline, read the data from Pub/Sub, and write the data to BigQuery.

You migrated your on-premises Apache Hadoop Distributed File System (HDFS) data lake to Cloud Storage. The data scientist team needs to process the data by using Apache Spark and SQL. Security policies need to be enforced at the column level. You need a cost-effective solution that can scale into a data mesh. What should you do?

A.

1. Deploy a long-living Dataproc cluster with Apache Hive and Ranger enabled.
2. Configure Ranger for column level security.
3. Process with Dataproc Spark or Hive SQL.

B.

1. Define a BigLake table.
2. Create a taxonomy of policy tags in Data Catalog.
3. Add policy tags to columns.
4. Process with the Spark-BigQuery connector or BigQuery SQL.

C.

1. Load the data to BigQuery tables.
2. Create a taxonomy of policy tags in Data Catalog.
3. Add policy tags to columns.
4. Process with the Spark-BigQuery connector or BigQuery SQL.

D.

1. Apply an Identity and Access Management (IAM) policy at the file level in Cloud Storage.
2. Define a BigQuery external table for SQL processing.

3. Use Dataproc Spark to process the Cloud Storage files.

One of your encryption keys stored in Cloud Key Management Service (Cloud KMS) was exposed. You need to re- encrypt all your CMEK protected Cloud Storage data that used that key, and then delete the compromised key. You also want to reduce the risk of objects getting written without customer-managed encryption key (CMEK) protection in the future. What should you do?

A. Rotate the Cloud KMS key version. Continue to use the same Cloud Storage bucket.

B. Create a new Cloud KMS key. Set the default CMEK key on the existing Cloud Storage bucket to the new one.

C. Create a new Cloud KMS key. Create a new Cloud Storage bucket. Copy all objects from the old bucket to the new one bucket while specifying the new Cloud KMS key in the copy command.

D. Create a new Cloud KMS key. Create a new Cloud Storage bucket configured to use the new key as the default CMEK key. Copy all objects from the old bucket to the new bucket without specifying a key.

You have an upstream process that writes data to Cloud Storage. This data is then read by an Apache Spark job that runs on Dataproc. These jobs are run in the us-central1 region, but the data could be stored anywhere in the United States. You need to have a recovery process in place in case of a catastrophic single region failure. You need an approach with a maximum of 15 minutes of data loss (RPO=15 mins). You want to ensure that there is minimal latency when reading the data. What should you do?

A.

1. Create two regional Cloud Storage buckets, one in the us-central1 region and one in the us-south1 region.
2. Have the upstream process write data to the us-central1 bucket. Use the Storage Transfer Service to copy data hourly from the us-central1 bucket to the us-south1 bucket.
3. Run the Dataproc cluster in a zone in the us-central1 region, reading from the bucket in that region.
4. In case of regional failure, redeploy your Dataproc clusters to the us-south1 region and read from the bucket in that region instead.

B.

1. Create a Cloud Storage bucket in the US multi-region.
2. Run the Dataproc cluster in a zone in the us-central1 region, reading data from the US multi-region bucket.
3. In case of a regional failure, redeploy the Dataproc cluster to the us-central2 region and continue reading from the same bucket.

C.

1. Create a dual-region Cloud Storage bucket in the us-central1 and us-south1 regions.

2. Enable turbo replication.
3. Run the Dataproc cluster in a zone in the us-central1 region, reading from the bucket in the us-south1 region.
4. In case of a regional failure, redeploy your Dataproc cluster to the us-south1 region and continue reading from the same bucket.

D.

1. Create a dual-region Cloud Storage bucket in the us-central1 and us-south1 regions.
2. Enable turbo replication.
3. Run the Dataproc cluster in a zone in the us-central1 region, reading from the bucket in the same region.
4. In case of a regional failure, redeploy the Dataproc clusters to the us-south1 region and read from the same bucket.

You currently have transactional data stored on-premises in a PostgreSQL database. To modernize your data environment, you want to run transactional workloads and support analytics needs with a single database. You need to move to Google Cloud without changing database management systems and minimize cost and complexity. What should you do?

A. Migrate and modernize your database with Cloud Spanner.

B. Migrate your workloads to AlloyDB for PostgreSQL.

C. Migrate to BigQuery to optimize analytics.

D. Migrate your PostgreSQL database to Cloud SQL for PostgreSQL.

You are architecting a data transformation solution for BigQuery. Your developers are proficient with SQL and want to use the ELT development technique. In addition, your developers need an intuitive coding environment and the ability to manage SQL as code. You need to identify a solution for your developers to build these pipelines. What should you do?

A. Use Dataform to build, manage, and schedule SQL pipelines.

B. Use Dataflow jobs to read data from Pub/Sub, transform the data, and load the data to BigQuery.

C. Use Data Fusion to build and execute ETL pipelines.

D. Use Cloud Composer to load data and run SQL pipelines by using the BigQuery job operators.

You work for a farming company. You have one BigQuery table named sensors, which is about 500 MB and contains the list of your 5000 sensors, with columns for id, name, and location. This table is updated every hour. Each sensor generates one metric every 30 seconds along with a timestamp, which you want to store in BigQuery. You want to run an analytical query on the data

once a week for monitoring purposes. You also want to minimize costs. What data model should you use?

A.

1. Create a metrics column in the sensors table.
2. Set RECORD type and REPEATED mode for the metrics column.
3. Use an UPDATE statement every 30 seconds to add new metrics.

B.

1. Create a metrics column in the sensors table.
2. Set RECORD type and REPEATED mode for the metrics column.
3. Use an INSERT statement every 30 seconds to add new metrics.

C.

1. Create a metrics table partitioned by timestamp.
2. Create a sensorId column in the metrics table, that points to the id column in the sensors table.
3. Use an INSERT statement every 30 seconds to append new metrics to the metrics table.
4. Join the two tables, if needed, when running the analytical query.

D.

1. Create a metrics table partitioned by timestamp.
2. Create a sensorId column in the metrics table, which points to the id column in the sensors table.
3. Use an UPDATE statement every 30 seconds to append new metrics to the metrics table.
4. Join the two tables, if needed, when running the analytical query.

You are managing a Dataplex environment with raw and curated zones. A data engineering team is uploading JSON and CSV files to a bucket asset in the curated zone but the files are not being automatically discovered by Dataplex. What should you do to ensure that the files are discovered by Dataplex?

A. Move the JSON and CSV files to the raw zone.

B. Enable auto-discovery of files for the curated zone.

C. Use the bg command-line tool to load the JSON and CSV files into BigQuery tables.

D. Grant object level access to the CSV and JSON files in Cloud Storage.

You have a table that contains millions of rows of sales data, partitioned by date. Various applications and users query this data many times a minute. The query requires aggregating values by using AVG, MAX, and SUM, and does not require joining to other tables. The required aggregations are only computed over the past year of data, though you need to retain full

historical data in the base tables. You want to ensure that the query results always include the latest data from the tables, while also reducing computation cost, maintenance overhead, and duration. What should you do?

A. Create a materialized view to aggregate the base table data. Include a filter clause to specify the last one year of partitions.

B. Create a materialized view to aggregate the base table data. Configure a partition expiration on the base table to retain only the last one year of partitions.

C. Create a view to aggregate the base table data. Include a filter clause to specify the last year of partitions.

D. Create a new table that aggregates the base table data. Include a filter clause to specify the last year of partitions. Set up a scheduled query to recreate the new table every hour.

Your organization uses a multi-cloud data storage strategy, storing data in Cloud Storage, and data in Amazon Web Services' (AWS) S3 storage buckets. All data resides in US regions. You want to query up-to-date data by using BigQuery, regardless of which cloud the data is stored in. You need to allow users to query the tables from BigQuery without giving direct access to the data in the storage buckets. What should you do?

A. Setup a BigQuery Omni connection to the AWS S3 bucket data. Create BigLake tables over the Cloud Storage and S3 data and query the data using BigQuery directly.

B. Set up a BigQuery Omni connection to the AWS S3 bucket data. Create external tables over the Cloud Storage and S3 data and query the data using BigQuery directly.

C. Use the Storage Transfer Service to copy data from the AWS S3 buckets to Cloud Storage buckets. Create BigLake tables over the Cloud Storage data and query the data using BigQuery directly.

D. Use the Storage Transfer Service to copy data from the AWS S3 buckets to Cloud Storage buckets. Create external tables over the Cloud Storage data and query the data using BigQuery directly.

You are preparing an organization-wide dataset. You need to preprocess customer data stored in a restricted bucket in Cloud Storage. The data will be used to create consumer analyses. You need to comply with data privacy requirements. What should you do?

A. Use Dataflow and the Cloud Data Loss Prevention API to mask sensitive data. Write the processed data in BigQuery.

B. Use customer-managed encryption keys (CMEK) to directly encrypt the data in Cloud Storage. Use federated queries from BigQuery. Share the encryption key by following the principle of least privilege.

C. Use the Cloud Data Loss Prevention API and Dataflow to detect and remove sensitive fields from the data in Cloud Storage. Write the filtered data in BigQuery.

D. Use Dataflow and Cloud KMS to encrypt sensitive fields and write the encrypted data in BigQuery. Share the encryption key by following the principle of least privilege.

You need to connect multiple applications with dynamic public IP addresses to a Cloud SQL instance. You configured users with strong passwords and enforced the SSL connection to your Cloud SQL instance. You want to use Cloud SQL public IP and ensure that you have secured connections. What should you do?

A. Add CIDR 0.0.0.0/0 network to Authorized Network. Use Identity and Access Management (IAM) to add users.

B. Add all application networks to Authorized Network and regularly update them.

C. Leave the Authorized Network empty. Use Cloud SQL Auth proxy on all applications.

D. Add CIDR 0.0.0.0/0 network to Authorized Network. Use Cloud SQL Auth proxy on all applications.

You are migrating a large number of files from a public HTTPS endpoint to Cloud Storage. The files are protected from unauthorized access using signed URLs. You created a TSV file that contains the list of object URLs and started a transfer job by using Storage Transfer Service. You notice that the job has run for a long time and eventually failed. Checking the logs of the transfer job reveals that the job was running fine until one point and then it failed due to HTTP 403 errors on the remaining files. You verified that there were no changes to the source system. You need to fix the problem to resume the migration process. What should you do?

A. Set up Cloud Storage FUSE and mount the Cloud Storage bucket on a Compute Engine instance. Remove the completed files from the TSV file. Use a shell script to iterate through the TSV file and download the remaining URLs to the FUSE mount point.

B. Renew the TLS certificate of the HTTPS endpoint. Remove the completed files from the TSV file and rerun the Storage Transfer Service job.

C. Create a new TSV file for the remaining files by generating signed URLs with a longer validity period. Split the TSV file into multiple smaller files and submit them as separate Storage Transfer Service jobs in parallel.

D. Update the file checksums in the TSV file from using MD5 to SHA256. Remove the completed files from the TSV file and rerun the Storage Transfer Service job.

You work for an airline and you need to store weather data in a BigQuery table. Weather data will be used as input to a machine learning model. The model only uses the last 30 days of weather data. You want to avoid storing unnecessary data and minimize costs. What should you do?

A. Create a BigQuery table where each record has an ingestion timestamp. Run a scheduled query to delete all the rows with an ingestion timestamp older than 30 days.

B. Create a BigQuery table partitioned by datetime value of the weather date. Set up partition expiration to 30 days.

C. Create a BigQuery table partitioned by ingestion time. Set up partition expiration to 30 days.

D. Create a BigQuery table with a datetime column for the day the weather data refers to. Run a scheduled query to delete rows with a datetime value older than 30 days.

You need to look at BigQuery data from a specific table multiple times a day. The underlying table you are querying is several petabytes in size, but you want to filter your data and provide simple aggregations to downstream users. You want to run queries faster and get up-to-date insights quicker. What should you do?

A. Run a scheduled query to pull the necessary data at specific intervals dally.

B. Use a cached query to accelerate time to results.

C. Limit the query columns being pulled in the final result.

D. Create a materialized view based off of the query being run.

Your chemical company needs to manually check documentation for customer order. You use a pull subscription in Pub/Sub so that sales agents get details from the order. You must ensure that you do not process orders twice with different sales agents and that you do not add more complexity to this workflow. What should you do?

A. Use a Deduplicate PTransform in Dataflow before sending the messages to the sales agents.

B. Create a transactional database that monitors the pending messages.

C. Use Pub/Sub exactly-once delivery in your pull subscription.

D. Create a new Pub/Sub push subscription to monitor the orders processed in the agent's system.