

Topic 1 - Question Set 1

Question #1

Topic 1

Your company has decided to make a major revision of their API in order to create better experiences for their developers. They need to keep the old version of the API available and deployable, while allowing new customers and testers to try out the new API. They want to keep the same SSL and DNS records in place to serve both APIs.

What should they do?

- A. Configure a new load balancer for the new version of the API
- B. Reconfigure old clients to use a new endpoint for the new API
- C. Have the old API forward traffic to the new API based on the path
- D. Use separate backend pools for each API path behind the load balancer

Question #2

Topic 1

Your company plans to migrate a multi-petabyte data set to the cloud. The data set must be available 24hrs a day. Your business analysts have experience only with using a SQL interface.

How should you store the data to optimize it for ease of analysis?

- A. Load data into Google BigQuery
- B. Insert data into Google Cloud SQL
- C. Put flat files into Google Cloud Storage
- D. Stream data into Google Cloud Datastore

Question #3

Topic 1

The operations manager asks you for a list of recommended practices that she should consider when migrating a J2EE application to the cloud. Which three practices should you recommend? (Choose three.)

- A. Port the application code to run on Google App Engine
- B. Integrate Cloud Dataflow into the application to capture real-time metrics
- C. Instrument the application with a monitoring tool like Stackdriver Debugger
- D. Select an automation framework to reliably provision the cloud infrastructure
- E. Deploy a continuous integration tool with automated testing in a staging environment
- F. Migrate from MySQL to a managed NoSQL database like Google Cloud Datastore or Bigtable

A news feed web service has the following code running on Google App Engine. During peak load, users report that they can see news articles they already viewed.

What is the most likely cause of this problem?

```
import news
from flask import Flask, redirect, request
from flask.ext.api import status
from google.appengine.api import users

app = Flask(__name__)
sessions = {}

@app.route("/")
def homepage():
    user = users.get_current_user()
    if not user:
        return "Invalid login",
        status.HTTP_401_UNAUTHORIZED

    if user not in sessions:
        sessions[user] = {"viewed": []}

    news_articles = news.get_new_news (user, sessions [user]
["viewed"])
    sessions [user] ["viewed"] += [n["id"] for n
in news_articles]

    return news.render(news_articles)

if __name__ == "__main__":
    app.run()
```

- A. The session variable is local to just a single instance
- B. The session variable is being overwritten in Cloud Datastore
- C. The URL of the API needs to be modified to prevent caching
- D. The HTTP Expires header needs to be set to -1 stop caching

An application development team believes their current logging tool will not meet their needs for their new cloud-based product. They want a better tool to capture errors and help them analyze their historical log data. You want to help them find a solution that meets their needs.

What should you do?

- A. Direct them to download and install the Google StackDriver logging agent
- B. Send them a list of online resources about logging best practices
- C. Help them define their requirements and assess viable logging tools
- D. Help them upgrade their current tool to take advantage of any new features

You need to reduce the number of unplanned rollbacks of erroneous production deployments in your company's web hosting platform.

Improvement to the QA/

Test processes accomplished an 80% reduction.

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

- A. Introduce a green-blue deployment model
- B. Replace the QA environment with canary releases
- C. Fragment the monolithic platform into microservices
- D. Reduce the platform's dependency on relational database systems
- E. Replace the platform's relational database systems with a NoSQL database

To reduce costs, the Director of Engineering has required all developers to move their development infrastructure resources from on-premises virtual machines

(VMs) to Google Cloud Platform. These resources go through multiple start/stop events during the day and require state to persist. You have been asked to design the process of running a development environment in Google Cloud while providing cost visibility to the finance department.

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

- A. Use the `--no-auto-delete` flag on all persistent disks and stop the VM
- B. Use the `--auto-delete` flag on all persistent disks and terminate the VM
- C. Apply VM CPU utilization label and include it in the BigQuery billing export
- D. Use Google BigQuery billing export and labels to associate cost to groups
- E. Store all state into local SSD, snapshot the persistent disks, and terminate the VM
- F. Store all state in Google Cloud Storage, snapshot the persistent disks, and terminate the VM

Your company wants to track whether someone is present in a meeting room reserved for a scheduled meeting. There are 1000 meeting rooms across 5 offices on 3 continents. Each room is equipped with a motion sensor that reports its status every second. The data from the motion detector includes only a sensor ID and several different discrete items of information. Analysts will use this data, together with information about account owners and office locations.

Which database type should you use?

- A. Flat file
- B. NoSQL
- C. Relational
- D. Blobstore

You set up an autoscaling instance group to serve web traffic for an upcoming launch. After configuring the instance group as a backend service to an HTTP(S) load balancer, you notice that virtual machine (VM) instances are being terminated and re-launched every minute. The instances do not have a public IP address.

You have verified the appropriate web response is coming from each instance using the curl command. You want to ensure the backend is configured correctly.

What should you do?

- A. Ensure that a firewall rules exists to allow source traffic on HTTP/HTTPS to reach the load balancer.
- B. Assign a public IP to each instance and configure a firewall rule to allow the load balancer to reach the instance public IP.
- C. Ensure that a firewall rule exists to allow load balancer health checks to reach the instances in the instance group.
- D. Create a tag on each instance with the name of the load balancer. Configure a firewall rule with the name of the load balancer as the source and the instance tag as the destination.

You write a Python script to connect to Google BigQuery from a Google Compute Engine virtual machine. The script is printing errors that it cannot connect to

BigQuery.

What should you do to fix the script?

- A. Install the latest BigQuery API client library for Python
- B. Run your script on a new virtual machine with the BigQuery access scope enabled
- C. Create a new service account with BigQuery access and execute your script with that user
- D. Install the bq component for gcloud with the command `gcloud components install bq`.

Your customer is moving an existing corporate application to Google Cloud Platform from an on-premises data center. The business owners require minimal user disruption. There are strict security team requirements for storing passwords.

What authentication strategy should they use?

- A. Use G Suite Password Sync to replicate passwords into Google
- B. Federate authentication via SAML 2.0 to the existing Identity Provider
- C. Provision users in Google using the Google Cloud Directory Sync tool
- D. Ask users to set their Google password to match their corporate password

Your company has successfully migrated to the cloud and wants to analyze their data stream to optimize operations. They do not have any existing code for this analysis, so they are exploring all their options. These options include a mix of batch and stream processing, as they are running some hourly jobs and live- processing some data as it comes in.

Which technology should they use for this?

- A. Google Cloud Dataproc
- B. Google Cloud Dataflow
- C. Google Container Engine with Bigtable
- D. Google Compute Engine with Google BigQuery

Your customer is receiving reports that their recently updated Google App Engine application is taking approximately 30 seconds to load for some of their users.

This behavior was not reported before the update.

What strategy should you take?

- A. Work with your ISP to diagnose the problem
- B. Open a support ticket to ask for network capture and flow data to diagnose the problem, then roll back your application
- C. Roll back to an earlier known good release initially, then use Stackdriver Trace and Logging to diagnose the problem in a development/test/staging environment
- D. Roll back to an earlier known good release, then push the release again at a quieter period to investigate. Then use Stackdriver Trace and Logging to diagnose the problem

A production database virtual machine on Google Compute Engine has an ext4-formatted persistent disk for data files. The database is about to run out of storage space.

How can you remediate the problem with the least amount of downtime?

- A. In the Cloud Platform Console, increase the size of the persistent disk and use the `resize2fs` command in Linux.
- B. Shut down the virtual machine, use the Cloud Platform Console to increase the persistent disk size, then restart the virtual machine
- C. In the Cloud Platform Console, increase the size of the persistent disk and verify the new space is ready to use with the `fdisk` command in Linux
- D. In the Cloud Platform Console, create a new persistent disk attached to the virtual machine, format and mount it, and configure the database service to move the files to the new disk
- E. In the Cloud Platform Console, create a snapshot of the persistent disk restore the snapshot to a new larger disk, unmount the old disk, mount the new disk and restart the database service

Your application needs to process credit card transactions. You want the smallest scope of Payment Card Industry (PCI) compliance without compromising the ability to analyze transactional data and trends relating to which payment methods are used.

How should you design your architecture?

- A. Create a tokenizer service and store only tokenized data
- B. Create separate projects that only process credit card data
- C. Create separate subnetworks and isolate the components that process credit card data
- D. Streamline the audit discovery phase by labeling all of the virtual machines (VMs) that process PCI data
- E. Enable Logging export to Google BigQuery and use ACLs and views to scope the data shared with the auditor

You have been asked to select the storage system for the click-data of your company's large portfolio of websites. This data is streamed in from a custom website analytics package at a typical rate of 6,000 clicks per minute. With bursts of up to 8,500 clicks per second. It must have been stored for future analysis by your data science and user experience teams.

Which storage infrastructure should you choose?

- A. Google Cloud SQL
- B. Google Cloud Bigtable
- C. Google Cloud Storage
- D. Google Cloud Datastore

You are creating a solution to remove backup files older than 90 days from your backup Cloud Storage bucket. You want to optimize ongoing Cloud Storage spend.

What should you do?

- A. Write a lifecycle management rule in XML and push it to the bucket with gsutil
- B. Write a lifecycle management rule in JSON and push it to the bucket with gsutil
- C. Schedule a cron script using `gsutil ls -l gs://backups/**` to find and remove items older than 90 days
- D. Schedule a cron script using `gsutil ls -l gs://backups/**` to find and remove items older than 90 days and schedule it with cron

Your company is forecasting a sharp increase in the number and size of Apache Spark and Hadoop jobs being run on your local datacenter. You want to utilize the cloud to help you scale this upcoming demand with the least amount of operations work and code change.

Which product should you use?

- A. Google Cloud Dataflow
- B. Google Cloud Dataproc
- C. Google Compute Engine
- D. Google Kubernetes Engine

The database administration team has asked you to help them improve the performance of their new database server running on Google Compute Engine. The database is for importing and normalizing their performance statistics and is built with MySQL running on Debian Linux. They have an n1-standard-8 virtual machine with 80 GB of SSD persistent disk.

What should they change to get better performance from this system?

- A. Increase the virtual machine's memory to 64 GB
- B. Create a new virtual machine running PostgreSQL
- C. Dynamically resize the SSD persistent disk to 500 GB
- D. Migrate their performance metrics warehouse to BigQuery
- E. Modify all of their batch jobs to use bulk inserts into the database

You want to optimize the performance of an accurate, real-time, weather-charting application. The data comes from 50,000 sensors sending 10 readings a second, in the format of a timestamp and sensor reading.

Where should you store the data?

- A. Google BigQuery
- B. Google Cloud SQL
- C. Google Cloud Bigtable
- D. Google Cloud Storage

Your company's user-feedback portal comprises a standard LAMP stack replicated across two zones. It is deployed in the us-central1 region and uses autoscaled managed instance groups on all layers, except the database. Currently, only a small group of select customers have access to the portal. The portal meets a

99,99% availability SLA under these conditions. However next quarter, your company will be making the portal available to all users, including unauthenticated users. You need to develop a resiliency testing strategy to ensure the system maintains the SLA once they introduce additional user load.

What should you do?

- A. Capture existing users input, and replay captured user load until autoscale is triggered on all layers. At the same time, terminate all resources in one of the zones
- B. Create synthetic random user input, replay synthetic load until autoscale logic is triggered on at least one layer, and introduce "chaos" to the system by terminating random resources on both zones
- C. Expose the new system to a larger group of users, and increase group size each day until autoscale logic is triggered on all layers. At the same time, terminate random resources on both zones
- D. Capture existing users input, and replay captured user load until resource utilization crosses 80%. Also, derive estimated number of users based on existing user's usage of the app, and deploy enough resources to handle 200% of expected load

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

```
FROM ubuntu:16.04

COPY . /src

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

RUN pip install -r requirements.txt
```

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

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

- A. Remove Python after running pip
- B. Remove dependencies from requirements.txt
- C. Use a slimmed-down base image like Alpine Linux
- D. Use larger machine types for your Google Container Engine node pools
- E. Copy the source after the package dependencies (Python and pip) are installed

Your solution is producing performance bugs in production that you did not see in staging and test environments. You want to adjust your test and deployment procedures to avoid this problem in the future.

What should you do?

- A. Deploy fewer changes to production
- B. Deploy smaller changes to production
- C. Increase the load on your test and staging environments
- D. Deploy changes to a small subset of users before rolling out to production

A small number of API requests to your microservices-based application take a very long time. You know that each request to the API can traverse many services.

You want to know which service takes the longest in those cases.

What should you do?

- A. Set timeouts on your application so that you can fail requests faster
- B. Send custom metrics for each of your requests to Stackdriver Monitoring
- C. Use Stackdriver Monitoring to look for insights that show when your API latencies are high
- D. Instrument your application with Stackdriver Trace in order to break down the request latencies at each microservice

During a high traffic portion of the day, one of your relational databases crashes, but the replica is never promoted to a master. You want to avoid this in the future.

What should you do?

- A. Use a different database
- B. Choose larger instances for your database
- C. Create snapshots of your database more regularly
- D. Implement routinely scheduled failovers of your databases

Your organization requires that metrics from all applications be retained for 5 years for future analysis in possible legal proceedings.

Which approach should you use?

- A. Grant the security team access to the logs in each Project
- B. Configure Stackdriver Monitoring for all Projects, and export to BigQuery
- C. Configure Stackdriver Monitoring for all Projects with the default retention policies
- D. Configure Stackdriver Monitoring for all Projects, and export to Google Cloud Storage

Your company has decided to build a backup replica of their on-premises user authentication PostgreSQL database on Google Cloud Platform.

The database is 4

TB, and large updates are frequent. Replication requires private address space communication.

Which networking approach should you use?

- A. Google Cloud Dedicated Interconnect
- B. Google Cloud VPN connected to the data center network
- C. A NAT and TLS translation gateway installed on-premises
- D. A Google Compute Engine instance with a VPN server installed connected to the data center network

Auditors visit your teams every 12 months and ask to review all the Google Cloud Identity and Access Management (Cloud IAM) policy changes in the previous 12 months. You want to streamline and expedite the analysis and audit process.

What should you do?

- A. Create custom Google Stackdriver alerts and send them to the auditor
- B. Enable Logging export to Google BigQuery and use ACLs and views to scope the data shared with the auditor
- C. Use cloud functions to transfer log entries to Google Cloud SQL and use ACLs and views to limit an auditor's view
- D. Enable Google Cloud Storage (GCS) log export to audit logs into a GCS bucket and delegate access to the bucket

You are designing a large distributed application with 30 microservices. Each of your distributed microservices needs to connect to a database back-end. You want to store the credentials securely.

Where should you store the credentials?

- A. In the source code
- B. In an environment variable
- C. In a secret management system
- D. In a config file that has restricted access through ACLs

A lead engineer wrote a custom tool that deploys virtual machines in the legacy data center. He wants to migrate the custom tool to the new cloud environment.

You want to advocate for the adoption of Google Cloud Deployment Manager.

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

- A. Cloud Deployment Manager uses Python
- B. Cloud Deployment Manager APIs could be deprecated in the future
- C. Cloud Deployment Manager is unfamiliar to the company's engineers
- D. Cloud Deployment Manager requires a Google APIs service account to run
- E. Cloud Deployment Manager can be used to permanently delete cloud resources
- F. Cloud Deployment Manager only supports automation of Google Cloud resources

A development manager is building a new application. He asks you to review his requirements and identify what cloud technologies he can use to meet them. The application must:

1. Be based on open-source technology for cloud portability
2. Dynamically scale compute capacity based on demand
3. Support continuous software delivery
4. Run multiple segregated copies of the same application stack
5. Deploy application bundles using dynamic templates
6. Route network traffic to specific services based on URL

Which combination of technologies will meet all of his requirements?

- A. Google Kubernetes Engine, Jenkins, and Helm
- B. Google Kubernetes Engine and Cloud Load Balancing
- C. Google Kubernetes Engine and Cloud Deployment Manager
- D. Google Kubernetes Engine, Jenkins, and Cloud Load Balancing

You have created several pre-emptible Linux virtual machine instances using Google Compute Engine. You want to properly shut down your application before the virtual machines are preempted.

What should you do?

- A. Create a shutdown script named k99.shutdown in the /etc/rc.6.d/ directory
- B. Create a shutdown script registered as a xinetd service in Linux and configure a Stackdriver endpoint check to call the service
- C. Create a shutdown script and use it as the value for a new metadata entry with the key shutdown-script in the Cloud Platform Console when you create the new virtual machine instance
- D. Create a shutdown script, registered as a xinetd service in Linux, and use the gcloud compute instances add-metadata command to specify the service URL as the value for a new metadata entry with the key shutdown-script-url

Your organization has a 3-tier web application deployed in the same network on Google Cloud Platform. Each tier (web, API, and database) scales independently of the others. Network traffic should flow through the web to the API tier and then on to the database tier. Traffic should not flow between the web and the database tier.

How should you configure the network?

- A. Add each tier to a different subnetwork
- B. Set up software based firewalls on individual VMs
- C. Add tags to each tier and set up routes to allow the desired traffic flow
- D. Add tags to each tier and set up firewall rules to allow the desired traffic flow

Your development team has installed a new Linux kernel module on the batch servers in Google Compute Engine (GCE) virtual machines (VMs) to speed up the nightly batch process. Two days after the installation, 50% of the batch servers failed the nightly batch run. You want to collect details on the failure to pass back to the development team.

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

- A. Use Stackdriver Logging to search for the module log entries
- B. Read the debug GCE Activity log using the API or Cloud Console
- C. Use gcloud or Cloud Console to connect to the serial console and observe the logs
- D. Identify whether a live migration event of the failed server occurred, using in the activity log
- E. Adjust the Google Stackdriver timeline to match the failure time, and observe the batch server metrics
- F. Export a debug VM into an image, and run the image on a local server where kernel log messages will be displayed on the native screen

Your company wants to try out the cloud with low risk. They want to archive approximately 100 TB of their log data to the cloud and test the analytics features available to them there, while also retaining that data as a long-term disaster recovery backup.

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

- A. Load logs into Google BigQuery
- B. Load logs into Google Cloud SQL
- C. Import logs into Google Stackdriver
- D. Insert logs into Google Cloud Bigtable
- E. Upload log files into Google Cloud Storage

You created a pipeline that can deploy your source code changes to your infrastructure in instance groups for self-healing. One of the changes negatively affects your key performance indicator. You are not sure how to fix it, and investigation could take up to a week.

What should you do?

- A. Log in to a server, and iterate on the fix locally
- B. Revert the source code change, and rerun the deployment pipeline
- C. Log into the servers with the bad code change, and swap in the previous code
- D. Change the instance group template to the previous one, and delete all instances

Your organization wants to control IAM policies for different departments independently, but centrally.

Which approach should you take?

- A. Multiple Organizations with multiple Folders
- B. Multiple Organizations, one for each department
- C. A single Organization with Folders for each department
- D. A single Organization with multiple projects, each with a central owner

You deploy your custom Java application to Google App Engine. It fails to deploy and gives you the following stack trace. What should you do?

```
java.lang.SecurityException: SHA1 digest error for
com/Altostrat/CloakedServlet.class
    at com.google.appengine.runtime.Request.process
-d36f818a24b8cf1d (Request.java)
    at
sun.security.util.ManifestEntryVerifier.verify
(ManifestEntryVerifier.java:210)
    at java.util.jar.JarVerifier.processEntry
(JarVerifier.java:218)
    at java.util.jar.JarVerifier.update
(JarVerifier.java:205)
    at
java.util.jar.JarVerifiersVerifierStream.read
(JarVerifier.java:428)
    at sun.misc.Resource.getBytes
(Resource.java:124)
    at java.net.URL.ClassLoader.defineClass
(URLClassLoader.java:273)
    at sun.reflect.GeneratedMethodAccessor5.invoke
(Unknown Source)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke
(Method.java:616)
    at java.lang.ClassLoader.loadClass
(ClassLoader.java:266)
```

- A. Upload missing JAR files and redeploy your application.
- B. Digitally sign all of your JAR files and redeploy your application
- C. Recompile the CLoakedServlet class using and MD5 hash instead of SHA1

You are designing a mobile chat application. You want to ensure people cannot spoof chat messages, by providing a message were sent by a specific user.

What should you do?

- A. Tag messages client side with the originating user identifier and the destination user.
- B. Encrypt the message client side using block-based encryption with a shared key.
- C. Use public key infrastructure (PKI) to encrypt the message client side using the originating user's private key.
- D. Use a trusted certificate authority to enable SSL connectivity between the client application and the server.

As part of implementing their disaster recovery plan, your company is trying to replicate their production MySQL database from their private data center to their GCP project using a Google Cloud VPN connection. They are experiencing latency issues and a small amount of packet loss that is disrupting the replication. What should they do?

- A. Configure their replication to use UDP.
- B. Configure a Google Cloud Dedicated Interconnect.
- C. Restore their database daily using Google Cloud SQL.
- D. Add additional VPN connections and load balance them.
- E. Send the replicated transaction to Google Cloud Pub/Sub.

Question #41

Topic 1

Your customer support tool logs all email and chat conversations to Cloud Bigtable for retention and analysis. What is the recommended approach for sanitizing this data of personally identifiable information or payment card information before initial storage?

- A. Hash all data using SHA256
- B. Encrypt all data using elliptic curve cryptography
- C. De-identify the data with the Cloud Data Loss Prevention API
- D. Use regular expressions to find and redact phone numbers, email addresses, and credit card numbers

Question #42

Topic 1

You are using Cloud Shell and need to install a custom utility for use in a few weeks. Where can you store the file so it is in the default execution path and persists across sessions?

- A. ~/bin
- B. Cloud Storage
- C. /google/scripts
- D. /usr/local/bin

Question #43

Topic 1

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

- A. Create a VPC and connect it to your on-premises data center using Dedicated Interconnect.
- B. Create a VPC and connect it to your on-premises data center using a single Cloud VPN.
- C. Create a Cloud Content Delivery Network (Cloud CDN) and connect it to your on-premises data center using Dedicated Interconnect.
- D. Create a Cloud Content Delivery Network (Cloud CDN) and connect it to your on-premises datacenter using a single Cloud VPN.

You are analyzing and defining business processes to support your startup's trial usage of GCP, and you don't yet know what consumer demand for your product will be. Your manager requires you to minimize GCP service costs and adhere to Google best practices. What should you do?

- A. Utilize free tier and sustained use discounts. Provision a staff position for service cost management.
- B. Utilize free tier and sustained use discounts. Provide training to the team about service cost management.
- C. Utilize free tier and committed use discounts. Provision a staff position for service cost management.
- D. Utilize free tier and committed use discounts. Provide training to the team about service cost management.

You are building a continuous deployment pipeline for a project stored in a Git source repository and want to ensure that code changes can be verified before deploying to production. What should you do?

- A. Use Spinnaker to deploy builds to production using the red/black deployment strategy so that changes can easily be rolled back.
- B. Use Spinnaker to deploy builds to production and run tests on production deployments.
- C. Use Jenkins to build the staging branches and the master branch. Build and deploy changes to production for 10% of users before doing a complete rollout.
- D. Use Jenkins to monitor tags in the repository. Deploy staging tags to a staging environment for testing. After testing, tag the repository for production and deploy that to the production environment.

You have an outage in your Compute Engine managed instance group: all instances keep restarting after 5 seconds. You have a health check configured, but autoscaling is disabled. Your colleague, who is a Linux expert, offered to look into the issue. You need to make sure that he can access the VMs. What should you do?

- A. Grant your colleague the IAM role of project Viewer
- B. Perform a rolling restart on the instance group
- C. Disable the health check for the instance group. Add his SSH key to the project-wide SSH Keys
- D. Disable autoscaling for the instance group. Add his SSH key to the project-wide SSH Keys

Your company is migrating its on-premises data center into the cloud. As part of the migration, you want to integrate Google Kubernetes Engine (GKE) for workload orchestration. Parts of your architecture must also be PCI DSS-compliant. Which of the following is most accurate?

- A. App Engine is the only compute platform on GCP that is certified for PCI DSS hosting.
- B. GKE cannot be used under PCI DSS because it is considered shared hosting.
- C. GKE and GCP provide the tools you need to build a PCI DSS-compliant environment.
- D. All Google Cloud services are usable because Google Cloud Platform is certified PCI-compliant.

Your company has multiple on-premises systems that serve as sources for reporting. The data has not been maintained well and has become degraded over time.

You want to use Google-recommended practices to detect anomalies in your company data. What should you do?

- A. Upload your files into Cloud Storage. Use Cloud Datalab to explore and clean your data.
- B. Upload your files into Cloud Storage. Use Cloud Dataprep to explore and clean your data.
- C. Connect Cloud Datalab to your on-premises systems. Use Cloud Datalab to explore and clean your data.
- D. Connect Cloud Dataprep to your on-premises systems. Use Cloud Dataprep to explore and clean your data.

Google Cloud Platform resources are managed hierarchically using organization, folders, and projects. When Cloud Identity and Access Management (IAM) policies exist at these different levels, what is the effective policy at a particular node of the hierarchy?

- A. The effective policy is determined only by the policy set at the node
- B. The effective policy is the policy set at the node and restricted by the policies of its ancestors
- C. The effective policy is the union of the policy set at the node and policies inherited from its ancestors
- D. The effective policy is the intersection of the policy set at the node and policies inherited from its ancestors

You are migrating your on-premises solution to Google Cloud in several phases. You will use Cloud VPN to maintain a connection between your on-premises systems and Google Cloud until the migration is completed. You want to make sure all your on-premise systems remain reachable during this period. How should you organize your networking in Google Cloud?

- A. Use the same IP range on Google Cloud as you use on-premises
- B. Use the same IP range on Google Cloud as you use on-premises for your primary IP range and use a secondary range that does not overlap with the range you use on-premises
- C. Use an IP range on Google Cloud that does not overlap with the range you use on-premises
- D. Use an IP range on Google Cloud that does not overlap with the range you use on-premises for your primary IP range and use a secondary range with the same IP range as you use on-premises

You have found an error in your App Engine application caused by missing Cloud Datastore indexes. You have created a YAML file with the required indexes and want to deploy these new indexes to Cloud Datastore. What should you do?

- A. Point gcloud datastore create-indexes to your configuration file
- B. Upload the configuration file to App Engine's default Cloud Storage bucket, and have App Engine detect the new indexes
- C. In the GCP Console, use Datastore Admin to delete the current indexes and upload the new configuration file
- D. Create an HTTP request to the built-in python module to send the index configuration file to your application

You have an application that will run on Compute Engine. You need to design an architecture that takes into account a disaster recovery plan that requires your application to fail over to another region in case of a regional outage. What should you do?

- A. Deploy the application on two Compute Engine instances in the same project but in a different region. Use the first instance to serve traffic, and use the HTTP load balancing service to fail over to the standby instance in case of a disaster.
- B. Deploy the application on a Compute Engine instance. Use the instance to serve traffic, and use the HTTP load balancing service to fail over to an instance on your premises in case of a disaster.
- C. Deploy the application on two Compute Engine instance groups, each in the same project but in a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance group in case of a disaster.
- D. Deploy the application on two Compute Engine instance groups, each in a separate project and a different region. Use the first instance group to serve traffic, and use the HTTP load balancing service to fail over to the standby instance group in case of a disaster.

You are deploying an application on App Engine that needs to integrate with an on-premises database. For security purposes, your on-premises database must not be accessible through the public internet. What should you do?

- A. Deploy your application on App Engine standard environment and use App Engine firewall rules to limit access to the open on-premises database.
- B. Deploy your application on App Engine standard environment and use Cloud VPN to limit access to the on-premises database.
- C. Deploy your application on App Engine flexible environment and use App Engine firewall rules to limit access to the on-premises database.
- D. Deploy your application on App Engine flexible environment and use Cloud VPN to limit access to the on-premises database.

You are working in a highly secured environment where public Internet access from the Compute Engine VMs is not allowed. You do not yet have a VPN connection to access an on-premises file server. You need to install specific software on a Compute Engine instance. How should you install the software?

- A. Upload the required installation files to Cloud Storage. Configure the VM on a subnet with a Private Google Access subnet. Assign only an internal IP address to the VM. Download the installation files to the VM using gsutil.
- B. Upload the required installation files to Cloud Storage and use firewall rules to block all traffic except the IP address range for Cloud Storage. Download the files to the VM using gsutil.
- C. Upload the required installation files to Cloud Source Repositories. Configure the VM on a subnet with a Private Google Access subnet. Assign only an internal IP address to the VM. Download the installation files to the VM using gcloud.
- D. Upload the required installation files to Cloud Source Repositories and use firewall rules to block all traffic except the IP address range for Cloud Source Repositories. Download the files to the VM using gsutil.

Your company is moving 75 TB of data into Google Cloud. You want to use Cloud Storage and follow Google-recommended practices. What should you do?

- A. Move your data onto a Transfer Appliance. Use a Transfer Appliance Rehydrator to decrypt the data into Cloud Storage.
- B. Move your data onto a Transfer Appliance. Use Cloud Dataprep to decrypt the data into Cloud Storage.
- C. Install gsutil on each server that contains data. Use resumable transfers to upload the data into Cloud Storage.
- D. Install gsutil on each server containing data. Use streaming transfers to upload the data into Cloud Storage.

You have an application deployed on Google Kubernetes Engine using a Deployment named echo-deployment. The deployment is exposed using a Service called echo-service. You need to perform an update to the application with minimal downtime to the application. What should you do?

- A. Use `kubectl set image deployment/echo-deployment <new-image>`
- B. Use the rolling update functionality of the Instance Group behind the Kubernetes cluster
- C. Update the deployment yaml file with the new container image. Use `kubectl delete deployment/echo-deployment` and `kubectl create -f <yaml-file>`
- D. Update the service yaml file with the new container image. Use `kubectl delete service/echo-service` and `kubectl create -f <yaml-file>`

Your company is using BigQuery as its enterprise data warehouse. Data is distributed over several Google Cloud projects. All queries on BigQuery need to be billed on a single project. You want to make sure that no query costs are incurred on the projects that contain the data. Users should be able to query the datasets, but not edit them.

How should you configure users' access roles?

- A. Add all users to a group. Grant the group the role of BigQuery user on the billing project and BigQuery dataViewer on the projects that contain the data.
- B. Add all users to a group. Grant the group the roles of BigQuery dataViewer on the billing project and BigQuery user on the projects that contain the data.
- C. Add all users to a group. Grant the group the roles of BigQuery jobUser on the billing project and BigQuery dataViewer on the projects that contain the data.
- D. Add all users to a group. Grant the group the roles of BigQuery dataViewer on the billing project and BigQuery jobUser on the projects that contain the data.

You have developed an application using Cloud ML Engine that recognizes famous paintings from uploaded images. You want to test the application and allow specific people to upload images for the next 24 hours. Not all users have a Google Account. How should you have users upload images?

- A. Have users upload the images to Cloud Storage. Protect the bucket with a password that expires after 24 hours.
- B. Have users upload the images to Cloud Storage using a signed URL that expires after 24 hours.
- C. Create an App Engine web application where users can upload images. Configure App Engine to disable the application after 24 hours. Authenticate users via Cloud Identity.
- D. Create an App Engine web application where users can upload images for the next 24 hours. Authenticate users via Cloud Identity.

Your web application must comply with the requirements of the European Union's General Data Protection Regulation (GDPR). You are responsible for the technical architecture of your web application. What should you do?

- A. Ensure that your web application only uses native features and services of Google Cloud Platform, because Google already has various certifications and provides “pass-on” compliance when you use native features.
- B. Enable the relevant GDPR compliance setting within the GCP Console for each of the services in use within your application.
- C. Ensure that Cloud Security Scanner is part of your test planning strategy in order to pick up any compliance gaps.
- D. Define a design for the security of data in your web application that meets GDPR requirements.

You need to set up Microsoft SQL Server on GCP. Management requires that there's no downtime in case of a data center outage in any of the zones within a GCP region. What should you do?

- A. Configure a Cloud SQL instance with high availability enabled.
- B. Configure a Cloud Spanner instance with a regional instance configuration.
- C. Set up SQL Server on Compute Engine, using Always On Availability Groups using Windows Failover Clustering. Place nodes in different subnets.
- D. Set up SQL Server Always On Availability Groups using Windows Failover Clustering. Place nodes in different zones.

The development team has provided you with a Kubernetes Deployment file. You have no infrastructure yet and need to deploy the application. What should you do?

- A. Use gcloud to create a Kubernetes cluster. Use Deployment Manager to create the deployment.
- B. Use gcloud to create a Kubernetes cluster. Use kubectl to create the deployment.
- C. Use kubectl to create a Kubernetes cluster. Use Deployment Manager to create the deployment.
- D. Use kubectl to create a Kubernetes cluster. Use kubectl to create the deployment.

You need to evaluate your team readiness for a new GCP project. You must perform the evaluation and create a skills gap plan which incorporates the business goal of cost optimization. Your team has deployed two GCP projects successfully to date. What should you do?

- A. Allocate budget for team training. Set a deadline for the new GCP project.
- B. Allocate budget for team training. Create a roadmap for your team to achieve Google Cloud certification based on job role.
- C. Allocate budget to hire skilled external consultants. Set a deadline for the new GCP project.
- D. Allocate budget to hire skilled external consultants. Create a roadmap for your team to achieve Google Cloud certification based on job role.

You are designing an application for use only during business hours. For the minimum viable product release, you'd like to use a managed product that automatically `scales to zero` so you don't incur costs when there is no activity.

Which primary compute resource should you choose?

- A. Cloud Functions
- B. Compute Engine
- C. Google Kubernetes Engine
- D. AppEngine flexible environment

You are creating an App Engine application that uses Cloud Datastore as its persistence layer. You need to retrieve several root entities for which you have the identifiers. You want to minimize the overhead in operations performed by Cloud Datastore. What should you do?

- A. Create the Key object for each Entity and run a batch get operation
- B. Create the Key object for each Entity and run multiple get operations, one operation for each entity
- C. Use the identifiers to create a query filter and run a batch query operation
- D. Use the identifiers to create a query filter and run multiple query operations, one operation for each entity

You need to upload files from your on-premises environment to Cloud Storage. You want the files to be encrypted on Cloud Storage using customer-supplied encryption keys. What should you do?

- A. Supply the encryption key in a .boto configuration file. Use gsutil to upload the files.
- B. Supply the encryption key using gcloud config. Use gsutil to upload the files to that bucket.
- C. Use gsutil to upload the files, and use the flag --encryption-key to supply the encryption key.
- D. Use gsutil to create a bucket, and use the flag --encryption-key to supply the encryption key. Use gsutil to upload the files to that bucket.

Your customer wants to capture multiple GBs of aggregate real-time key performance indicators (KPIs) from their game servers running on Google Cloud Platform and monitor the KPIs with low latency. How should they capture the KPIs?

- A. Store time-series data from the game servers in Google Bigtable, and view it using Google Data Studio.
- B. Output custom metrics to Stackdriver from the game servers, and create a Dashboard in Stackdriver Monitoring Console to view them.
- C. Schedule BigQuery load jobs to ingest analytics files uploaded to Cloud Storage every ten minutes, and visualize the results in Google Data Studio.
- D. Insert the KPIs into Cloud Datastore entities, and run ad hoc analysis and visualizations of them in Cloud Datalab.

You have a Python web application with many dependencies that requires 0.1 CPU cores and 128 MB of memory to operate in production. You want to monitor and maximize machine utilization. You also want to reliably deploy new versions of the application. Which set of steps should you take?

- A. Perform the following: 1. Create a managed instance group with f1-micro type machines. 2. Use a startup script to clone the repository, check out the production branch, install the dependencies, and start the Python app. 3. Restart the instances to automatically deploy new production releases.
- B. Perform the following: 1. Create a managed instance group with n1-standard-1 type machines. 2. Build a Compute Engine image from the production branch that contains all of the dependencies and automatically starts the Python app. 3. Rebuild the Compute Engine image, and update the instance template to deploy new production releases.
- C. Perform the following: 1. Create a Google Kubernetes Engine (GKE) cluster with n1-standard-1 type machines. 2. Build a Docker image from the production branch with all of the dependencies, and tag it with the version number. 3. Create a Kubernetes Deployment with the imagePullPolicy set to 'IfNotPresent' in the staging namespace, and then promote it to the production namespace after testing.
- D. Perform the following: 1. Create a GKE cluster with n1-standard-4 type machines. 2. Build a Docker image from the master branch with all of the dependencies, and tag it with 'latest'. 3. Create a Kubernetes Deployment in the default namespace with the imagePullPolicy set to 'Always'. Restart the pods to automatically deploy new production releases.

Your company wants to start using Google Cloud resources but wants to retain their on-premises Active Directory domain controller for identity management.

What should you do?

- A. Use the Admin Directory API to authenticate against the Active Directory domain controller.
- B. Use Google Cloud Directory Sync to synchronize Active Directory usernames with cloud identities and configure SAML SSO.
- C. Use Cloud Identity-Aware Proxy configured to use the on-premises Active Directory domain controller as an identity provider.
- D. Use Compute Engine to create an Active Directory (AD) domain controller that is a replica of the on-premises AD domain controller using Google Cloud Directory Sync.

You are running a cluster on Kubernetes Engine (GKE) to serve a web application. Users are reporting that a specific part of the application is not responding anymore. You notice that all pods of your deployment keep restarting after 2 seconds. The application writes logs to standard output. You want to inspect the logs to find the cause of the issue. Which approach can you take?

- A. Review the Stackdriver logs for each Compute Engine instance that is serving as a node in the cluster.
- B. Review the Stackdriver logs for the specific GKE container that is serving the unresponsive part of the application.
- C. Connect to the cluster using gcloud credentials and connect to a container in one of the pods to read the logs.
- D. Review the Serial Port logs for each Compute Engine instance that is serving as a node in the cluster.

You are using a single Cloud SQL instance to serve your application from a specific zone. You want to introduce high availability. What should you do?

- A. Create a read replica instance in a different region
- B. Create a failover replica instance in a different region
- C. Create a read replica instance in the same region, but in a different zone
- D. Create a failover replica instance in the same region, but in a different zone

Your company is running a stateless application on a Compute Engine instance. The application is used heavily during regular business hours and lightly outside of business hours. Users are reporting that the application is slow during peak hours. You need to optimize the application's performance. What should you do?

- A. Create a snapshot of the existing disk. Create an instance template from the snapshot. Create an autoscaled managed instance group from the instance template.
- B. Create a snapshot of the existing disk. Create a custom image from the snapshot. Create an autoscaled managed instance group from the custom image.
- C. Create a custom image from the existing disk. Create an instance template from the custom image. Create an autoscaled managed instance group from the instance template.
- D. Create an instance template from the existing disk. Create a custom image from the instance template. Create an autoscaled managed instance group from the custom image.

Your web application has several VM instances running within a VPC. You want to restrict communications between instances to only the paths and ports you authorize, but you don't want to rely on static IP addresses or subnets because the app can autoscale. How should you restrict communications?

- A. Use separate VPCs to restrict traffic
- B. Use firewall rules based on network tags attached to the compute instances
- C. Use Cloud DNS and only allow connections from authorized hostnames
- D. Use service accounts and configure the web application to authorize particular service accounts to have access

You are using Cloud SQL as the database backend for a large CRM deployment. You want to scale as usage increases and ensure that you don't run out of storage, maintain 75% CPU usage cores, and keep replication lag below 60 seconds. What are the correct steps to meet your requirements?

- A. 1. Enable automatic storage increase for the instance. 2. Create a Stackdriver alert when CPU usage exceeds 75%, and change the instance type to reduce CPU usage. 3. Create a Stackdriver alert for replication lag, and shard the database to reduce replication time.
- B. 1. Enable automatic storage increase for the instance. 2. Change the instance type to a 32-core machine type to keep CPU usage below 75%. 3. Create a Stackdriver alert for replication lag, and deploy memcache to reduce load on the master.
- C. 1. Create a Stackdriver alert when storage exceeds 75%, and increase the available storage on the instance to create more space. 2. Deploy memcached to reduce CPU load. 3. Change the instance type to a 32-core machine type to reduce replication lag.
- D. 1. Create a Stackdriver alert when storage exceeds 75%, and increase the available storage on the instance to create more space. 2. Deploy memcached to reduce CPU load. 3. Create a Stackdriver alert for replication lag, and change the instance type to a 32-core machine type to reduce replication lag.

You are tasked with building an online analytical processing (OLAP) marketing analytics and reporting tool. This requires a relational database that can operate on hundreds of terabytes of data. What is the Google-recommended tool for such applications?

- A. Cloud Spanner, because it is globally distributed
- B. Cloud SQL, because it is a fully managed relational database
- C. Cloud Firestore, because it offers real-time synchronization across devices
- D. BigQuery, because it is designed for large-scale processing of tabular data

You have deployed an application to Google Kubernetes Engine (GKE), and are using the Cloud SQL proxy container to make the Cloud SQL database available to the services running on Kubernetes. You are notified that the application is reporting database connection issues. Your company policies require a post-mortem. What should you do?

- A. Use `gcloud sql instances restart`.
- B. Validate that the Service Account used by the Cloud SQL proxy container still has the Cloud Build Editor role.
- C. In the GCP Console, navigate to Stackdriver Logging. Consult logs for (GKE) and Cloud SQL.
- D. In the GCP Console, navigate to Cloud SQL. Restore the latest backup. Use `kubectl` to restart all pods.

Your company pushes batches of sensitive transaction data from its application server VMs to Cloud Pub/Sub for processing and storage. What is the Google-recommended way for your application to authenticate to the required Google Cloud services?

- A. Ensure that VM service accounts are granted the appropriate Cloud Pub/Sub IAM roles.
- B. Ensure that VM service accounts do not have access to Cloud Pub/Sub, and use VM access scopes to grant the appropriate Cloud Pub/Sub IAM roles.
- C. Generate an OAuth2 access token for accessing Cloud Pub/Sub, encrypt it, and store it in Cloud Storage for access from each VM.
- D. Create a gateway to Cloud Pub/Sub using a Cloud Function, and grant the Cloud Function service account the appropriate Cloud Pub/Sub IAM roles.

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

How should you deploy the VPN?

- A. Use VPC Network Peering between the VPC and the on-premises network.
- B. Expose the VPC to the on-premises network using IAM and VPC Sharing.
- C. Create a global Cloud VPN Gateway with VPN tunnels from each region to the on-premises peer gateway.
- D. Deploy Cloud VPN Gateway in each region. Ensure that each region has at least one VPN tunnel to the on-premises peer gateway.

Your applications will be writing their logs to BigQuery for analysis. Each application should have its own table. Any logs older than 45 days should be removed.

You want to optimize storage and follow Google-recommended practices. What should you do?

- A. Configure the expiration time for your tables at 45 days
- B. Make the tables time-partitioned, and configure the partition expiration at 45 days
- C. Rely on BigQuery's default behavior to prune application logs older than 45 days
- D. Create a script that uses the BigQuery command line tool (bq) to remove records older than 45 days

You want your Google Kubernetes Engine cluster to automatically add or remove nodes based on CPU load.

What should you do?

- A. Configure a HorizontalPodAutoscaler with a target CPU usage. Enable the Cluster Autoscaler from the GCP Console.
- B. Configure a HorizontalPodAutoscaler with a target CPU usage. Enable autoscaling on the managed instance group for the cluster using the gcloud command.
- C. Create a deployment and set the maxUnavailable and maxSurge properties. Enable the Cluster Autoscaler using the gcloud command.
- D. Create a deployment and set the maxUnavailable and maxSurge properties. Enable autoscaling on the cluster managed instance group from the GCP Console.

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

What should you do?

- A. Verify that Dedicated Interconnect can replicate files to GCP. Verify that direct peering can establish a secure connection between your networks if Dedicated Interconnect fails.
- B. Verify that Dedicated Interconnect can replicate files to GCP. Verify that Cloud VPN can establish a secure connection between your networks if Dedicated Interconnect fails.
- C. Verify that the Transfer Appliance can replicate files to GCP. Verify that direct peering can establish a secure connection between your networks if the Transfer Appliance fails.
- D. Verify that the Transfer Appliance can replicate files to GCP. Verify that Cloud VPN can establish a secure connection between your networks if the Transfer Appliance fails.

Your company operates nationally and plans to use GCP for multiple batch workloads, including some that are not time-critical. You also need to use GCP services that are HIPAA-certified and manage service costs.

How should you design to meet Google best practices?

- A. Provision preemptible VMs to reduce cost. Discontinue use of all GCP services and APIs that are not HIPAA-compliant.
- B. Provision preemptible VMs to reduce cost. Disable and then discontinue use of all GCP services and APIs that are not HIPAA-compliant.
- C. Provision standard VMs in the same region to reduce cost. Discontinue use of all GCP services and APIs that are not HIPAA-compliant.
- D. Provision standard VMs to the same region to reduce cost. Disable and then discontinue use of all GCP services and APIs that are not HIPAA-compliant.

Your customer wants to do resilience testing of their authentication layer. This consists of a regional managed instance group serving a public REST API that reads from and writes to a Cloud SQL instance.

What should you do?

- A. Engage with a security company to run web scrapers that look your for users' authentication data om malicious websites and notify you if any is found.
- B. Deploy intrusion detection software to your virtual machines to detect and log unauthorized access.
- C. Schedule a disaster simulation exercise during which you can shut off all VMs in a zone to see how your application behaves.
- D. Configure a read replica for your Cloud SQL instance in a different zone than the master, and then manually trigger a failover while monitoring KPIs for our REST API.

Your BigQuery project has several users. For audit purposes, you need to see how many queries each user ran in the last month. What should you do?

- A. Connect Google Data Studio to BigQuery. Create a dimension for the users and a metric for the amount of queries per user.
- B. In the BigQuery interface, execute a query on the JOBS table to get the required information.
- C. Use 'bq show' to list all jobs. Per job, use 'bq ls' to list job information and get the required information.
- D. Use Cloud Audit Logging to view Cloud Audit Logs, and create a filter on the query operation to get the required information.

You want to automate the creation of a managed instance group. The VMs have many OS package dependencies. You want to minimize the startup time for new

VMs in the instance group.

What should you do?

- A. Use Terraform to create the managed instance group and a startup script to install the OS package dependencies.
- B. Create a custom VM image with all OS package dependencies. Use Deployment Manager to create the managed instance group with the VM image.
- C. Use Puppet to create the managed instance group and install the OS package dependencies.
- D. Use Deployment Manager to create the managed instance group and Ansible to install the OS package dependencies.

Your company captures all web traffic data in Google Analytics 360 and stores it in BigQuery. Each country has its own dataset. Each dataset has multiple tables.

You want analysts from each country to be able to see and query only the data for their respective countries.

How should you configure the access rights?

- A. Create a group per country. Add analysts to their respective country-groups. Create a single group 'all_analysts', and add all country-groups as members. Grant the 'all_analysts' group the IAM role of BigQuery jobUser. Share the appropriate dataset with view access with each respective analyst country-group.
- B. Create a group per country. Add analysts to their respective country-groups. Create a single group 'all_analysts', and add all country-groups as members. Grant the 'all_analysts' group the IAM role of BigQuery jobUser. Share the appropriate tables with view access with each respective analyst country-group.
- C. Create a group per country. Add analysts to their respective country-groups. Create a single group 'all_analysts', and add all country-groups as members. Grant the 'all_analysts' group the IAM role of BigQuery dataViewer. Share the appropriate dataset with view access with each respective analyst country- group.
- D. Create a group per country. Add analysts to their respective country-groups. Create a single group 'all_analysts', and add all country-groups as members. Grant the 'all_analysts' group the IAM role of BigQuery dataViewer. Share the appropriate table with view access with each respective analyst country-group.

You have been engaged by your client to lead the migration of their application infrastructure to GCP. One of their current problems is that the on-premises high performance SAN is requiring frequent and expensive upgrades to keep up with the variety of workloads that are identified as follows: 20 TB of log archives retained for legal reasons; 500 GB of VM boot/data volumes and templates; 500 GB of image thumbnails; 200 GB of customer session state data that allows customers to restart sessions even if off-line for several days.

Which of the following best reflects your recommendations for a cost-effective storage allocation?

- A. Local SSD for customer session state data. Lifecycle-managed Cloud Storage for log archives, thumbnails, and VM boot/data volumes.
- B. Memcache backed by Cloud Datastore for the customer session state data. Lifecycle-managed Cloud Storage for log archives, thumbnails, and VM boot/data volumes.
- C. Memcache backed by Cloud SQL for customer session state data. Assorted local SSD-backed instances for VM boot/data volumes. Cloud Storage for log archives and thumbnails.
- D. Memcache backed by Persistent Disk SSD storage for customer session state data. Assorted local SSD-backed instances for VM boot/data volumes. Cloud Storage for log archives and thumbnails.

Your web application uses Google Kubernetes Engine to manage several workloads. One workload requires a consistent set of hostnames even after pod scaling and relaunches.

Which feature of Kubernetes should you use to accomplish this?

- A. StatefulSets
- B. Role-based access control
- C. Container environment variables
- D. Persistent Volumes

You are using Cloud CDN to deliver static HTTP(S) website content hosted on a Compute Engine instance group. You want to improve the cache hit ratio.

What should you do?

- A. Customize the cache keys to omit the protocol from the key.
- B. Shorten the expiration time of the cached objects.
- C. Make sure the HTTP(S) header `Cache-Region` points to the closest region of your users.
- D. Replicate the static content in a Cloud Storage bucket. Point CloudCDN toward a load balancer on that bucket.

Your architecture calls for the centralized collection of all admin activity and VM system logs within your project.

How should you collect these logs from both VMs and services?

- A. All admin and VM system logs are automatically collected by Stackdriver.
- B. Stackdriver automatically collects admin activity logs for most services. The Stackdriver Logging agent must be installed on each instance to collect system logs.
- C. Launch a custom syslogd compute instance and configure your GCP project and VMs to forward all logs to it.
- D. Install the Stackdriver Logging agent on a single compute instance and let it collect all audit and access logs for your environment.

You have an App Engine application that needs to be updated. You want to test the update with production traffic before replacing the current application version.

What should you do?

- A. Deploy the update using the Instance Group Updater to create a partial rollout, which allows for canary testing.
- B. Deploy the update as a new version in the App Engine application, and split traffic between the new and current versions.
- C. Deploy the update in a new VPC, and use Google's global HTTP load balancing to split traffic between the update and current applications.
- D. Deploy the update as a new App Engine application, and use Google's global HTTP load balancing to split traffic between the new and current applications.

All Compute Engine instances in your VPC should be able to connect to an Active Directory server on specific ports. Any other traffic emerging from your instances is not allowed. You want to enforce this using VPC firewall rules.

How should you configure the firewall rules?

- A. Create an egress rule with priority 1000 to deny all traffic for all instances. Create another egress rule with priority 100 to allow the Active Directory traffic for all instances.
- B. Create an egress rule with priority 100 to deny all traffic for all instances. Create another egress rule with priority 1000 to allow the Active Directory traffic for all instances.
- C. Create an egress rule with priority 1000 to allow the Active Directory traffic. Rely on the implied deny egress rule with priority 100 to block all traffic for all instances.
- D. Create an egress rule with priority 100 to allow the Active Directory traffic. Rely on the implied deny egress rule with priority 1000 to block all traffic for all instances.

Your customer runs a web service used by e-commerce sites to offer product recommendations to users. The company has begun experimenting with a machine learning model on Google Cloud Platform to improve the quality of results.

What should the customer do to improve their model's results over time?

- A. Export Cloud Machine Learning Engine performance metrics from Stackdriver to BigQuery, to be used to analyze the efficiency of the model.
- B. Build a roadmap to move the machine learning model training from Cloud GPUs to Cloud TPUs, which offer better results.
- C. Monitor Compute Engine announcements for availability of newer CPU architectures, and deploy the model to them as soon as they are available for additional performance.
- D. Save a history of recommendations and results of the recommendations in BigQuery, to be used as training data.

A development team at your company has created a dockerized HTTPS web application. You need to deploy the application on Google Kubernetes Engine (GKE) and make sure that the application scales automatically.

How should you deploy to GKE?

- A. Use the Horizontal Pod Autoscaler and enable cluster autoscaling. Use an Ingress resource to load-balance the HTTPS traffic.
- B. Use the Horizontal Pod Autoscaler and enable cluster autoscaling on the Kubernetes cluster. Use a Service resource of type LoadBalancer to load-balance the HTTPS traffic.
- C. Enable autoscaling on the Compute Engine instance group. Use an Ingress resource to load-balance the HTTPS traffic.
- D. Enable autoscaling on the Compute Engine instance group. Use a Service resource of type LoadBalancer to load-balance the HTTPS traffic.

You need to design a solution for global load balancing based on the URL path being requested. You need to ensure operations reliability and end-to-end in-transit encryption based on Google best practices.

What should you do?

- A. Create a cross-region load balancer with URL Maps.
- B. Create an HTTPS load balancer with URL Maps.
- C. Create appropriate instance groups and instances. Configure SSL proxy load balancing.
- D. Create a global forwarding rule. Configure SSL proxy load balancing.

You have an application that makes HTTP requests to Cloud Storage. Occasionally the requests fail with HTTP status codes of 5xx and 429. How should you handle these types of errors?

- A. Use gRPC instead of HTTP for better performance.
- B. Implement retry logic using a truncated exponential backoff strategy.
- C. Make sure the Cloud Storage bucket is multi-regional for geo-redundancy.
- D. Monitor <https://status.cloud.google.com/feed.atom> and only make requests if Cloud Storage is not reporting an incident.

You need to develop procedures to test a disaster plan for a mission-critical application. You want to use Google-recommended practices and native capabilities within GCP.

What should you do?

- A. Use Deployment Manager to automate service provisioning. Use Activity Logs to monitor and debug your tests.
- B. Use Deployment Manager to automate service provisioning. Use Stackdriver to monitor and debug your tests.
- C. Use gcloud scripts to automate service provisioning. Use Activity Logs to monitor and debug your tests.
- D. Use gcloud scripts to automate service provisioning. Use Stackdriver to monitor and debug your tests.

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

How should you store the files?

- A. Save the files in a Multi-Regional Cloud Storage bucket.
- B. Save the files in a Regional Cloud Storage bucket, one bucket per zone of the region.
- C. Save the files in multiple Regional Cloud Storage buckets, one bucket per zone per region.
- D. Save the files in multiple Multi-Regional Cloud Storage buckets, one bucket per multi-region.

Your company acquired a healthcare startup and must retain its customers' medical information for up to 4 more years, depending on when it was created. Your corporate policy is to securely retain this data, and then delete it as soon as regulations allow.

Which approach should you take?

- A. Store the data in Google Drive and manually delete records as they expire.
- B. Anonymize the data using the Cloud Data Loss Prevention API and store it indefinitely.
- C. Store the data in Cloud Storage and use lifecycle management to delete files when they expire.
- D. Store the data in Cloud Storage and run a nightly batch script that deletes all expired data.

You are deploying a PHP App Engine Standard service with Cloud SQL as the backend. You want to minimize the number of queries to the database.

What should you do?

- A. Set the memcache service level to dedicated. Create a key from the hash of the query, and return database values from memcache before issuing a query to Cloud SQL.
- B. Set the memcache service level to dedicated. Create a cron task that runs every minute to populate the cache with keys containing query results.
- C. Set the memcache service level to shared. Create a cron task that runs every minute to save all expected queries to a key called `λcached_queriesλ`.
- D. Set the memcache service level to shared. Create a key called `λcached_queriesλ`, and return database values from the key before using a query to Cloud SQL.

You need to ensure reliability for your application and operations by supporting reliable task scheduling for compute on GCP. Leveraging Google best practices, what should you do?

- A. Using the Cron service provided by App Engine, publish messages directly to a message-processing utility service running on Compute Engine instances.
- B. Using the Cron service provided by App Engine, publish messages to a Cloud Pub/Sub topic. Subscribe to that topic using a message-processing utility service running on Compute Engine instances.
- C. Using the Cron service provided by Google Kubernetes Engine (GKE), publish messages directly to a message-processing utility service running on Compute Engine instances.
- D. Using the Cron service provided by GKE, publish messages to a Cloud Pub/Sub topic. Subscribe to that topic using a message-processing utility service running on Compute Engine instances.

Question #101

Topic 1

Your company is building a new architecture to support its data-centric business focus. You are responsible for setting up the network. Your company's mobile and web-facing applications will be deployed on-premises, and all data analysis will be conducted in GCP. The plan is to process and load 7 years of archived .csv files totaling 900 TB of data and then continue loading 10 TB of data daily. You currently have an existing 100-MB internet connection.

What actions will meet your company's needs?

- A. Compress and upload both archived files and files uploaded daily using the gsutil `-m` option.
- B. Lease a Transfer Appliance, upload archived files to it, and send it to Google to transfer archived data to Cloud Storage. Establish a connection with Google using a Dedicated Interconnect or Direct Peering connection and use it to upload files daily.
- C. Lease a Transfer Appliance, upload archived files to it, and send it to Google to transfer archived data to Cloud Storage. Establish one Cloud VPN Tunnel to VPC networks over the public internet, and compress and upload files daily using the gsutil `-m` option.
- D. Lease a Transfer Appliance, upload archived files to it, and send it to Google to transfer archived data to Cloud Storage. Establish a Cloud VPN Tunnel to VPC networks over the public internet, and compress and upload files daily.

Question #102

Topic 1

You are developing a globally scaled frontend for a legacy streaming backend data API. This API expects events in strict chronological order with no repeat data for proper processing.

Which products should you deploy to ensure guaranteed-once FIFO (first-in, first-out) delivery of data?

- A. Cloud Pub/Sub alone
- B. Cloud Pub/Sub to Cloud Dataflow
- C. Cloud Pub/Sub to Stackdriver
- D. Cloud Pub/Sub to Cloud SQL

Your company is planning to perform a lift and shift migration of their Linux RHEL 6.5+ virtual machines. The virtual machines are running in an on-premises

VMware environment. You want to migrate them to Compute Engine following Google-recommended practices. What should you do?

- A. 1. Define a migration plan based on the list of the applications and their dependencies. 2. Migrate all virtual machines into Compute Engine individually with Migrate for Compute Engine.
- B. 1. Perform an assessment of virtual machines running in the current VMware environment. 2. Create images of all disks. Import disks on Compute Engine. 3. Create standard virtual machines where the boot disks are the ones you have imported.
- C. 1. Perform an assessment of virtual machines running in the current VMware environment. 2. Define a migration plan, prepare a Migrate for Compute Engine migration RunBook, and execute the migration.
- D. 1. Perform an assessment of virtual machines running in the current VMware environment. 2. Install a third-party agent on all selected virtual machines. 3. Migrate all virtual machines into Compute Engine.

Question #104

Topic 1

You need to deploy an application to Google Cloud. The application receives traffic via TCP and reads and writes data to the filesystem. The application does not support horizontal scaling. The application process requires full control over the data on the file system because concurrent access causes corruption. The business is willing to accept a downtime when an incident occurs, but the application must be available 24/7 to support their business operations. You need to design the architecture of this application on Google Cloud. What should you do?

- A. Use a managed instance group with instances in multiple zones, use Cloud Filestore, and use an HTTP load balancer in front of the instances.
- B. Use a managed instance group with instances in multiple zones, use Cloud Filestore, and use a network load balancer in front of the instances.
- C. Use an unmanaged instance group with an active and standby instance in different zones, use a regional persistent disk, and use an HTTP load balancer in front of the instances.
- D. Use an unmanaged instance group with an active and standby instance in different zones, use a regional persistent disk, and use a network load balancer in front of the instances.

Question #105

Topic 1

Your company has an application running on multiple Compute Engine instances. You need to ensure that the application can communicate with an on-premises service that requires high throughput via internal IPs, while minimizing latency. What should you do?

- A. Use OpenVPN to configure a VPN tunnel between the on-premises environment and Google Cloud.
- B. Configure a direct peering connection between the on-premises environment and Google Cloud.
- C. Use Cloud VPN to configure a VPN tunnel between the on-premises environment and Google Cloud.
- D. Configure a Cloud Dedicated Interconnect connection between the on-premises environment and Google Cloud.

You are managing an application deployed on Cloud Run for Anthos, and you need to define a strategy for deploying new versions of the application. You want to evaluate the new code with a subset of production traffic to decide whether to proceed with the rollout. What should you do?

- A. Deploy a new revision to Cloud Run with the new version. Configure traffic percentage between revisions.
- B. Deploy a new service to Cloud Run with the new version. Add a Cloud Load Balancing instance in front of both services.
- C. In the Google Cloud Console page for Cloud Run, set up continuous deployment using Cloud Build for the development branch. As part of the Cloud Build trigger, configure the substitution variable TRAFFIC_PERCENTAGE with the percentage of traffic you want directed to a new version.
- D. In the Google Cloud Console, configure Traffic Director with a new Service that points to the new version of the application on Cloud Run. Configure Traffic Director to send a small percentage of traffic to the new version of the application.

You are monitoring Google Kubernetes Engine (GKE) clusters in a Cloud Monitoring workspace. As a Site Reliability Engineer (SRE), you need to triage incidents quickly. What should you do?

- A. Navigate the predefined dashboards in the Cloud Monitoring workspace, and then add metrics and create alert policies.
- B. Navigate the predefined dashboards in the Cloud Monitoring workspace, create custom metrics, and install alerting software on a Compute Engine instance.
- C. Write a shell script that gathers metrics from GKE nodes, publish these metrics to a Pub/Sub topic, export the data to BigQuery, and make a Data Studio dashboard.
- D. Create a custom dashboard in the Cloud Monitoring workspace for each incident, and then add metrics and create alert policies.

You are implementing a single Cloud SQL MySQL second-generation database that contains business-critical transaction data. You want to ensure that the minimum amount of data is lost in case of catastrophic failure. Which two features should you implement? (Choose two.)

- A. Sharding
- B. Read replicas
- C. Binary logging
- D. Automated backups
- E. Semisynchronous replication

You are working at a sports association whose members range in age from 8 to 30. The association collects a large amount of health data, such as sustained injuries. You are storing this data in BigQuery. Current legislation requires you to delete such information upon request of the subject. You want to design a solution that can accommodate such a request. What should you do?

- A. Use a unique identifier for each individual. Upon a deletion request, delete all rows from BigQuery with this identifier.
- B. When ingesting new data in BigQuery, run the data through the Data Loss Prevention (DLP) API to identify any personal information. As part of the DLP scan, save the result to Data Catalog. Upon a deletion request, query Data Catalog to find the column with personal information.
- C. Create a BigQuery view over the table that contains all data. Upon a deletion request, exclude the rows that affect the subject's data from this view. Use this view instead of the source table for all analysis tasks.
- D. Use a unique identifier for each individual. Upon a deletion request, overwrite the column with the unique identifier with a salted SHA256 of its value.

Question #110

Topic 1

Your company has announced that they will be outsourcing operations functions. You want to allow developers to easily stage new versions of a cloud-based application in the production environment and allow the outsourced operations team to autonomously promote staged versions to production. You want to minimize the operational overhead of the solution. Which Google Cloud product should you migrate to?

- A. App Engine
- B. GKE On-Prem
- C. Compute Engine
- D. Google Kubernetes Engine

Question #111

Topic 1

Your company is running its application workloads on Compute Engine. The applications have been deployed in production, acceptance, and development environments. The production environment is business-critical and is used 24/7, while the acceptance and development environments are only critical during office hours. Your CFO has asked you to optimize these environments to achieve cost savings during idle times. What should you do?

- A. Create a shell script that uses the `gcloud` command to change the machine type of the development and acceptance instances to a smaller machine type outside of office hours. Schedule the shell script on one of the production instances to automate the task.
- B. Use Cloud Scheduler to trigger a Cloud Function that will stop the development and acceptance environments after office hours and start them just before office hours.
- C. Deploy the development and acceptance applications on a managed instance group and enable autoscaling.
- D. Use regular Compute Engine instances for the production environment, and use preemptible VMs for the acceptance and development environments.

You are moving an application that uses MySQL from on-premises to Google Cloud. The application will run on Compute Engine and will use Cloud SQL. You want to cut over to the Compute Engine deployment of the application with minimal downtime and no data loss to your customers. You want to migrate the application with minimal modification. You also need to determine the cutover strategy. What should you do?

- A. 1. Set up Cloud VPN to provide private network connectivity between the Compute Engine application and the on-premises MySQL server. 2. Stop the on-premises application. 3. Create a mysqldump of the on-premises MySQL server. 4. Upload the dump to a Cloud Storage bucket. 5. Import the dump into Cloud SQL. 6. Modify the source code of the application to write queries to both databases and read from its local database. 7. Start the Compute Engine application. 8. Stop the on-premises application.
- B. 1. Set up Cloud SQL proxy and MySQL proxy. 2. Create a mysqldump of the on-premises MySQL server. 3. Upload the dump to a Cloud Storage bucket. 4. Import the dump into Cloud SQL. 5. Stop the on-premises application. 6. Start the Compute Engine application.
- C. 1. Set up Cloud VPN to provide private network connectivity between the Compute Engine application and the on-premises MySQL server. 2. Stop the on-premises application. 3. Start the Compute Engine application, configured to read and write to the on-premises MySQL server. 4. Create the replication configuration in Cloud SQL. 5. Configure the source database server to accept connections from the Cloud SQL replica. 6. Finalize the Cloud SQL replica configuration. 7. When replication has been completed, stop the Compute Engine application. 8. Promote the Cloud SQL replica to a standalone instance. 9. Restart the Compute Engine application, configured to read and write to the Cloud SQL standalone instance.
- D. 1. Stop the on-premises application. 2. Create a mysqldump of the on-premises MySQL server. 3. Upload the dump to a Cloud Storage bucket. 4. Import the dump into Cloud SQL. 5. Start the application on Compute Engine.

Question #113

Topic 1

Your organization has decided to restrict the use of external IP addresses on instances to only approved instances. You want to enforce this requirement across all of your Virtual Private Clouds (VPCs). What should you do?

- A. Remove the default route on all VPCs. Move all approved instances into a new subnet that has a default route to an internet gateway.
- B. Create a new VPC in custom mode. Create a new subnet for the approved instances, and set a default route to the internet gateway on this new subnet.
- C. Implement a Cloud NAT solution to remove the need for external IP addresses entirely.
- D. Set an Organization Policy with a constraint on constraints/compute.vmExternallpAccess. List the approved instances in the allowedValues list.

Question #114

Topic 1

Your company uses the Firewall Insights feature in the Google Network Intelligence Center. You have several firewall rules applied to Compute Engine instances.

You need to evaluate the efficiency of the applied firewall ruleset. When you bring up the Firewall Insights page in the Google Cloud Console, you notice that there are no log rows to display. What should you do to troubleshoot the issue?

- A. Enable Virtual Private Cloud (VPC) flow logging.
- B. Enable Firewall Rules Logging for the firewall rules you want to monitor.
- C. Verify that your user account is assigned the compute.networkAdmin Identity and Access Management (IAM) role.
- D. Install the Google Cloud SDK, and verify that there are no Firewall logs in the command line output.

Your company has sensitive data in Cloud Storage buckets. Data analysts have Identity Access Management (IAM) permissions to read the buckets. You want to prevent data analysts from retrieving the data in the buckets from outside the office network. What should you do?

- A. 1. Create a VPC Service Controls perimeter that includes the projects with the buckets. 2. Create an access level with the CIDR of the office network.
- B. 1. Create a firewall rule for all instances in the Virtual Private Cloud (VPC) network for source range. 2. Use the Classless Inter-domain Routing (CIDR) of the office network.
- C. 1. Create a Cloud Function to remove IAM permissions from the buckets, and another Cloud Function to add IAM permissions to the buckets. 2. Schedule the Cloud Functions with Cloud Scheduler to add permissions at the start of business and remove permissions at the end of business.
- D. 1. Create a Cloud VPN to the office network. 2. Configure Private Google Access for on-premises hosts.

You have developed a non-critical update to your application that is running in a managed instance group, and have created a new instance template with the update that you want to release. To prevent any possible impact to the application, you don't want to update any running instances. You want any new instances that are created by the managed instance group to contain the new update. What should you do?

- A. Start a new rolling restart operation.
- B. Start a new rolling replace operation.
- C. Start a new rolling update. Select the Proactive update mode.
- D. Start a new rolling update. Select the Opportunistic update mode.

Your company is designing its application landscape on Compute Engine. Whenever a zonal outage occurs, the application should be restored in another zone as quickly as possible with the latest application data. You need to design the solution to meet this requirement. What should you do?

- A. Create a snapshot schedule for the disk containing the application data. Whenever a zonal outage occurs, use the latest snapshot to restore the disk in the same zone.
- B. Configure the Compute Engine instances with an instance template for the application, and use a regional persistent disk for the application data. Whenever a zonal outage occurs, use the instance template to spin up the application in another zone in the same region. Use the regional persistent disk for the application data.
- C. Create a snapshot schedule for the disk containing the application data. Whenever a zonal outage occurs, use the latest snapshot to restore the disk in another zone within the same region.
- D. Configure the Compute Engine instances with an instance template for the application, and use a regional persistent disk for the application data. Whenever a zonal outage occurs, use the instance template to spin up the application in another region. Use the regional persistent disk for the application data.

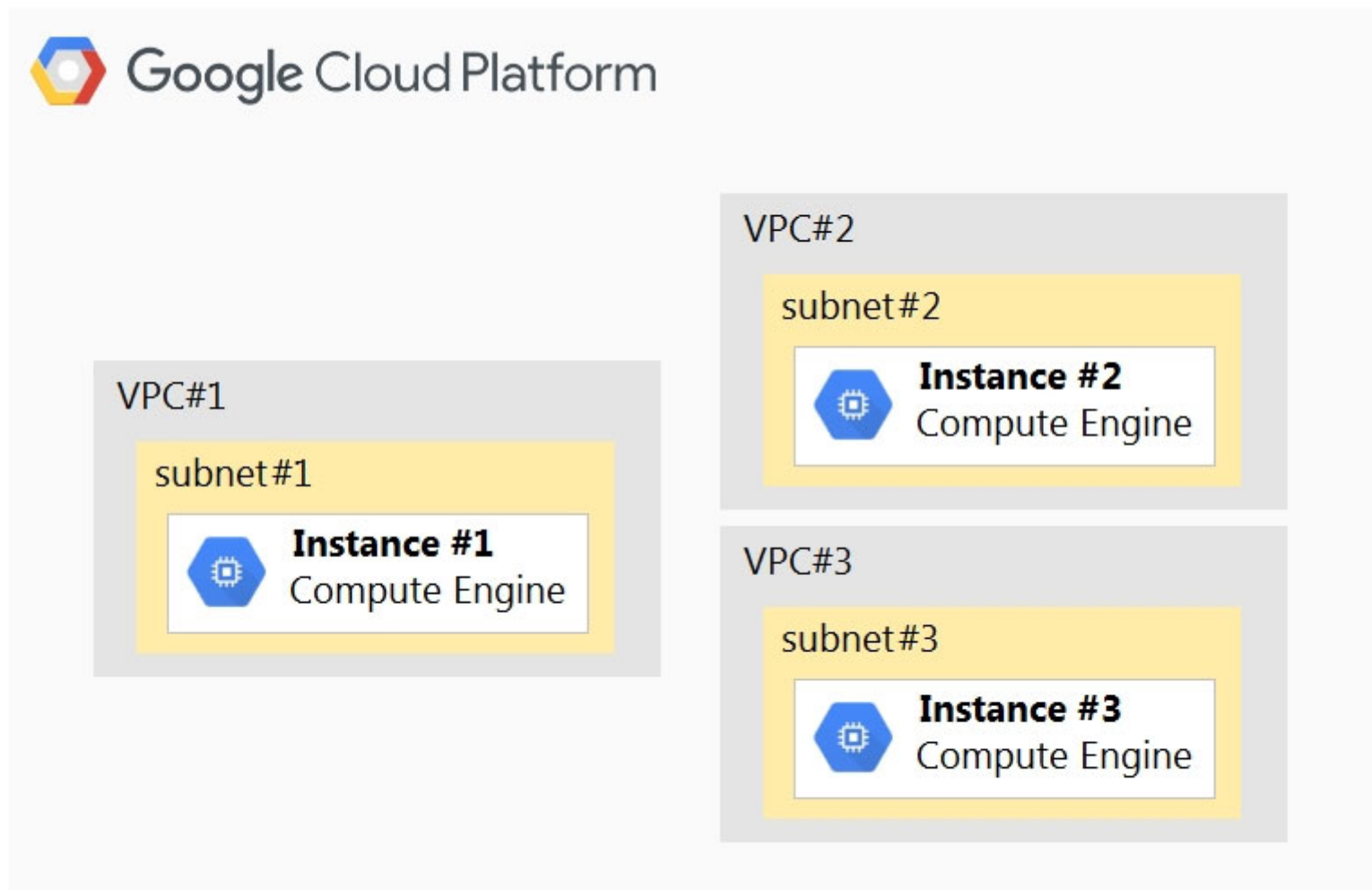
Your company has just acquired another company, and you have been asked to integrate their existing Google Cloud environment into your company's data center. Upon investigation, you discover that some of the RFC 1918 IP ranges being used in the new company's Virtual Private Cloud (VPC) overlap with your data center IP space. What should you do to enable connectivity and make sure that there are no routing conflicts when connectivity is established?

- A. Create a Cloud VPN connection from the new VPC to the data center, create a Cloud Router, and apply new IP addresses so there is no overlapping IP space.
- B. Create a Cloud VPN connection from the new VPC to the data center, and create a Cloud NAT instance to perform NAT on the overlapping IP space.
- C. Create a Cloud VPN connection from the new VPC to the data center, create a Cloud Router, and apply a custom route advertisement to block the overlapping IP space.
- D. Create a Cloud VPN connection from the new VPC to the data center, and apply a firewall rule that blocks the overlapping IP space.

You need to migrate Hadoop jobs for your company's Data Science team without modifying the underlying infrastructure. You want to minimize costs and infrastructure management effort. What should you do?

- A. Create a Dataproc cluster using standard worker instances.
- B. Create a Dataproc cluster using preemptible worker instances.
- C. Manually deploy a Hadoop cluster on Compute Engine using standard instances.
- D. Manually deploy a Hadoop cluster on Compute Engine using preemptible instances.

Your company has a project in Google Cloud with three Virtual Private Clouds (VPCs). There is a Compute Engine instance on each VPC. Network subnets do not overlap and must remain separated. The network configuration is shown below.



Instance #1 is an exception and must communicate directly with both Instance #2 and Instance #3 via internal IPs. How should you accomplish this?

- A. Create a cloud router to advertise subnet #2 and subnet #3 to subnet #1.
- B. Add two additional NICs to Instance #1 with the following configuration: NIC1 -> VPC: VPC #2 -> SUBNETWORK: subnet #2 NIC2 -> VPC: VPC #3 -> SUBNETWORK: subnet #3 Update firewall rules to enable traffic between instances.
- C. Create two VPN tunnels via CloudVPN: 1 -> between VPC #1 and VPC #2. 1 -> between VPC #2 and VPC #3. Update firewall rules to enable traffic between the instances.
- D. Peer all three VPCs: -> Peer VPC #1 with VPC #2. -> Peer VPC #2 with VPC #3. Update firewall rules to enable traffic between the instances.

You need to deploy an application on Google Cloud that must run on a Debian Linux environment. The application requires extensive configuration in order to operate correctly. You want to ensure that you can install Debian distribution updates with minimal manual intervention whenever they become available. What should you do?

- A. Create a Compute Engine instance template using the most recent Debian image. Create an instance from this template, and install and configure the application as part of the startup script. Repeat this process whenever a new Google-managed Debian image becomes available.
- B. Create a Debian-based Compute Engine instance, install and configure the application, and use OS patch management to install available updates.
- C. Create an instance with the latest available Debian image. Connect to the instance via SSH, and install and configure the application on the instance. Repeat this process whenever a new Google-managed Debian image becomes available.
- D. Create a Docker container with Debian as the base image. Install and configure the application as part of the Docker image creation process. Host the container on Google Kubernetes Engine and restart the container whenever a new update is available.

You have an application that runs in Google Kubernetes Engine (GKE). Over the last 2 weeks, customers have reported that a specific part of the application returns errors very frequently. You currently have no logging or monitoring solution enabled on your GKE cluster. You want to diagnose the problem, but you have not been able to replicate the issue. You want to cause minimal disruption to the application. What should you do?

- A. 1. Update your GKE cluster to use Cloud Operations for GKE. 2. Use the GKE Monitoring dashboard to investigate logs from affected Pods.
- B. 1. Create a new GKE cluster with Cloud Operations for GKE enabled. 2. Migrate the affected Pods to the new cluster, and redirect traffic for those Pods to the new cluster. 3. Use the GKE Monitoring dashboard to investigate logs from affected Pods.
- C. 1. Update your GKE cluster to use Cloud Operations for GKE, and deploy Prometheus. 2. Set an alert to trigger whenever the application returns an error.
- D. 1. Create a new GKE cluster with Cloud Operations for GKE enabled, and deploy Prometheus. 2. Migrate the affected Pods to the new cluster, and redirect traffic for those Pods to the new cluster. 3. Set an alert to trigger whenever the application returns an error.

You need to deploy a stateful workload on Google Cloud. The workload can scale horizontally, but each instance needs to read and write to the same POSIX filesystem. At high load, the stateful workload needs to support up to 100 MB/s of writes. What should you do?

- A. Use a persistent disk for each instance.
- B. Use a regional persistent disk for each instance.
- C. Create a Cloud Filestore instance and mount it in each instance.
- D. Create a Cloud Storage bucket and mount it in each instance using gcsfuse.

Your company has an application deployed on Anthos clusters (formerly Anthos GKE) that is running multiple microservices. The cluster has both Anthos Service

Mesh and Anthos Config Management configured. End users inform you that the application is responding very slowly. You want to identify the microservice that is causing the delay. What should you do?

- A. Use the Service Mesh visualization in the Cloud Console to inspect the telemetry between the microservices.
- B. Use Anthos Config Management to create a ClusterSelector selecting the relevant cluster. On the Google Cloud Console page for Google Kubernetes Engine, view the Workloads and filter on the cluster. Inspect the configurations of the filtered workloads.
- C. Use Anthos Config Management to create a namespaceSelector selecting the relevant cluster namespace. On the Google Cloud Console page for Google Kubernetes Engine, visit the workloads and filter on the namespace. Inspect the configurations of the filtered workloads.
- D. Reinstall istio using the default istio profile in order to collect request latency. Evaluate the telemetry between the microservices in the Cloud Console.

You are working at a financial institution that stores mortgage loan approval documents on Cloud Storage. Any change to these approval documents must be uploaded as a separate approval file, so you want to ensure that these documents cannot be deleted or overwritten for the next 5 years. What should you do?

- A. Create a retention policy on the bucket for the duration of 5 years. Create a lock on the retention policy.
- B. Create the bucket with uniform bucket-level access, and grant a service account the role of Object Writer. Use the service account to upload new files.
- C. Use a customer-managed key for the encryption of the bucket. Rotate the key after 5 years.
- D. Create the bucket with fine-grained access control, and grant a service account the role of Object Writer. Use the service account to upload new files.

Your team will start developing a new application using microservices architecture on Kubernetes Engine. As part of the development lifecycle, any code change that has been pushed to the remote develop branch on your GitHub repository should be built and tested automatically. When the build and test are successful, the relevant microservice will be deployed automatically in the development environment. You want to ensure that all code deployed in the development environment follows this process. What should you do?

- A. Have each developer install a pre-commit hook on their workstation that tests the code and builds the container when committing on the development branch. After a successful commit, have the developer deploy the newly built container image on the development cluster.
- B. Install a post-commit hook on the remote git repository that tests the code and builds the container when code is pushed to the development branch. After a successful commit, have the developer deploy the newly built container image on the development cluster.
- C. Create a Cloud Build trigger based on the development branch that tests the code, builds the container, and stores it in Container Registry. Create a deployment pipeline that watches for new images and deploys the new image on the development cluster. Ensure only the deployment tool has access to deploy new versions.
- D. Create a Cloud Build trigger based on the development branch to build a new container image and store it in Container Registry. Rely on Vulnerability Scanning to ensure the code tests succeed. As the final step of the Cloud Build process, deploy the new container image on the development cluster. Ensure only Cloud Build has access to deploy new versions.

Your operations team has asked you to help diagnose a performance issue in a production application that runs on Compute Engine. The application is dropping requests that reach it when under heavy load. The process list for affected instances shows a single application process that is consuming all available CPU, and autoscaling has reached the upper limit of instances. There is no abnormal load on any other related systems, including the database. You want to allow production traffic to be served again as quickly as possible. Which action should you recommend?

- A. Change the autoscaling metric to `agent.googleapis.com/memory/percent_used`.
- B. Restart the affected instances on a staggered schedule.
- C. SSH to each instance and restart the application process.
- D. Increase the maximum number of instances in the autoscaling group.

You are implementing the infrastructure for a web service on Google Cloud. The web service needs to receive and store the data from 500,000 requests per second. The data will be queried later in real time, based on exact matches of a known set of attributes. There will be periods where the web service will not receive any requests. The business wants to keep costs low. Which web service platform and database should you use for the application?

- A. Cloud Run and BigQuery
- B. Cloud Run and Cloud Bigtable
- C. A Compute Engine autoscaling managed instance group and BigQuery
- D. A Compute Engine autoscaling managed instance group and Cloud Bigtable

You are developing an application using different microservices that should remain internal to the cluster. You want to be able to configure each microservice with a specific number of replicas. You also want to be able to address a specific microservice from any other microservice in a uniform way, regardless of the number of replicas the microservice scales to. You need to implement this solution on Google Kubernetes Engine. What should you do?

- A. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using a Service, and use the Service DNS name to address it from other microservices within the cluster.
- B. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using an Ingress, and use the Ingress IP address to address the Deployment from other microservices within the cluster.
- C. Deploy each microservice as a Pod. Expose the Pod in the cluster using a Service, and use the Service DNS name to address the microservice from other microservices within the cluster.
- D. Deploy each microservice as a Pod. Expose the Pod in the cluster using an Ingress, and use the Ingress IP address name to address the Pod from other microservices within the cluster.

Your company has a networking team and a development team. The development team runs applications on Compute Engine instances that contain sensitive data. The development team requires administrative permissions for Compute Engine. Your company requires all network resources to be managed by the networking team. The development team does not want the networking team to have access to the sensitive data on the instances. What should you do?

- A. 1. Create a project with a standalone VPC and assign the Network Admin role to the networking team. 2. Create a second project with a standalone VPC and assign the Compute Admin role to the development team. 3. Use Cloud VPN to join the two VPCs.
- B. 1. Create a project with a standalone Virtual Private Cloud (VPC), assign the Network Admin role to the networking team, and assign the Compute Admin role to the development team.
- C. 1. Create a project with a Shared VPC and assign the Network Admin role to the networking team. 2. Create a second project without a VPC, configure it as a Shared VPC service project, and assign the Compute Admin role to the development team.
- D. 1. Create a project with a standalone VPC and assign the Network Admin role to the networking team. 2. Create a second project with a standalone VPC and assign the Compute Admin role to the development team. 3. Use VPC Peering to join the two VPCs.

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

- A. Store static content such as HTML and images in Cloud CDN. Host the APIs on App Engine and store the user data in Cloud SQL.
- B. Store static content such as HTML and images in a Cloud Storage bucket. Host the APIs on a zonal Google Kubernetes Engine cluster with worker nodes in multiple zones, and save the user data in Cloud Spanner.
- C. Store static content such as HTML and images in Cloud CDN. Use Cloud Run to host the APIs and save the user data in Cloud SQL.
- D. Store static content such as HTML and images in a Cloud Storage bucket. Use Cloud Functions to host the APIs and save the user data in Firestore.

Your company sends all Google Cloud logs to Cloud Logging. Your security team wants to monitor the logs. You want to ensure that the security team can react quickly if an anomaly such as an unwanted firewall change or server breach is detected. You want to follow Google-recommended practices. What should you do?

- A. Schedule a cron job with Cloud Scheduler. The scheduled job queries the logs every minute for the relevant events.
- B. Export logs to BigQuery, and trigger a query in BigQuery to process the log data for the relevant events.
- C. Export logs to a Pub/Sub topic, and trigger Cloud Function with the relevant log events.
- D. Export logs to a Cloud Storage bucket, and trigger Cloud Run with the relevant log events.

You have deployed several instances on Compute Engine. As a security requirement, instances cannot have a public IP address. There is no VPN connection between Google Cloud and your office, and you need to connect via SSH into a specific machine without violating the security requirements. What should you do?

- A. Configure Cloud NAT on the subnet where the instance is hosted. Create an SSH connection to the Cloud NAT IP address to reach the instance.
- B. Add all instances to an unmanaged instance group. Configure TCP Proxy Load Balancing with the instance group as a backend. Connect to the instance using the TCP Proxy IP.
- C. Configure Identity-Aware Proxy (IAP) for the instance and ensure that you have the role of IAP-secured Tunnel User. Use the gcloud command line tool to ssh into the instance.
- D. Create a bastion host in the network to SSH into the bastion host from your office location. From the bastion host, SSH into the desired instance.

Your company is using Google Cloud. You have two folders under the Organization: Finance and Shopping. The members of the development team are in a

Google Group. The development team group has been assigned the Project Owner role on the Organization. You want to prevent the development team from creating resources in projects in the Finance folder. What should you do?

- A. Assign the development team group the Project Viewer role on the Finance folder, and assign the development team group the Project Owner role on the Shopping folder.
- B. Assign the development team group only the Project Viewer role on the Finance folder.
- C. Assign the development team group the Project Owner role on the Shopping folder, and remove the development team group Project Owner role from the Organization.
- D. Assign the development team group only the Project Owner role on the Shopping folder.

You are developing your microservices application on Google Kubernetes Engine. During testing, you want to validate the behavior of your application in case a specific microservice should suddenly crash. What should you do?

- A. Add a taint to one of the nodes of the Kubernetes cluster. For the specific microservice, configure a pod anti-affinity label that has the name of the tainted node as a value.
- B. Use Istio's fault injection on the particular microservice whose faulty behavior you want to simulate.
- C. Destroy one of the nodes of the Kubernetes cluster to observe the behavior.
- D. Configure Istio's traffic management features to steer the traffic away from a crashing microservice.

Your company is developing a new application that will allow globally distributed users to upload pictures and share them with other selected users. The application will support millions of concurrent users. You want to allow developers to focus on just building code without having to create and maintain the underlying infrastructure. Which service should you use to deploy the application?

- A. App Engine
- B. Cloud Endpoints
- C. Compute Engine
- D. Google Kubernetes Engine

Your company provides a recommendation engine for retail customers. You are providing retail customers with an API where they can submit a user ID and the

API returns a list of recommendations for that user. You are responsible for the API lifecycle and want to ensure stability for your customers in case the API makes backward-incompatible changes. You want to follow Google-recommended practices. What should you do?

- A. Create a distribution list of all customers to inform them of an upcoming backward-incompatible change at least one month before replacing the old API with the new API.
- B. Create an automated process to generate API documentation, and update the public API documentation as part of the CI/CD process when deploying an update to the API.
- C. Use a versioning strategy for the APIs that increases the version number on every backward-incompatible change.
- D. Use a versioning strategy for the APIs that adds the suffix `λDEPRECATEDλ` to the current API version number on every backward-incompatible change. Use the current version number for the new API.

Question #138

Topic 1

Your company has developed a monolithic, 3-tier application to allow external users to upload and share files. The solution cannot be easily enhanced and lacks reliability. The development team would like to re-architect the application to adopt microservices and a fully managed service approach, but they need to convince their leadership that the effort is worthwhile. Which advantage(s) should they highlight to leadership?

- A. The new approach will be significantly less costly, make it easier to manage the underlying infrastructure, and automatically manage the CI/CD pipelines.
- B. The monolithic solution can be converted to a container with Docker. The generated container can then be deployed into a Kubernetes cluster.
- C. The new approach will make it easier to decouple infrastructure from application, develop and release new features, manage the underlying infrastructure, manage CI/CD pipelines and perform A/B testing, and scale the solution if necessary.
- D. The process can be automated with Migrate for Compute Engine.

Question #139

Topic 1

Your team is developing a web application that will be deployed on Google Kubernetes Engine (GKE). Your CTO expects a successful launch and you need to ensure your application can handle the expected load of tens of thousands of users. You want to test the current deployment to ensure the latency of your application stays below a certain threshold. What should you do?

- A. Use a load testing tool to simulate the expected number of concurrent users and total requests to your application, and inspect the results.
- B. Enable autoscaling on the GKE cluster and enable horizontal pod autoscaling on your application deployments. Send curl requests to your application, and validate if the auto scaling works.
- C. Replicate the application over multiple GKE clusters in every Google Cloud region. Configure a global HTTP(S) load balancer to expose the different clusters over a single global IP address.
- D. Use Cloud Debugger in the development environment to understand the latency between the different microservices.

Your company has a Kubernetes application that pulls messages from Pub/Sub and stores them in Filestore. Because the application is simple, it was deployed as a single pod. The infrastructure team has analyzed Pub/Sub metrics and discovered that the application cannot process the messages in real time. Most of them wait for minutes before being processed. You need to scale the elaboration process that is I/O-intensive. What should you do?

- A. Use `kubectl autoscale deployment APP_NAME --max 6 --min 2 --cpu-percent 50` to configure Kubernetes autoscaling deployment.
- B. Configure a Kubernetes autoscaling deployment based on the `subscription/push_request_latencies` metric.
- C. Use the `--enable-autoscaling` flag when you create the Kubernetes cluster.
- D. Configure a Kubernetes autoscaling deployment based on the `subscription/num_undelivered_messages` metric.

Your company is developing a web-based application. You need to make sure that production deployments are linked to source code commits and are fully auditable. What should you do?

- A. Make sure a developer is tagging the code commit with the date and time of commit.
- B. Make sure a developer is adding a comment to the commit that links to the deployment.
- C. Make the container tag match the source code commit hash.
- D. Make sure the developer is tagging the commits with `latest`.

An application development team has come to you for advice. They are planning to write and deploy an HTTP(S) API using Go 1.12. The API will have a very unpredictable workload and must remain reliable during peaks in traffic. They want to minimize operational overhead for this application. Which approach should you recommend?

- A. Develop the application with containers, and deploy to Google Kubernetes Engine.
- B. Develop the application for App Engine standard environment.
- C. Use a Managed Instance Group when deploying to Compute Engine.
- D. Develop the application for App Engine flexible environment, using a custom runtime.

Your company is designing its data lake on Google Cloud and wants to develop different ingestion pipelines to collect unstructured data from different sources.

After the data is stored in Google Cloud, it will be processed in several data pipelines to build a recommendation engine for end users on the website. The structure of the data retrieved from the source systems can change at any time. The data must be stored exactly as it was retrieved for reprocessing purposes in case the data structure is incompatible with the current processing pipelines. You need to design an architecture to support the use case after you retrieve the data. What should you do?

- A. Send the data through the processing pipeline, and then store the processed data in a BigQuery table for reprocessing.
- B. Store the data in a BigQuery table. Design the processing pipelines to retrieve the data from the table.
- C. Send the data through the processing pipeline, and then store the processed data in a Cloud Storage bucket for reprocessing.
- D. Store the data in a Cloud Storage bucket. Design the processing pipelines to retrieve the data from the bucket.

You are responsible for the Google Cloud environment in your company. Multiple departments need access to their own projects, and the members within each department will have the same project responsibilities. You want to structure your Google Cloud environment for minimal maintenance and maximum overview of IAM permissions as each department's projects start and end. You want to follow Google-recommended practices. What should you do?

- A. Grant all department members the required IAM permissions for their respective projects.
- B. Create a Google Group per department and add all department members to their respective groups. Create a folder per department and grant the respective group the required IAM permissions at the folder level. Add the projects under the respective folders.
- C. Create a folder per department and grant the respective members of the department the required IAM permissions at the folder level. Structure all projects for each department under the respective folders.
- D. Create a Google Group per department and add all department members to their respective groups. Grant each group the required IAM permissions for their respective projects.

Your company has an application running as a Deployment in a Google Kubernetes Engine (GKE) cluster. You have separate clusters for development, staging, and production. You have discovered that the team is able to deploy a Docker image to the production cluster without first testing the deployment in development and then staging. You want to allow the team to have autonomy but want to prevent this from happening. You want a Google Cloud solution that can be implemented quickly with minimal effort. What should you do?

- A. Configure a Kubernetes lifecycle hook to prevent the container from starting if it is not approved for usage in the given environment.
- B. Implement a corporate policy to prevent teams from deploying Docker images to an environment unless the Docker image was tested in an earlier environment.
- C. Configure binary authorization policies for the development, staging, and production clusters. Create attestations as part of the continuous integration pipeline.
- D. Create a Kubernetes admissions controller to prevent the container from starting if it is not approved for usage in the given environment.

Your company wants to migrate their 10-TB on-premises database export into Cloud Storage. You want to minimize the time it takes to complete this activity, the overall cost, and database load. The bandwidth between the on-premises environment and Google Cloud is 1 Gbps. You want to follow Google-recommended practices. What should you do?

- A. Develop a Dataflow job to read data directly from the database and write it into Cloud Storage.
- B. Use the Data Transfer appliance to perform an offline migration.
- C. Use a commercial partner ETL solution to extract the data from the on-premises database and upload it into Cloud Storage.
- D. Compress the data and upload it with `gsutil -m` to enable multi-threaded copy.

Your company has an enterprise application running on Compute Engine that requires high availability and high performance. The application has been deployed on two instances in two zones in the same region in active-passive mode. The application writes data to a persistent disk. In the case of a single zone outage, that data should be immediately made available to the other instance in the other zone. You want to maximize performance while minimizing downtime and data loss.

What should you do?

- A. 1. Attach a persistent SSD disk to the first instance. 2. Create a snapshot every hour. 3. In case of a zone outage, recreate a persistent SSD disk in the second instance where data is coming from the created snapshot.
- B. 1. Create a Cloud Storage bucket. 2. Mount the bucket into the first instance with gcs-fuse. 3. In case of a zone outage, mount the Cloud Storage bucket to the second instance with gcs-fuse.
- C. 1. Attach a regional SSD persistent disk to the first instance. 2. In case of a zone outage, force-attach the disk to the other instance.
- D. 1. Attach a local SSD to the first instance disk. 2. Execute an rsync command every hour where the target is a persistent SSD disk attached to the second instance. 3. In case of a zone outage, use the second instance.

You are designing a Data Warehouse on Google Cloud and want to store sensitive data in BigQuery. Your company requires you to generate the encryption keys outside of Google Cloud. You need to implement a solution. What should you do?

- A. Generate a new key in Cloud Key Management Service (Cloud KMS). Store all data in Cloud Storage using the customer-managed key option and select the created key. Set up a Dataflow pipeline to decrypt the data and to store it in a new BigQuery dataset.
- B. Generate a new key in Cloud KMS. Create a dataset in BigQuery using the customer-managed key option and select the created key.
- C. Import a key in Cloud KMS. Store all data in Cloud Storage using the customer-managed key option and select the created key. Set up a Dataflow pipeline to decrypt the data and to store it in a new BigQuery dataset.
- D. Import a key in Cloud KMS. Create a dataset in BigQuery using the customer-supplied key option and select the created key.

Your organization has stored sensitive data in a Cloud Storage bucket. For regulatory reasons, your company must be able to rotate the encryption key used to encrypt the data in the bucket. The data will be processed in Dataproc. You want to follow Google-recommended practices for security. What should you do?

- A. Create a key with Cloud Key Management Service (KMS). Encrypt the data using the encrypt method of Cloud KMS.
- B. Create a key with Cloud Key Management Service (KMS). Set the encryption key on the bucket to the Cloud KMS key.
- C. Generate a GPG key pair. Encrypt the data using the GPG key. Upload the encrypted data to the bucket.
- D. Generate an AES-256 encryption key. Encrypt the data in the bucket using the customer-supplied encryption keys feature.

Your team needs to create a Google Kubernetes Engine (GKE) cluster to host a newly built application that requires access to third-party services on the internet.

Your company does not allow any Compute Engine instance to have a public IP address on Google Cloud. You need to create a deployment strategy that adheres to these guidelines. What should you do?

- A. Configure the GKE cluster as a private cluster, and configure Cloud NAT Gateway for the cluster subnet.
- B. Configure the GKE cluster as a private cluster. Configure Private Google Access on the Virtual Private Cloud (VPC).
- C. Configure the GKE cluster as a route-based cluster. Configure Private Google Access on the Virtual Private Cloud (VPC).
- D. Create a Compute Engine instance, and install a NAT Proxy on the instance. Configure all workloads on GKE to pass through this proxy to access third-party services on the Internet.

Your company has a support ticketing solution that uses App Engine Standard. The project that contains the App Engine application already has a Virtual Private

Cloud (VPC) network fully connected to the company's on-premises environment through a Cloud VPN tunnel. You want to enable the App Engine application to communicate with a database that is running in the company's on-premises environment. What should you do?

- A. Configure private Google access for on-premises hosts only.
- B. Configure private Google access.
- C. Configure private services access.
- D. Configure serverless VPC access.

Your company is planning to upload several important files to Cloud Storage. After the upload is completed, they want to verify that the uploaded content is identical to what they have on-premises. You want to minimize the cost and effort of performing this check. What should you do?

- A. 1. Use Linux shasum to compute a digest of files you want to upload. 2. Use gsutil -m to upload all the files to Cloud Storage. 3. Use gsutil cp to download the uploaded files. 4. Use Linux shasum to compute a digest of the downloaded files. 5. Compare the hashes.
- B. 1. Use gsutil -m to upload the files to Cloud Storage. 2. Develop a custom Java application that computes CRC32C hashes. 3. Use gsutil ls -L gs://[YOUR_BUCKET_NAME] to collect CRC32C hashes of the uploaded files. 4. Compare the hashes.
- C. 1. Use gsutil -m to upload all the files to Cloud Storage. 2. Use gsutil cp to download the uploaded files. 3. Use Linux diff to compare the content of the files.
- D. 1. Use gsutil -m to upload the files to Cloud Storage. 2. Use gsutil hash -c FILE_NAME to generate CRC32C hashes of all on-premises files. 3. Use gsutil ls -L gs://[YOUR_BUCKET_NAME] to collect CRC32C hashes of the uploaded files. 4. Compare the hashes.

You have deployed an application on Anthos clusters (formerly Anthos GKE). According to the SRE practices at your company, you need to be alerted if request latency is above a certain threshold for a specified amount of time. What should you do?

- A. Install Anthos Service Mesh on your cluster. Use the Google Cloud Console to define a Service Level Objective (SLO), and create an alerting policy based on this SLO.
- B. Enable the Cloud Trace API on your project, and use Cloud Monitoring Alerts to send an alert based on the Cloud Trace metrics.
- C. Use Cloud Profiler to follow up the request latency. Create a custom metric in Cloud Monitoring based on the results of Cloud Profiler, and create an Alerting policy in case this metric exceeds the threshold.
- D. Configure Anthos Config Management on your cluster, and create a yaml file that defines the SLO and alerting policy you want to deploy in your cluster.

Your company has a stateless web API that performs scientific calculations. The web API runs on a single Google Kubernetes Engine (GKE) cluster. The cluster is currently deployed in us-central1. Your company has expanded to offer your API to customers in Asia. You want to reduce the latency for users in Asia.

What should you do?

- A. Create a second GKE cluster in asia-southeast1, and expose both APIs using a Service of type LoadBalancer. Add the public IPs to the Cloud DNS zone.
- B. Use a global HTTP(s) load balancer with Cloud CDN enabled.
- C. Create a second GKE cluster in asia-southeast1, and use kubemci to create a global HTTP(s) load balancer.
- D. Increase the memory and CPU allocated to the application in the cluster.

You are migrating third-party applications from optimized on-premises virtual machines to Google Cloud. You are unsure about the optimum CPU and memory options. The applications have a consistent usage pattern across multiple weeks. You want to optimize resource usage for the lowest cost. What should you do?

- A. Create an instance template with the smallest available machine type, and use an image of the third-party application taken from a current on-premises virtual machine. Create a managed instance group that uses average CPU utilization to autoscale the number of instances in the group. Modify the average CPU utilization threshold to optimize the number of instances running.
- B. Create an App Engine flexible environment, and deploy the third-party application using a Dockerfile and a custom runtime. Set CPU and memory options similar to your application's current on-premises virtual machine in the app.yaml file.
- C. Create multiple Compute Engine instances with varying CPU and memory options. Install the Cloud Monitoring agent, and deploy the third-party application on each of them. Run a load test with high traffic levels on the application, and use the results to determine the optimal settings.
- D. Create a Compute Engine instance with CPU and memory options similar to your application's current on-premises virtual machine. Install the Cloud Monitoring agent, and deploy the third-party application. Run a load test with normal traffic levels on the application, and follow the Rightsizing Recommendations in the Cloud Console.

Your company has a Google Cloud project that uses BigQuery for data warehousing. They have a VPN tunnel between the on-premises environment and Google

Cloud that is configured with Cloud VPN. The security team wants to avoid data exfiltration by malicious insiders, compromised code, and accidental oversharing.

What should they do?

- A. Configure Private Google Access for on-premises only.
- B. Perform the following tasks: 1. Create a service account. 2. Give the BigQuery JobUser role and Storage Reader role to the service account. 3. Remove all other IAM access from the project.
- C. Configure VPC Service Controls and configure Private Google Access.
- D. Configure Private Google Access.

You are working at an institution that processes medical data. You are migrating several workloads onto Google Cloud. Company policies require all workloads to run on physically separated hardware, and workloads from different clients must also be separated. You created a sole-tenant node group and added a node for each client. You need to deploy the workloads on these dedicated hosts. What should you do?

- A. Add the node group name as a network tag when creating Compute Engine instances in order to host each workload on the correct node group.
- B. Add the node name as a network tag when creating Compute Engine instances in order to host each workload on the correct node.
- C. Use node affinity labels based on the node group name when creating Compute Engine instances in order to host each workload on the correct node group.
- D. Use node affinity labels based on the node name when creating Compute Engine instances in order to host each workload on the correct node.

Your company's test suite is a custom C++ application that runs tests throughout each day on Linux virtual machines. The full test suite takes several hours to complete, running on a limited number of on-premises servers reserved for testing. Your company wants to move the testing infrastructure to the cloud, to reduce the amount of time it takes to fully test a change to the system, while changing the tests as little as possible. Which cloud infrastructure should you recommend?

- A. Google Compute Engine unmanaged instance groups and Network Load Balancer
- B. Google Compute Engine managed instance groups with auto-scaling
- C. Google Cloud Dataproc to run Apache Hadoop jobs to process each test
- D. Google App Engine with Google StackDriver for logging

A lead software engineer tells you that his new application design uses websockets and HTTP sessions that are not distributed across the web servers. You want to help him ensure his application will run properly on Google Cloud Platform.

What should you do?

- A. Help the engineer to convert his websocket code to use HTTP streaming
- B. Review the encryption requirements for websocket connections with the security team
- C. Meet with the cloud operations team and the engineer to discuss load balancer options
- D. Help the engineer redesign the application to use a distributed user session service that does not rely on websockets and HTTP sessions.

The application reliability team at your company this added a debug feature to their backend service to send all server events to Google Cloud Storage for eventual analysis. The event records are at least 50 KB and at most 15 MB and are expected to peak at 3,000 events per second. You want to minimize data loss.

Which process should you implement?

- A. Append metadata to file body Compress individual files Name files with serverName " Timestamp Create a new bucket if bucket is older than 1 hour and save individual files to the new bucket. Otherwise, save files to existing bucket.
- B. Batch every 10,000 events with a single manifest file for metadata Compress event files and manifest file into a single archive file Name files using serverName " EventSequence Create a new bucket if bucket is older than 1 day and save the single archive file to the new bucket. Otherwise, save the single archive file to existing bucket.
- C. Compress individual files Name files with serverName " EventSequence Save files to one bucket Set custom metadata headers for each object after saving
- D. Append metadata to file body Compress individual files Name files with a random prefix pattern Save files to one bucket

A recent audit revealed that a new network was created in your GCP project. In this network, a GCE instance has an SSH port open to the world. You want to discover this network's origin.

What should you do?

- A. Search for Create VM entry in the Stackdriver alerting console
- B. Navigate to the Activity page in the Home section. Set category to Data Access and search for Create VM entry
- C. In the Logging section of the console, specify GCE Network as the logging section. Search for the Create Insert entry
- D. Connect to the GCE instance using project SSH keys. Identify previous logins in system logs, and match these with the project owners list

You want to make a copy of a production Linux virtual machine in the US-Central region. You want to manage and replace the copy easily if there are changes on the production virtual machine. You will deploy the copy as a new instance in a different project in the US-East region.

What steps must you take?

- A. Use the Linux `dd` and `netcat` commands to copy and stream the root disk contents to a new virtual machine instance in the US-East region.
- B. Create a snapshot of the root disk and select the snapshot as the root disk when you create a new virtual machine instance in the US-East region.
- C. Create an image file from the root disk with Linux `dd` command, create a new virtual machine instance in the US-East region
- D. Create a snapshot of the root disk, create an image file in Google Cloud Storage from the snapshot, and create a new virtual machine instance in the US-East region using the image file the root disk.

Your company runs several databases on a single MySQL instance. They need to take backups of a specific database at regular intervals. The backup activity needs to complete as quickly as possible and cannot be allowed to impact disk performance.

How should you configure the storage?

- A. Configure a cron job to use the `gcloud` tool to take regular backups using persistent disk snapshots.
- B. Mount a Local SSD volume as the backup location. After the backup is complete, use `gsutil` to move the backup to Google Cloud Storage.
- C. Use `gcsfuse` to mount a Google Cloud Storage bucket as a volume directly on the instance and write backups to the mounted location using `mysqldump`.
- D. Mount additional persistent disk volumes onto each virtual machine (VM) instance in a RAID10 array and use LVM to create snapshots to send to Cloud Storage

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

Bigtable.

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

- A. Ensure that the load tests validate the performance of Cloud Bigtable
- B. Create a separate Google Cloud project to use for the load-testing environment
- C. Schedule the load-testing tool to regularly run against the production environment
- D. Ensure all third-party systems your services use is capable of handling high load
- E. Instrument the production services to record every transaction for replay by the load-testing tool
- F. Instrument the load-testing tool and the target services with detailed logging and metrics collection

Your customer is moving their corporate applications to Google Cloud Platform. The security team wants detailed visibility of all projects in the organization. You provision the Google Cloud Resource Manager and set up yourself as the org admin.

What Google Cloud Identity and Access Management (Cloud IAM) roles should you give to the security team?

- A. Org viewer, project owner
- B. Org viewer, project viewer
- C. Org admin, project browser
- D. Project owner, network admin

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

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

- A. Ensure every code check-in is peer reviewed by a security SME
- B. Use source code security analyzers as part of the CI/CD pipeline
- C. Ensure you have stubs to unit test all interfaces between components
- D. Enable code signing and a trusted binary repository integrated with your CI/CD pipeline
- E. Run a vulnerability security scanner as part of your continuous-integration /continuous-delivery (CI/CD) pipeline

You want to enable your running Google Kubernetes Engine cluster to scale as demand for your application changes.

What should you do?

- A. Add additional nodes to your Kubernetes Engine cluster using the following command: `gcloud container clusters resize CLUSTER_Name --size 10`
- B. Add a tag to the instances in the cluster with the following command: `gcloud compute instances add-tags INSTANCE --tags enable-autoscaling max-nodes=10`
- C. Update the existing Kubernetes Engine cluster with the following command: `gcloud alpha container clusters update mycluster --enable-autoscaling --min-nodes=1 --max-nodes=10`
- D. Create a new Kubernetes Engine cluster with the following command: `gcloud alpha container clusters create mycluster --enable-autoscaling --min-nodes=1 --max-nodes=10` and redeploy your application

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

Which infrastructure should you recommend? (Choose two.)

- A. Use Google App Engine to serve the website and Google Cloud Datastore to store user data.
- B. Use a Google Container Engine cluster to serve the website and store data to persistent disk.
- C. Use a managed instance group to serve the website and Google Cloud Bigtable to store user data.
- D. Use a single Compute Engine virtual machine (VM) to host a web server, backend by Google Cloud SQL.

Your company just finished a rapid lift and shift to Google Compute Engine for your compute needs. You have another 9 months to design and deploy a more cloud-native solution. Specifically, you want a system that is no-ops and auto-scaling.

Which two compute products should you choose? (Choose two.)

- A. Compute Engine with containers
- B. Google Kubernetes Engine with containers
- C. Google App Engine Standard Environment
- D. Compute Engine with custom instance types
- E. Compute Engine with managed instance groups

One of your primary business objectives is being able to trust the data stored in your application. You want to log all changes to the application data.

How can you design your logging system to verify authenticity of your logs?

- A. Write the log concurrently in the cloud and on premises
- B. Use a SQL database and limit who can modify the log table
- C. Digitally sign each timestamp and log entry and store the signature
- D. Create a JSON dump of each log entry and store it in Google Cloud Storage

Your company has a Google Workspace account and Google Cloud Organization. Some developers in the company have created Google Cloud projects outside of the Google Cloud Organization.

You want to create an Organization structure that allows developers to create projects, but prevents them from modifying production projects. You want to manage policies for all projects centrally and be able to set more restrictive policies for production projects.

You want to minimize disruption to users and developers when business needs change in the future. You want to follow Google-recommended practices. Now should you design the Organization structure?

- A. 1. Create a second Google Workspace account and Organization. 2. Grant all developers the Project Creator IAM role on the new Organization. 3. Move the developer projects into the new Organization. 4. Set the policies for all projects on both Organizations. 5. Additionally, set the production policies on the original Organization.
- B. 1. Create a folder under the Organization resource named "Production". 2. Grant all developers the Project Creator IAM role on the new Organization. 3. Move the developer projects into the new Organization. 4. Set the policies for all projects on the Organization. 5. Additionally, set the production policies on the "Production" folder.
- C. 1. Create folders under the Organization resource named "Development" and "Production". 2. Grant all developers the Project Creator IAM role on the "Development" folder. 3. Move the developer projects into the "Development" folder. 4. Set the policies for all projects on the Organization. 5. Additionally, set the production policies on the "Production" folder.
- D. 1. Designate the Organization for production projects only. 2. Ensure that developers do not have the Project Creator IAM role on the Organization. 3. Create development projects outside of the Organization using the developer Google Workspace accounts. 4. Set the policies for all projects on the Organization. 5. Additionally, set the production policies on the individual production projects.

Question #172

Topic 1

Your company has an application running on Compute Engine that allows users to play their favorite music. There are a fixed number of instances. Files are stored in Cloud Storage, and data is streamed directly to users. Users are reporting that they sometimes need to attempt to play popular songs multiple times before they are successful. You need to improve the performance of the application. What should you do?

- A. 1. Mount the Cloud Storage bucket using gcsfuse on all backend Compute Engine instances. 2. Serve music files directly from the backend Compute Engine instance.
- B. 1. Create a Cloud Filestore NFS volume and attach it to the backend Compute Engine instances. 2. Download popular songs in Cloud Filestore. 3. Serve music files directly from the backend Compute Engine instance.
- C. 1. Copy popular songs into CloudSQL as a blob. 2. Update application code to retrieve data from CloudSQL when Cloud Storage is overloaded.
- D. 1. Create a managed instance group with Compute Engine instances. 2. Create a global load balancer and configure it with two backends: a Managed instance group and a Cloud Storage bucket. 3. Enable Cloud CDN on the bucket backend.

Question #173

Topic 1

The operations team in your company wants to save Cloud VPN log events for one year. You need to configure the cloud infrastructure to save the logs. What should you do?

- A. Set up a filter in Cloud Logging and a Cloud Storage bucket as an export target for the logs you want to save.
- B. Enable the Compute Engine API, and then enable logging on the firewall rules that match the traffic you want to save.
- C. Set up a Cloud Logging Dashboard titled Cloud VPN Logs, and then add a chart that queries for the VPN metrics over a one-year time period.
- D. Set up a filter in Cloud Logging and a topic in Pub/Sub to publish the logs.

You are working with a data warehousing team that performs data analysis. The team needs to process data from external partners, but the data contains personally identifiable information (PII). You need to process and store the data without storing any of the PII data. What should you do?

- A. Create a Dataflow pipeline to retrieve the data from the external sources. As part of the pipeline, use the Cloud Data Loss Prevention (Cloud DLP) API to remove any PII data. Store the result in BigQuery.
- B. Create a Dataflow pipeline to retrieve the data from the external sources. As part of the pipeline, store all non-PII data in BigQuery and store all PII data in a Cloud Storage bucket that has a retention policy set.
- C. Ask the external partners to upload all data on Cloud Storage. Configure Bucket Lock for the bucket. Create a Dataflow pipeline to read the data from the bucket. As part of the pipeline, use the Cloud Data Loss Prevention (Cloud DLP) API to remove any PII data. Store the result in BigQuery.
- D. Ask the external partners to import all data in your BigQuery dataset. Create a dataflow pipeline to copy the data into a new table. As part of the Dataflow bucket, skip all data in columns that have PII data

You want to allow your operations team to store logs from all the production projects in your Organization, without including logs from other projects. All of the production projects are contained in a folder. You want to ensure that all logs for existing and new production projects are captured automatically. What should you do?

- A. Create an aggregated export on the Production folder. Set the log sink to be a Cloud Storage bucket in an operations project.
- B. Create an aggregated export on the Organization resource. Set the log sink to be a Cloud Storage bucket in an operations project.
- C. Create log exports in the production projects. Set the log sinks to be a Cloud Storage bucket in an operations project.
- D. Create log exports in the production projects. Set the log sinks to be BigQuery datasets in the production projects, and grant IAM access to the operations team to run queries on the datasets.

Your company has an application that is running on multiple instances of Compute Engine. It generates 1 TB per day of logs. For compliance reasons, the logs need to be kept for at least two years. The logs need to be available for active query for 30 days. After that, they just need to be retained for audit purposes. You want to implement a storage solution that is compliant, minimizes costs, and follows Google-recommended practices. What should you do?

- A. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month. 4. Configure a retention policy at the bucket level using bucket lock.
- B. 1. Write a daily cron job, running on all instances, that uploads logs into a Cloud Storage bucket. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month.
- C. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a partitioned BigQuery table. 3. Set a `time_partitioning_expiration` of 30 days.
- D. 1. Create a daily cron job, running on all instances, that uploads logs into a partitioned BigQuery table. 2. Set a `time_partitioning_expiration` of 30 days.

Your company has just recently activated Cloud Identity to manage users. The Google Cloud Organization has been configured as well. The security team needs to secure projects that will be part of the Organization. They want to prohibit IAM users outside the domain from gaining permissions from now on. What should they do?

- A. Configure an organization policy to restrict identities by domain.
- B. Configure an organization policy to block creation of service accounts.
- C. Configure Cloud Scheduler to trigger a Cloud Function every hour that removes all users that don't belong to the Cloud Identity domain from all projects.
- D. Create a technical user (e.g., crawler@yourdomain.com), and give it the project owner role at root organization level. Write a bash script that:
 - λ€ç Lists all the IAM rules of all projects within the organization.
 - λ€ç Deletes all users that do not belong to the company domain.Create a Compute Engine instance in a project within the Organization and configure gcloud to be executed with technical user credentials. Configure a cron job that executes the bash script every hour.

Question #178

Topic 1

Your company has an application running on Google Cloud that is collecting data from thousands of physical devices that are globally distributed. Data is published to Pub/Sub and streamed in real time into an SSD Cloud Bigtable cluster via a Dataflow pipeline. The operations team informs you that your Cloud Bigtable cluster has a hotspot, and queries are taking longer than expected. You need to resolve the problem and prevent it from happening in the future. What should you do?

- A. Advise your clients to use HBase APIs instead of NodeJS APIs.
- B. Delete records older than 30 days.
- C. Review your RowKey strategy and ensure that keys are evenly spread across the alphabet.
- D. Double the number of nodes you currently have.

Question #179

Topic 1

Your company has a Google Cloud project that uses BigQuery for data warehousing. There are some tables that contain personally identifiable information (PII).

Only the compliance team may access the PII. The other information in the tables must be available to the data science team. You want to minimize cost and the time it takes to assign appropriate access to the tables. What should you do?

- A. 1. From the dataset where you have the source data, create views of tables that you want to share, excluding PII. 2. Assign an appropriate project-level IAM role to the members of the data science team. 3. Assign access controls to the dataset that contains the view.
- B. 1. From the dataset where you have the source data, create materialized views of tables that you want to share, excluding PII. 2. Assign an appropriate project-level IAM role to the members of the data science team. 3. Assign access controls to the dataset that contains the view.
- C. 1. Create a dataset for the data science team. 2. Create views of tables that you want to share, excluding PII. 3. Assign an appropriate project-level IAM role to the members of the data science team. 4. Assign access controls to the dataset that contains the view. 5. Authorize the view to access the source dataset.
- D. 1. Create a dataset for the data science team. 2. Create materialized views of tables that you want to share, excluding PII. 3. Assign an appropriate project-level IAM role to the members of the data science team. 4. Assign access controls to the dataset that contains the view. 5. Authorize the view to access the source dataset.

Your operations team currently stores 10 TB of data in an object storage service from a third-party provider. They want to move this data to a Cloud Storage bucket as quickly as possible, following Google-recommended practices. They want to minimize the cost of this data migration. Which approach should they use?

- A. Use the `gsutil mv` command to move the data.
- B. Use the Storage Transfer Service to move the data.
- C. Download the data to a Transfer Appliance, and ship it to Google.
- D. Download the data to the on-premises data center, and upload it to the Cloud Storage bucket.

You have a Compute Engine managed instance group that adds and removes Compute Engine instances from the group in response to the load on your application. The instances have a shutdown script that removes REDIS database entries associated with the instance. You see that many database entries have not been removed, and you suspect that the shutdown script is the problem. You need to ensure that the commands in the shutdown script are run reliably every time an instance is shut down. You create a Cloud Function to remove the database entries. What should you do next?

- A. Modify the shutdown script to wait for 30 seconds before triggering the Cloud Function.
- B. Do not use the Cloud Function. Modify the shutdown script to restart if it has not completed in 30 seconds.
- C. Set up a Cloud Monitoring sink that triggers the Cloud Function after an instance removal log message arrives in Cloud Logging.
- D. Modify the shutdown script to wait for 30 seconds and then publish a message to a Pub/Sub queue.

You are managing several projects on Google Cloud and need to interact on a daily basis with BigQuery, Bigtable, and Kubernetes Engine using the `gcloud` CL tool. You are travelling a lot and work on different workstations during the week. You want to avoid having to manage the `gcloud` CLI manually. What should you do?

- A. Use Google Cloud Shell in the Google Cloud Console to interact with Google Cloud.
- B. Create a Compute Engine instance and install `gcloud` on the instance. Connect to this instance via SSH to always use the same `gcloud` installation when interacting with Google Cloud.
- C. Install `gcloud` on all of your workstations. Run the command `gcloud components auto-update` on each workstation
- D. Use a package manager to install `gcloud` on your workstations instead of installing it manually.

Your company recently acquired a company that has infrastructure in Google Cloud. Each company has its own Google Cloud organization. Each company is using a Shared Virtual Private Cloud (VPC) to provide network connectivity for its applications. Some of the subnets used by both companies overlap. In order for both businesses to integrate, the applications need to have private network connectivity. These applications are not on overlapping subnets. You want to provide connectivity with minimal re-engineering. What should you do?

- A. Set up VPC peering and peer each Shared VPC together.
- B. Migrate the projects from the acquired company into your company's Google Cloud organization. Re-launch the instances in your company's Shared VPC.
- C. Set up a Cloud VPN gateway in each Shared VPC and peer Cloud VPNs.
- D. Configure SSH port forwarding on each application to provide connectivity between applications in the different Shared VPCs.

You are managing several internal applications that are deployed on Compute Engine. Business users inform you that an application has become very slow over the past few days. You want to find the underlying cause in order to solve the problem. What should you do first?

- A. Inspect the logs and metrics from the instances in Cloud Logging and Cloud Monitoring.
- B. Change the Compute Engine Instances behind the application to a machine type with more CPU and memory.
- C. Restore a backup of the application database from a time before the application became slow.
- D. Deploy the applications on a managed instance group with autoscaling enabled. Add a load balancer in front of the managed instance group, and have the users connect to the IP of the load balancer.

Your company has an application running as a Deployment in a Google Kubernetes Engine (GKE) cluster. When releasing new versions of the application via a rolling deployment, the team has been causing outages. The root cause of the outages is misconfigurations with parameters that are only used in production. You want to put preventive measures for this in the platform to prevent outages. What should you do?

- A. Configure liveness and readiness probes in the Pod specification.
- B. Configure health checks on the managed instance group.
- C. Create a Scheduled Task to check whether the application is available.
- D. Configure an uptime alert in Cloud Monitoring.

Your company uses Google Kubernetes Engine (GKE) as a platform for all workloads. Your company has a single large GKE cluster that contains batch, stateful, and stateless workloads. The GKE cluster is configured with a single node pool with 200 nodes. Your company needs to reduce the cost of this cluster but does not want to compromise availability. What should you do?

- A. Create a second GKE cluster for the batch workloads only. Allocate the 200 original nodes across both clusters.
- B. Configure CPU and memory limits on the namespaces in the cluster. Configure all Pods to have a CPU and memory limits.
- C. Configure a HorizontalPodAutoscaler for all stateless workloads and for all compatible stateful workloads. Configure the cluster to use node auto scaling.
- D. Change the node pool to use preemptible VMs.

Your company has a Google Cloud project that uses BigQuery for data warehousing on a pay-per-use basis. You want to monitor queries in real time to discover the most costly queries and which users spend the most. What should you do?

- A. 1. In the BigQuery dataset that contains all the tables to be queried, add a label for each user that can launch a query. 2. Open the Billing page of the project. 3. Select Reports. 4. Select BigQuery as the product and filter by the user you want to check.
- B. 1. Create a Cloud Logging sink to export BigQuery data access logs to BigQuery. 2. Perform a BigQuery query on the generated table to extract the information you need.
- C. 1. Create a Cloud Logging sink to export BigQuery data access logs to Cloud Storage. 2. Develop a Dataflow pipeline to compute the cost of queries split by users.
- D. 1. Activate billing export into BigQuery. 2. Perform a BigQuery query on the billing table to extract the information you need.

Your company and one of its partners each have a Google Cloud project in separate organizations. Your company's project (prj-a) runs in Virtual Private Cloud (vpc-a). The partner's project (prj-b) runs in vpc-b. There are two instances running on vpc-a and one instance running on vpc-b. Subnets defined in both VPCs are not overlapping. You need to ensure that all instances communicate with each other via internal IPs, minimizing latency and maximizing throughput. What should you do?

- A. Set up a network peering between vpc-a and vpc-b.
- B. Set up a VPN between vpc-a and vpc-b using Cloud VPN.
- C. Configure IAP TCP forwarding on the instance in vpc-b, and then launch the following gcloud command from one of the instances in vpc-a
gcloud: gcloud compute start-iap-tunnel INSTANCE_NAME_IN_VPC_8 22 \ --local-host-port=localhost:22
- D. 1. Create an additional instance in vpc-a. 2. Create an additional instance in vpc-b. 3. Install OpenVPN in newly created instances. 4. Configure a VPN tunnel between vpc-a and vpc-b with the help of OpenVPN.

You want to store critical business information in Cloud Storage buckets. The information is regularly changed, but previous versions need to be referenced on a regular basis. You want to ensure that there is a record of all changes to any information in these buckets. You want to ensure that accidental edits or deletions can be easily rolled back. Which feature should you enable?

- A. Bucket Lock
- B. Object Versioning
- C. Object change notification
- D. Object Lifecycle Management

You have a Compute Engine application that you want to autoscale when total memory usage exceeds 80%. You have installed the Cloud Monitoring agent and configured the autoscaling policy as follows:

↻ Metric identifier: agent.googleapis.com/memory/percent_used

↻ Filter: metric.label.state = 'used'

↻ Target utilization level: 80

↻ Target type: GAUGE

You observe that the application does not scale under high load. You want to resolve this. What should you do?

- A. Change the Target type to DELTA_PER_MINUTE.
- B. Change the Metric identifier to agent.googleapis.com/memory/bytes_used.
- C. Change the filter to metric.label.state = 'used' AND metric.label.state = 'buffered' AND metric.label.state = 'cached' AND metric.label.state = 'slab'.
- D. Change the filter to metric.label.state = 'free' and the Target utilization to 20.

You are deploying an application to Google Cloud. The application is part of a system. The application in Google Cloud must communicate over a private network with applications in a non-Google Cloud environment. The expected average throughput is 200 kbps. The business requires:

☞ as close to 100% system availability as possible

☞ cost optimization

You need to design the connectivity between the locations to meet the business requirements. What should you provision?

- A. An HA Cloud VPN gateway connected with two tunnels to an on-premises VPN gateway
- B. Two Classic Cloud VPN gateways connected to two on-premises VPN gateways Configure each Classic Cloud VPN gateway to have two tunnels, each connected to different on-premises VPN gateways
- C. Two HA Cloud VPN gateways connected to two on-premises VPN gateways Configure each HA Cloud VPN gateway to have two tunnels, each connected to different on-premises VPN gateways
- D. A single Cloud VPN gateway connected to an on-premises VPN gateway

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

Modify applications for the cloud

Move applications to a new infrastructure

Leverage managed services wherever feasible

Sunset 20% of capacity in existing data centers

Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question

The JencoMart security team requires that all Google Cloud Platform infrastructure is deployed using a least privilege model with separation of duties for administration between production and development resources.

What Google domain and project structure should you recommend?

- A. Create two G Suite accounts to manage users: one for development/test/staging and one for production. Each account should contain one project for every application
- B. Create two G Suite accounts to manage users: one with a single project for all development applications and one with a single project for all production applications
- C. Create a single G Suite account to manage users with each stage of each application in its own project
- D. Create a single G Suite account to manage users with one project for the development/test/staging environment and one project for the production environment

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

Modify applications for the cloud
Move applications to a new infrastructure
Leverage managed services wherever feasible
Sunset 20% of capacity in existing data centers
Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question

A few days after JencoMart migrates the user credentials database to Google Cloud Platform and shuts down the old server, the new database server stops responding to SSH connections. It is still serving database requests to the application servers correctly.

What three steps should you take to diagnose the problem? (Choose three.)

- A. Delete the virtual machine (VM) and disks and create a new one
- B. Delete the instance, attach the disk to a new VM, and investigate
- C. Take a snapshot of the disk and connect to a new machine to investigate
- D. Check inbound firewall rules for the network the machine is connected to
- E. Connect the machine to another network with very simple firewall rules and investigate
- F. Print the Serial Console output for the instance for troubleshooting, activate the interactive console, and investigate

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

Modify applications for the cloud
Move applications to a new infrastructure
Leverage managed services wherever feasible
Sunset 20% of capacity in existing data centers
Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question

JencoMart has decided to migrate user profile storage to Google Cloud Datastore and the application servers to Google Compute Engine (GCE). During the migration, the existing infrastructure will need access to Datastore to upload the data.
What service account key-management strategy should you recommend?

- A. Provision service account keys for the on-premises infrastructure and for the GCE virtual machines (VMs)
- B. Authenticate the on-premises infrastructure with a user account and provision service account keys for the VMs
- C. Provision service account keys for the on-premises infrastructure and use Google Cloud Platform (GCP) managed keys for the VMs
- D. Deploy a custom authentication service on GCE/Google Kubernetes Engine (GKE) for the on-premises infrastructure and use GCP managed keys for the VMs

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

Modify applications for the cloud
Move applications to a new infrastructure
Leverage managed services wherever feasible
Sunset 20% of capacity in existing data centers
Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question

JencoMart has built a version of their application on Google Cloud Platform that serves traffic to Asia. You want to measure success against their business and technical goals.

Which metrics should you track?

- A. Error rates for requests from Asia
- B. Latency difference between US and Asia
- C. Total visits, error rates, and latency from Asia
- D. Total visits and average latency for users from Asia
- E. The number of character sets present in the database

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

- Modify applications for the cloud
- Move applications to a new infrastructure
- Leverage managed services wherever feasible
- Sunset 20% of capacity in existing data centers
- Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

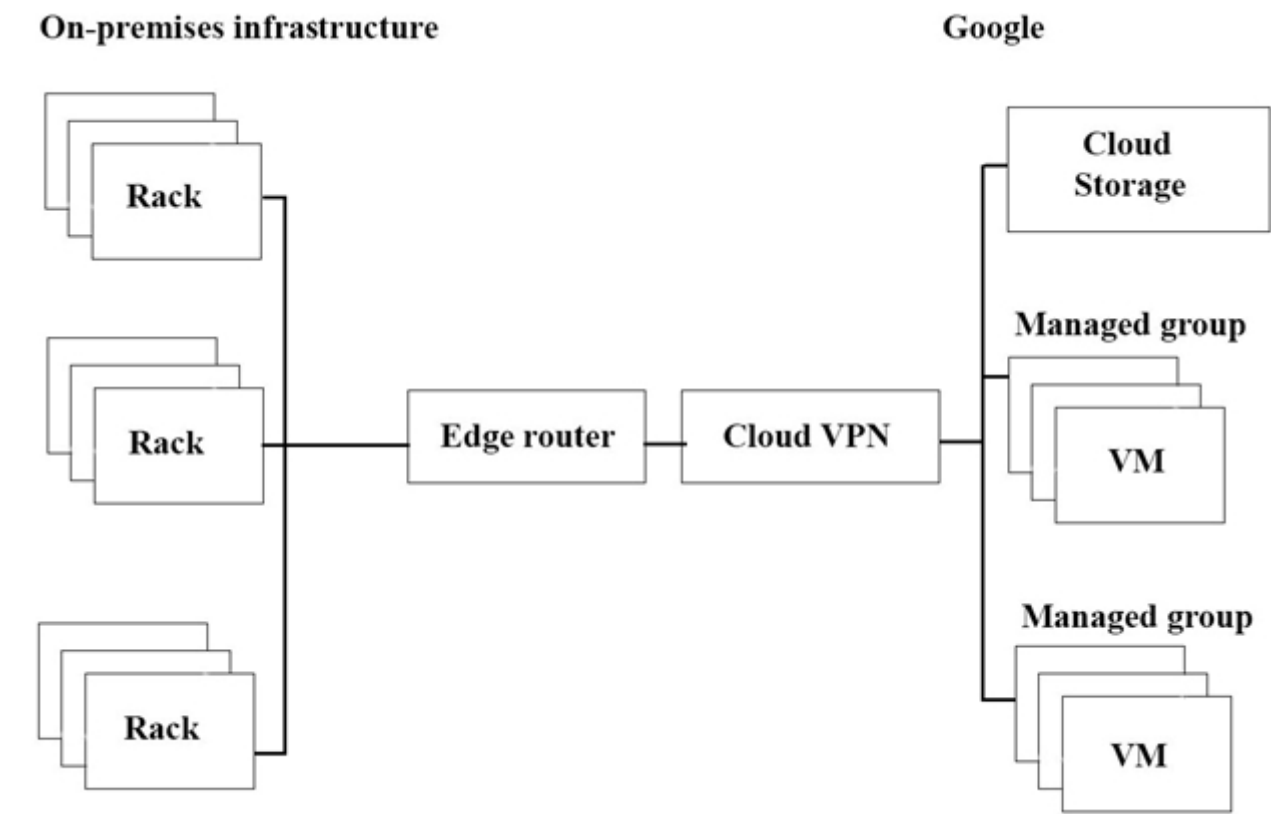
CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question



The migration of JencoMart's application to Google Cloud Platform (GCP) is progressing too slowly. The infrastructure is shown in the diagram. You want to maximize throughput.

What are three potential bottlenecks? (Choose three.)

- A. A single VPN tunnel, which limits throughput
- B. A tier of Google Cloud Storage that is not suited for this task
- C. A copy command that is not suited to operate over long distances
- D. Fewer virtual machines (VMs) in GCP than on-premises machines
- E. A separate storage layer outside the VMs, which is not suited for this task
- F. Complicated internet connectivity between the on-premises infrastructure and GCP

Introductory Info

Company Overview -

JencoMart is a global retailer with over 10,000 stores in 16 countries. The stores carry a range of goods, such as groceries, tires, and jewelry. One of the company's core values is excellent customer service. In addition, they recently introduced an environmental policy to reduce their carbon output by 50% over the next 5 years.

Company Background -

JencoMart started as a general store in 1931, and has grown into one of the world's leading brands, known for great value and customer service. Over time, the company transitioned from only physical stores to a stores and online hybrid model, with 25% of sales online. Currently, JencoMart has little presence in Asia, but considers that market key for future growth.

Solution Concept -

JencoMart wants to migrate several critical applications to the cloud but has not completed a technical review to determine their suitability for the cloud and the engineering required for migration. They currently host all of these applications on infrastructure that is at its end of life and is no longer supported.

Existing Technical Environment -

JencoMart hosts all of its applications in 4 data centers: 3 in North American and 1 in Europe; most applications are dual-homed.

JencoMart understands the dependencies and resource usage metrics of their on-premises architecture.

Application: Customer loyalty portal

LAMP (Linux, Apache, MySQL and PHP) application served from the two JencoMart-owned U.S. data centers.

Database -

Oracle Database stores user profiles

- 20 TB
- Complex table structure
- Well maintained, clean data
- Strong backup strategy

PostgreSQL database stores user credentials

- Single-homed in US West
- No redundancy
- Backed up every 12 hours
- 100% uptime service level agreement (SLA)
- Authenticates all users

Compute -

30 machines in US West Coast, each machine has:

- Twin, dual core CPUs
- 32 GB of RAM
- Twin 250 GB HDD (RAID 1)

20 machines in US East Coast, each machine has:

- Single, dual-core CPU
- 24 GB of RAM
- Twin 250 GB HDD (RAID 1)

Storage -

Access to shared 100 TB SAN in each location

Tape backup every week

Business Requirements -

Optimize for capacity during peak periods and value during off-peak periods

Guarantee service availability and support

Reduce on-premises footprint and associated financial and environmental impact

Move to outsourcing model to avoid large upfront costs associated with infrastructure purchase

Expand services into Asia

Technical Requirements -

Assess key application for cloud suitability

Modify applications for the cloud
Move applications to a new infrastructure
Leverage managed services wherever feasible
Sunset 20% of capacity in existing data centers
Decrease latency in Asia

CEO Statement -

JencoMart will continue to develop personal relationships with our customers as more people access the web. The future of our retail business is in the global market and the connection between online and in-store experiences. As a large, global company, we also have a responsibility to the environment through `green` initiatives and policies.

CTO Statement -

The challenges of operating data centers prevent focus on key technologies critical to our long-term success. Migrating our data services to a public cloud infrastructure will allow us to focus on big data and machine learning to improve our service to customers.

CFO Statement -

Since its founding, JencoMart has invested heavily in our data services infrastructure. However, because of changing market trends, we need to outsource our infrastructure to ensure our long-term success. This model will allow us to respond to increasing customer demand during peak periods and reduce costs.

Question

JencoMart wants to move their User Profiles database to Google Cloud Platform.

Which Google Database should they use?

- A. Cloud Spanner
- B. Google BigQuery
- C. Google Cloud SQL
- D. Google Cloud Datastore

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. Your team is in charge of creating a payment card data vault for card numbers used to bill tens of thousands of viewers, merchandise consumers, and season ticket holders. You need to implement a custom card tokenization service that meets the following requirements:

* It must provide low latency at minimal cost.

* It must be able to identify duplicate credit cards and must not store plaintext card numbers.

* It should support annual key rotation.

Which storage approach should you adopt for your tokenization service?

- A. Store the card data in Secret Manager after running a query to identify duplicates.
- B. Encrypt the card data with a deterministic algorithm stored in Firestore using Datastore mode.
- C. Encrypt the card data with a deterministic algorithm and shard it across multiple Memorystore instances.
- D. Use column-level encryption to store the data in Cloud SQL.

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. Recently HRL started a new regional racing league in Cape Town, South Africa. In an effort to give customers in Cape Town a better user experience, HRL has partnered with the Content Delivery Network provider, Fastly. HRL needs to allow traffic coming from all of the Fastly IP address ranges into their Virtual Private Cloud network (VPC network). You are a member of the HRL security team and you need to configure the update that will allow only the Fastly IP address ranges through the External HTTP(S) load balancer. Which command should you use?

A.

```
gcloud compute security-policies rules update 1000 \
  --security-policy from-fastly \
  --src-ip-ranges * \
  --action "allow"
```

B.

```
gcloud compute firewall rules update sourceiplist-fastly \  
  --priority 1000 \  
  --allow tcp:443
```

C.

```
gcloud compute firewall rules update hlr-policy \  
  --priority 1000 \  
  --target-tags=sourceiplist-fastly \  
  --allow tcp:443
```

D.

```
gcloud compute security-policies rules update 1000 \  
  --security-policy hlr-policy \  
  --expression "evaluatePreconfiguredExpr('sourceiplist-fastly')" \  
  --action "allow"
```

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. The HRL development team releases a new version of their predictive capability application every Tuesday evening at 3 a.m. UTC to a repository. The security team at HRL has developed an in-house penetration test Cloud Function called

Airwolf. The security team wants to run Airwolf against the predictive capability application as soon as it is released every Tuesday. You need to set up Airwolf to run at the recurring weekly cadence. What should you do?

- A. Set up Cloud Tasks and a Cloud Storage bucket that triggers a Cloud Function.
- B. Set up a Cloud Logging sink and a Cloud Storage bucket that triggers a Cloud Function.
- C. Configure the deployment job to notify a Pub/Sub queue that triggers a Cloud Function.

D. Set up Identity and Access Management (IAM) and Confidential Computing to trigger a Cloud Function.

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. HRL wants better prediction accuracy from their ML prediction models. They want you to use Google's AI Platform so HRL can understand and interpret the predictions. What should you do?

- A. Use Explainable AI.
- B. Use Vision AI.
- C. Use Google Cloud's operations suite.
- D. Use Jupyter Notebooks.

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. HRL is looking for a cost-effective approach for storing their race data such as telemetry. They want to keep all historical records, train models using only the previous season's data, and plan for data growth in terms of volume and information collected. You need to propose a data solution. Considering HRL business requirements and the goals expressed by CEO S. Hawke, what should you do?

- A. Use Firestore for its scalable and flexible document-based database. Use collections to aggregate race data by season and event.
- B. Use Cloud Spanner for its scalability and ability to version schemas with zero downtime. Split race data using season as a primary key.
- C. Use BigQuery for its scalability and ability to add columns to a schema. Partition race data based on season.

D. Use Cloud SQL for its ability to automatically manage storage increases and compatibility with MySQL. Use separate database instances for each season.

Introductory Info

Company overview -

Helicopter Racing League (HRL) is a global sports league for competitive helicopter racing. Each year HRL holds the world championship and several regional league competitions where teams compete to earn a spot in the world championship. HRL offers a paid service to stream the races all over the world with live telemetry and predictions throughout each race.

Solution concept -

HRL wants to migrate their existing service to a new platform to expand their use of managed AI and ML services to facilitate race predictions. Additionally, as new fans engage with the sport, particularly in emerging regions, they want to move the serving of their content, both real-time and recorded, closer to their users.

Existing technical environment -

HRL is a public cloud-first company; the core of their mission-critical applications runs on their current public cloud provider. Video recording and editing is performed at the race tracks, and the content is encoded and transcoded, where needed, in the cloud. Enterprise-grade connectivity and local compute is provided by truck-mounted mobile data centers. Their race prediction services are hosted exclusively on their existing public cloud provider. Their existing technical environment is as follows:

Existing content is stored in an object storage service on their existing public cloud provider.

Video encoding and transcoding is performed on VMs created for each job.

Race predictions are performed using TensorFlow running on VMs in the current public cloud provider.

Business requirements -

HRL's owners want to expand their predictive capabilities and reduce latency for their viewers in emerging markets. Their requirements are:

Support ability to expose the predictive models to partners.

Increase predictive capabilities during and before races:

λ→ Race results

λ→ Mechanical failures

λ→ Crowd sentiment

Increase telemetry and create additional insights.

Measure fan engagement with new predictions.

Enhance global availability and quality of the broadcasts.

Increase the number of concurrent viewers.

Minimize operational complexity.

Ensure compliance with regulations.

Create a merchandising revenue stream.

Technical requirements -

Maintain or increase prediction throughput and accuracy.

Reduce viewer latency.

Increase transcoding performance.

Create real-time analytics of viewer consumption patterns and engagement.

Create a data mart to enable processing of large volumes of race data.

Executive statement -

Our CEO, S. Hawke, wants to bring high-adrenaline racing to fans all around the world. We listen to our fans, and they want enhanced video streams that include predictions of events within the race (e.g., overtaking). Our current platform allows us to predict race outcomes but lacks the facility to support real-time predictions during races and the capacity to process season-long results.

Question

For this question, refer to the Helicopter Racing League (HRL) case study. A recent finance audit of cloud infrastructure noted an exceptionally high number of

Compute Engine instances are allocated to do video encoding and transcoding. You suspect that these Virtual Machines are zombie machines that were not deleted after their workloads completed. You need to quickly get a list of which VM instances are idle. What should you do?

- A. Log into each Compute Engine instance and collect disk, CPU, memory, and network usage statistics for analysis.
- B. Use the gcloud compute instances list to list the virtual machine instances that have the idle: true label set.
- C. Use the gcloud recommender command to list the idle virtual machine instances.

D. From the Google Console, identify which Compute Engine instances in the managed instance groups are no longer responding to health check probes.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

For this question, refer to the EHR Healthcare case study. You are responsible for ensuring that EHR's use of Google Cloud will pass an upcoming privacy compliance audit. What should you do? (Choose two.)

- A. Verify EHR's product usage against the list of compliant products on the Google Cloud compliance page.
- B. Advise EHR to execute a Business Associate Agreement (BAA) with Google Cloud.
- C. Use Firebase Authentication for EHR's user facing applications.
- D. Implement Prometheus to detect and prevent security breaches on EHR's web-based applications.

E. Use GKE private clusters for all Kubernetes workloads.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

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

- A. Enable Binary Authorization on GKE, and sign containers as part of a CI/CD pipeline.
- B. Configure Jenkins to utilize Kritis to cryptographically sign a container as part of a CI/CD pipeline.
- C. Configure Container Registry to only allow trusted service accounts to create and deploy containers from the registry.
- D. Configure Container Registry to use vulnerability scanning to confirm that there are no vulnerabilities before deploying the workload.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

You need to upgrade the EHR connection to comply with their requirements. The new connection design must support business-critical needs and meet the same network and security policy requirements. What should you do?

- A. Add a new Dedicated Interconnect connection.
- B. Upgrade the bandwidth on the Dedicated Interconnect connection to 100 G.
- C. Add three new Cloud VPN connections.
- D. Add a new Carrier Peering connection.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

For this question, refer to the EHR Healthcare case study. You need to define the technical architecture for hybrid connectivity between EHR's on-premises systems and Google Cloud. You want to follow Google's recommended practices for production-level applications. Considering the EHR Healthcare business and technical requirements, what should you do?

- A. Configure two Partner Interconnect connections in one metro (City), and make sure the Interconnect connections are placed in different metro zones.
- B. Configure two VPN connections from on-premises to Google Cloud, and make sure the VPN devices on-premises are in separate racks.
- C. Configure Direct Peering between EHR Healthcare and Google Cloud, and make sure you are peering at least two Google locations.
- D. Configure two Dedicated Interconnect connections in one metro (City) and two connections in another metro, and make sure the Interconnect connections are placed in different metro zones.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

For this question, refer to the EHR Healthcare case study. You are a developer on the EHR customer portal team. Your team recently migrated the customer portal application to Google Cloud. The load has increased on the application servers, and now the application is logging many timeout errors. You recently incorporated Pub/Sub into the application architecture, and the application is not logging any Pub/Sub publishing errors. You want to improve publishing latency.

What should you do?

- A. Increase the Pub/Sub Total Timeout retry value.
- B. Move from a Pub/Sub subscriber pull model to a push model.
- C. Turn off Pub/Sub message batching.
- D. Create a backup Pub/Sub message queue.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

For this question, refer to the EHR Healthcare case study. In the past, configuration errors put public IP addresses on backend servers that should not have been accessible from the Internet. You need to ensure that no one can put external IP addresses on backend Compute Engine instances and that external IP addresses can only be configured on frontend Compute Engine instances. What should you do?

- A. Create an Organizational Policy with a constraint to allow external IP addresses only on the frontend Compute Engine instances.
- B. Revoke the compute.networkAdmin role from all users in the project with front end instances.
- C. Create an Identity and Access Management (IAM) policy that maps the IT staff to the compute.networkAdmin role for the organization.
- D. Create a custom Identity and Access Management (IAM) role named GCE_FRONTEND with the compute.addresses.create permission.

Introductory Info

Company overview -

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi- national medical offices, hospitals, and insurance providers.

Solution concept -

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google

Cloud has been chosen to replace their current colocation facilities.

Existing technical environment -

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Business requirements -

- * On-board new insurance providers as quickly as possible.
- * Provide a minimum 99.9% availability for all customer-facing systems.
- * Provide centralized visibility and proactive action on system performance and usage.
- * Increase ability to provide insights into healthcare trends.
- * Reduce latency to all customers.
- * Maintain regulatory compliance.
- * Decrease infrastructure administration costs.
- * Make predictions and generate reports on industry trends based on provider data.

Technical requirements -

- * Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers.
- * Provide a consistent way to manage customer-facing applications that are container-based.
- * Provide a secure and high-performance connection between on-premises systems and Google Cloud.
- * Provide consistent logging, log retention, monitoring, and alerting capabilities.
- * Maintain and manage multiple container-based environments.
- * Dynamically scale and provision new environments.
- * Create interfaces to ingest and process data from new providers.

Executive statement -

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

Question

For this question, refer to the EHR Healthcare case study. You are responsible for designing the Google Cloud network architecture for Google Kubernetes

Engine. You want to follow Google best practices. Considering the EHR Healthcare business and technical requirements, what should you do to reduce the attack surface?

- A. Use a private cluster with a private endpoint with master authorized networks configured.
- B. Use a public cluster with firewall rules and Virtual Private Cloud (VPC) routes.
- C. Use a private cluster with a public endpoint with master authorized networks configured.

D. Use a public cluster with master authorized networks enabled and firewall rules.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

1. Dynamically scale up or down based on game activity

2. Connect to a managed NoSQL database service

3. Run customize Linux distro

Requirements for Game Analytics Platform

1. Dynamically scale up or down based on game activity

2. Process incoming data on the fly directly from the game servers

3. Process data that arrives late because of slow mobile networks

4. Allow SQL queries to access at least 10 TB of historical data

5. Process files that are regularly uploaded by users' mobile devices

6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation.

Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games wants you to design their new testing strategy. How should the test coverage differ from their existing backends on the other platforms?

- A. Tests should scale well beyond the prior approaches
- B. Unit tests are no longer required, only end-to-end tests
- C. Tests should be applied after the release is in the production environment
- D. Tests should include directly testing the Google Cloud Platform (GCP) infrastructure

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

1. Dynamically scale up or down based on game activity
2. Connect to a managed NoSQL database service
3. Run customize Linux distro

Requirements for Game Analytics Platform

1. Dynamically scale up or down based on game activity
2. Process incoming data on the fly directly from the game servers
3. Process data that arrives late because of slow mobile networks
4. Allow SQL queries to access at least 10 TB of historical data
5. Process files that are regularly uploaded by users' mobile devices
6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games has deployed their new backend on Google Cloud Platform (GCP). You want to create a through testing process for new versions of the backend before they are released to the public. You want the testing environment to scale in an economical way. How should you design the process?

- A. Create a scalable environment in GCP for simulating production load
- B. Use the existing infrastructure to test the GCP-based backend at scale
- C. Build stress tests into each component of your application using resources internal to GCP to simulate load

D. Create a set of static environments in GCP to test different levels of load λ for example, high, medium, and low

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

- 1. Dynamically scale up or down based on game activity
- 2. Connect to a managed NoSQL database service
- 3. Run customize Linux distro

Requirements for Game Analytics Platform

- 1. Dynamically scale up or down based on game activity
- 2. Process incoming data on the fly directly from the game servers
- 3. Process data that arrives late because of slow mobile networks
- 4. Allow SQL queries to access at least 10 TB of historical data
- 5. Process files that are regularly uploaded by users' mobile devices
- 6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games wants to set up a continuous delivery pipeline. Their architecture includes many small services that they want to be able to update and roll back quickly. Mountkirk Games has the following requirements:

- ☞ Services are deployed redundantly across multiple regions in the US and Europe
- ☞ Only frontend services are exposed on the public internet
- ☞ They can provide a single frontend IP for their fleet of services
- ☞ Deployment artifacts are immutable

Which set of products should they use?

- A. Google Cloud Storage, Google Cloud Dataflow, Google Compute Engine
- B. Google Cloud Storage, Google App Engine, Google Network Load Balancer

C. Google Kubernetes Registry, Google Container Engine, Google HTTP(S) Load Balancer

D. Google Cloud Functions, Google Cloud Pub/Sub, Google Cloud Deployment Manager

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

1. Dynamically scale up or down based on game activity
2. Connect to a managed NoSQL database service
3. Run customize Linux distro

Requirements for Game Analytics Platform

1. Dynamically scale up or down based on game activity
2. Process incoming data on the fly directly from the game servers
3. Process data that arrives late because of slow mobile networks
4. Allow SQL queries to access at least 10 TB of historical data
5. Process files that are regularly uploaded by users' mobile devices
6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games' gaming servers are not automatically scaling properly. Last month, they rolled out a new feature, which suddenly became very popular. A record number of users are trying to use the service, but many of them are getting 503 errors and very slow response times. What should they investigate first?

- A. Verify that the database is online
- B. Verify that the project quota hasn't been exceeded
- C. Verify that the new feature code did not introduce any performance bugs

D. Verify that the load-testing team is not running their tool against production

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

1. Dynamically scale up or down based on game activity
2. Connect to a managed NoSQL database service
3. Run customize Linux distro

Requirements for Game Analytics Platform

1. Dynamically scale up or down based on game activity
2. Process incoming data on the fly directly from the game servers
3. Process data that arrives late because of slow mobile networks
4. Allow SQL queries to access at least 10 TB of historical data
5. Process files that are regularly uploaded by users' mobile devices
6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games needs to create a repeatable and configurable mechanism for deploying isolated application environments. Developers and testers can access each other's environments and resources, but they cannot access staging or production resources. The staging environment needs access to some services from production.

What should you do to isolate development environments from staging and production?

- A. Create a project for development and test and another for staging and production
- B. Create a network for development and test and another for staging and production
- C. Create one subnetwork for development and another for staging and production
- D. Create one project for development, a second for staging and a third for production

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for the most popular mobile platforms. They build all of their games using some server-side integration. Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

1. Dynamically scale up or down based on game activity
2. Connect to a managed NoSQL database service
3. Run customize Linux distro

Requirements for Game Analytics Platform

1. Dynamically scale up or down based on game activity
2. Process incoming data on the fly directly from the game servers
3. Process data that arrives late because of slow mobile networks
4. Allow SQL queries to access at least 10 TB of historical data
5. Process files that are regularly uploaded by users' mobile devices
6. Use only fully managed services

CEO Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users.

CTO Statement -

Our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

CFO Statement -

We are not capturing enough user demographic data, usage metrics, and other KPIs. As a result, we do not engage the right users, we are not confident that our marketing is targeting the right users, and we are not selling enough premium Blast-Ups inside the games, which dramatically impacts our revenue.

Question

Mountkirk Games wants to set up a real-time analytics platform for their new game. The new platform must meet their technical requirements. Which combination of Google technologies will meet all of their requirements?

- A. Kubernetes Engine, Cloud Pub/Sub, and Cloud SQL
- B. Cloud Dataflow, Cloud Storage, Cloud Pub/Sub, and BigQuery
- C. Cloud SQL, Cloud Storage, Cloud Pub/Sub, and Cloud Dataflow
- D. Cloud Dataproc, Cloud Pub/Sub, Cloud SQL, and Cloud Dataflow

E. Cloud Pub/Sub, Compute Engine, Cloud Storage, and Cloud Dataproc

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation.

Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. Mountkirk Games wants to migrate from their current analytics and statistics reporting model to one that meets their technical requirements on Google Cloud Platform.

Which two steps should be part of their migration plan? (Choose two.)

- A. Evaluate the impact of migrating their current batch ETL code to Cloud Dataflow.
- B. Write a schema migration plan to denormalize data for better performance in BigQuery.
- C. Draw an architecture diagram that shows how to move from a single MySQL database to a MySQL cluster.
- D. Load 10 TB of analytics data from a previous game into a Cloud SQL instance, and run test queries against the full dataset to confirm that they complete successfully.
- E. Integrate Cloud Armor to defend against possible SQL injection attacks in analytics files uploaded to Cloud Storage.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. You need to analyze and define the technical architecture for the compute workloads for your company, Mountkirk Games. Considering the Mountkirk Games business and technical requirements, what should you do?

- A. Create network load balancers. Use preemptible Compute Engine instances.
- B. Create network load balancers. Use non-preemptible Compute Engine instances.
- C. Create a global load balancer with managed instance groups and autoscaling policies. Use preemptible Compute Engine instances.
- D. Create a global load balancer with managed instance groups and autoscaling policies. Use non-preemptible Compute Engine instances.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. Mountkirk Games wants to design their solution for the future in order to take advantage of cloud and technology improvements as they become available. Which two steps should they take? (Choose two.)

- A. Store as much analytics and game activity data as financially feasible today so it can be used to train machine learning models to predict user behavior in the future.
- B. Begin packaging their game backend artifacts in container images and running them on Google Kubernetes Engine to improve the ability to scale up or down based on game activity.
- C. Set up a CI/CD pipeline using Jenkins and Spinnaker to automate canary deployments and improve development velocity.
- D. Adopt a schema versioning tool to reduce downtime when adding new game features that require storing additional player data in the database.
- E. Implement a weekly rolling maintenance process for the Linux virtual machines so they can apply critical kernel patches and package updates and reduce the risk of 0-day vulnerabilities.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. Mountkirk Games wants you to design a way to test the analytics platform's resilience to changes in mobile network latency. What should you do?

- A. Deploy failure injection software to the game analytics platform that can inject additional latency to mobile client analytics traffic.
- B. Build a test client that can be run from a mobile phone emulator on a Compute Engine virtual machine, and run multiple copies in Google Cloud Platform regions all over the world to generate realistic traffic.
- C. Add the ability to introduce a random amount of delay before beginning to process analytics files uploaded from mobile devices.
- D. Create an opt-in beta of the game that runs on players' mobile devices and collects response times from analytics endpoints running in Google Cloud Platform regions all over the world.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. You need to analyze and define the technical architecture for the database workloads for your company, Mountkirk Games. Considering the business and technical requirements, what should you do?

- A. Use Cloud SQL for time series data, and use Cloud Bigtable for historical data queries.
- B. Use Cloud SQL to replace MySQL, and use Cloud Spanner for historical data queries.
- C. Use Cloud Bigtable to replace MySQL, and use BigQuery for historical data queries.
- D. Use Cloud Bigtable for time series data, use Cloud Spanner for transactional data, and use BigQuery for historical data queries.

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

For this question, refer to the Mountkirk Games case study. Which managed storage option meets Mountkirk's technical requirement for storing game activity in a time series database service?

- A. Cloud Bigtable
- B. Cloud Spanner
- C. BigQuery
- D. Cloud Datastore

Introductory Info

Company Overview -

Mountkirk Games makes online, session-based, multiplayer games for mobile platforms. They build all of their games using some server-side integration.

Historically, they have used cloud providers to lease physical servers.

Due to the unexpected popularity of some of their games, they have had problems scaling their global audience, application servers, MySQL databases, and analytics tools.

Their current model is to write game statistics to files and send them through an ETL tool that loads them into a centralized MySQL database for reporting.

Solution Concept -

Mountkirk Games is building a new game, which they expect to be very popular. They plan to deploy the game's backend on Google Compute Engine so they can capture streaming metrics, run intensive analytics, and take advantage of its autoscaling server environment and integrate with a managed NoSQL database.

Business Requirements -

Increase to a global footprint

Improve uptime ``" downtime is loss of players

Increase efficiency of the cloud resources we use

Reduce latency to all customers

Technical Requirements -

Requirements for Game Backend Platform

Dynamically scale up or down based on game activity

Connect to a transactional database service to manage user profiles and game state

Store game activity in a timeseries database service for future analysis

As the system scales, ensure that data is not lost due to processing backlogs

Run hardened Linux distro

Requirements for Game Analytics Platform

Dynamically scale up or down based on game activity

Process incoming data on the fly directly from the game servers

Process data that arrives late because of slow mobile networks

Allow queries to access at least 10 TB of historical data

Process files that are regularly uploaded by users' mobile devices

Executive Statement -

Our last successful game did not scale well with our previous cloud provider, resulting in lower user adoption and affecting the game's reputation. Our investors want more key performance indicators (KPIs) to evaluate the speed and stability of the game, as well as other metrics that provide deeper insight into usage patterns so we can adapt the game to target users. Additionally, our current technology stack cannot provide the scale we need, so we want to replace MySQL and move to an environment that provides autoscaling, low latency load balancing, and frees us up from managing physical servers.

Question

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

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

- A. Create an instance template for the backend. For every region, deploy it on a multi-zone managed instance group. Use an L4 load balancer.
- B. Create an instance template for the backend. For every region, deploy it on a single-zone managed instance group. Use an L4 load balancer.
- C. Create an instance template for the backend. For every region, deploy it on a multi-zone managed instance group. Use an L7 load balancer.
- D. Create an instance template for the backend. For every region, deploy it on a single-zone managed instance group. Use an L7 load balancer.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

You need to optimize batch file transfers into Cloud Storage for Mountkirk Games' new Google Cloud solution. The batch files contain game statistics that need to be staged in Cloud Storage and be processed by an extract transform load (ETL) tool. What should you do?

- A. Use gsutil to batch move files in sequence.
- B. Use gsutil to batch copy the files in parallel.
- C. Use gsutil to extract the files as the first part of ETL.
- D. Use gsutil to load the files as the last part of ETL.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

You are implementing Firestore for Mountkirk Games. Mountkirk Games wants to give a new game programmatic access to a legacy game's Firestore database.

Access should be as restricted as possible. What should you do?

- A. Create a service account (SA) in the legacy game's Google Cloud project, add a second SA in the new game's IAM page, and then give the Organization Admin role to both SAs.
- B. Create a service account (SA) in the legacy game's Google Cloud project, give the SA the Organization Admin role, and then give it the Firebase Admin role in both projects.
- C. Create a service account (SA) in the legacy game's Google Cloud project, add this SA in the new game's IAM page, and then give it the Firebase Admin role in both projects.

D. Create a service account (SA) in the legacy game's Google Cloud project, give it the Firebase Admin role, and then migrate the new game to the legacy game's project.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

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

Technical requirements -

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

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

Mountkirk Games wants to limit the physical location of resources to their operating Google Cloud regions. What should you do?

- A. Configure an organizational policy which constrains where resources can be deployed.
- B. Configure IAM conditions to limit what resources can be configured.
- C. Configure the quotas for resources in the regions not being used to 0.
- D. Configure a custom alert in Cloud Monitoring so you can disable resources as they are created in other regions.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

You need to implement a network ingress for a new game that meets the defined business and technical requirements. Mountkirk Games wants each regional game instance to be located in multiple Google Cloud regions. What should you do?

- A. Configure a global load balancer connected to a managed instance group running Compute Engine instances.
- B. Configure kubemci with a global load balancer and Google Kubernetes Engine.
- C. Configure a global load balancer with Google Kubernetes Engine.
- D. Configure Ingress for Anthos with a global load balancer and Google Kubernetes Engine.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

Your development teams release new versions of games running on Google Kubernetes Engine (GKE) daily. You want to create service level indicators (SLIs) to evaluate the quality of the new versions from the user's perspective. What should you do?

- A. Create CPU Utilization and Request Latency as service level indicators.
- B. Create GKE CPU Utilization and Memory Utilization as service level indicators.
- C. Create Request Latency and Error Rate as service level indicators.
- D. Create Server Uptime and Error Rate as service level indicators.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

Mountkirk Games wants you to secure the connectivity from the new gaming application platform to Google Cloud. You want to streamline the process and follow

Google-recommended practices. What should you do?

A. Configure Workload Identity and service accounts to be used by the application platform.

B. Use Kubernetes Secrets, which are obfuscated by default. Configure these Secrets to be used by the application platform.

C. Configure Kubernetes Secrets to store the secret, enable Application-Layer Secrets Encryption, and use Cloud Key Management Service (Cloud KMS) to manage the encryption keys. Configure these Secrets to be used by the application platform.

D. Configure HashiCorp Vault on Compute Engine, and use customer managed encryption keys and Cloud Key Management Service (Cloud KMS) to manage the encryption keys. Configure these Secrets to be used by the application platform.

Introductory Info

Company overview -

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

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

Solution concept -

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

Existing technical environment -

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

Business requirements -

Support multiple gaming platforms.

Support multiple regions.

Support rapid iteration of game features.

Minimize latency.

Optimize for dynamic scaling.

Use managed services and pooled resources.

Minimize costs.

Technical requirements -

Dynamically scale based on game activity.

Publish scoring data on a near real-time global leaderboard.

Store game activity logs in structured files for future analysis.

Use GPU processing to render graphics server-side for multi-platform support.

Support eventual migration of legacy games to this new platform.

Executive statement -

Our last game was the first time we used Google Cloud, and it was a tremendous success. We were able to analyze player behavior and game telemetry in ways that we never could before. This success allowed us to bet on a full migration to the cloud and to start building all-new games using cloud-native design principles.

Our new game is our most ambitious to date and will open up doors for us to support more gaming platforms beyond mobile. Latency is our top priority, although cost management is the next most important challenge. As with our first cloud-based game, we have grown to expect the cloud to enable advanced analytics capabilities so we can rapidly iterate on our deployments of bug fixes and new functionality.

Question

Your development team has created a mobile game app. You want to test the new mobile app on Android and iOS devices with a variety of configurations. You need to ensure that testing is efficient and cost-effective. What should you do?

- A. Upload your mobile app to the Firebase Test Lab, and test the mobile app on Android and iOS devices.
- B. Create Android and iOS VMs on Google Cloud, install the mobile app on the VMs, and test the mobile app.
- C. Create Android and iOS containers on Google Kubernetes Engine (GKE), install the mobile app on the containers, and test the mobile app.
- D. Upload your mobile app with different configurations to Firebase Hosting and test each configuration.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

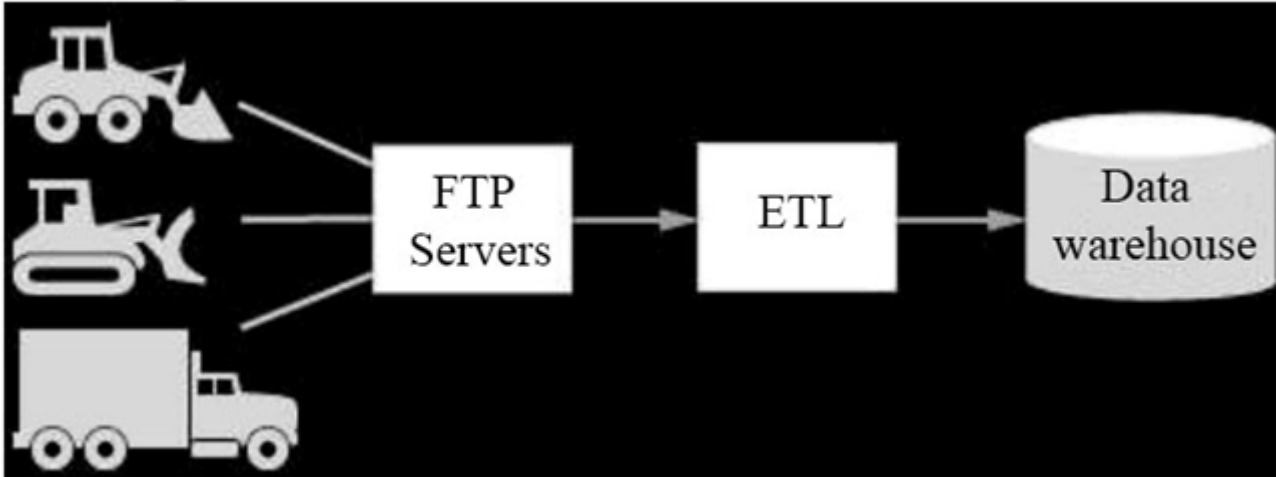
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

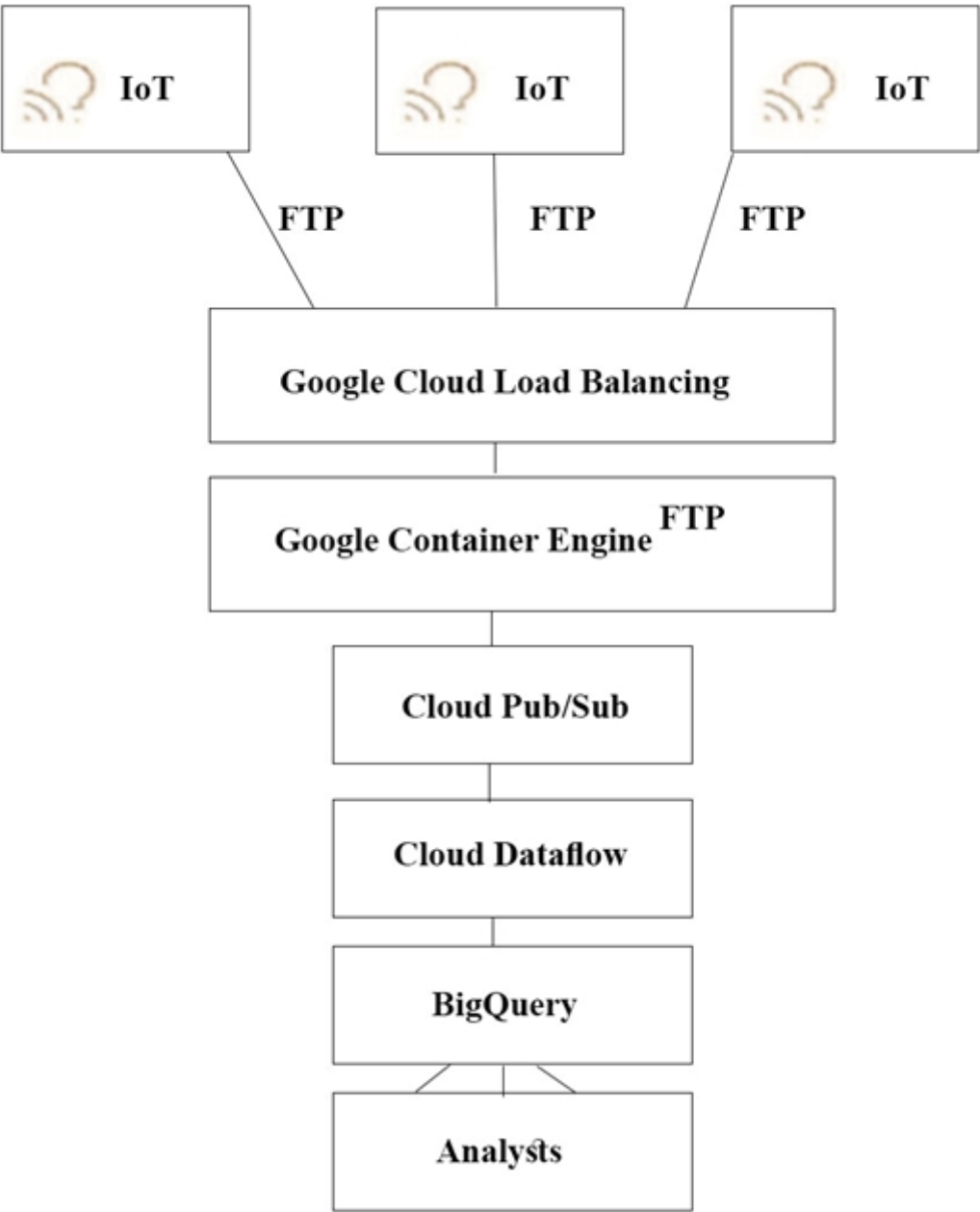
Question

TerramEarth's CTO wants to use the raw data from connected vehicles to help identify approximately when a vehicle in the field will have a catastrophic failure.

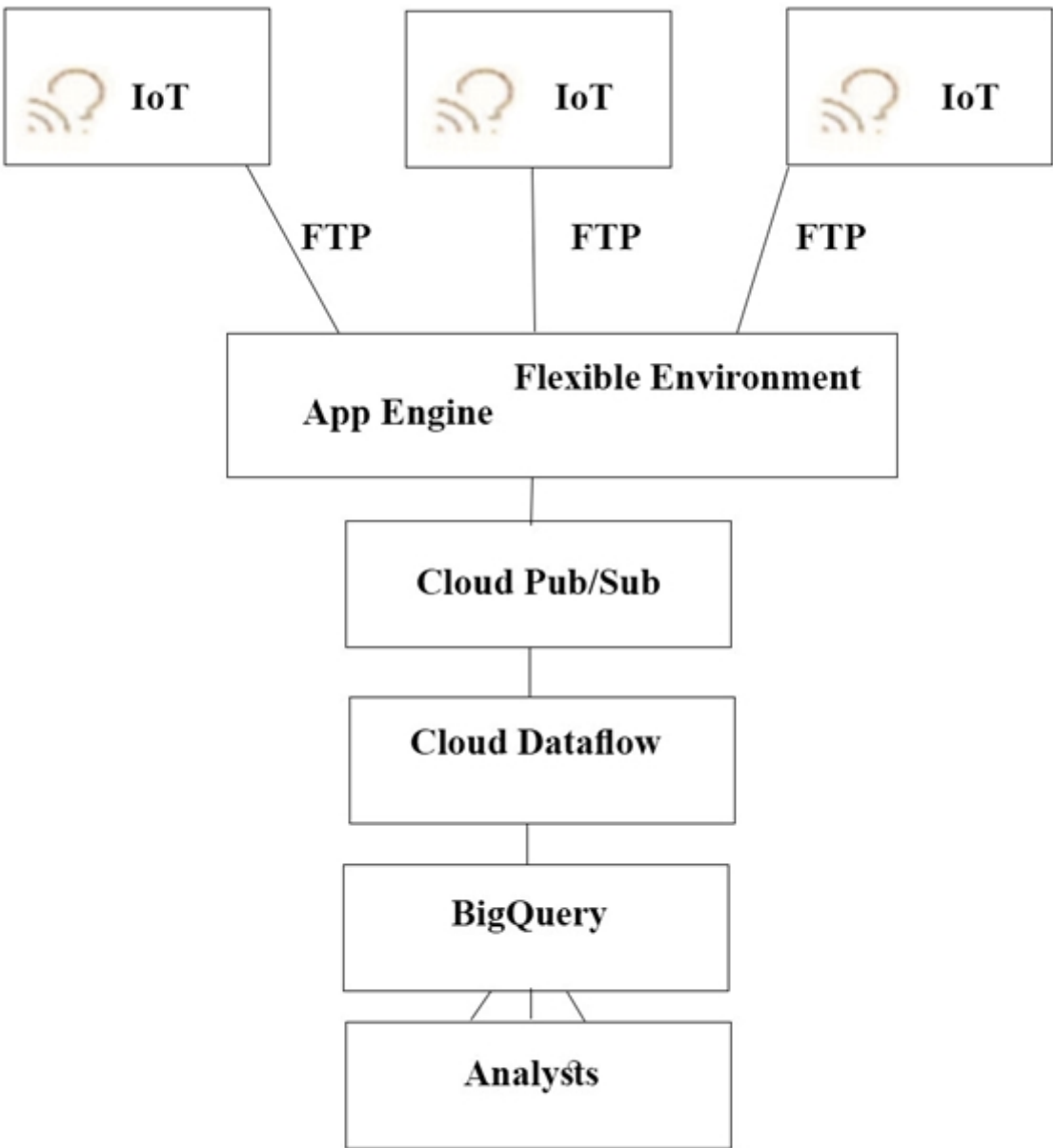
You want to allow analysts to centrally query the vehicle data.

Which architecture should you recommend?

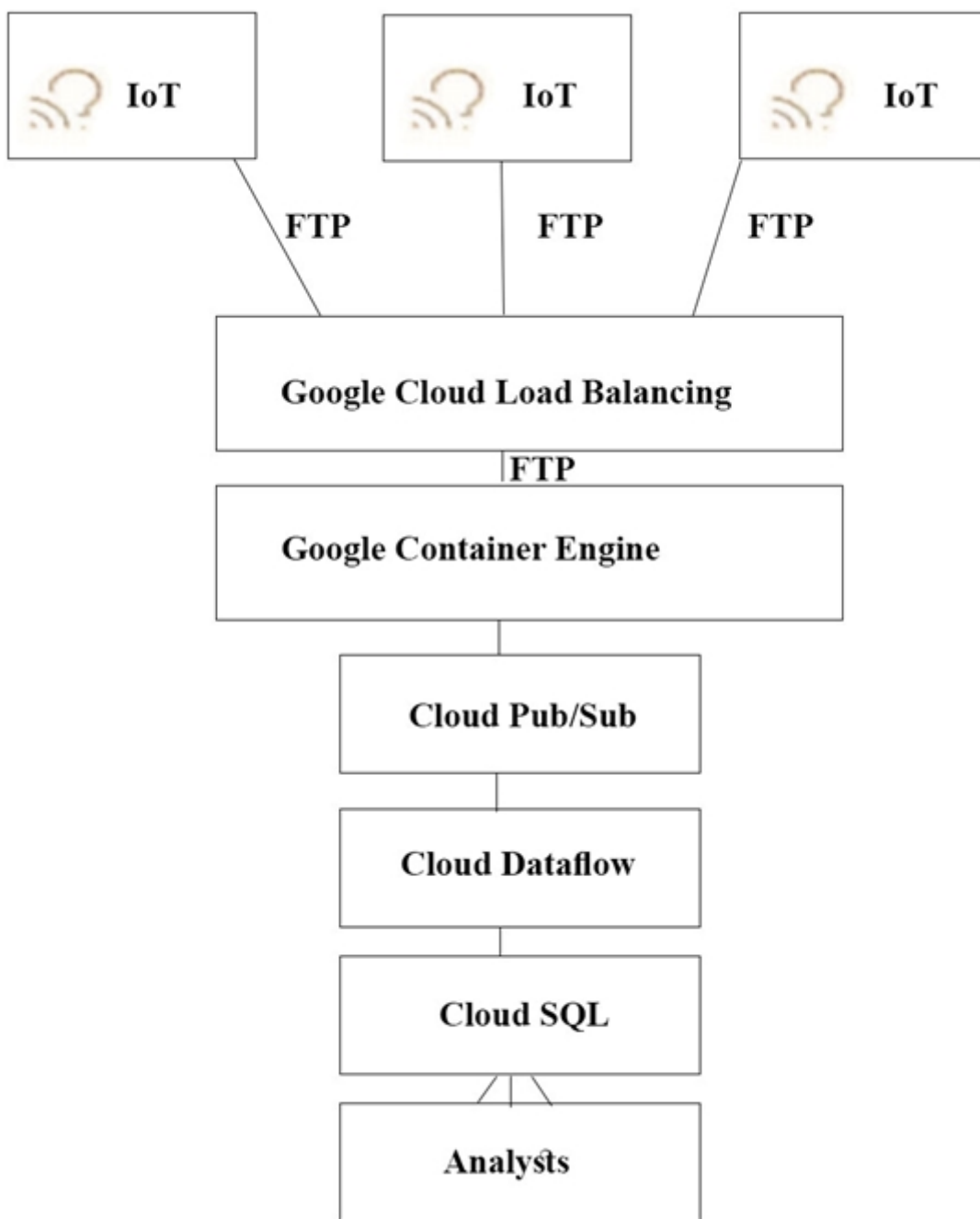
A.



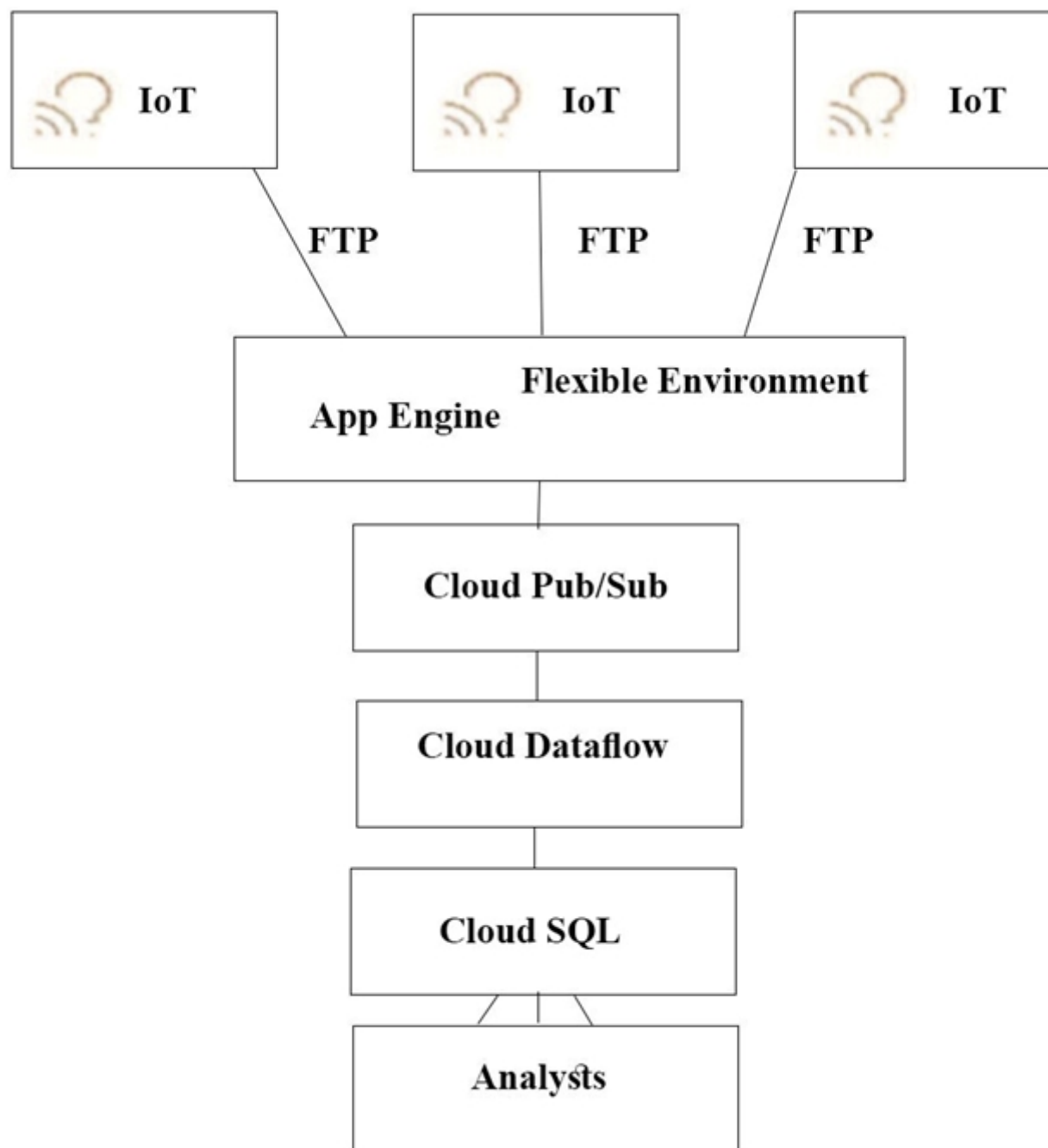
B.



C.



D.



Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

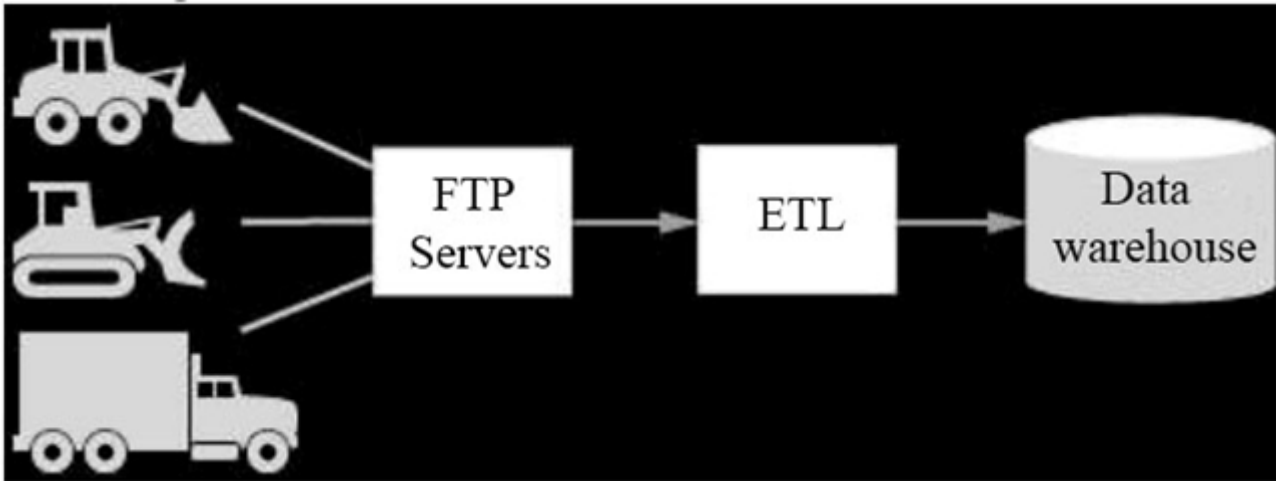
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

The TerramEarth development team wants to create an API to meet the company's business requirements. You want the development team to focus their development effort on business value versus creating a custom framework.

Which method should they use?

- A. Use Google App Engine with Google Cloud Endpoints. Focus on an API for dealers and partners
- B. Use Google App Engine with a JAX-RS Jersey Java-based framework. Focus on an API for the public
- C. Use Google App Engine with the Swagger (Open API Specification) framework. Focus on an API for the public
- D. Use Google Container Engine with a Django Python container. Focus on an API for the public
- E. Use Google Container Engine with a Tomcat container with the Swagger (Open API Specification) framework. Focus on an API for dealers and partners

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

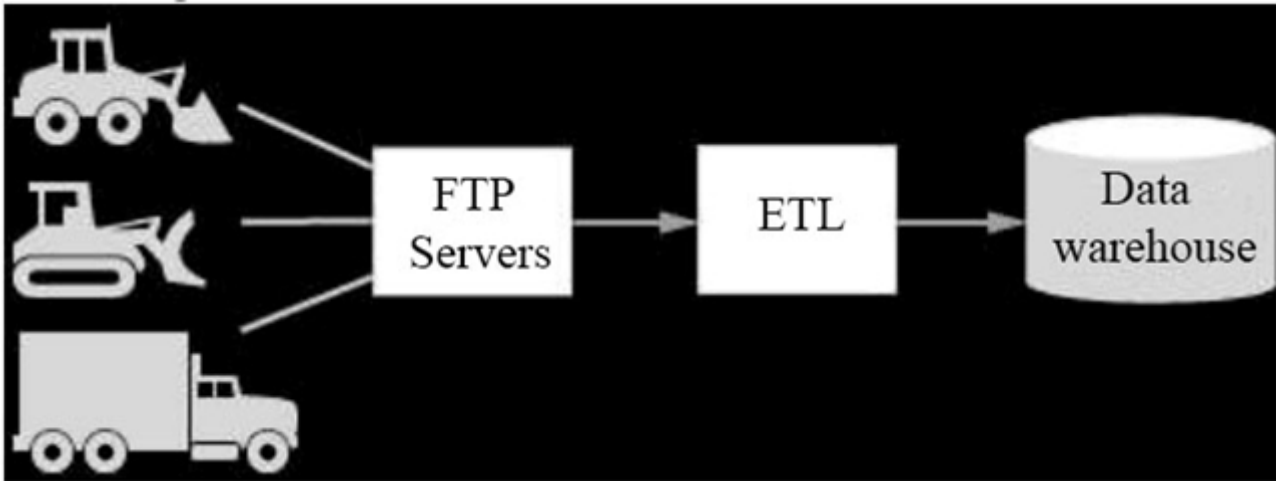
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

Your development team has created a structured API to retrieve vehicle data. They want to allow third parties to develop tools for dealerships that use this vehicle event data. You want to support delegated authorization against this data.

What should you do?

- A. Build or leverage an OAuth-compatible access control system
- B. Build SAML 2.0 SSO compatibility into your authentication system
- C. Restrict data access based on the source IP address of the partner systems
- D. Create secondary credentials for each dealer that can be given to the trusted third party

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

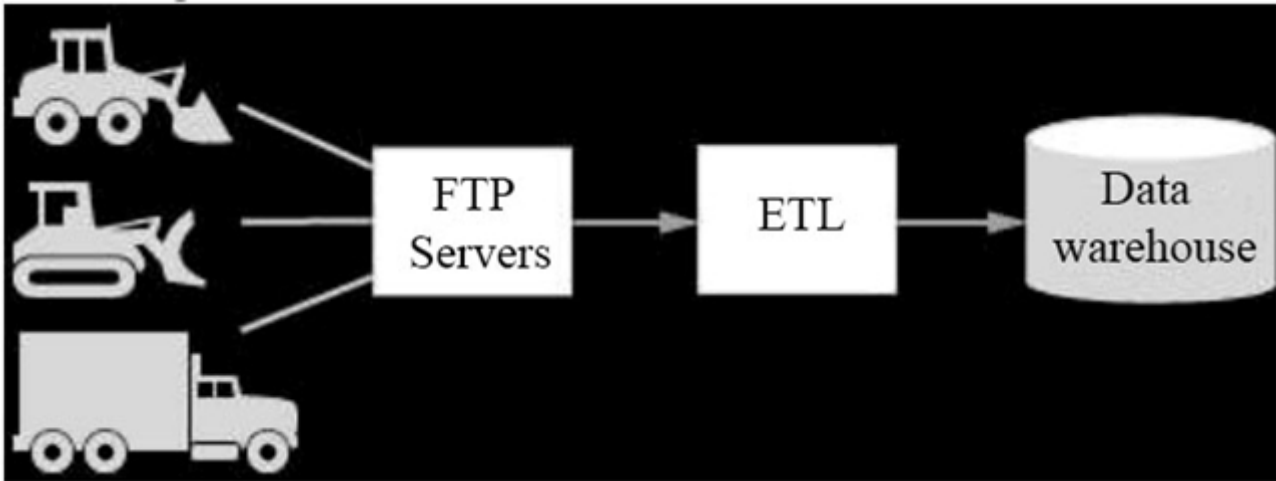
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

TerramEarth plans to connect all 20 million vehicles in the field to the cloud. This increases the volume to 20 million 600 byte records a second for 40 TB an hour.

How should you design the data ingestion?

- A. Vehicles write data directly to GCS
- B. Vehicles write data directly to Google Cloud Pub/Sub
- C. Vehicles stream data directly to Google BigQuery
- D. Vehicles continue to write data using the existing system (FTP)

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

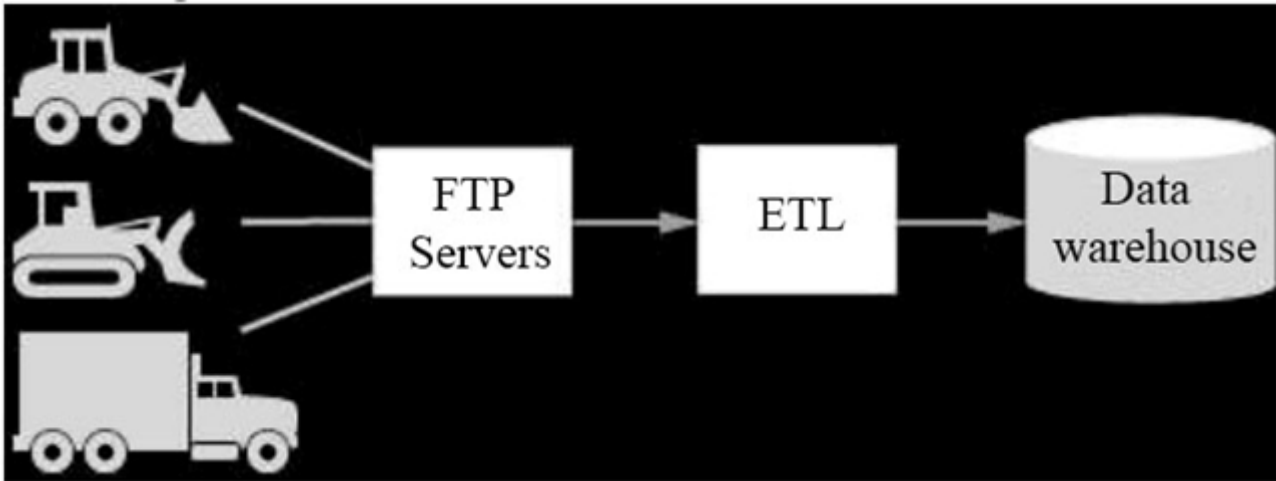
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

You analyzed TerramEarth's business requirement to reduce downtime, and found that they can achieve a majority of time saving by reducing customer's wait time for parts. You decided to focus on reduction of the 3 weeks aggregate reporting time.

Which modifications to the company's processes should you recommend?

- A. Migrate from CSV to binary format, migrate from FTP to SFTP transport, and develop machine learning analysis of metrics
- B. Migrate from FTP to streaming transport, migrate from CSV to binary format, and develop machine learning analysis of metrics
- C. Increase fleet cellular connectivity to 80%, migrate from FTP to streaming transport, and develop machine learning analysis of metrics
- D. Migrate from FTP to SFTP transport, develop machine learning analysis of metrics, and increase dealer local inventory by a fixed factor

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

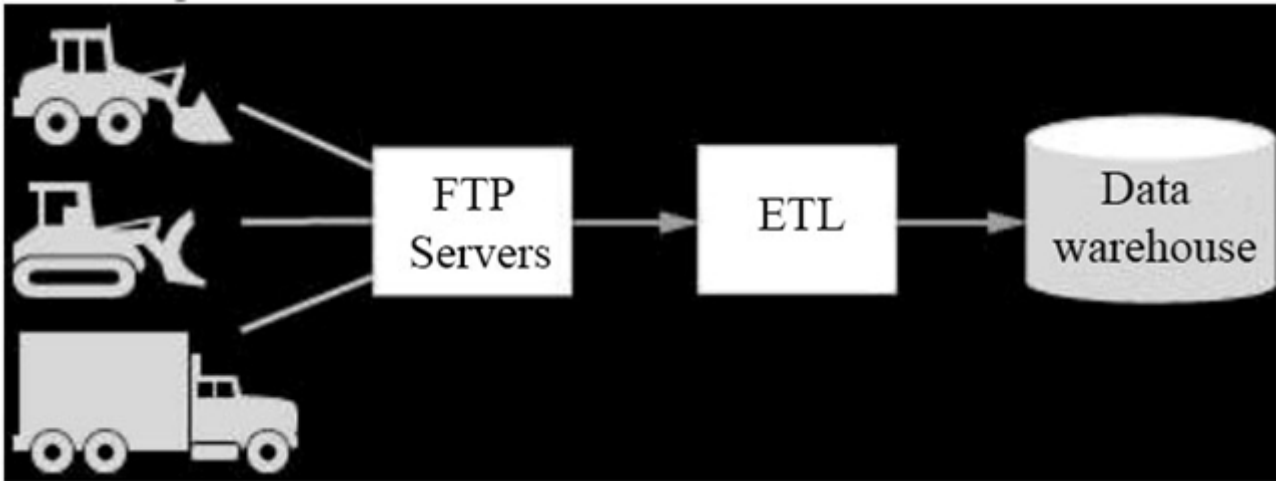
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

Which of TerramEarth's legacy enterprise processes will experience significant change as a result of increased Google Cloud Platform adoption?

- A. Opex/capex allocation, LAN changes, capacity planning
- B. Capacity planning, TCO calculations, opex/capex allocation
- C. Capacity planning, utilization measurement, data center expansion
- D. Data Center expansion, TCO calculations, utilization measurement

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

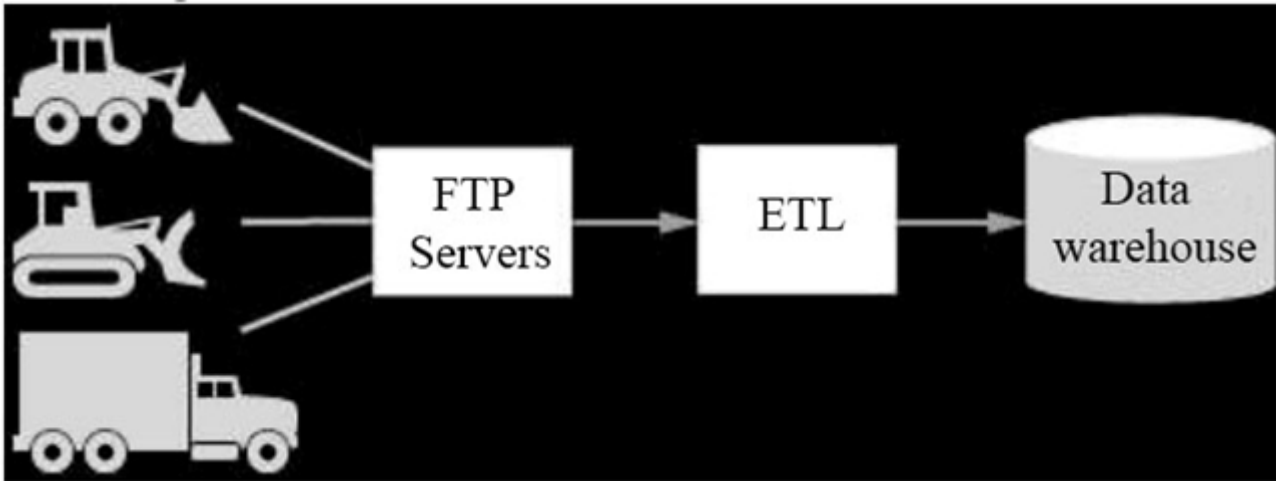
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

To speed up data retrieval, more vehicles will be upgraded to cellular connections and be able to transmit data to the ETL process. The current FTP process is error-prone and restarts the data transfer from the start of the file when connections fail, which happens often. You want to improve the reliability of the solution and minimize data transfer time on the cellular connections.

What should you do?

- A. Use one Google Container Engine cluster of FTP servers. Save the data to a Multi-Regional bucket. Run the ETL process using data in the bucket
- B. Use multiple Google Container Engine clusters running FTP servers located in different regions. Save the data to Multi-Regional buckets in US, EU, and Asia. Run the ETL process using the data in the bucket
- C. Directly transfer the files to different Google Cloud Multi-Regional Storage bucket locations in US, EU, and Asia using Google APIs over HTTP(S). Run the ETL process using the data in the bucket
- D. Directly transfer the files to a different Google Cloud Regional Storage bucket location in US, EU, and Asia using Google APIs over HTTP(S). Run the ETL process to retrieve the data from each Regional bucket

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

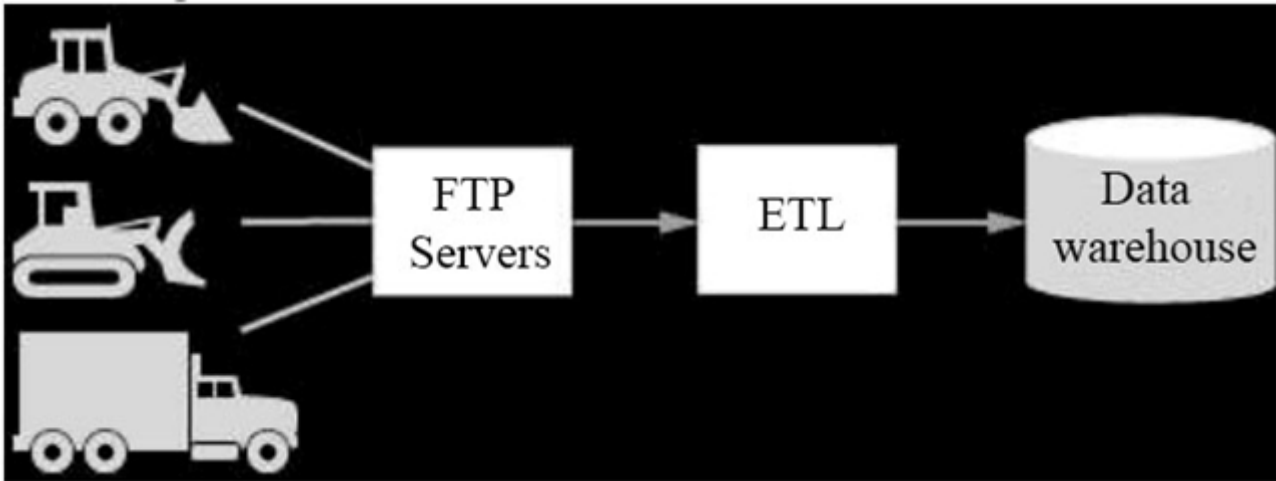
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

TerramEarth's 20 million vehicles are scattered around the world. Based on the vehicle's location, its telemetry data is stored in a Google Cloud Storage (GCS) regional bucket (US, Europe, or Asia). The CTO has asked you to run a report on the raw telemetry data to determine why vehicles are breaking down after 100 K miles. You want to run this job on all the data.

What is the most cost-effective way to run this job?

- A. Move all the data into 1 zone, then launch a Cloud Dataproc cluster to run the job
- B. Move all the data into 1 region, then launch a Google Cloud Dataproc cluster to run the job
- C. Launch a cluster in each region to preprocess and compress the raw data, then move the data into a multi-region bucket and use a Dataproc cluster to finish the job
- D. Launch a cluster in each region to preprocess and compress the raw data, then move the data into a region bucket and use a Cloud Dataproc cluster to finish the job

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

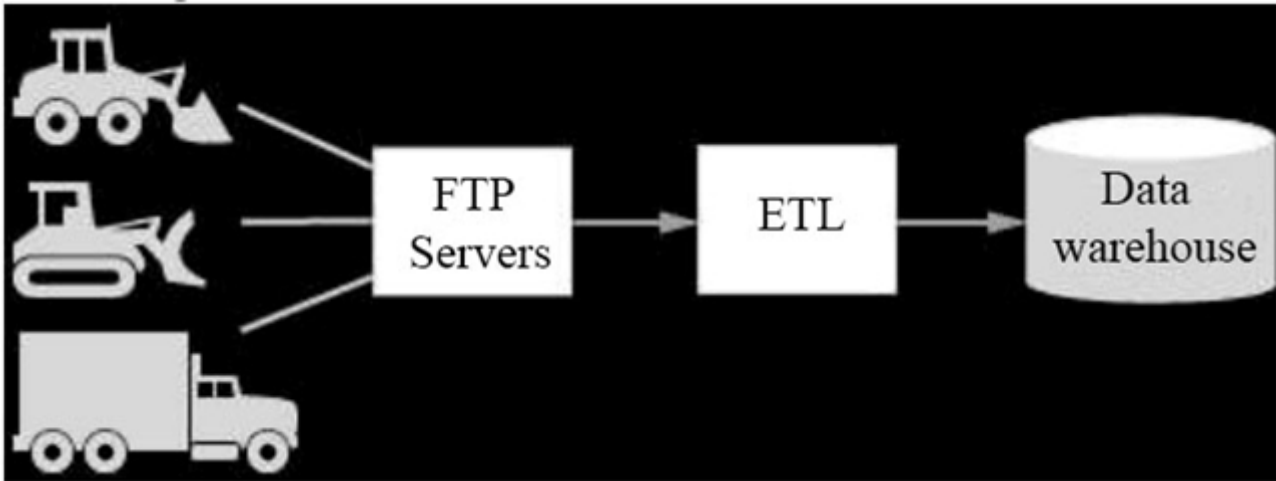
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

TerramEarth has equipped all connected trucks with servers and sensors to collect telemetry data. Next year they want to use the data to train machine learning models. They want to store this data in the cloud while reducing costs.

What should they do?

- A. Have the vehicle's computer compress the data in hourly snapshots, and store it in a Google Cloud Storage (GCS) Nearline bucket
- B. Push the telemetry data in real-time to a streaming dataflow job that compresses the data, and store it in Google BigQuery
- C. Push the telemetry data in real-time to a streaming dataflow job that compresses the data, and store it in Cloud Bigtable
- D. Have the vehicle's computer compress the data in hourly snapshots, and store it in a GCS Coldline bucket

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

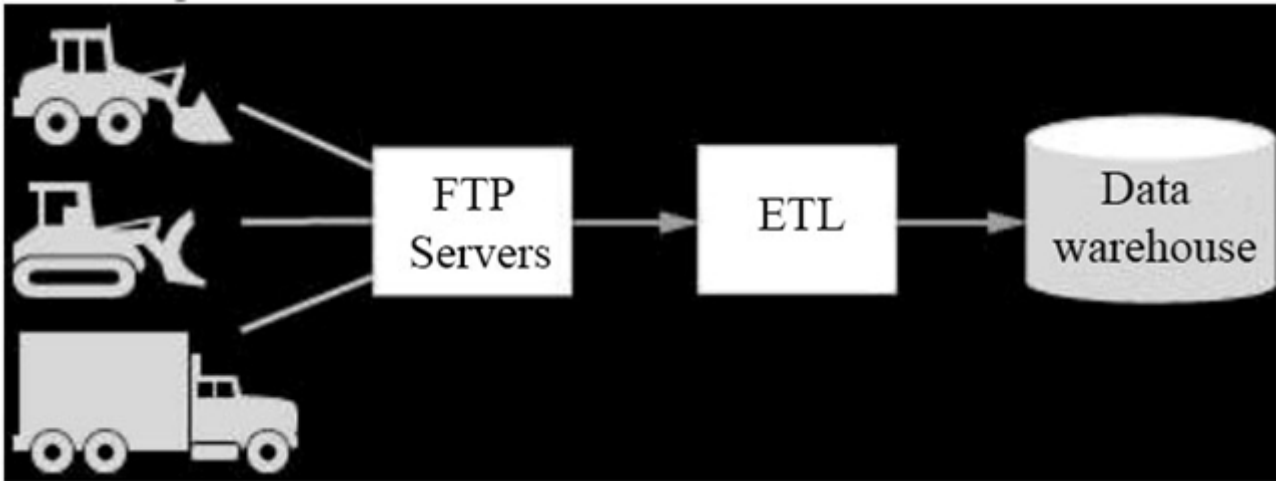
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

Your agricultural division is experimenting with fully autonomous vehicles. You want your architecture to promote strong security during vehicle operation.

Which two architectures should you consider? (Choose two.)

- A. Treat every micro service call between modules on the vehicle as untrusted.
- B. Require IPv6 for connectivity to ensure a secure address space.
- C. Use a trusted platform module (TPM) and verify firmware and binaries on boot.
- D. Use a functional programming language to isolate code execution cycles.
- E. Use multiple connectivity subsystems for redundancy.
- F. Enclose the vehicle's drive electronics in a Faraday cage to isolate chips.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries: about 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Company background -

TerramEarth was formed in 1946, when several small, family owned companies combined to retool after World War II. The company cares about their employees and customers and considers them to be extended members of their family.

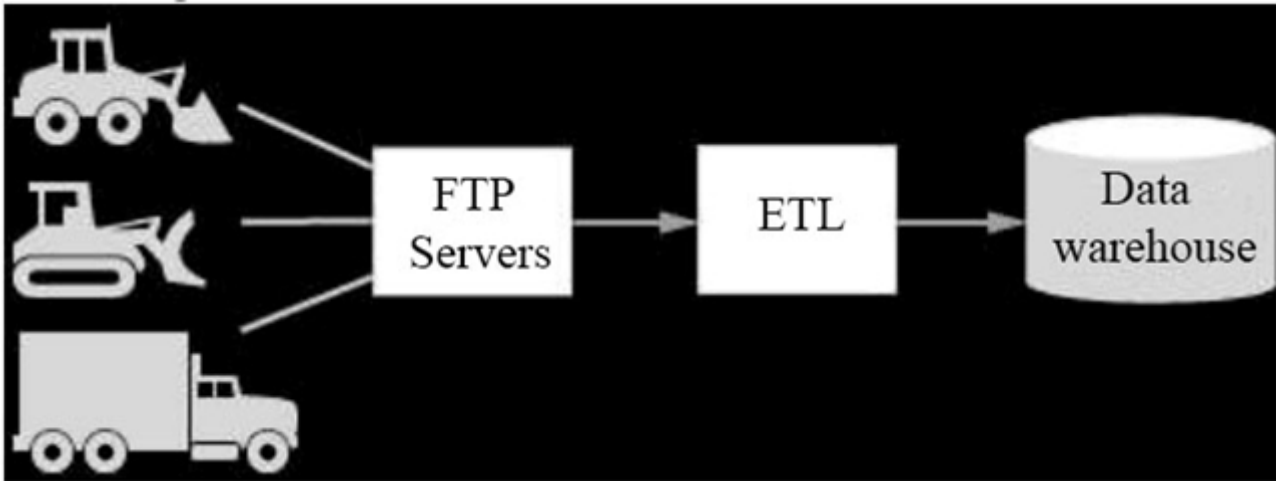
TerramEarth is proud of their ability to innovate on their core products and find new markets as their customers' needs change. For the past 20 years, trends in the industry have been largely toward increasing productivity by using larger vehicles with a human operator.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second with 22 hours of operation per day, Terram Earth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -



TerramEarth's existing architecture is composed of Linux-based systems that reside in a data center. These systems gzip CSV files from the field and upload via

FTP, transform and aggregate them, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week, without increasing the cost of carrying surplus inventory

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers.

CEO Statement -

We have been successful in capitalizing on the trend toward larger vehicles to increase the productivity of our customers. Technological change is occurring rapidly, and TerramEarth has taken advantage of connected devices technology to provide our customers with better services, such as our intelligent farming equipment. With this technology, we have been able to increase farmers' yields by 25%, by using past trends to adjust how our vehicles operate. These advances have led to the rapid growth of our agricultural product line, which we expect will generate 50% of our revenues by 2020.

CTO Statement -

Our competitive advantage has always been in the manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. Unfortunately, our CEO doesn't take technology obsolescence seriously and he considers the many new companies in our industry to be niche players. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

Operational parameters such as oil pressure are adjustable on each of TerramEarth's vehicles to increase their efficiency, depending on their environmental conditions. Your primary goal is to increase the operating efficiency of all 20 million cellular and unconnected vehicles in the field. How can you accomplish this goal?

- A. Have you engineers inspect the data for patterns, and then create an algorithm with rules that make operational adjustments automatically
- B. Capture all operating data, train machine learning models that identify ideal operations, and run locally to make operational adjustments automatically
- C. Implement a Google Cloud Dataflow streaming job with a sliding window, and use Google Cloud Messaging (GCM) to make operational adjustments automatically
- D. Capture all operating data, train machine learning models that identify ideal operations, and host in Google Cloud Machine Learning (ML) Platform to make operational adjustments automatically

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules. Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with 22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center. These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old. With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%. However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week
Support the dealer network with more data on how their customers use their equipment to better position new products and services
Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast
Create a backup strategy
Increase security of data transfer from equipment to the datacenter
Improve data in the data warehouse
Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2
- 16 CPUs
- 128 GB of RAM
- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs
- Windows Server 2008 R2
- 16 CPUs
- 32 GB of RAM
- 500 GB HDD

Data warehouse:

A single PostgreSQL server
- RedHat Linux
- 64 CPUs

- 128 GB of RAM
- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. To be compliant with European GDPR regulation, TerramEarth is required to delete data generated from its

European customers after a period of 36 months when it contains personal data. In the new architecture, this data will be stored in both Cloud Storage and

BigQuery. What should you do?

- A. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.
- B. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action when with an Age condition of 36 months.
- C. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.
- D. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action with an Age condition of 36 months.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with

22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center.

These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast

Create a backup strategy

Increase security of data transfer from equipment to the datacenter

Improve data in the data warehouse

Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2

- 16 CPUs

- 128 GB of RAM

- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs

- Windows Server 2008 R2

- 16 CPUs

- 32 GB of RAM

- 500 GB HDD

Data warehouse:

A single PostgreSQL server

- RedHat Linux

- 64 CPUs

- 128 GB of RAM

- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. TerramEarth has decided to store data files in Cloud Storage. You need to configure Cloud Storage lifecycle rule to store 1 year of data and minimize file storage cost.

Which two actions should you take?

- A. Create a Cloud Storage lifecycle rule with Age: 30, Storage Class: Standard, and Action: Set to Coldline, and create a second GCS life-cycle rule with Age: 365, Storage Class: Coldline, and Action: Delete.
- B. Create a Cloud Storage lifecycle rule with Age: 30, Storage Class: Coldline, and Action: Set to Nearline, and create a second GCS life-cycle rule with Age: 91, Storage Class: Coldline, and Action: Set to Nearline.
- C. Create a Cloud Storage lifecycle rule with Age: 90, Storage Class: Standard, and Action: Set to Nearline, and create a second GCS life-cycle rule with Age: 91, Storage Class: Nearline, and Action: Set to Coldline.
- D. Create a Cloud Storage lifecycle rule with Age: 30, Storage Class: Standard, and Action: Set to Coldline, and create a second GCS life-cycle rule with Age: 365, Storage Class: Nearline, and Action: Delete.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with

22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center. These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast

Create a backup strategy

Increase security of data transfer from equipment to the datacenter

Improve data in the data warehouse

Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2

- 16 CPUs

- 128 GB of RAM

- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs

- Windows Server 2008 R2

- 16 CPUs

- 32 GB of RAM

- 500 GB HDD

Data warehouse:

A single PostgreSQL server

- RedHat Linux

- 64 CPUs

- 128 GB of RAM

- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. You need to implement a reliable, scalable GCP solution for the data warehouse for your company, TerramEarth.

Considering the TerramEarth business and technical requirements, what should you do?

- A. Replace the existing data warehouse with BigQuery. Use table partitioning.
- B. Replace the existing data warehouse with a Compute Engine instance with 96 CPUs.
- C. Replace the existing data warehouse with BigQuery. Use federated data sources.
- D. Replace the existing data warehouse with a Compute Engine instance with 96 CPUs. Add an additional Compute Engine preemptible instance with 32 CPUs.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules. Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with 22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center. These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old. With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%. However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast

Create a backup strategy

Increase security of data transfer from equipment to the datacenter

Improve data in the data warehouse

Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2

- 16 CPUs
- 128 GB of RAM
- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs

- Windows Server 2008 R2
- 16 CPUs
- 32 GB of RAM
- 500 GB HDD

Data warehouse:

A single PostgreSQL server

- RedHat Linux
- 64 CPUs
- 128 GB of RAM
- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. A new architecture that writes all incoming data to BigQuery has been introduced. You notice that the data is dirty, and want to ensure data quality on an automated daily basis while managing cost.

What should you do?

- A. Set up a streaming Cloud Dataflow job, receiving data by the ingestion process. Clean the data in a Cloud Dataflow pipeline.
- B. Create a Cloud Function that reads data from BigQuery and cleans it. Trigger the Cloud Function from a Compute Engine instance.
- C. Create a SQL statement on the data in BigQuery, and save it as a view. Run the view daily, and save the result to a new table.
- D. Use Cloud Dataprep and configure the BigQuery tables as the source. Schedule a daily job to clean the data.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with

22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center. These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast

Create a backup strategy

Increase security of data transfer from equipment to the datacenter

Improve data in the data warehouse

Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2

- 16 CPUs

- 128 GB of RAM

- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs

- Windows Server 2008 R2

- 16 CPUs

- 32 GB of RAM

- 500 GB HDD

Data warehouse:

A single PostgreSQL server

- RedHat Linux

- 64 CPUs

- 128 GB of RAM

- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. Considering the technical requirements, how should you reduce the unplanned vehicle downtime in GCP?

- A. Use BigQuery as the data warehouse. Connect all vehicles to the network and stream data into BigQuery using Cloud Pub/Sub and Cloud Dataflow. Use Google Data Studio for analysis and reporting.
- B. Use BigQuery as the data warehouse. Connect all vehicles to the network and upload gzip files to a Multi-Regional Cloud Storage bucket using gcloud. Use Google Data Studio for analysis and reporting.
- C. Use Cloud Dataproc Hive as the data warehouse. Upload gzip files to a Multi-Regional Cloud Storage bucket. Upload this data into BigQuery using gcloud. Use Google Data Studio for analysis and reporting.
- D. Use Cloud Dataproc Hive as the data warehouse. Directly stream data into partitioned Hive tables. Use Pig scripts to analyze data.

Introductory Info

Company Overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. About 80% of their business is from mining and 20% from agriculture. They currently have over 500 dealers and service centers in 100 countries. Their mission is to build products that make their customers more productive.

Solution Concept -

There are 20 million TerramEarth vehicles in operation that collect 120 fields of data per second. Data is stored locally on the vehicle and can be accessed for analysis when a vehicle is serviced. The data is downloaded via a maintenance port. This same port can be used to adjust operational parameters, allowing the vehicles to be upgraded in the field with new computing modules.

Approximately 200,000 vehicles are connected to a cellular network, allowing TerramEarth to collect data directly. At a rate of 120 fields of data per second, with

22 hours of operation per day, TerramEarth collects a total of about 9 TB/day from these connected vehicles.

Existing Technical Environment -

TerramEarth's existing architecture is composed of Linux and Windows-based systems that reside in a single U.S, west coast based data center. These systems gzip CSV files from the field and upload via FTP, and place the data in their data warehouse. Because this process takes time, aggregated reports are based on data that is 3 weeks old.

With this data, TerramEarth has been able to preemptively stock replacement parts and reduce unplanned downtime of their vehicles by 60%.

However, because the data is stale, some customers are without their vehicles for up to 4 weeks while they wait for replacement parts.

Business Requirements -

Decrease unplanned vehicle downtime to less than 1 week

Support the dealer network with more data on how their customers use their equipment to better position new products and services

Have the ability to partner with different companies `` especially with seed and fertilizer suppliers in the fast-growing agricultural business `` to create compelling joint offerings for their customers

Technical Requirements -

Expand beyond a single datacenter to decrease latency to the American midwest and east coast

Create a backup strategy

Increase security of data transfer from equipment to the datacenter

Improve data in the data warehouse

Use customer and equipment data to anticipate customer needs

Application 1: Data ingest -

A custom Python application reads uploaded datafiles from a single server, writes to the data warehouse.

Compute:

Windows Server 2008 R2

- 16 CPUs

- 128 GB of RAM

- 10 TB local HDD storage

Application 2: Reporting -

An off the shelf application that business analysts use to run a daily report to see what equipment needs repair. Only 2 analysts of a team of 10 (5 west coast, 5 east coast) can connect to the reporting application at a time.

Compute:

Off the shelf application. License tied to number of physical CPUs

- Windows Server 2008 R2

- 16 CPUs

- 32 GB of RAM

- 500 GB HDD

Data warehouse:

A single PostgreSQL server

- RedHat Linux

- 64 CPUs

- 128 GB of RAM

- 4x 6TB HDD in RAID 0

Executive Statement -

Our competitive advantage has always been in our manufacturing process, with our ability to build better vehicles for lower cost than our competitors. However, new products with different approaches are constantly being developed, and I'm concerned that we lack the skills to undergo the next wave of transformations in our industry. My goals are to build our skills while addressing immediate market needs through incremental innovations.

Question

For this question, refer to the TerramEarth case study. You are asked to design a new architecture for the ingestion of the data of the 200,000 vehicles that are connected to a cellular network. You want to follow Google-recommended practices.

Considering the technical requirements, which components should you use for the ingestion of the data?

- A. Google Kubernetes Engine with an SSL Ingress
- B. Cloud IoT Core with public/private key pairs
- C. Compute Engine with project-wide SSH keys
- D. Compute Engine with specific SSH keys

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

For this question, refer to the TerramEarth case study. You start to build a new application that uses a few Cloud Functions for the backend. One use case requires a Cloud Function `func_display` to invoke another Cloud Function `func_query`. You want `func_query` only to accept invocations from `func_display`. You also want to follow Google's recommended best practices. What should you do?

- A. Create a token and pass it in as an environment variable to `func_display`. When invoking `func_query`, include the token in the request. Pass the same token to `func_query` and reject the invocation if the tokens are different.
- B. Make `func_query` 'Require authentication.' Create a unique service account and associate it to `func_display`. Grant the service account invoker role for `func_query`. Create an id token in `func_display` and include the token to the request when invoking `func_query`.
- C. Make `func_query` 'Require authentication' and only accept internal traffic. Create those two functions in the same VPC. Create an ingress firewall rule for `func_query` to only allow traffic from `func_display`.

D. Create those two functions in the same project and VPC. Make func_query only accept internal traffic. Create an ingress firewall for func_query to only allow traffic from func_display. Also, make sure both functions use the same service account.

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

For this question, refer to the TerramEarth case study. You have broken down a legacy monolithic application into a few containerized RESTful microservices.

You want to run those microservices on Cloud Run. You also want to make sure the services are highly available with low latency to your customers. What should you do?

- A. Deploy Cloud Run services to multiple availability zones. Create Cloud Endpoints that point to the services. Create a global HTTP(S) Load Balancing instance and attach the Cloud Endpoints to its backend.
- B. Deploy Cloud Run services to multiple regions. Create serverless network endpoint groups pointing to the services. Add the serverless NEG's to a backend service that is used by a global HTTP(S) Load Balancing instance.
- C. Deploy Cloud Run services to multiple regions. In Cloud DNS, create a latency-based DNS name that points to the services.
- D. Deploy Cloud Run services to multiple availability zones. Create a TCP/IP global load balancer. Add the Cloud Run Endpoints to its backend service.

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

For this question, refer to the TerramEarth case study. You are migrating a Linux-based application from your private data center to Google Cloud. The

TerramEarth security team sent you several recent Linux vulnerabilities published by Common Vulnerabilities and Exposures (CVE). You need assistance in understanding how these vulnerabilities could impact your migration. What should you do? (Choose two.)

- A. Open a support case regarding the CVE and chat with the support engineer.
- B. Read the CVEs from the Google Cloud Status Dashboard to understand the impact.
- C. Read the CVEs from the Google Cloud Platform Security Bulletins to understand the impact.
- D. Post a question regarding the CVE in Stack Overflow to get an explanation.
- E. Post a question regarding the CVE in a Google Cloud discussion group to get an explanation.

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

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

- A. Create a scheduled job in Cloud Run to invoke a container every minute. The container will check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.
- B. Create a cron job on a Compute Engine VM that runs every minute. The cron job invokes a Python program to check the application URL. If the application is down, switch the URL to the "Site is unavailable" page, and notify the Ops team.
- C. Create a Cloud Monitoring uptime check to validate the application URL. If it fails, put a message in a Pub/Sub queue that triggers a Cloud Function to switch the URL to the "Site is unavailable" page, and notify the Ops team.

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

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

For this question, refer to the TerramEarth case study. You are building a microservice-based application for TerramEarth. The application is based on Docker containers. You want to follow Google-recommended practices to build the application continuously and store the build artifacts. What should you do?

- A. Configure a trigger in Cloud Build for new source changes. Invoke Cloud Build to build container images for each microservice, and tag them using the code commit hash. Push the images to the Container Registry.
- B. Configure a trigger in Cloud Build for new source changes. The trigger invokes build jobs and build container images for the microservices. Tag the images with a version number, and push them to Cloud Storage.
- C. Create a Scheduler job to check the repo every minute. For any new change, invoke Cloud Build to build container images for the microservices. Tag the images using the current timestamp, and push them to the Container Registry.
- D. Configure a trigger in Cloud Build for new source changes. Invoke Cloud Build to build one container image, and tag the image with the label 'latest.' Push the image to the Container Registry.

Introductory Info

Company overview -

TerramEarth manufactures heavy equipment for the mining and agricultural industries. They currently have over 500 dealers and service centers in 100 countries.

Their mission is to build products that make their customers more productive.

Solution concept -

There are 2 million TerramEarth vehicles in operation currently, and we see 20% yearly growth. Vehicles collect telemetry data from many sensors during operation. A small subset of critical data is transmitted from the vehicles in real time to facilitate fleet management. The rest of the sensor data is collected, compressed, and uploaded daily when the vehicles return to home base. Each vehicle usually generates 200 to 500 megabytes of data per day.

Existing technical environment -

TerramEarth's vehicle data aggregation and analysis infrastructure resides in Google Cloud and serves clients from all around the world. A growing amount of sensor data is captured from their two main manufacturing plants and sent to private data centers that contain their legacy inventory and logistics management systems. The private data centers have multiple network interconnects configured to Google Cloud. The web frontend for dealers and customers is running in Google Cloud and allows access to stock management and analytics.

Business requirements -

- * Predict and detect vehicle malfunction and rapidly ship parts to dealerships for just-in-time repair where possible.
- * Decrease cloud operational costs and adapt to seasonality.
- * Increase speed and reliability of development workflow.
- * Allow remote developers to be productive without compromising code or data security.
- * Create a flexible and scalable platform for developers to create custom API services for dealers and partners.

Technical requirements -

- * Create a new abstraction layer for HTTP API access to their legacy systems to enable a gradual move into the cloud without disrupting operations.
- * Modernize all CI/CD pipelines to allow developers to deploy container-based workloads in highly scalable environments.
- * Allow developers to run experiments without compromising security and governance requirements.
- * Create a self-service portal for internal and partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints.
- * Use cloud-native solutions for keys and secrets management and optimize for identity-based access.
- * Improve and standardize tools necessary for application and network monitoring and troubleshooting.

Executive statement -

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize vehicle downtimes.

After moving multiple systems into Google Cloud, we are seeking new ways to provide best-in-class online fleet management services to our customers and improve operations of our dealerships. Our 5-year strategic plan is to create a partner ecosystem of new products by enabling access to our data, increasing autonomous operation capabilities of our vehicles, and creating a path to move the remaining legacy systems to the cloud.

Question

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

- A. Request Transfer Appliances from Google Cloud, export the data to appliances, and return the appliances to Google Cloud.
- B. Configure the Storage Transfer service from Google Cloud to send the data from your data center to Cloud Storage.
- C. Make sure there are no other users consuming the 1Gbps link, and use multi-thread transfer to upload the data to Cloud Storage.
- D. Export files to an encrypted USB device, send the device to Google Cloud, and request an import of the data to Cloud Storage.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

The Dress4Win security team has disabled external SSH access into production virtual machines (VMs) on Google Cloud Platform (GCP).

The operations team needs to remotely manage the VMs, build and push Docker containers, and manage Google Cloud Storage objects.

What can they do?

- A. Grant the operations engineer access to use Google Cloud Shell.
- B. Configure a VPN connection to GCP to allow SSH access to the cloud VMs.
- C. Develop a new access request process that grants temporary SSH access to cloud VMs when an operations engineer needs to perform a task.
- D. Have the development team build an API service that allows the operations team to execute specific remote procedure calls to accomplish their tasks.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

- Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.
- Improve business agility and speed of innovation through rapid provisioning of new resources.
- Analyze and optimize architecture for performance in the cloud.
- Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

- Evaluate and choose an automation framework for provisioning resources in cloud.
- Support failover of the production environment to cloud during an emergency.
- Identify production services that can migrate to cloud to save capacity.
- Use managed services whenever possible.
- Encrypt data on the wire and at rest.
- Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

At Dress4Win, an operations engineer wants to create a tow-cost solution to remotely archive copies of database backup files.

The database files are compressed tar files stored in their current data center.

How should he proceed?

- A. Create a cron script using gsutil to copy the files to a Coldline Storage bucket.
- B. Create a cron script using gsutil to copy the files to a Regional Storage bucket.
- C. Create a Cloud Storage Transfer Service Job to copy the files to a Coldline Storage bucket.
- D. Create a Cloud Storage Transfer Service job to copy the files to a Regional Storage bucket.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

- Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.
- Improve business agility and speed of innovation through rapid provisioning of new resources.
- Analyze and optimize architecture for performance in the cloud.
- Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

- Evaluate and choose an automation framework for provisioning resources in cloud.
- Support failover of the production environment to cloud during an emergency.
- Identify production services that can migrate to cloud to save capacity.
- Use managed services whenever possible.
- Encrypt data on the wire and at rest.
- Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

Dress4Win has asked you to recommend machine types they should deploy their application servers to.

How should you proceed?

- A. Perform a mapping of the on-premises physical hardware cores and RAM to the nearest machine types in the cloud.
- B. Recommend that Dress4Win deploy application servers to machine types that offer the highest RAM to CPU ratio available.
- C. Recommend that Dress4Win deploy into production with the smallest instances available, monitor them over time, and scale the machine type up until the desired performance is reached.
- D. Identify the number of virtual cores and RAM associated with the application server virtual machines align them to a custom machine type in the cloud, monitor performance, and scale the machine types up until the desired performance is reached.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

As part of Dress4Win's plans to migrate to the cloud, they want to be able to set up a managed logging and monitoring system so they can handle spikes in their traffic load.

They want to ensure that:

- * The infrastructure can be notified when it needs to scale up and down to handle the ebb and flow of usage throughout the day
- * Their administrators are notified automatically when their application reports errors.
- * They can filter their aggregated logs down in order to debug one piece of the application across many hosts

Which Google StackDriver features should they use?

- A. Logging, Alerts, Insights, Debug
- B. Monitoring, Trace, Debug, Logging
- C. Monitoring, Logging, Alerts, Error Reporting
- D. Monitoring, Logging, Debug, Error Report

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

- Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.
- Improve business agility and speed of innovation through rapid provisioning of new resources.
- Analyze and optimize architecture for performance in the cloud.
- Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

- Evaluate and choose an automation framework for provisioning resources in cloud.
- Support failover of the production environment to cloud during an emergency.
- Identify production services that can migrate to cloud to save capacity.
- Use managed services whenever possible.
- Encrypt data on the wire and at rest.
- Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

Dress4Win would like to become familiar with deploying applications to the cloud by successfully deploying some applications quickly, as is. They have asked for your recommendation.

What should you advise?

- A. Identify self-contained applications with external dependencies as a first move to the cloud.
- B. Identify enterprise applications with internal dependencies and recommend these as a first move to the cloud.
- C. Suggest moving their in-house databases to the cloud and continue serving requests to on-premise applications.
- D. Recommend moving their message queuing servers to the cloud and continue handling requests to on-premise applications.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

Dress4Win has asked you for advice on how to migrate their on-premises MySQL deployment to the cloud.

They want to minimize downtime and performance impact to their on-premises solution during the migration.

Which approach should you recommend?

- A. Create a dump of the on-premises MySQL master server, and then shut it down, upload it to the cloud environment, and load into a new MySQL cluster.
- B. Setup a MySQL replica server/slave in the cloud environment, and configure it for asynchronous replication from the MySQL master server on-premises until cutover.
- C. Create a new MySQL cluster in the cloud, configure applications to begin writing to both on premises and cloud MySQL masters, and destroy the original cluster at cutover.
- D. Create a dump of the MySQL replica server into the cloud environment, load it into: Google Cloud Datastore, and configure applications to read/write to Cloud Datastore at cutover.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

Dress4Win has configured a new uptime check with Google Stackdriver for several of their legacy services. The Stackdriver dashboard is not reporting the services as healthy.

What should they do?

- A. Install the Stackdriver agent on all of the legacy web servers.
- B. In the Cloud Platform Console download the list of the uptime servers' IP addresses and create an inbound firewall rule
- C. Configure their load balancer to pass through the User-Agent HTTP header when the value matches GoogleStackdriverMonitoring-UptimeChecks ([https:// cloud.google.com/monitoring](https://cloud.google.com/monitoring))
- D. Configure their legacy web servers to allow requests that contain user-Agent HTTP header when the value matches GoogleStackdriverMonitoring- UptimeChecks (<https://cloud.google.com/monitoring>)

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

▪

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

As part of their new application experience, Dress4Wm allows customers to upload images of themselves.

The customer has exclusive control over who may view these images.

Customers should be able to upload images with minimal latency and also be shown their images quickly on the main application page when they log in.

Which configuration should Dress4Win use?

- A. Store image files in a Google Cloud Storage bucket. Use Google Cloud Datastore to maintain metadata that maps each customer's ID and their image files.
- B. Store image files in a Google Cloud Storage bucket. Add custom metadata to the uploaded images in Cloud Storage that contains the customer's unique ID.
- C. Use a distributed file system to store customers' images. As storage needs increase, add more persistent disks and/or nodes. Assign each customer a unique ID, which sets each file's owner attribute, ensuring privacy of images.
- D. Use a distributed file system to store customers' images. As storage needs increase, add more persistent disks and/or nodes. Use a Google Cloud SQL database to maintain metadata that maps each customer's ID to their image files.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

Dress4Win has end-to-end tests covering 100% of their endpoints.

They want to ensure that the move to the cloud does not introduce any new bugs.

Which additional testing methods should the developers employ to prevent an outage?

- A. They should enable Google Stackdriver Debugger on the application code to show errors in the code.
- B. They should add additional unit tests and production scale load tests on their cloud staging environment.
- C. They should run the end-to-end tests in the cloud staging environment to determine if the code is working as intended.
- D. They should add canary tests so developers can measure how much of an impact the new release causes to latency.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

- Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.
- Improve business agility and speed of innovation through rapid provisioning of new resources.
- Analyze and optimize architecture for performance in the cloud.
- Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

- Evaluate and choose an automation framework for provisioning resources in cloud.
- Support failover of the production environment to cloud during an emergency.
- Identify production services that can migrate to cloud to save capacity.
- Use managed services whenever possible.
- Encrypt data on the wire and at rest.
- Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

You want to ensure Dress4Win's sales and tax records remain available for infrequent viewing by auditors for at least 10 years.

Cost optimization is your top priority.

Which cloud services should you choose?

- A. Google Cloud Storage Coldline to store the data, and gsutil to access the data.
- B. Google Cloud Storage Nearline to store the data, and gsutil to access the data.
- C. Google Bigtabte with US or EU as location to store the data, and gcloud to access the data.
- D. BigQuery to store the data, and a web server cluster in a managed instance group to access the data. Google Cloud SQL mirrored across two distinct regions to store the data, and a Redis cluster in a managed instance group to access the data.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a website and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a premium app model.

Company Background -

Dress4Win's application has grown from a few servers in the founder's garage to several hundred servers and appliances in a collocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is considering moving their development and test environments. They are also considering building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location.

Databases:

- MySQL - user data, inventory, static data
- Redis - metadata, social graph, caching

Application servers:

- Tomcat - Java micro-services
- Nginx - static content
- Apache Beam - Batch processing

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- NAS - image storage, logs, backups

Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations

MQ servers:

- Messaging
- Social notifications
- Events

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

-

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Migrate fully to the cloud if all other requirements are met.

Technical Requirements -

Evaluate and choose an automation framework for provisioning resources in cloud.

Support failover of the production environment to cloud during an emergency.

Identify production services that can migrate to cloud to save capacity.

Use managed services whenever possible.

Encrypt data on the wire and at rest.

Support multiple VPN connections between the production data center and cloud environment.

CEO Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a new

competitor could use a public cloud platform to offset their up-front investment and freeing them to focus on developing better features.

CTO Statement -

We have invested heavily in the current infrastructure, but much of the equipment is approaching the end of its useful life. We are consistently waiting weeks for new gear to be racked before we can start new projects. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

CFO Statement -

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years puts a cloud strategy between 30 to 50% lower than our current model.

Question

The current Dress4Win system architecture has high latency to some customers because it is located in one data center.

As of a future evaluation and optimizing for performance in the cloud, Dressss4Win wants to distribute its system architecture to multiple locations when Google cloud platform.

Which approach should they use?

- A. Use regional managed instance groups and a global load balancer to increase performance because the regional managed instance group can grow instances in each region separately based on traffic.
- B. Use a global load balancer with a set of virtual machines that forward the requests to a closer group of virtual machines managed by your operations team.
- C. Use regional managed instance groups and a global load balancer to increase reliability by providing automatic failover between zones in different regions.
- D. Use a global load balancer with a set of virtual machines that forward the requests to a closer group of virtual machines as part of a separate managed instance groups.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. Dress4Win is expected to grow to 10 times its size in 1 year with a corresponding growth in data and traffic that mirrors the existing patterns of usage. The CIO has set the target of migrating production infrastructure to the cloud within the next 6 months. How will you configure the solution to scale for this growth without making major application changes and still maximize the ROI?

- A. Migrate the web application layer to App Engine, and MySQL to Cloud Datastore, and NAS to Cloud Storage. Deploy RabbitMQ, and deploy Hadoop servers using Deployment Manager.
- B. Migrate RabbitMQ to Cloud Pub/Sub, Hadoop to BigQuery, and NAS to Compute Engine with Persistent Disk storage. Deploy Tomcat, and deploy Nginx using Deployment Manager.
- C. Implement managed instance groups for Tomcat and Nginx. Migrate MySQL to Cloud SQL, RabbitMQ to Cloud Pub/Sub, Hadoop to Cloud Dataproc, and NAS to Compute Engine with Persistent Disk storage.
- D. Implement managed instance groups for the Tomcat and Nginx. Migrate MySQL to Cloud SQL, RabbitMQ to Cloud Pub/Sub, Hadoop to Cloud Dataproc, and NAS to Cloud Storage.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. Considering the given business requirements, how would you automate the deployment of web and transactional data layers?

A. Deploy Nginx and Tomcat using Cloud Deployment Manager to Compute Engine. Deploy a Cloud SQL server to replace MySQL. Deploy Jenkins using Cloud Deployment Manager.

B. Deploy Nginx and Tomcat using Cloud Launcher. Deploy a MySQL server using Cloud Launcher. Deploy Jenkins to Compute Engine using Cloud Deployment Manager scripts.

C. Migrate Nginx and Tomcat to App Engine. Deploy a Cloud Datastore server to replace the MySQL server in a high-availability configuration. Deploy Jenkins to Compute Engine using Cloud Launcher.

D. Migrate Nginx and Tomcat to App Engine. Deploy a MySQL server using Cloud Launcher. Deploy Jenkins to Compute Engine using Cloud Launcher.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. Which of the compute services should be migrated as-is and would still be an optimized architecture for performance in the cloud?

- A. Web applications deployed using App Engine standard environment
- B. RabbitMQ deployed using an unmanaged instance group
- C. Hadoop/Spark deployed using Cloud Dataproc Regional in High Availability mode
- D. Jenkins, monitoring, bastion hosts, security scanners services deployed on custom machine types

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. To be legally compliant during an audit, Dress4Win must be able to give insights in all administrative actions that modify the configuration or metadata of resources on Google Cloud.

What should you do?

- A. Use Stackdriver Trace to create a Trace list analysis.
- B. Use Stackdriver Monitoring to create a dashboard on the project's activity.
- C. Enable Cloud Identity-Aware Proxy in all projects, and add the group of Administrators as a member.
- D. Use the Activity page in the GCP Console and Stackdriver Logging to provide the required insight.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. You are responsible for the security of data stored in Cloud Storage for your company, Dress4Win. You have already created a set of Google Groups and assigned the appropriate users to those groups. You should use Google best practices and implement the simplest design to meet the requirements.

Considering Dress4Win's business and technical requirements, what should you do?

A. Assign custom IAM roles to the Google Groups you created in order to enforce security requirements. Encrypt data with a customer-supplied encryption key when storing files in Cloud Storage.

B. Assign custom IAM roles to the Google Groups you created in order to enforce security requirements. Enable default storage encryption before storing files in Cloud Storage.

C. Assign predefined IAM roles to the Google Groups you created in order to enforce security requirements. Utilize Google's default encryption at rest when storing files in Cloud Storage.

D. Assign predefined IAM roles to the Google Groups you created in order to enforce security requirements. Ensure that the default Cloud KMS key is set before storing files in Cloud Storage.

Introductory Info

Company Overview -

Dress4Win is a web-based company that helps their users organize and manage their personal wardrobe using a web app and mobile application. The company also cultivates an active social network that connects their users with designers and retailers. They monetize their services through advertising, e-commerce, referrals, and a freemium app model. The application has grown from a few servers in the founder's garage to several hundred servers and appliances in a colocated data center. However, the capacity of their infrastructure is now insufficient for the application's rapid growth. Because of this growth and the company's desire to innovate faster, Dress4Win is committing to a full migration to a public cloud.

Solution Concept -

For the first phase of their migration to the cloud, Dress4Win is moving their development and test environments. They are also building a disaster recovery site, because their current infrastructure is at a single location. They are not sure which components of their architecture they can migrate as is and which components they need to change before migrating them.

Existing Technical Environment -

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

MySQL. 1 server for user data, inventory, static data:

- MySQL 5.8
- 8 core CPUs
- 128 GB of RAM
- 2x 5 TB HDD (RAID 1)

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM

Compute:

40 Web Application servers providing micro-services based APIs and static content.

~"

- Tomcat

Java -

- Nginx
- 4 core CPUs
- 32 GB of RAM

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM

Miscellaneous servers:

- Jenkins, monitoring, bastion hosts, security scanners
- 8 core CPUs
- 32GB of RAM

Storage appliances:

iSCSI for VM hosts

Fiber channel SAN ~" MySQL databases

- 1 PB total storage; 400 TB available

NAS ~" image storage, logs, backups

- 100 TB total storage; 35 TB available

Business Requirements -

Build a reliable and reproducible environment with scaled parity of production.

Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.

Improve business agility and speed of innovation through rapid provisioning of new resources.

Analyze and optimize architecture for performance in the cloud.

Technical Requirements -

Easily create non-production environments in the cloud.

Implement an automation framework for provisioning resources in cloud.

Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.

Support failover of the production environment to cloud during an emergency.

Encrypt data on the wire and at rest.

Support multiple private connections between the production data center and cloud environment.

Executive Statement -

Our investors are concerned about our ability to scale and contain costs with our current infrastructure. They are also concerned that a competitor could use a public cloud platform to offset their up-front investment and free them to focus on developing better features. Our traffic patterns are highest in the mornings and weekend evenings; during other times, 80% of our capacity is sitting idle.

Our capital expenditure is now exceeding our quarterly projections. Migrating to the cloud will likely cause an initial increase in spending, but we expect to fully transition before our next hardware refresh cycle. Our total cost of ownership (TCO) analysis over the next 5 years for a public cloud strategy achieves a cost reduction between 30% and 50% over our current model.

Question

For this question, refer to the Dress4Win case study. You want to ensure that your on-premises architecture meets business requirements before you migrate your solution.

What change in the on-premises architecture should you make?

- A. Replace RabbitMQ with Google Pub/Sub.
- B. Downgrade MySQL to v5.7, which is supported by Cloud SQL for MySQL.
- C. Resize compute resources to match predefined Compute Engine machine types.
- D. Containerize the micro-services and host them in Google Kubernetes Engine.