1. Question

A company wants to build an application that stores images in a Cloud Storage bucket and wants to generate thumbnails as well as resize the images. They want to use a google managed service that can scale up and scale down to zero automatically with minimal effort. You have been asked to recommend a service. Which GCP service would you suggest?

- Google Compute Engine
- Google Kubernetes Engine
- **Cloud Functions**
- **Google App Engine**

Incorrect

Cloud Functions. is the right answer.
Cloud Functions is Google Cloud's event-driven serverless compute platform. It automatically scales based on the load and requires no additional configuration. You pay only for the resources used.
Ref: https://cloud.google.com/functions
While all other options i.e. Google Compute Engine, Google Kubernetes Engine, Google App Engine support autoscaling, it needs to be configured explicitly based on the load and is not as trivial as the scale up or scale down offered by Google's cloud functions.

2. Question

A team of data scientists infrequently needs to use a Google Kubernetes Engine (GKE) cluster that you manage. They require GPUs for some long-running, non- restartable jobs. You want to minimize cost. What should you do?

- Enable node auto-provisioning on the GKE cluster.
- Create a VerticalPodAutscaler for those workloads.
- Create a node pool with preemptible VMs and GPUs attached to those VMs.
- **Create a node pool of instances with GPUs, and enable autoscaling on this node pool with a minimum size of 1.**

Unattempted

Enable node auto-provisioning on the GKE cluster. is not right.
Node auto-provisioning automatically manages a set of node pools on the user's behalf. Without Node auto-provisioning, GKE considers starting new nodes only from the set of user-created node pools. With node auto-provisioning, new node pools can be created and deleted automatically. This in no way helps us with our requirements.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-provisioning

Create a VerticalPodAutscaler for those workloads. is not right.
Vertical pod autoscaling (VPA) frees you from having to think about what values to specify for a container's CPU and memory requests. The autoscaler can recommend values for CPU and memory requests and limits, or it can automatically update the values. This doesn't help us with the GPU requirement. Moreover, due to Kubernetes limitations, the only way to modify the resource requests of a running Pod is to recreate the Pod. This has the negative effect of killing the non-restartable jobs which is undesirable.
https://cloud.google.com/kubernetes-engine/docs/concepts/verticalpodautoscaler#overview

Create a node pool with preemptible VMs and GPUs attached to those VMs. is not right.
You can use preemptible VMs in your GKE clusters or node pools to run batch or fault-tolerant jobs that are less sensitive to the ephemeral, non-guaranteed nature of preemptible VMs. Whereas we have long-running and non-restartable jobs so preemptible VMs aren't suitable for our requirement.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/preemptible-vms

Create a node pool of instances with GPUs, and enable autoscaling on this node pool with a minimum size of 1. is the right answer.
A node pool is a group of nodes within a cluster that all have the same configuration. Our requirement is GPUs, so we create a node pool with GPU enabled and have the scientist's applications deployed to the cluster and use this node pool. At the same time, you want to minimize cost so you start with 1 instance and scale up as needed. It is important to note that the scale down needs to take into consideration if there are any running jobs otherwise the scale down may terminate the nonrestartable job.
Ref: https://cloud.google.com/kubernetes-engine/docs/concepts/node-pools

- **3. Question**

- An employee was terminated, but their access to Google Cloud Platform (GCP) was not removed until 2 weeks later. You need to find out this employee accessed any sensitive customer information after their termination. What should you do?

.

- View System Event Logs in Stackdriver. Search for the user's email as the principal.
- View System Event Logs in Stackdriver. Search for the service account associated with the user.
- **View Data Access audit logs in Stackdriver. Search for the user's email as the principal.**
- View the Admin Activity log in Stackdriver. Search for the service account associated with the user.

Unattempted

View the Admin Activity log in Stackdriver. Search for the service account associated with the user. is not right.
Admin Activity logs do not contain log entries for reading resource data. Admin Activity audit logs contain log entries for API calls or other administrative actions that modify the configuration or metadata of resources.
Ref: https://cloud.google.com/logging/docs/audit#admin-activity

View System Event Logs in Stackdriver. Search for the user's email as the principal. is not right.
System Event audit logs do not contain log entries for reading resource data. System Event audit logs contain log entries for Google Cloud administrative actions that modify the configuration of resources. System Event audit logs are generated by Google systems; they are not driven by direct user action.
Ref: https://cloud.google.com/logging/docs/audit#system-event

View System Event Logs in Stackdriver. Search for the service account associated with the user. is not right.
System Event audit logs do not contain log entries for reading resource data. System Event audit logs contain log entries for Google Cloud administrative actions that modify the configuration of resources. System Event audit logs are generated by Google systems; they are not driven by direct user action.
Ref: https://cloud.google.com/logging/docs/audit#system-event

View Data Access audit logs in Stackdriver. Search for the user's email as the principal. is the right answer.
Data Access audit logs contain API calls that read the configuration or metadata of resources, as well as user-driven API calls that create, modify, or read user-provided resource data.
Ref: https://cloud.google.com/logging/docs/audit#data-access

4. Question

An engineer from your team accidentally deployed several new versions of NodeJS application on Google App Engine Standard. You are concerned the new versions are serving traffic. You have been asked to produce a list of all the versions of the application that are receiving traffic as well the percent traffic split between them. What should you do?

- **gcloud app versions list --hide-no-traffic**
- gcloud app versions list --show-traffic
- gcloud app versions list
- gcloud app versions list --traffic

Unattempted

gcloud app versions list. is not right
This command lists all the versions of all services that are currently deployed to the App Engine server. While this list includes all versions that are receiving traffic, it also includes versions that are not receiving traffic.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/list

gcloud app versions list –traffic. is not right
gcloud app versions list command does not support –traffic flag.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/list

gcloud app versions list –show-traffic. is not right
gcloud app versions list command does not support –show-traffic flag.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/list

gcloud app versions list –hide-no-traffic. is the right answer.
This command correctly lists just the versions that are receiving traffic by hiding versions that do not receive traffic. This is the only command that fits our requirements.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/list

5. Question

An intern joined your team recently and needs access to Google Compute Engine in your sandbox project to explore various settings and spin up compute instances to test features. You have been asked to facilitate this. How should you give your intern access to compute engine without giving more permissions than is necessary?

.

- Grant Project Editor IAM role for sandbox project.
- Grant Compute Engine Admin Role for sandbox project.
- Create a shared VPC to enable the intern access Compute resources.
- **Grant Compute Engine Instance Admin Role for the sandbox project.**

Unattempted

Create a shared VPC to enable the intern access Compute resources. is not right.
Creating a shared VPC is not sufficient to grant intern access to compute resources. Shared VPCs are primarily used by organizations to connect resources from multiple projects to a common Virtual Private Cloud (VPC) network, so that they can communicate with each other securely and efficiently using internal IPs from that network.
Ref: https://cloud.google.com/vpc/docs/shared-vpc

Grant Project Editor IAM role for sandbox project. is not right.
Project editor role grants all viewer permissions, plus permissions for actions that modify state, such as changing existing resources. While this role lets the intern explore compute engine settings and spin up compute instances, it grants more permissions than what is needed. Our intern can modify any resource in the project.
https://cloud.google.com/iam/docs/understanding-roles#primitive_roles

Grant Compute Engine Admin Role for sandbox project. is not right.
Compute Engine Admin Role grants full control of all Compute Engine resources; including networks, load balancing, service accounts etc. While this role lets the intern explore compute engine settings and spin up compute instances, it grants more permissions than what is needed.
Ref: https://cloud.google.com/compute/docs/access/iam#compute.storageAdmin

Grant Compute Engine Instance Admin Role for the sandbox project. is the right answer.
Compute Engine Instance Admin Role grants full control of Compute Engine instances, instance groups, disks, snapshots, and images. It also provides read access to all Compute Engine networking resources. This provides just the required permissions to the intern.
Ref: https://cloud.google.com/compute/docs/access/iam#compute.storageAdmin

.

6. Question

.

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?

.

- Enable Logging export to Google Cloud Storage (GCS) bucket and delegate access to the bucket
- **Enable Logging export to Google BigQuery and use ACLs and views to scope the data shared with the auditor**
- Create custom Google Stackdriver alerts and send them to the auditor
- Use Cloud Functions to transfer log entries to Google Cloud SQL and use ACLs and views to limit an auditor's view

Unattempted

Create custom Google Stackdriver alerts and send them in an email to the auditor. is not right.
Stackdriver Alerting gives timely awareness to problems in your cloud applications so you can resolve the problems quickly. Sending alerts to your auditor is not of much use during audits.
Ref: https://cloud.google.com/monitoring/alerts

Use Cloud Functions to transfer log entries to Google Cloud SQL and use ACLs and views to limit an auditor's view. is not right.
Using Cloud Functions to transfer log entries to Google Cloud SQL is expensive in comparison to audit logs export feature which exports logs to various destinations with minimal configuration.
Ref: https://cloud.google.com/logging/docs/export/
Auditors spend a lot of time reviewing log messages. And you want to expedite the audit process!! So you want to make it easier for the auditor to extract the information easily from the logs.

Between the two remaining options, the only difference is the log export sink destination
Ref: https://cloud.google.com/logging/docs/export/

One option exports to Google Cloud Storage (GCS) bucket whereas other exports to BigQuery. Querying information out of files in a bucket is much harder compared to querying information from BigQuery Dataset where it is as simple as running a job or set of jobs to extract just the required information and in the format required. By enabling

the auditor to run jobs in Big Queries, you streamline the log extraction process and the auditor can review the extracted logs much quicker. While as good as the other option (bucket) is, Enable Logging export to Google BigQuery and use ACLs and views to scope the data shared with the auditor is the right answer.

You need to configure log sinks before you can receive any logs, and you can't retroactively export logs that were written before the sink was created.

.

7. Question

.

Every employee of your company has a Google account. Your operational team needs to manage a large number of instances on the Compute Engine. Each member of this team needs only administrative access to the servers. Your security team wants to ensure that the deployment of credentials is operationally efficient and must be able to determine who accessed a given instance. What should you do?

.

- Ask each member of the team to generate a new SSH key pair and to send you their public key. Use a configuration management tool to deploy those keys on each instance.
- **Ask each member of the team to generate a new SSH key pair and to add the public key to their Google account. Grant the "compute.osAdminLogin" role to the Google group corresponding to this team.**
- Generate a new SSH key pair. Give the private key to each member of your team. Configure the public key in the metadata of each instance.
- Generate a new SSH key pair. Give the private key to each member of your team. Configure the public key as a project-wide public SSH key in your Cloud Platform project and allow project-wide public SSH keys on each instance.

Unattempted

Generate a new SSH key pair. Give the private key to each member of your team. Configure the public key in the metadata of each instance. is not right.
Reuse of a single SSH key pair by all employees is a very bad security practice as auditing becomes very impossible.

Generate a new SSH key pair. Give the private key to each member of your team. Configure the public key as a project-wide public SSH key in your Cloud Platform project and allow project-wide public SSH keys on each instance. is not right.

Reuse of a single SSH key pair by all employees is a very bad security practice as auditing becomes very impossible.

Ask each member of the team to generate a new SSH key pair and to send you their public key. Use a configuration management tool to deploy those keys on each instance. is not right.
While this can be done, it is not operationally efficient. Let's say a user leaves the company, you then have to remove their SSH key from all instances where it has been added (can't be removed at a single place). Similarly, when a user joins the company, you have to add their SSH key to all the instances. This is very tedious and not operationally efficient.

Ask each member of the team to generate a new SSH key pair and to add the public key to their Google account. Grant the "compute.osAdminLogin" role to the Google group corresponding to this team. is the right answer.
By letting users manage their own SSH key pair (and it's rotation etc), you delete the operational burden of managing SSH keys to individual users. Secondly, granting compute.osAdminLogin grants the group administrator permissions (as opposed to granting compute.osLogin, which does not grant administrator permissions). Finally, managing provisioning and de-provisioning is as simple as adding or removing the user from the group.

OS Login lets you use Compute Engine IAM roles to efficiently manage SSH access to Linux instances and is an alternative to manually managing instance access by adding and removing SSH keys in the metadata. Before you can manage instance access using IAM roles, you must enable the OS Login feature by setting a metadata key-value pair in your project or in your instance's metadata: enable-oslogin=TRUE. After you enable OS Login on one or more instances in your project, those instances accept connections only from user accounts that have the necessary IAM roles in your project or organization. There are two predefined roles.
? roles/compute.osLogin, which does not grant administrator permissions
? roles/compute.osAdminLogin, which grants administrator permissions
At any point, to revoke user access to instances that are enabled to use OS Login, remove the user roles from that user account
Ref: https://cloud.google.com/compute/docs/instances/managing-instance-access#enable_oslogin

- **8. Question**
-

For service discovery, you need to associate each of the Compute Engine instances of your VPC with an internal (DNS) record in a custom zone. You want to follow Google recommended practices. What should you do?

.

- Create a new VPC, block all external traffic with a firewall rule and create 2 Cloud DNS zones - a first zone in the new VPC and a second zone in the main VPC that is forwarding requests to the first Cloud DNS zone. Create records for each instance in the first zone.
- Deploy the BIND DNS server in the VPC, and create a Cloud DNS forwarding zone to forward the DNS requests to BIND. Create records for each instance in the BIND DNS server.
- **Create a Cloud DNS zone, set its visibility to private and associate it with your VPC. Create records for each instance in that zone.**
- Create your Compute Engine instances with custom hostnames.

Unattempted

Our requirements here are 1. Internal and 2. Custom Zone

Create your Compute Engine instances with custom hostnames. is not right.
This doesn't put them in a custom zone.

Deploy the BIND DNS server in the VPC, and create a Cloud DNS forwarding zone to forward the DNS requests to BIND. Create records for each instance in the BIND DNS server. is not right.
This might be possible but not something Google recommends. The Cloud DNS service offering from Google already offers these features so it is pointless installing a custom DNS server to do that.

Create a new VPC, block all external traffic with a firewall rule and create 2 Cloud DNS zones – a first zone in the new VPC and a second zone in the main VPC that is forwarding requests to the first Cloud DNS zone. Create records for each instance in the first zone. is not right.
This doesn't make any sense, moreover, the two VPCs can't communicate without VPC peering.
Ref: https://cloud.google.com/dns/docs/overview#concepts

Create a Cloud DNS zone, set its visibility to private and associate it with your VPC. Create records for each instance in that zone. is the right answer.
You should absolutely do this when you want internal DNS records in a custom zone.

Cloud DNS gives you the option of private zones and internal DNS names.
Ref: https://cloud.google.com/dns/docs/overview#concepts

.

9. Question

.

In Cloud Shell, your active gcloud configuration is as shown below.
$ gcloud config list
[component_manager]
disable_update_check = True
[compute]
gce_metadata_read_timeout_sec = 5
zone = europe-west2-a
[core]
account = gcp-ace-lab-user@gmail.com
disable_usage_reporting = False
project = gcp-ace-lab-266520
[metrics]
environment = devshell
You want to create two compute instances – one in europe-west2-a and another in europe-west2-b. What should you do? (Select 2)

.

- gcloud compute instances create instance1 gcloud compute instances create instance2
- **gcloud compute instances create instance1 gcloud config set compute/zone europe-west2-b gcloud compute instances create instance2**
- **gcloud compute instances create instance1 gcloud compute instances create instance2 --zone=europe-west2-b**
- gcloud compute instances create instance1 gcloud config set zone europe-west2-b gcloud compute instances create instance2
- gcloud compute instances create instance1 gcloud configuration set compute/zone europe-west2-b gcloud compute instances create instance2

Unattempted

gcloud compute instances create instance1
gcloud compute instances create instance2. is not right.
The default compute/zone property is set to europe-west2-a in the current gcloud configuration. Executing the two commands above would create two compute instances in the default zone i.e. europe-west2-a which doesn't satisfy our requirement.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create

gcloud compute instances create instance1
gcloud config set zone europe-west2-b
gcloud compute instances create instance2. is not right.
The approach is right but the syntax is wrong. gcloud config does not have a core/zone property. The syntax for this command is gcloud config set SECTION/PROPERTY VALUE. If SECTION is missing, SECTION is defaulted to core. We are effectively trying to run gcloud config set core/zone europe-west2-b but the core section doesn't have a property called zone, so this command fails.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/set

gcloud compute instances create instance1
gcloud configuration set compute/zone europe-west2-b
gcloud compute instances create instance2. is not right.
Like above, the approach is right but the syntax is wrong. You want to set the default compute/zone property in gcloud configuration to europe-west2-b but it needs to be done via the command gcloud config set and not gcloud configuration set.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/set

gcloud compute instances create instance1
gcloud config set compute/zone europe-west2-b
gcloud compute instances create instance2. is the right answer.
The default compute/zone property is europe-west2-a in the current gcloud configuration so executing the first gcloud compute instances create command creates the instance in europe-west2-a zone. Next, executing the gcloud config set compute/zone europe-west2-b changes the default compute/zone property in default configuration to europe-west2-b. Executing the second gcloud compute instances create command creates a compute instance in europe-west2-b which is what we want.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/set
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create

gcloud compute instances create instance1
gcloud compute instances create instance2 –zone=europe-west2-b. is the right answer.
The default compute/zone property is europe-west2-a in the current gcloud configuration so executing the first gcloud compute instances create command creates the instance in europe-west2-a zone. Next, executing the second gcloud compute instances create command with –zone property creates a compute instance in provided zone i.e. europe-west2-b instead of using the default zone from the current active configuration.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create

.

10. Question

.

In Regional Storage buckets with object versioning enabled, what is the effect of deleting the live version of an object and deleting a noncurrent version of an object?

.

- **1. The live version becomes a noncurrent version. 2. The noncurrent version is deleted permanently.**
- 1. The live version becomes a noncurrent version and a lifecycle rule is applied to delete after 30 days. 2. A lifecycle rule is applied on the noncurrent version to delete after 30 days.
- 1. The live version becomes a noncurrent version and a lifecycle rule is applied to transition to Nearline Storage after 30 days. 2. A lifecycle rule is applied on the noncurrent version to transition to Nearline Storage after 30 days.
- 1. The live version is deleted permanently. 2. The noncurrent version is deleted permanently.

Unattempted

1. The live version becomes a noncurrent version.
2. The noncurrent version is deleted permanently. is the right answer.

In buckets with object versioning enabled, deleting the live version of an object creates a noncurrent version while deleting a noncurrent version deletes that version permanently.
Ref: https://cloud.google.com/storage/docs/lifecycle#actions

.

11. Question

.

Several employees at your company have been creating projects with Cloud Platform and paying for it with their personal credit cards, which the company reimburses. The company wants to centralize all these projects under a single, new billing account. What should you do?

.

- Contact cloud-billing@google.com with your bank account details and request a corporate billing account for your company.
- In the Google Cloud Platform Console, create a new billing account and set up a payment method.
- **In the Google Platform Console, go to the Resource Manage and move all projects to the root Organization.**

- Create a ticket with Google Support and wait for their call to share your credit card details over the phone.

Unattempted

Contact cloud-billing@google.com with your bank account details and request a corporate billing account for your company. is not right.
That is not how we set up billing for the organization.
Ref: https://cloud.google.com/billing/docs/concepts

Create a ticket with Google Support and wait for their call to share your credit card details over the phone. is not right.
That is not how we set up billing for the organization.
Ref: https://cloud.google.com/billing/docs/concepts

In the Google Cloud Platform Console, create a new billing account and set up a payment method. is not right.
Unless all projects are modified to use the new billing account, this doesn't work.
Ref: https://cloud.google.com/billing/docs/concepts

In the Google Platform Console, go to the Resource Manage and move all projects to the root Organization. is the right answer.
If we move all projects under the root organization hierarchy, they need to use a billing account within the root organization. We can then consolidate all the costs under different billing accounts as needed e.g. per project, or one for dev work and another billing account for production usage, etc.
Ref: https://cloud.google.com/billing/docs/concepts

.

12. Question

.

The storage costs for your application logs have far exceeded the project budget. The logs are currently being retained indefinitely in the Cloud Storage bucket myapp-gcp-ace-logs. You have been asked to remove logs older than 90 days from your Cloud Storage bucket. You want to optimize ongoing Cloud Storage spend. What should you do?

.

- Write a script that runs gsutil ls -l gs://myapp-gcp-ace-logs/** to find and remove items older than 90 days. Schedule the script with cron.
- Write a script that runs gsutil ls -lr gs://myapp-gcp-ace-logs/** to find and remove items older than 90 days. Repeat this process every morning.

- Write a lifecycle management rule in XML and push it to the bucket with gsutil lifecycle set config-xml-file.
- **Write a lifecycle management rule in JSON and push it to the bucket with gsutil lifecycle set config-json-file.**

Unattempted

You write a lifecycle management rule in XML and push it to the bucket with gsutil lifecycle set config-xml-file. is not right.
gsutil lifecycle set enables you to set the lifecycle configuration on one or more buckets based on the configuration file provided. However, XML is not a valid supported type for the configuration file.
Ref: https://cloud.google.com/storage/docs/gsutil/commands/lifecycle

Write a script that runs gsutil ls -lr gs://myapp-gcp-ace-logs/** to find and remove items older than 90 days. Repeat this process every morning. is not right.
This manual approach is error-prone, time-consuming and expensive. GCP Cloud Storage provides lifecycle management rules that let you achieve this with minimal effort.

Write a script that runs gsutil ls -l gs://myapp-gcp-ace-logs/** to find and remove items older than 90 days. Schedule the script with cron. is not right.
This manual approach is error-prone, time-consuming and expensive. GCP Cloud Storage provides lifecycle management rules that let you achieve this with minimal effort.

Write a lifecycle management rule in JSON and push it to the bucket with gsutil lifecycle set config-json-file. is the right answer.
You can assign a lifecycle management configuration to a bucket. The configuration contains a set of rules which apply to current and future objects in the bucket. When an object meets the criteria of one of the rules, Cloud Storage automatically performs a specified action on the object. One of the supported actions is to Delete objects. You can set up a lifecycle management to delete objects older than 90 days. "gsutil lifecycle set" enables you to set the lifecycle configuration on the bucket based on the configuration file. JSON is the only supported type for the configuration file. The config-json-file specified on the command line should be a path to a local file containing the lifecycle configuration JSON document.
Ref: https://cloud.google.com/storage/docs/gsutil/commands/lifecycle
Ref: https://cloud.google.com/storage/docs/lifecycle

.

13. Question

.

Users of your application are complaining of slowness when loading the application. You realize the slowness is because the App Engine deployment serving the application is deployed in us-central whereas all users of this application are closest to europe-west3. You want to change the region of the App Engine application to europe-west3 to minimize latency. What's the best way to change the App Engine region?

.

- **Create a new project and create an App Engine instance in europe-west3**
- Contact Google Cloud Support and request the change.
- Use the gcloud app region set command and supply the name of the new region.
- From the console, under the App Engine page, click edit, and change the region drop-down.

Unattempted

Use the gcloud app region set command and supply the name of the new region. is not right.
gcloud app region command does not provide a set action. The only action gcloud app region command currently supports is list which lists the availability of flex and standard environments for each region.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/regions/list

Contact Google Cloud Support and request the change. is not right.
Unfortunately, Google Cloud Support isn't of much use here as they would not be able to change the region of an App Engine Deployment. App engine is a regional service, which means the infrastructure that runs your app(s) is located in a specific region and is managed by Google to be redundantly available across all the zones within that region. Once an app engine deployment is created in a region, it can't be changed.
Ref: https://cloud.google.com/appengine/docs/locations

From the console, Click edit in App Engine dashboard page and change the region drop-down. is not right.
The settings mentioned in this option aren't available in the App Engine dashboard. App engine is a regional service. Once an app engine deployment is created in a region, it can't be changed. As shown in the screenshot below, Region is greyed out.

Create a new project and create an App Engine instance in europe-west3. is the right answer.
App engine is a regional service, which means the infrastructure that runs your app(s) is located in a specific region and is managed by Google to be redundantly available across all the zones within that region. Once an app engine deployment is created in a region, it

can't be changed. The only way is to create a new project and create an App Engine instance in europe-west3, send all user traffic to this instance and delete the app engine instance in us-central.
Ref: https://cloud.google.com/appengine/docs/locations

.

14. Question

.

You are analyzing Google Cloud Platform service costs from three separate projects. You want to use the information to create service costs estimates grouped by service type, daily and monthly, for the next six months using standard query syntax. What should you do?

.

- **Export your bill to a BigQuery dataset and then write time window based SQL queries for analysis.**
- Export your bill to a Cloud Storage bucket and then import into Cloud Bigtable for analysis.
- Export your bill to a Cloud Storage bucket and then import into Google Sheets for analysis
- Export your transactions to a local file and perform analysis with a suitable desktop tool.

Unattempted

Requirements
1. use query syntax
2. need the billing data of all three projects

Export your bill to a Cloud Storage bucket and then import into Cloud Bigtable for analysis. is not right.
BigTable is a NoSQL database and doesn't offer query syntax support.

Export your bill to a Cloud Storage bucket and then import into Google Sheets for analysis. is not right.
Google Sheets don't offer full support for query syntax. Moreover, export to Cloud Storage bucket captures a smaller dataset than export to BigQuery. For example, the exported billing data does not include resource labels or any invoice-level charges such as taxes accrued or adjustment memos.

Export your transactions to a local file and perform analysis with a suitable desktop tool. is not right.
Billing data can't be exported to a local file, it can only be exported to a BigQuery Dataset or Cloud Storage bucket.

Export your bill to a BigQuery dataset and then write time window based SQL queries for analysis. is the right answer.
You can export billing information from multiple projects into a BigQuery dataset. Unlike the export to Cloud Storage bucket, export to BigQuery dataset includes all information making it easy and straightforward to construct queries in BigQuery to estimate the cost. BigQuery supports Standard SQL so you can join tables and group by fields (labels in this case) as needed
Ref: https://cloud.google.com/billing/docs/how-to/export-data-bigquery.

.

15. Question

.

You are building a new version of an application hosted in an App Engine environment. You want to test the new version with 1% of users before you completely switch your application over to the new version. What should you do?

.

- Deploy a new version of your application in Google Kubernetes Engine instead of App Engine and then use GCP Console to split traffic.
- Deploy a new version of your application in a Compute Engine instance instead of App Engine and then use GCP Console to split traffic.
- Deploy a new version as a separate app in App Engine. Then configure App Engine using GCP Console to split traffic between the two apps.
- **Deploy a new version of your application in App Engine. Then go to App Engine settings in GCP Console and split traffic between the current version and newly deployed versions accordingly.**

Unattempted

Deploy a new version of your application in Google Kubernetes Engine instead of App Engine and then use GCP Console to split traffic. is not right.
When you can achieve this natively in GCP app engine using versions, there is no need to do it outside App Engine.
Ref: https://cloud.google.com/appengine/docs/standard/python/splitting-traffic

Deploy a new version of your application in a Compute Engine instance instead of App Engine and then use GCP Console to split traffic. is not right.
When you can achieve this natively in GCP app engine using versions, there is no need to do it outside App Engine.
Ref: https://cloud.google.com/appengine/docs/standard/python/splitting-traffic

Deploy a new version as a separate app in App Engine. Then configure App Engine using GCP Console to split traffic between the two apps. is not right.
You can achieve this natively in GCP app engine using versions but App Engine doesn't let you split traffic between apps. If you need to do it between apps, you are probably looking at doing this at the load balancer layer or at the DNS layer – either increasing the cost/complexity or introduce other problems such as caching issues.
Ref: https://cloud.google.com/appengine/docs/standard/python/splitting-traffic

Deploy a new version of your application in App Engine. Then go to App Engine settings in GCP Console and split traffic between the current version and newly deployed versions accordingly. is the right answer.
GCP App Engine natively offers traffic splitting functionality between versions. You can use traffic splitting to specify a percentage distribution of traffic across two or more of the versions within a service. Splitting traffic allows you to conduct A/B testing between your versions and provides control over the pace when rolling out features.
Ref: https://cloud.google.com/appengine/docs/standard/python/splitting-traffic

.

16. Question

.

You are building a pipeline to process time-series data. Which Google Cloud Platform services should you put in boxes 1,2,3, and 4?
Larger image

.

- Firebase Messages, Cloud Pub/Sub, Cloud Spanner, BigQuery
- Cloud Pub/Sub, Cloud Storage, BigQuery, Cloud Bigtable
- Cloud Pub/Sub, Cloud Dataflow, Cloud Datastore, BigQuery
- **Cloud Pub/Sub, Cloud Dataflow, Cloud Bigtable, BigQuery**

Unattempted

For box 1 where you want to ingest time series data, your best bet is Cloud Pub/Sub.
For box 2 where you want to process the data in pipelines, your best bet is Cloud Dataflow.

That leaves us with two remaining options, both have BigQuery as no 4. For (storage) 3, it is a choice between Bigtable and Datastore. Bigtable provides out of the box support for time series data. So using Bigtable for Storage is the right answer.
Ref: https://cloud.google.com/bigtable/docs/schema-design-time-series

The answer is Cloud Pub/Sub, Cloud Dataflow, Cloud Bigtable, BigQuery

.

17. Question

.

You are building a product on top of Google Kubernetes Engine (GKE). You have a single GKE cluster. For each of your customers, a Pod is running in that cluster, and your customers can run arbitrary code inside their Pod. You want to maximize the isolation between your customers' Pods. What should you do?

.

- Use Binary Authorization and whitelist only the container images used by your customers' Pods.
- Use the Container Analysis API to detect vulnerabilities in the containers used by your customers' Pods.
- **Create a GKE node pool with a sandbox type configured to gvisor. Add the parameter runtimeClassName: gvisor to the specification of your customers' Pods.**
- Use the cos_containerd image for your GKE nodes. Add a nodeSelector with the value cloud.google.com/gke-os-distribution: cos_containerd to the specification of your customers' Pods.

Unattempted

Use Binary Authorization and whitelist only the container images used by your customers' Pods. is not right.
Binary Authorization is a deploy-time security control that ensures only trusted container images are deployed on Google Kubernetes Engine (GKE). With Binary Authorization, you can require images to be signed by trusted authorities during the development process and then enforce signature validation when deploying. By enforcing validation, you can gain tighter control over your container environment by ensuring only verified images are integrated into the build-and-release process.
Ref: https://cloud.google.com/binary-authorization

Use the Container Analysis API to detect vulnerabilities in the containers used by your customers' Pods. is not right.
Container Analysis is a service that provides vulnerability scanning and metadata storage

for software artifacts. The scanning service performs vulnerability scans on images in Container Registry, then stores the resulting metadata and makes it available for consumption through an API. Metadata storage allows storing information from different sources, including vulnerability scanning, other Cloud services, and third-party providers.
Ref: https://cloud.google.com/container-registry/docs/container-analysis

Use the cos_containerd image for your GKE nodes. Add a nodeSelector with the value cloud.google.com/gke-os-distribution: cos_containerd to the specification of your customers' Pods. is not right.
The cos_containerd and ubuntu_containerdimages let you use containerd as the container runtime in your GKE cluster. This doesn't directly provide the isolation we require.
https://cloud.google.com/kubernetes-engine/docs/concepts/using-containerd

Create a GKE node pool with a sandbox type configured to gvisor. Add the parameter runtimeClassName: gvisor to the specification of your customers' Pods. is the right answer.
GKE Sandbox provides an extra layer of security to prevent untrusted code from affecting the host kernel on your cluster nodes when containers in the Pod execute unknown or untrusted code. Multi-tenant clusters and clusters whose containers run untrusted workloads are more exposed to security vulnerabilities than other clusters. Examples include SaaS providers, web-hosting providers, or other organizations that allow their users to upload and run code. When you enable GKE Sandbox on a node pool, a sandbox is created for each Pod running on a node in that node pool. In addition, nodes running sandboxed Pods are prevented from accessing other Google Cloud services or cluster metadata. Each sandbox uses its own userspace kernel. With this in mind, you can make decisions about how to group your containers into Pods, based on the level of isolation you require and the characteristics of your applications.
Ref: https://cloud.google.com/kubernetes-engine/docs/concepts/sandbox-pods

.

18. Question

.

You are building an application that stores relational data from users. Users across the globe will use this application. Your CTO is concerned about the scaling requirements because the size of the user base is unknown. You need to implement a database solution that can scale with your user growth with minimum configuration changes. Which storage solution should you use?

.

- Cloud Datastore
- Cloud SQL

- Cloud Spanner
- Cloud Firestore

Unattempted

Our requirements are relational data, global users, scaling

Cloud Firestore is not right.
Cloud Firestore is not a relational database. Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
Ref: https://firebase.google.com/docs/firestore

Cloud Datastore is not right.
Cloud Datastore is not a relational database. Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development
Ref: https://cloud.google.com/datastore/docs/concepts/overview

Cloud SQL is not right.
While Cloud SQL is a relational database, it does not offer infinite automated scaling with minimum configuration changes. Cloud SQL is a fully-managed database service that makes it easy to set up, maintain, manage, and administer your relational databases on Google Cloud Platform
Ref: https://cloud.google.com/sql/docs

Cloud Spanner is the right answer.
Cloud Spanner is a relational database and is highly scalable. Cloud Spanner is a highly scalable, enterprise-grade, globally-distributed, and strongly consistent database service built for the cloud specifically to combine the benefits of relational database structure with a non-relational horizontal scale. This combination delivers high-performance transactions and strong consistency across rows, regions, and continents with an industry-leading 99.999% availability SLA, no planned downtime, and enterprise-grade security
https://cloud.google.com/spanner

19. Question

You are building an application that will run in your data center. The application will use Google Cloud Platform (GCP) services like AutoML. You created a service account that has appropriate access to AutoML. You need to enable authentication to the APIs from your on-premises environment. What should you do?

.

- Use service account credentials in your on-premises application.
- **Use gcloud to create a key file for the service account that has appropriate permissions.**
- Set up direct interconnect between your data center and Google Cloud Platform to enable authentication for your on-premises applications.
- Go to the IAM & admin console, grant a user account permissions similar to the service account permissions, and use this user account for authentication from your data center.

Unattempted

Use service account credentials in your on-premises application. is not right.
Service accounts do not have passwords
Ref: https://cloud.google.com/iam/docs/service-accounts

Go to the IAM & admin console, grant a user account permissions similar to the service account permissions, and use this user account for authentication from your data center. is not right.
While granting Users a similar set of permissions lets them impersonate service accounts and access all resources the service account has access to, you should use a service account to represent a non-human user that needs to authenticate and be authorized to access data in Google APIs. Typically, service accounts are used in scenarios such as:
Running workloads on virtual machines (VMs).
Running workloads on on-premises workstations or data centers that call Google APIs.
Running workloads that are not tied to the lifecycle of a human user.
Your application assumes the identity of the service account to call Google APIs so that the users aren't directly involved.
Ref: https://cloud.google.com/iam/docs/understanding-service-accounts

Set up direct interconnect between your data center and Google Cloud Platform to enable authentication for your on-premises applications. is not right.
While setting up interconnect provides a direct physical connection between your on-premises network and Google's network, it doesn't directly help us authenticate our application running in the data center. You can configure Private Google Access for on-premises hosts by sending requests to restricted.googleapis.com and advertise a custom route on cloud router but this only lets you reach Google API and doesn't help with authentication.
Ref: https://cloud.google.com/interconnect/docs/support/faq

Use gcloud to create a key file for the service account that has appropriate permissions. is the right answer.
To use a service account outside of Google Cloud, such as on other platforms or on-

premises, you must first establish the identity of the service account. Public/private key pairs provide a secure way of accomplishing this goal. You can create a service account key using the Cloud Console, the gcloud tool, the serviceAccounts.keys.create() method, or one of the client libraries.
Ref: https://cloud.google.com/iam/docs/creating-managing-service-account-keys

.

20. Question

.

You are building an archival solution for your data warehouse and have selected Cloud Storage to archive your data. Your users need to be able to access this archived data once a quarter for some regulatory requirements. You want to select a cost-efficient option. Which storage option should you use?

.

- **Coldline Storage**
- Nearline Storage
- Regional Storage
- Multi-Regional Storage

Unattempted

Nearline Storage. is not right.
Nearline Storage is a low-cost, highly durable storage service for storing infrequently accessed data. Nearline Storage is ideal for data you plan to read or modify on average once per month or less. Nearline storage is more expensive than Coldline Storage which is more suitable for our requirements.
https://cloud.google.com/storage/docs/storage-classes#nearline

Regional Storage. is not right.
While this would certainly let you access your files once a quarter, it would be too expensive compared to Coldline storage which is more suitable for our requirement.
https://cloud.google.com/storage/docs/storage-classes#standard

Multi-Regional Storage. is not right.
While this would certainly let you access your files once a quarter, it would be too expensive compared to Coldline storage which is more suitable for our requirement.
https://cloud.google.com/storage/docs/storage-classes#standard

Coldline Storage. is the right answer.
Coldline Storage is a very-low-cost, highly durable storage service for storing infrequently

accessed data. Coldline Storage is ideal for data you plan to read or modify at most once a quarter. Since we have a requirement to access data once a quarter and want to go with the most cost-efficient option, we should select Coldline Storage.
Ref: https://cloud.google.com/storage/docs/storage-classes#coldline

.

21. Question

.

You are configuring service accounts for an application that spans multiple projects. Virtual machines (VMs) running in the web-applications project need access to BigQuery datasets in crm-databases project. You want to follow Google-recommended practices to give access to the service account in the web-applications project. What should you do?

.

- Grant project owner role on web-applications project to the service account in crm-databases project.
- Grant project owner role on crm-databases project to the service account in web-applications project.
- Grant project owner role on crm-databases project and bigquery.dataViewer role to the service account in web-applications.
- **Grant bigquery.dataViewer role on crm-databases project to the service account in web-applications.**

Unattempted

Grant project owner role on web-applications project to the service account in crm-databases project. is not right.
Our requirement is to identify the access needed for service account in the web-applications project, not the service account in crm-databases project

Grant project owner role on crm-databases project to the service account in web-applications project. is not right.
The primitive project owner role provides permissions to manage all resources within the project. For this scenario, the service account in the web-applications project needs access to BigQuery datasets in crm-databases project. Granting the project owner role would fall foul of least privilege principle.
Ref: https://cloud.google.com/iam/docs/recommender-overview

Grant project owner role on crm-databases project and bigquery.dataViewer role to the service account in web-applications. is not right.
The primitive project owner role provides permissions to manage all resources within the

project. For this scenario, the service account in the web-applications project needs access to BigQuery datasets in crm-databases project. Granting the project owner role would fall foul of least privilege principle.
Ref: https://cloud.google.com/iam/docs/recommender-overview

Grant bigquery.dataViewer role on crm-databases project to the service account in web-applications. is the right answer.
bigquery.dataViewer role provides permissions to read the dataset's metadata and list tables in the dataset as well as Read data and metadata from the dataset's tables. This is exactly what we need to fulfil this requirement and follows the least privilege principle.
Ref: https://cloud.google.com/iam/docs/understanding-roles#bigquery-roles

.

22. Question

.

You are creating a Google Kubernetes Engine (GKE) cluster with a cluster autoscaler feature enabled. You need to make sure that each node of the cluster will run a monitoring pod that sends container metrics to a third-party monitoring solution. What should you do?

.

- Deploy the monitoring pod in a StatefulSet object.
- Reference the monitoring pod in a Deployment object.
- Reference the monitoring pod in a cluster initializer at the GKE cluster creation time.
- **Deploy the monitoring pod in a DaemonSet object.**

Unattempted

Reference the monitoring pod in a Deployment object. is not right.
In our scenario, we need just 1 instance of the monitoring pod running on each node. Bundling the monitoring pod with a deployment object may result in multiple pod instances on the same node. In GKE, deployments represent a set of multiple, identical Pods with no unique identities. Deployment runs multiple replicas of your application and automatically replaces any instances that fail or become unresponsive. In this way, Deployments help ensure that one or more instances of your application are available to serve user requests.
https://cloud.google.com/kubernetes-engine/docs/concepts/deployment

Reference the monitoring pod in a cluster initializer at the GKE cluster creation time. is not right.

You can not use gcloud init to initialize a monitoring pod. gcloud initializer performs the following setup steps.
? Authorizes gcloud and other SDK tools to access Google Cloud Platform using your user account credentials, or from an account of your choosing whose credentials are already available.
? Sets up a new or existing configuration.
? Sets properties in that configuration, including the current project and optionally, the default Google Compute Engine region and zone you'd like to use.
Ref: https://cloud.google.com/sdk/gcloud/reference/init

Deploy the monitoring pod in a StatefulSet object. is not right.
In GKE, StatefulSets represents a set of Pods with unique, persistent identities and stable hostnames that GKE maintains regardless of where they are scheduled. The state information and other resilient data for any given StatefulSet Pod is maintained in persistent disk storage associated with the StatefulSet. The main purpose of StatefulSets is to set up persistent storage for pods that are deployed across multiple zones.
Ref: https://cloud.google.com/kubernetes-engine/docs/concepts/statefulset
Although persistent volumes can be used, they are limited to two zones and you'd have to get into node affinity if you want to use a persistent volume with a pod on a zone that is not covered by the persistent volumes zones.
See this for more information https://kubernetes.io/docs/setup/best-practices/multiple-zones/

Deploy the monitoring pod in a DaemonSet object. is the right answer.
In GKE, DaemonSets manage groups of replicated Pods and adhere to a one-Pod-per-node model, either across the entire cluster or a subset of nodes. As you add nodes to a node pool, DaemonSets automatically add Pods to the new nodes as needed. So, this is a perfect fit for our monitoring pod.
https://cloud.google.com/kubernetes-engine/docs/concepts/daemonset
DaemonSets are useful for deploying ongoing background tasks that you need to run on all or certain nodes, and which do not require user intervention. Examples of such tasks include storage daemons like ceph, log collection daemons like fluentd, and node monitoring daemons like collectd. For example, you could have DaemonSets for each type of daemon run on all of your nodes. Alternatively, you could run multiple DaemonSets for a single type of daemon, but have them use different configurations for different hardware types and resource needs.

- **23. Question**

-

You are deploying an application to a Compute Engine VM in a managed instance group. The application must be running at all times, but only a single instance of the VM should run per GCP project. How should you configure the instance group?

.

- Set autoscaling to On, set the minimum number of instances to 1, and then set the maximum number of instances to 2.
- **Set autoscaling to On, set the minimum number of instances to 1, and then set the maximum number of instances to 1.**
- Set autoscaling to Off, set the minimum number of instances to 1, and then set the maximum number of instances to 1.
- Set autoscaling to Off, set the minimum number of instances to 1, and then set the maximum number of instances to 2.

Unattempted

Requirements
1. Since we need the application running at all times, we need a minimum 1 instance.
2. Only a single instance of the VM should run, we need a maximum 1 instance.
3. We want the application running at all times. If the VM crashes due to any underlying hardware failure, we want another instance to be added to MIG so that application can continue to serve requests. We can achieve this by enabling autoscaling.

The only option that satisfies these three is Set autoscaling to On, set the minimum number of instances to 1, and then set the maximum number of instances to 1.
Ref: https://cloud.google.com/compute/docs/autoscaler

.

24. Question

.

You are deploying an application to the App Engine. You want the number of instances to scale based on request rate. You need at least 3 unoccupied instances at all times. Which scaling type should you use?

.

- Basic Scaling with min_instances set to 3.
- Manual Scaling with 3 instances.
- **Automatic Scaling with min_idle_instances set to 3.**
- Basic Scaling with max_instances set to 3.

Unattempted

Manual Scaling with 3 instances. is not right.
Manual scaling uses resident instances that continuously run the specified number of instances regardless of the load level. This allows tasks such as complex initializations and applications that rely on the state of the memory over time. This does not autoscale based on the request rate so doesn't fit our requirements.
Ref: https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed

Basic Scaling with min_instances set to 3. is not right.
Basic scaling creates dynamic instances when your application receives requests. Each instance will be shut down when the app becomes idle. Basic scaling is ideal for work that is intermittent or driven by user activity. In absence of any load, the App engine may shut down all instances so it is not suitable for our requirement of "at least 3 instances at all times".
Ref: https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed

Basic Scaling with max_instances set to 3. is not right.
Basic scaling creates dynamic instances when your application receives requests. Each instance will be shut down when the app becomes idle. Basic scaling is ideal for work that is intermittent or driven by user activity. In absence of any load, the App engine may shut down all instances so it is not suitable for our requirement of "at least 3 instances at all times".
Ref: https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed

Automatic Scaling with min_idle_instances set to 3. is the right answer.
Automatic scaling creates dynamic instances based on request rate, response latencies, and other application metrics. However, if you specify the number of minimum idle instances, that specified number of instances run as resident instances while any additional instances are dynamic.
Ref: https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed

- **25. Question**

- You are designing an application that lets users upload and share photos. You expect your application to grow really fast and you are targeting a worldwide audience. You want to

delete uploaded photos after 30 days. You want to minimize costs while ensuring your application is highly available. Which GCP storage solution should you choose?

.

- Persistent SSD on VM instances.
- Cloud Filestore.
- **Multiregional Cloud Storage bucket.**
- Cloud Datastore database.

Unattempted

Cloud Datastore database. is not right.
Cloud Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. We want to store objects/files and Cloud Datastore is not a suitable storage option for such data.
Ref: https://cloud.google.com/datastore/docs/concepts/overview

Cloud Filestore. is not right.
Cloud Filestore is a managed file storage service based on NFSv3 protocol. While Cloud Filestore can be used to store images, Cloud Filestore is a zonal service and can not scale easily to support a worldwide audience. Also, Cloud Filestore costs a lot (10 times) more than some of the storage classes offered by Google Cloud Storage.
Ref: https://cloud.google.com/filestore, Ref: https://cloud.google.com/storage/pricing

Persistent SSD on VM instances. is not right.
Persistent SSD is a regional service and doesn't automatically scale to other regions to support a worldwide user base. Moreover, Persistent SSD disks are very expensive. A regional persistent SSD costs $0.34 per GB per month. In comparison, Google Cloud Storage offers several storage classes that are significantly cheaper.
Ref: https://cloud.google.com/persistent-disk
Ref: https://cloud.google.com/filestore/pricing

Multiregional Cloud Storage bucket. is the right answer.
Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. We don't need to set up auto-scaling ourselves. Cloud Storage autoscaling is managed by GCP. Cloud Storage is an object store so it is suitable for storing photos. Cloud Storage allows world-wide storage and retrieval so cater well to our worldwide audience. Cloud storage provides us lifecycle rules that can be configured to automatically delete objects older than 30 days. This also fits our requirements. Finally, Google Cloud Storage offers several storage classes such as Nearline Storage ($0.01 per GB per Month) Coldline Storage ($0.007 per GB per Month) and Archive Storage ($0.004 per GB per month) which are significantly cheaper than any of the options above.

Ref: https://cloud.google.com/storage/docs
Ref: https://cloud.google.com/storage/pricing

.

26. Question

.

You are designing an application that uses WebSockets and HTTP sessions that are not distributed across the web servers. You want to ensure the application runs properly on Google Cloud Platform. What should you do?

.

- **Meet with the cloud enablement team to discuss load balancer options.**
- Redesign the application to use a distributed user session service that does not rely on WebSockets and HTTP sessions.
- Review the encryption requirements for WebSocket connections with the security team.
- Convert the WebSocket code to use HTTP streaming.

Unattempted

Google HTTP(S) Load Balancing has native support for the WebSocket protocol when you use HTTP or HTTPS, not HTTP/2, as the protocol to the backend.
Ref: https://cloud.google.com/load-balancing/docs/https#websocket_proxy_support
So the next possible step is to Meet with the cloud enablement team to discuss load balancer options.
We don't need to convert WebSocket code to use HTTP streaming or Redesign the application, as WebSocket support is offered by Google HTTP(S) Load Balancing. Reviewing the encryption requirements is a good idea but it has nothing to do with WebSockets.

.

27. Question

.

You are given a project with a single virtual private cloud (VPC) and a single subnet in the us-central1 region. There is a Compute Engine instance hosting an application in this subnet. You need to deploy a new instance in the same project in the europe-west1 region. This new instance needs access to the application. You want to follow Google-recommended practices. What should you do?

.

- 1. Create a VPC and a subnet in europe-west1. 2. Expose the application with an internal load balancer. 3. Create the new instance in the new subnet and use the load balancer's address as the endpoint.
- 1. Create a VPC and a subnet in europe-west1. 2. Peer the 2 VPCs. 3. Create the new instance in the new subnet and use the first instance's private address as the endpoint.
- **1. Create a subnet in the same VPC, in europe-west1. 2. Create the new instance in the new subnet and use the first instance subnet's private address as the endpoint.**
- 1. Create a subnet in the same VPC, in europe-west1. 2. Use Cloud VPN to connect the two subnets. 3. Create the new instance in the new subnet and use the first instance's private address as the endpoint.

Unattempted

Our requirements are to connect the instance in europe-west1 region with the application running in us-central1 region following Google-recommended practices. The two instances are in the same project.

1. Create a VPC and a subnet in europe-west1.
2. Expose the application with an internal load balancer.
3. Create the new instance in the new subnet and use the load balancer's address as the endpoint. is not right.
We have two different VPCs. There is no mention of the CIDR range so let's assume the two subnets in two VPCs use different CIDR ranges. However, there is no communication route between the two VPCs. If we create an internal load balancer, that load balancer is not visible outside the VPC. So the new instance cannot connect to the load balancer's internal address.
Ref: https://cloud.google.com/load-balancing/docs/internal

1. Create a subnet in the same VPC, in europe-west1.
2. Use Cloud VPN to connect the two subnets.
3. Create the new instance in the new subnet and use the first instance's private address as the endpoint. is not right.
Cloud VPN securely connects your on-premises network to your Google Cloud (GCP) Virtual Private Cloud (VPC) network through an IPsec VPN connection. It is not meant to connect two subnets within the same VPC. Moreover, subnets within the same VPC can communicate with each other by setting up relevant firewall rules.

1. Create a VPC and a subnet in europe-west1.
2. Peer the 2 VPCs.
3. Create the new instance in the new subnet and use the first instance's private address as the endpoint. is not right.

Given that the new instance wants to access the application on the existing compute engine instance, these applications seem to be related so they should be within the same VPC. It is possible to have them in different VPCs and peer the VPCs but this is a lot of additional work and we can simplify this by choosing the option below (which is the answer)

1. Create a subnet in the same VPC, in europe-west1.
2. Create the new instance in the new subnet and use the first instance subnet's private address as the endpoint. is the right answer.
We can create another subnet in the same VPC and this subnet is located in europe-west1. We can then spin up a new instance in this subnet. We also have to set up a firewall rule to allow communication between the two subnets. All instances in the two subnets with the same VPC can communicate through the internal IP Address
Ref: https://cloud.google.com/vpc

.

28. Question

.

You are hosting an application on bare metal servers in your data center. The application needs access to Cloud Storage. However, security policies prevent the servers hosting the application from having public IP addresses or access to the internet. You want to follow Google recommended practices to provide the application with access to Cloud Storage. What should you do?

.

- Use nslookup to get the IP addresses for storage.googleapis.com Negotiate with the security team to be able to give public IP addresses to the servers. Only allow egress traffic from those servers to the IP addresses for storage.googleapis.com
- Using Cloud VPN, create a VPN tunnel to a Virtual Private Cloud (VPC) in Google Cloud Platform (GCP) In this VPC, create a Compute Engine instance and install the Squid proxy server on this instance Configure your servers to use that instance as a proxy to access cloud storage
- Use Migrate for Compute Engine (formerly known as Velostrata) to migrate these servers to Compute Engine Create an internal load balancer (ILB) that uses storage.googleapis.com as backend Configure your new instances to use the ILB as a proxy
- **Using Cloud VPN or Interconnect, create a tunnel to a VPC in GCP Using Cloud Router to create a custom route advertisement for 199.36.153.4/30. Announce that network to your on-premises network through the VPN tunnel. In your on-premises**

network, configure your DNS server to resolve *.googleapis.com as a CNAME to restricted.googleapis.com

Unattempted

Our requirement is to follow Google recommended practices to achieve the end result.

Configuring Private Google Access for On-Premises Hosts is best achieved by VPN/Interconnect + Advertise Routes + Use restricted Google IP Range.
Using Cloud VPN or Interconnect, create a tunnel to a VPC in GCP
Using Cloud Router to create a custom route advertisement for 199.36.153.4/30.
Announce that network to your on-premises network through the VPN tunnel.
In your on-premises network, configure your DNS server to resolve *.googleapis.com as a CNAME to restricted.googleapis.com is the right answer right, and it is what Google recommends.
Ref: https://cloud.google.com/vpc/docs/configure-private-google-access-hybrid
"You must configure routes so that Google API traffic is forwarded through your Cloud VPN or Cloud Interconnect connection, firewall rules on your on-premises firewall to allow the outgoing traffic, and DNS so that traffic to Google APIs resolves to the IP range you've added to your routes."
"You can use Cloud Router Custom Route Advertisement to announce the Restricted Google APIs IP addresses through Cloud Router to your on-premises network. The Restricted Google APIs IP range is 199.36.153.4/30. While this is technically a public IP range, Google does not announce it publicly. This IP range is only accessible to hosts that can reach your Google Cloud projects through internal IP ranges, such as through a Cloud VPN or Cloud Interconnect connection."

Without having a public IP address or access to the internet, the only way you could connect to cloud storage is if you have an internal route to it. So Negotiate with the security team to be able to give public IP addresses to the servers is not right.

Following "Google recommended practices" is synonymous with "using Google's services" (Not quite, but it is – at least for the exam !!). So In this VPC, create a Compute Engine instance and install the Squid proxy server on this instance is not right.

Migrating the VM to Compute Engine is a bit drastic when Google says it is perfectly fine to have Hybrid Connectivity architectures https://cloud.google.com/hybrid-connectivity. So, Use Migrate for Compute Engine (formerly known as Velostrata) to migrate these servers to Compute Engine is not right.

.

29. Question

.

You are managing several Google Cloud Platform (GCP) projects and need access to all logs for the past 60 days. You want to be able to explore and quickly analyze the log contents. You want to follow Google-recommended practices to obtain the combined logs for all projects. What should you do?

.

- Navigate to Stackdriver Logging and select resource.labels.project_id="*"
- **Create a Stackdriver Logging Export with a Sink destination to a BigQuery dataset. Configure the table expiration to 60 days.**
- Create a Stackdriver Logging Export with a Sink destination to Cloud Storage. Create a lifecycle rule to delete objects after 60 days.
- Configure a Cloud Scheduler job to read from Stackdriver and store the logs in BigQuery. Configure the table expiration to 60 days.

Unattempted

Navigate to Stackdriver Logging and select resource.labels.project_id=''*''. is not right.
Log entries are held in Stackdriver Logging for a limited time known as the retention period – which is 30 days (default configuration). After that, the entries are deleted. To keep log entries longer, you need to export them outside of Stackdriver Logging by configuring log sinks.
https://cloud.google.com/blog/products/gcp/best-practices-for-working-with-google-cloud-audit-logging

Configure a Cloud Scheduler job to read from Stackdriver and store the logs in BigQuery. Configure the table expiration to 60 days. is not right.
While this works, it makes no sense to use Cloud Scheduler job to read from Stackdriver and store the logs in BigQuery when Google provides a feature (export sinks) that does exactly the same thing and works out of the box.
Ref: https://cloud.google.com/logging/docs/export/configure_export_v2

Create a Stackdriver Logging Export with a Sink destination to Cloud Storage. Create a lifecycle rule to delete objects after 60 days. is not right.
You can export logs by creating one or more sinks that include a logs query and an export destination. Supported destinations for exported log entries are Cloud Storage, BigQuery, and Pub/Sub.
Ref: https://cloud.google.com/logging/docs/export/configure_export_v2
Sinks are limited to exporting log entries from the exact resource in which the sink was created: a Google Cloud project, organization, folder, or billing account. If it makes it easier to exporting from all projects of an organication, you can create an aggregated sink that can export log entries from all the projects, folders, and billing accounts of a Google Cloud organization.

https://cloud.google.com/logging/docs/export/aggregated_sinks
Either way, we now have the data in Cloud Storage, but querying logs information from Cloud Storage is harder than Querying information from BigQuery dataset. For this reason, we should prefer Big Query over Cloud Storage.

Create a Stackdriver Logging Export with a Sink destination to a BigQuery dataset. Configure the table expiration to 60 days. is the right answer.
You can export logs by creating one or more sinks that include a logs query and an export destination. Supported destinations for exported log entries are Cloud Storage, BigQuery, and Pub/Sub.
Ref: https://cloud.google.com/logging/docs/export/configure_export_v2
Sinks are limited to exporting log entries from the exact resource in which the sink was created: a Google Cloud project, organization, folder, or billing account. If it makes it easier to exporting from all projects of an organication, you can create an aggregated sink that can export log entries from all the projects, folders, and billing accounts of a Google Cloud organization.
https://cloud.google.com/logging/docs/export/aggregated_sinks
Either way, we now have the data in a BigQuery Dataset. Querying information from a Big Query dataset is easier and quicker than analyzing contents in Cloud Storage bucket. As our requirement is to "Quickly analyze the log contents", we should prefer Big Query over Cloud Storage.

Also, You can control storage costs and optimize storage usage by setting the default table expiration for newly created tables in a dataset. If you set the property when the dataset is created, any table created in the dataset is deleted after the expiration period. If you set the property after the dataset is created, only new tables are deleted after the expiration period.
For example, if you set the default table expiration to 7 days, older data is automatically deleted after 1 week.
Ref: https://cloud.google.com/bigquery/docs/best-practices-storage

.

30. Question

.

You are migrating a mission critical on-premises application to cloud. The application requires 96 vCPUs to perform its task. You want to make sure the application runs in a similar environment on GCP. What should you do?

.

- **When creating the VM, use machine type n1-standard-96.**

- When creating the VM, use Intel Skylake as the CPU platform.
- Create the VM using Compute Engine default settings. Use gcloud to modify the running instance to have 96 vCPUs.
- Start the VM using Compute Engine default settings, and adjust as you go based on Rightsizing Recommendations.

Unattempted

Create the VM using Compute Engine default settings. Use gcloud to modify the running instance to have 96 vCPUs. is not right.
You can't increase the vCPUs to 96 without changing the machine type. While it is possible to set machine type using gcloud, this would mean downtime for the mission-critical application while the upgrade happens which is undesirable.
Ref: https://cloud.google.com/compute/docs/instances/changing-machine-type-of-stopped-instance

Start the VM using Compute Engine default settings, and adjust as you go based on Rightsizing Recommendations. is not right.
Since the application is mission-critical, we want to ensure that this application has all the required resources from the beginning. Starting with the default settings provisions a n1-standard-1 machine that has just 1 vCPU and our mission-critical application would be severely constrained for resources.

When creating the VM, use Intel Skylake as the CPU platform. is not right.
Intel Skylake is only offered in E2 machine types that are cost-optimized machine types and offer sizing between 2 to 16 vCPUs which is insufficient for our mission-critical application.
Ref: https://cloud.google.com/compute/docs/machine-types#e2_machine_types

When creating the VM, use machine type n1-standard-96. is the right answer.
n1-standard-96 offers 96 vCPUs and 624 GB of memory. This fits our requirements.
https://cloud.google.com/compute/docs/machine-types#n1_machine_type

.

31. Question

.

You are operating a Google Kubernetes Engine (GKE) cluster for your company where different teams can run non-production workloads. Your Machine Learning (ML) team needs access to Nvidia Tesla P100 GPUs to train their models. You want to minimize effort and cost. What should you do?

.

- Ask your ML team to add the "accelerator: gpu" annotation to their pod specification.
- Recreate all the nodes of the GKE cluster to enable GPUs on all of them.
- Create your own Kubernetes cluster on top of Compute Engine with nodes that have GPUs. Dedicate this cluster to your ML team.
- **Add a new, GPU-enabled, node pool to the GKE cluster. Ask your ML team to add the cloud.google.com/gke-accelerator: nvidia-tesla-p100 nodeSelector to their pod specification.**

Unattempted

Ask your ML team to add the ''accelerator: gpu'' annotation to their pod specification. is not right.
There are two issues with this approach. One – the syntax is invalid. Two – You cannot add GPUs to existing node pools.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/gpus

Recreate all the nodes of the GKE cluster to enable GPUs on all of them. is not right.
There are two issues with this approach. One – recreating all nodes to enable GPUs makes the cluster very expensive. Only the ML team needs access to GPUs to train their models. Recreating all nodes to enable GPUs helps your ML team use them but they are left unused for all other workloads yet cost you money. Two – Even though your nodes have GPUs enabled, you still have to modify pod specifications to request GPU. This step isn't performed in this option.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/gpus

Create your own Kubernetes cluster on top of Compute Engine with nodes that have GPUs. Dedicate this cluster to your ML team. is not right.
While this works, it increases the cost as you now pay the Kubernetes cluster management fee for two clusters instead of one. GKE clusters accrue a management fee that is per cluster per hour, irrespective of cluster size or topology.
Ref: https://cloud.google.com/kubernetes-engine/pricing

Add a new, GPU-enabled, node pool to the GKE cluster. Ask your ML team to add the cloud.google.com/gke-accelerator: nvidia-tesla-p100 nodeSelector to their pod specification. is the right answer.
This is the most optimal solution. Rather than recreating all nodes, you create a new node pool with GPU enabled. You then modify the pod specification to target particular GPU types by adding node selector to your workload's Pod specification. YOu still have a single cluster so you pay Kubernetes cluster management fee for just one cluster thus minimizing the cost.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/gpus

Ref: https://cloud.google.com/kubernetes-engine/pricing
Example:
apiVersion: v1
kind: Pod
metadata:
name: my-gpu-pod
spec:
containers:
– name: my-gpu-container
image: nvidia/cuda:10.0-runtime-ubuntu18.04
command: [“/bin/bash”]
resources:
limits:
nvidia.com/gpu: 2
nodeSelector:
cloud.google.com/gke-accelerator: nvidia-tesla-k80 # or nvidia-tesla-p100 or nvidia-tesla-p4 or nvidia-tesla-v100 or nvidia-tesla-t4

.

32. Question

.

You are running an application on multiple virtual machines within a managed instance group and have auto-scaling enabled. The autoscaling policy is configured so that additional instances are added to the group if the CPU utilization of instances goes above 80%. VMs are added until the instance group reaches its maximum limit of five VMs or until CPU utilization of instances lowers to 80%. The initial delay for HTTP health checks against the instances is set to 30 seconds. The virtual machine instances take around three minutes to become available for users. You observe that when the instance group auto-scales, it adds more instances then necessary to support the levels of end-user traffic. You want to properly maintain instance group sizes when autoscaling. What should you do?

.

- Decrease the maximum number of instances to 3.
- **Increase the initial delay of the HTTP health check to 200 seconds.**
- Set the maximum number of instances to 1.
- Use a TCP health check instead of an HTTP health check.

Unattempted

Scenario
? Autoscaling is enabled and kicks off the scale-up
? Scaling policy is based on target CPU utilization of 80%
? The initial delay is 30 seconds

? VM startup time is 3 minutes.
? Auto-scaling creates more instances than necessary.

Set the maximum number of instances to 1. is not right.
Setting the maximum number of instances to 1 effectively limits the scale up to 1 instance which is undesirable as in this case we may still be struggling with the CPU usage but we can't scale up. Therefore this is not the right answer.

Decrease the maximum number of instances to 3. is not right.
Setting the maximum number of instances to 3 effectively limits the scale up to 3 instances which is undesirable as in this case we may still be struggling with the CPU usage but we can't scale up. Therefore this is not the right answer.

Use a TCP health check instead of an HTTP health check. is not right.
TCP health check is a legacy health check, whereas HTTP health check is more advanced and "non-legacy". It is possible a TCP health check might say the application is UP when it is not as it only listens on application servers TCP port and doesn't validate the application health through a HTTP check on its health endpoint. This results in the load balancer sending requests to the application server when it is still loading the application resulting in failures.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/health-checks/create/tcp
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/health-checks/create/http

Increase the initial delay of the HTTP health check to 200 seconds. is the right answer.
The reason why our autoscaling is adding more instances than needed is that it checks 30 seconds after launching the instance and at this point, the instance isn't up and isn't ready to serve traffic. So our autoscaling policy starts another instance – again checks this after 30 seconds and the cycle repeats until it gets to the maximum instances or the instances launched earlier are healthy and start processing traffic – which happens after 180 seconds (3 minutes). This can be easily rectified by adjusting the initial delay to be higher than the time it takes for the instance to become available for processing traffic. So setting this to 200 ensures that it waits until the instance is up (around 180-second mark) and then starts forwarding traffic to this instance. Even after a cool out period, if the CPU utilization is still high, the autoscaler can again scale up but this scale-up is genuine and is based on the actual load.
"Initial Delay Seconds" – This setting delays autohealing from potentially prematurely recreating the instance if the instance is in the process of starting up. The initial delay timer starts when the currentAction of the instance is VERIFYING.
Ref: https://cloud.google.com/compute/docs/instance-groups/autohealing-instances-in-migs

33. Question

You are setting up a Windows VM on Compute Engine and want to make sure you can log in to the VM via RDP. What should you do?

- After the VM has been created, use your Google Account credentials to log in into the VM.
- **After the VM has been created, use gcloud compute reset-windows-password to retrieve the login credentials for the VM.**
- When creating the VM, add metadata to the instance using 'windows-password' as the key and a password as the value.
- After the VM has been created, download the JSON private key for the default Compute Engine service account. Use the credentials in the JSON file to log in to the VM.

Unattempted

When creating the VM, add metadata to the instance using 'windows-password' as the key and a password as the value. is not right.
It is not possible to specify a windows password at the time of creating windows VM instance. You can generate Windows passwords using either the Google Cloud Console or the gcloud command-line tool. Alternatively, you can generate passwords programmatically with API commands but all these methods assume that you have an existing windows instance.
Ref: https://cloud.google.com/compute/docs/instances/windows/creating-passwords-for-windows-instances#gcloud

After the VM has been created, use your Google Account credentials to log in into the VM. is not right.
You can generate Windows passwords using either the Google Cloud Console or the gcloud command-line tool. Alternatively, you can generate passwords programmatically with API commands but you can't use your gcloud account credentials to log into the VM.
Ref: https://cloud.google.com/compute/docs/instances/windows/creating-passwords-for-windows-instances#gcloud

After the VM has been created, download the JSON private key for the default Compute Engine service account. Use the credentials in the JSON file to log in to the VM. is not right.
This is not a supported method of authentication for logging into the VM. You can generate Windows passwords using either the Google Cloud Console or the gcloud

command-line tool. Alternatively, you can generate passwords programmatically with API commands.
Ref: https://cloud.google.com/compute/docs/instances/windows/creating-passwords-for-windows-instances#gcloud

After the VM has been created, use gcloud compute reset-windows-password to retrieve the login credentials for the VM. is the right answer.
You can generate Windows passwords using either the Google Cloud Console or the gcloud command-line tool. This option uses the right syntax to reset the windows password.
gcloud compute reset-windows-password windows-instance
Ref: https://cloud.google.com/compute/docs/instances/windows/creating-passwords-for-windows-instances#gcloud

.

34. Question

.

You are the organization and billing administrator for your company. The engineering team has the Project Creator role at the organization level. You do not want the engineering team to be able to link projects to the billing account. Only the finance team should be able to link a project to a billing account, but they should not be able to make any other changes to projects. What should you do?

.

- Assign the engineering team the Billing Account User role on the billing account and the Project Billing Manager role on the organization.
- Assign the finance team only the Billing Account User role on the billing account.
- Assign the engineering team only the Billing Account User role on the billing account.
- **Assign the finance team the Billing Account User role on the billing account and the Project Billing Manager role on the organization.**

Unattempted

Assign the finance team only the Billing Account User role on the billing account. is not right.
In order to link a project to a billing account, you need the necessary roles at the project level as well as at the billing account level. In this scenario, we are granting just the Billing Account User role on the billing account to the Finance team which allows them to link projects to the billing account on which the role is granted. But we haven't granted them any role at the project level. So they would not be unable to link projects.

Assign the engineering team only the Billing Account User role on the billing account. is not right.
In order to link a project to a billing account, you need the necessary roles at the project level as well as at the billing account level. In this scenario, we are granting just the Billing Account User role on the billing account to the Engineering team which allows them to link projects to the billing account and our question clearly states we do not want to do that.

Assign the engineering team the Billing Account User role on the billing account and the Project Billing Manager role on the organization. is not right.
In order to link a project to a billing account, you need the necessary roles at the project level as well as at the billing account level. In this scenario, we are assigning the engineering team the Billing Account User role on the billing account which allows them to create new projects linked to the billing account on which the role is granted. We are also assigning them the Project Billing Manager role on the organization (trickles down to the project as well) which lets them attach the project to the billing account. But we don't want the engineering team to link projects to the billing account.

Assign the finance team the Billing Account User role on the billing account and the Project Billing Manager role on the organization. is the right answer.
In order to link a project to a billing account, you need the necessary roles at the project level as well as at the billing account level. In this scenario, we are assigning the finance team the Billing Account User role on the billing account which allows them to create new projects linked to the billing account on which the role is granted. We are also assigning them the Project Billing Manager role on the organization (trickles down to the project as well) which lets them attach the project to the billing account, but does not grant any rights over resources. This is exactly what we want.

.

35. Question

.

You are the project owner of a GCP project and want to delegate control to colleagues to manage buckets and files in Cloud Storage. You want to follow Google recommended practices. Which IAM roles should you grant your colleagues?

.

- Project Editor
- Storage Object Creator
- **Storage Admin**
- Storage Object Admin

Unattempted

Project Editor is not right. is not right.
Project editor is a primitive role that grants a lot more than what we need here. Google doesn't recommend using Primitive roles.
Ref: https://cloud.google.com/iam/docs/understanding-roles#primitive_role_definitions
All viewer permissions, plus permissions for actions that modify state, such as changing existing resources.

Storage Object Admin. is not right.
While this role grants full access to the objects, it does not grant access to the buckets so users of this role can not "manage buckets".
This role grants full control over objects, including listing, creating, viewing, and deleting objects.
Ref: https://cloud.google.com/iam/docs/understanding-roles#storage-roles

Storage Object Creator. is not right.
This role allows users to create objects. It does not give permission to view, delete, or overwrite objects.
Ref: https://cloud.google.com/iam/docs/understanding-roles#storage-roles

Storage Admin. is the right answer.
This role grants full control of buckets and objects. When applied to an individual bucket, control applies only to the specified bucket and objects within the bucket.
Ref: https://cloud.google.com/iam/docs/understanding-roles#storage-roles

.

36. Question

.

You are using Container Registry to centrally store your company's container images in a separate project. In another project, you want to create a Google Kubernetes Engine (GKE) cluster. You want to ensure that Kubernetes can download images from Container Registry. What should you do?

.

- **In the project where the images are stored, grant the Storage Object Viewer IAM role to the service account used by the Kubernetes nodes.**
- When you create the GKE cluster, choose the Allow full access to all Cloud APIs option under 'Access scopes'.

- Create a service account, and give it access to Cloud Storage. Create a P12 key for this service account and use it as an imagePullSecrets in Kubernetes.
- Configure the ACLs on each image in Cloud Storage to give read-only access to the default Compute Engine service account.

Unattempted

Here's some info about where Container Registry stores images and how access is controlled.
Container Registry uses Cloud Storage buckets as the underlying storage for container images. You control access to your images by granting appropriate Cloud Storage permissions to a user, group, service account, or another identity. Cloud Storage permissions granted at the project level apply to all storage buckets in the project, not just the buckets used by Container Registry. To configure permissions specific to Container Registry, grant permissions on the storage bucket used by the registry. Container Registry ignores permissions set on individual objects within the storage bucket.
Ref: https://cloud.google.com/container-registry/docs/access-control

Configure the ACLs on each image in Cloud Storage to give read-only access to the default Compute Engine service account. is not right.
As mentioned above, Container Registry ignores permissions set on individual objects within the storage bucket so this isn't going to work.
Ref: https://cloud.google.com/container-registry/docs/access-control

When you create the GKE cluster, choose the Allow full access to all Cloud APIs option under 'Access scopes'. is not right.
Selecting Allow full access to all Cloud APIs does not provide access to GCR images in a different project. If the Google Kubernetes Engine cluster and the Container Registry storage bucket are in the same Google Cloud project, the Compute Engine default service account is configured with the appropriate permissions to push or pull images. But if the cluster is in a different project or if the VMs in the cluster use a different service account, you must grant the service account the appropriate permissions to access the storage bucket used by Container Registry.
Ref: https://cloud.google.com/container-registry/docs/using-with-google-cloud-platform
In this case, since there is no mention of a service account, we have to assume we are using a default service account that hasn't been provided permissions to access the storage bucket used by Container Registry in another project so the image pull isn't going to work. You would end up with an error like:
Failed to pull image "gcr.io/kubernetes2-278322/simple-python-image": rpc error: code = Unknown desc = Error response from daemon: pull access denied for gcr.io/kubernetes2-278322/simple-python-image, repository does not exist or may require 'docker login'

Create a service account, and give it access to Cloud Storage. Create a P12 key for this service account and use it as an imagePullSecrets in Kubernetes. is not right.
It is technically possible to do it this way but using the JSON key and not P12 key as mentioned in this option. If you would like to understand how to do this, please look at these blogs.
Ref: https://medium.com/hackernoon/today-i-learned-pull-docker-image-from-gcr-google-container-registry-in-any-non-gcp-kubernetes-5f8298f28969
Ref: https://medium.com/@michaelmorrissey/using-cross-project-gcr-images-in-gke-1ddc36de3d42
Moreover, this approach is suitable for accessing GCR images in a non-Google Cloud Kubernetes environment. While it can be used in GKE too, it is not as secure as using Role Bindings since it involves downloading service account keys and setting them up as secret in Kubernetes.

In the project where the images are stored, grant the Storage Object Viewer IAM role to the service account used by the Kubernetes nodes. is the right answer.
Granting the storage object viewer IAM role in the project where images are stored to the service account used by the Kubernetes cluster ensures that the nodes in the cluster can Read Images from the storage bucket. It would be ideal to further restrict the role binding to provide access just to the Cloud Storage bucket that is used as the underlying storage for container images. This follows the principle of least privilege.
For more information about Storage Object Viewer IAM Role for GCR refer: https://cloud.google.com/container-registry/docs/access-control#permissions_and_roles

.

37. Question

.

You are using Deployment Manager to create a Google Kubernetes Engine cluster. Using the same Deployment Manager deployment, you also want to create a DaemonSet in the kube-system namespace of the cluster. You want a solution that uses the fewest possible services. What should you do?

.

- **Add the cluster's API as a new Type Provider in Deployment Manager, and use the new type to create the DaemonSet.**
- Use the Deployment Manager Runtime Configurator to create a new Config resource that contains the DaemonSet definition.
- With Deployment Manager, create a Compute Engine instance with a startup script that uses kubectl to create the DaemonSet.

- In the cluster's definition in Deployment Manager, add a metadata that has kube-system as key and the DaemonSet manifest as value.

Unattempted

In the cluster's definition in Deployment Manager, add a metadata that has kube-system as key and the DaemonSet manifest as value. is not right.
Metadata entries are key-value pairs and do not influence this behavior.
Ref: https://cloud.google.com/compute/docs/storing-retrieving-metadata

With Deployment Manager, create a Compute Engine instance with a startup script that uses kubectl to create the DaemonSet. is not right.
It is possible to spin up a compute engine instance with a startup script that executes kubectl to create a DaemonSet deployment.
kubectl apply -f https://k8s.io/examples/controllers/daemonset.yaml
Ref: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
But this involves using the compute engine service which is an additional service. Our requirement is to achieve using the fewest possible services and as you'll notice later, the correct answer uses fewer services.

Use the Deployment Manager Runtime Configurator to create a new Config resource that contains the DaemonSet definition. is not right.
You can configure the GKE nodes (provisioned by Deployment manager) to report their status to the Runtime Configurator, and when they are UP, you can run a task to create a DaemonSet. While this is possible, it involves one additional service – to run a task e.g. using Cloud Functions, etc. Our requirement is to achieve using the fewest possible services and as you'll notice later, the correct answer uses fewer services.
Here is some more info about Runtime Configurator. The Runtime Configurator feature lets you define and store data as a hierarchy of key-value pairs in Google Cloud Platform.
You can use these key-value pairs as a way to:
1. Dynamically configure services
2. Communicate service states
3. Send notification of changes to data
4. Share information between multiple tiers of services
For example, imagine a scenario where you have a cluster of nodes that run a startup procedure. During startup, you can configure your nodes to report their status to the Runtime Configurator, and then have another application query the Runtime Configurator and run specific tasks based on the status of the nodes.
The Runtime Configurator also offers a Watcher service and a Waiter service. The Watcher service watches a specific key pair and returns when the value of the key pair changes, while the Waiter service waits for a specific end condition and returns a response once that end condition has been met.
Ref: https://cloud.google.com/deployment-manager/runtime-configurator

Add the cluster's API as a new Type Provider in Deployment Manager, and use the new type to create the DaemonSet. is the right answer.
A type provider exposes all resources of a third-party API to Deployment Manager as base types that you can use in your configurations. If you have a cluster running on Google Kubernetes Engine, you could add the cluster as a type provider and access the Kubernetes API using Deployment Manager. Using these inherited API, you can create a DaemonSet.
This option uses just the Deployment Manager to create a DaemonSet and is, therefore, the right answer.
Ref: https://cloud.google.com/deployment-manager/docs/configuration/type-providers/creating-type-provider

.

38. Question

.

You are using Google Kubernetes Engine with autoscaling enabled to host a new application. You want to expose this new application to the public, using HTTPS on a public IP address. What should you do?

.

- **Create a Kubernetes Service of type NodePort for your application, and a Kubernetes Ingress to expose this Service via a Cloud Load Balancer.**
- Create a Kubernetes Service of type ClusterIP for your application. Configure the public DNS name of your application using the IP of this Service.
- Create a Kubernetes Service of type NodePort to expose the application on port 443 of each node of the Kubernetes cluster. Configure the public DNS name of your application with the IP of every node of the cluster to achieve load-balancing.
- Create a HAProxy pod in the cluster to load-balance the traffic to all the pods of the application. Forward the public traffic to HAProxy with an iptable rule. Configure the DNS name of your application using the public IP of the nodeHAProxy is running on.

Unattempted

Create a Kubernetes Service of type ClusterIP for your application. Configure the public DNS name of your application using the IP of this Service. is not right.
Kubernetes Service of type ClusterIP exposes the Service on a cluster-internal IP. Choosing this value makes the Service only reachable from within the cluster so you can not route external traffic to this IP.
Ref: https://kubernetes.io/docs/concepts/services-networking/service/

Create a HAProxy pod in the cluster to load-balance the traffic to all the pods of the application. Forward the public traffic to HAProxy with an iptable rule. Configure the DNS name of your application using the public IP of the nodeHAProxy is running on. is not right.
HAProxy is a popular Kubernetes ingress controller. An Ingress object is an independent resource, apart from Service objects, that configures external access to a service's pods. Ingress Controllers still need a way to receive external traffic. This can be done by exposing the Ingress Controller as a Kubernetes service with either NodePort or LoadBalancer type. You can't use public IP of the node the HAProxy is running on as this may be running in any node in the Kubernetes Cluster and in most cases, these nodes do not have public IPs. They are meant to be private and the pods/deployments are accessed through Service objects.
Ref: https://www.haproxy.com/blog/dissecting-the-haproxy-kubernetes-ingress-controller/

Create a Kubernetes Service of type NodePort to expose the application on port 443 of each node of the Kubernetes cluster. Configure the public DNS name of your application with the IP of every node of the cluster to achieve load-balancing. is not right.
Kubernetes Service of type NodePort uses a port in the range 30000-32767. Assuming that all the nodes have public IP addresses, enabling NodePort would expose a port such as 32000 so the application is accessible on https://IP:32000 which is not ideal. You want your application/website to be reachable directly on port 443. This also requires downstream clients to have awareness of all of your nodes' IP addresses, since they will need to connect to those addresses directly. In other words, they won't be able to connect to a single, proxied IP address. And this is against our requirement of "a public IP address".
Ref: https://kubernetes.io/docs/concepts/services-networking/service/
Ref: https://www.haproxy.com/blog/dissecting-the-haproxy-kubernetes-ingress-controller/

Create a Kubernetes Service of type NodePort for your application, and a Kubernetes Ingress to expose this Service via a Cloud Load Balancer. is the right answer.
This meets all our requirements. With (Global) Cloud Load Balancing, a single anycast IP front-ends all your backend instances in regions around the world. It provides cross-region load balancing, including automatic multi-region failover, which gently moves traffic in fractions if backends become unhealthy.
Ref: https://cloud.google.com/load-balancing/
The ingress accepts traffic from the cloud load balancer and can distribute the traffic across the pods in the cluster.
Ref: https://kubernetes.io/docs/concepts/services-networking/ingress/

.

39. Question

.

You are using multiple configurations for gcloud. You want to review the configured Kubernetes Engine cluster of an inactive configuration using the fewest possible steps. What should you do?

.

- Use gcloud config configurations describe to review the output.
- Use gcloud config configurations activate and gcloud config list to review the output.
- **Use kubectl config get-contexts to review the output.**
- Use kubectl config use-context and kubectl config view to review the output.

Unattempted

Our requirement is to get to the end goal with the fewest possible steps.

Use gcloud config configurations describe to review the output. is not right.
gcloud config configurations describe – describes a named configuration by listing its properties. This does not return any Kubernetes cluster details.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/configurations/describe

Use gcloud config configurations activate and gcloud config list to review the output. is not right.
gcloud config configurations activate – activates an existing named configuration. This does not return any Kubernetes cluster details.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/configurations/activate

Use kubectl config get-contexts to review the output. is the right answer.
kubectl config get-contexts displays a list of contexts as well as the clusters that use them. Here's a sample output.
$ kubectl config get-contexts
CURRENT NAME CLUSTER
gke_kubernetes-260922_us-central1-a_standard-cluster-1 gke_kubernetes-260922_us-central1-a_standard-cluster-1
gke_kubernetes-260922_us-central1-a_your-first-cluster-1 gke_kubernetes-260922_us-central1-a_your-first-cluster-1
* gke_kubernetes-260922_us-central1_standard-cluster-1 gke_kubernetes-260922_us-central1_standard-cluster-1

The output shows the clusters and the configurations they use. Using this information, it is possible to find out the cluster using the inactive configuration with just 1 step.

Use kubectl config use-context and kubectl config view to review the output. is not right. kubectl config use-context [my-cluster-name] is used to set the default context to [my-cluster-name]. But in order to do this, we first need a list of contexts and if you have multiple contexts, you'd need to execute kubectl config use-context [my-cluster-name] against each context. So that is at least 2+ steps. Further to that, the kubectl config view is used to get a full list of config. The output of the kubectl config view can be used to verify which clusters use what configuration but that is one additional step. Moreover, the output of the kubectl config view doesn't change much from one context to other – other than the current-context field. So our earlier steps of determining the contexts and using each context are of not much use. Though this can be used to achieve the same outcome, it involves more steps than the other option.

Here's a sample execution
Step 1: First get a list of contexts
kubectl config get-contexts -o=name
gke_kubernetes-260922_us-central1-a_standard-cluster-1
gke_kubernetes-260922_us-central1-a_your-first-cluster-1
gke_kubernetes-260922_us-central1_standard-cluster-1

Step 2: Use each context and view the config.
kubectl config use-context gke_kubernetes-260922_us-central1-a_standard-cluster-1
Switched to context "gke_kubernetes-260922_us-central1-a_standard-cluster-1".
kubectl config view > 1.out (this saves the output in of config view in 1.out)

kubectl config use-context gke_kubernetes-260922_us-central1-a_your-first-cluster-1
Switched to context "gke_kubernetes-260922_us-central1-a_your-first-cluster-1".
kubectl config view > 2.out (this saves the output in of config view in 2.out)

kubectl config use-context gke_kubernetes-260922_us-central1_standard-cluster-1
Switched to context "gke_kubernetes-260922_us-central1_standard-cluster-1".
kubectl config view > 3.out (this saves the output in of config view in 3.out)

diff 1.out 2.out
28c28
< current-context: gke_kubernetes-260922_us-central1-a_standard-cluster-1 --- >
current-context: gke_kubernetes-260922_us-central1-a_your-first-cluster-1

```
diff 2.out 3.out
28c28
< current-context: gke_kubernetes-260922_us-central1-a_your-first-cluster-1 --- >
current-context: gke_kubernetes-260922_us-central1_standard-cluster-1
```

Step 3: Determine the inactive configuration and the cluster using that configuration. The config itself has details about the clusters and contexts as shown below.

```
$ kubectl config view
apiVersion: v1
clusters:
- cluster:
certificate-authority-data: DATA+OMITTED
server: https://35.222.130.166
name: gke_kubernetes-260922_us-central1-a_standard-cluster-1
- cluster:
certificate-authority-data: DATA+OMITTED
server: https://35.225.14.172
name: gke_kubernetes-260922_us-central1-a_your-first-cluster-1
- cluster:
certificate-authority-data: DATA+OMITTED
server: https://34.69.212.109
name: gke_kubernetes-260922_us-central1_standard-cluster-1
contexts:
- context:
cluster: gke_kubernetes-260922_us-central1-a_standard-cluster-1
user: gke_kubernetes-260922_us-central1-a_standard-cluster-1
name: gke_kubernetes-260922_us-central1-a_standard-cluster-1
- context:
cluster: gke_kubernetes-260922_us-central1-a_your-first-cluster-1
user: gke_kubernetes-260922_us-central1-a_your-first-cluster-1
name: gke_kubernetes-260922_us-central1-a_your-first-cluster-1
- context:
cluster: gke_kubernetes-260922_us-central1_standard-cluster-1
user: gke_kubernetes-260922_us-central1_standard-cluster-1
name: gke_kubernetes-260922_us-central1_standard-cluster-1
current-context: gke_kubernetes-260922_us-central1-a_standard-cluster-1
```

-

40. Question

-

You built an application on Google Cloud Platform that uses Cloud Spanner. The support team needs to monitor the environment but should not have access to the data. You need a streamlined solution to grant the correct permissions to your support team, and you want to follow Google recommended practices. What should you do?

- Add the support team group to the roles/spanner.database.reader role
- Add the support team group to the roles/stackdriver.accounts.viewer role
- **Add the support team group to the roles/monitoring.viewer role**
- Add the support team group to the roles/spanner.database.user role

Unattempted

Requirements –
1. Monitoring access but no data access
2. Streamlined solution
3. Google recommended practices (i.e. look for something out of the box).

roles/spanner.databaseReader provides permission to read from the Spanner database, execute SQL queries on the database, and view the schema. Since this provides read access to data, roles/spanner.databaseReader. is not right.

roles/spanner.databaseUser provides permission to read from and write to the Spanner database, execute SQL queries on the database, and view and update the schema. Since this provides both read and write access to data, roles/spanner.databaseUser. is not right.

roles/stackdriver.accounts.viewer read-only access to get and list information about Stackdriver account structure. Since this does not provide monitor access to Cloud Spanner, roles/stackdriver.accounts.viewer. is not right.

roles/monitoring.viewer provides read-only access to get and list information about all monitoring data and configurations. This role provides monitoring access and fits our requirements. roles/monitoring.viewer. is the right answer.
Ref: https://cloud.google.com/iam/docs/understanding-roles#cloud-spanner-roles

41. Question

You create a Deployment with 2 replicas in a Google Kubernetes Engine cluster that has a single preemptible node pool. After a few minutes, you use kubectl to examine the status

of your Pod and observe that one of them is still in Pending status:
NAME READY STATUS RESTART AGE
myapp-deployment-58ddbbb995-lp86m 0/1 Pending 0 9m
myapp-deployment-58ddbbb995-qjpkg 1/1 Running 0 9m
What is the most likely cause?

.

- The pending Pod's resource requests are too large to fit on a single node of the cluster.
- **Too many Pods are already running in the cluster, and there are not enough resources left to schedule the pending Pod.**
- The node pool is configured with a service account that does not have permission to pull the container image used by the pending Pod.
- The pending Pod was originally scheduled on a node that has been preempted between the creation of the Deployment and your verification of the Pod status. It is currently being rescheduled on a new node.

Unattempted

The pending Pod was originally scheduled on a node that has been preempted between the creation of the Deployment and your verification of the Pod status. It is currently being rescheduled on a new node. is not right.
Our question states that we provisioned a Google Kubernetes Engine cluster with a single preemptible node pool.

The node pool is configured with a service account that does not have permission to pull the container image used by the pending Pod. is not right.
If the node pool has permission issues when pulling the container image, the other pod would not be in Running status. And the status would have been ImagePullBackOff if there was a problem pulling the image.

The pending Pod's resource requests are too large to fit on a single node of the cluster. is not right.
If the resource requests in Pod specification are too large to fit on the node, the other pod would not be in Running status, i.e. both pods should have been in pending status if this was the case.
Ref: The pending Pod's resource requests are too large to fit on a single node of the cluster.

Too many Pods are already running in the cluster, and there are not enough resources left to schedule the pending Pod. is the right answer.
When you have a deployment with some pods in running and other pods in the pending state, more often than not it is a problem with resources on the nodes. Here's a sample

output of this use case. We see that the problem is with insufficient CPU on the Kubernetes nodes so we have to either enable auto-scaling or manually scale up the nodes.
kubectl describe pod myapp-deployment-58ddbbb995-lp86m
Events:
Type Reason Age From Message
---- ------ ---- ---- -------
Warning FailedScheduling 28s (x4 over 3m1s) default-scheduler 0/1 nodes are available: 1 Insufficient cpu.

.

42. Question

.

You create a new Google Kubernetes Engine (GKE) cluster and want to make sure that it always runs a supported and stable version of Kubernetes. What should you do?

.

- Select "Container-Optimized OS (cos)" as a node image for your GKE cluster.
- Select the latest available cluster version for your GKE cluster.
- **Enable the Node Auto-Upgrades feature for your GKE cluster.**
- Enable the Node Auto-Repair feature for your GKE cluster.

Unattempted

Enable the Node Auto-Repair feature for your GKE cluster. is not right.
GKE's node auto-repair feature helps you keep the nodes in your cluster in a healthy, running state. When enabled, GKE makes periodic checks on the health state of each node in your cluster. If a node fails consecutive health checks over an extended time period, GKE initiates a repair process for that node.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-repair

Select the latest available cluster version for your GKE cluster. is not right.
We can certainly select the latest available cluster version at the time of GKE cluster provisioning, however, this does not automatically upgrade the cluster if new versions become available.

Select "Container-Optimized OS (cos)" as a node image for your GKE cluster. is not right.
Container-Optimized OS comes with the Docker container runtime and all Kubernetes components pre-installed for out of the box deployment, management, and orchestration of your containers. But these do not help with automatically upgrading GKE cluster

versions.
Ref: https://cloud.google.com/container-optimized-os

Enable the Node Auto-Upgrades feature for your GKE cluster. is the right answer.
Node auto-upgrades help you keep the nodes in your cluster up to date with the cluster master version when your master is updated on your behalf. When you create a new cluster or node pool with Google Cloud Console or the gcloud command, node auto-upgrade is enabled by default.
Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-upgrades

.

43. Question

.

You created a cluster.YAML file containing
resources:
– name: cluster
type: container.v1.cluster
properties:
zone: europe-west1-b
cluster:
description: "My GCP ACE cluster"
initialNodeCount: 2
You want to use Cloud Deployment Manager to create this cluster in GKE. What should you do?

.

- **gcloud deployment-manager deployments create my-gcp-ace-cluster --config cluster.yaml**
- gcloud deployment-manager deployments create my-gcp-ace-cluster --type container.v1.cluster --config cluster.yaml
- gcloud deployment-manager deployments apply my-gcp-ace-cluster --config cluster.yaml
- gcloud deployment-manager deployments apply my-gcp-ace-cluster --type container.v1.cluster --config cluster.yaml

Unattempted

gcloud deployment-manager deployments apply my-gcp-ace-cluster –config cluster.yaml. is not right.
"gcloud deployment-manager deployments" doesn't support action apply. With Google cloud in general, the action for creating is create and the action for retrieving is list. With Kubernetes resources, the corresponding actions are apply and get respectively.

Ref: https://cloud.google.com/sdk/gcloud/reference/deployment-manager/deployments/create

gcloud deployment-manager deployments apply my-gcp-ace-cluster –type container.v1.cluster –config cluster.yaml. is not right.
"gcloud deployment-manager deployments" doesn't support action apply. With Google cloud in general, the action for creating is create and the action for retrieving is list. With Kubernetes resources, the corresponding actions are apply and get respectively.
Ref: https://cloud.google.com/sdk/gcloud/reference/deployment-manager/deployments/create

gcloud deployment-manager deployments create my-gcp-ace-cluster –type container.v1.cluster –config cluster.yaml. is not right.
"gcloud deployment-manager deployments create" creates deployments based on the configuration file. (Infrastructure as code). It doesn't expect the parameter type passed to it directly and fails when executed with the type parameter.
Ref: https://cloud.google.com/sdk/gcloud/reference/deployment-manager/deployments/create

gcloud deployment-manager deployments create my-gcp-ace-cluster –config cluster.yaml. is the right answer.
"gcloud deployment-manager deployments create" creates deployments based on the configuration file. (Infrastructure as code). All the configuration related to the artifacts is in the configuration file. This command correctly creates a cluster based on the provided cluster.yaml configuration file.
Ref: https://cloud.google.com/sdk/gcloud/reference/deployment-manager/deployments/create

- **44. Question**

- You created a compute instance by running gcloud compute instances create instance1. You intended to create the instance in project gcp-ace-proj-266520 but the instance got created in a different project. Your cloud shell gcloud configuration is as shown.
$ gcloud config list
[component_manager]
disable_update_check = True
[compute]
gce_metadata_read_timeout_sec = 5
zone = europe-west2-a

[core]
account = gcp-ace-lab-user@gmail.com
disable_usage_reporting = False
project = gcp-ace-lab-266520
[metrics]
environment = devshell
What should you do to delete the instance that was created in the wrong project and recreate it in gcp-ace-proj-266520 project?

.

- gcloud compute instances delete instance1 gcloud config set compute/project gcp-ace-proj-266520 gcloud compute instances create instance1
- gcloud config set project gcp-ace-proj-266520 gcloud compute instances recreate instance1 --previous-project gcp-ace-lab-266520
- gcloud compute instances delete instance1 gcloud compute instances create instance1
- **gcloud compute instances delete instance1 gcloud config set project gcp-ace-proj-266520 gcloud compute instances create instance1**

Unattempted

gcloud compute instances delete instance1
gcloud compute instances create instance1. is not right.
The default core/project property is set to gcp-ace-lab-266520 in our current configuration so the instance would have been created in this project. Running the first command to delete the instance correctly deletes it from this project but we haven't modified the core/project property before executing the second command so the instance is recreated in the same project which is not what we want.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/delete

gcloud config set project gcp-ace-proj-266520
gcloud compute instances recreate instance1 –previous-project gcp-ace-lab-266520. is not right.
gcloud compute instances command doesn't support recreate action. It supports create/delete which is what we are supposed to use for this requirement.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances

gcloud compute instances delete instance1
gcloud config set compute/project gcp-ace-proj-266520
gcloud compute instances create instance1. is not right.
The approach is right but the syntax is wrong. gcloud config does not have a compute/project property. The project property is part of the core/ section as seen in the

output of gcloud configuration list in the question. In this scenario, we are trying to set compute/project property that doesn't exist in the compute section so the command fails.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/set

gcloud compute instances delete instance1
gcloud config set project gcp-ace-proj-266520
gcloud compute instances create instance1. is the right answer.
This sequence of commands correctly deletes the instance from gcp-ace-lab-266520 which is the default project in the active gcloud configuration, then modifies the current configuration to set the default project to gcp-ace-proj-266520, and finally creates the instance in the project gcp-ace-proj-266520 which is the default project in active gcloud configuration at the time of running the command. This produces the intended outcome of deleting the instance from gcp-ace-lab-266520 project and recreating it in gcp-ace-prod-266520
Ref: https://cloud.google.com/sdk/gcloud/reference/config/set
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/create
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/delete

.

45. Question

.

You created a Google Cloud Platform project with an App Engine application inside the project. You initially configured the application to be served from the us-central region. Now you want the application to be served from the asia-northeast1 region. What should you do?

.

- **Create a new GCP project and create an App Engine application inside this new project. Specify asia-northeast1 as the region to serve your application.**
- Create a second App Engine application in the existing GCP project and specify asia-northeast1 as the region to serve your application.
- Change the default region property setting in the existing GCP project to asia-northeast1.
- Change the region property setting in the existing App Engine application from us-central to asia-northeast1.

Unattempted

Change the default region property setting in the existing GCP project to asia-northeast1. is not right.
App Engine is regional, which means the infrastructure that runs your apps is located in a specific region and is managed by Google to be redundantly available across all the zones

within that region. You cannot change an app's region after you set it.
Ref: https://cloud.google.com/appengine/docs/locations

Change the region property setting in the existing App Engine application from us-central to asia-northeast1. is not right.
App Engine is regional, which means the infrastructure that runs your apps is located in a specific region and is managed by Google to be redundantly available across all the zones within that region. You cannot change an app's region after you set it.
Ref: https://cloud.google.com/appengine/docs/locations

Create a second App Engine application in the existing GCP project and specify asia-northeast1 as the region to serve your application. is not right.
App Engine is regional and you cannot change an app's region after you set it. You can deploy additional services in the App Engine but they will all be targeted to the same region.
Ref: https://cloud.google.com/appengine/docs/locations

Create a new GCP project and create an App Engine application inside this new project. Specify asia-northeast1 as the region to serve your application. is the right answer.
App Engine is regional and you cannot change an app's region after you set it. Therefore, the only way to have an app run in another region is by creating a new project and targeting the app engine to run in the required region (asia-northeast1 in our case).
Ref: https://cloud.google.com/appengine/docs/locations

.

46. Question

.

You created a Kubernetes deployment by running kubectl run nginx –image=nginx –labels=''app=prod''. Your Kubernetes cluster is also used by a number of other deployments. How can you find the identifier of the pods for this nginx deployment?

.

- kubectl get deployments --output=pods
- gcloud get pods --selector="app=prod"
- gcloud list gke-deployments --filter={ pod }
- **kubectl get pods -l "app=prod"**

Unattempted

gcloud get pods –selector=''app=prod''. is not right.
You can not retrieve pods from the Kubernetes cluster by using gcloud. You can list pods

by using Kubernetes CLI – kubectl get pods.
Ref: https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/

gcloud list gke-deployments –filter={ pod }. is not right.
You can not retrieve pods from the Kubernetes cluster by using gcloud. You can list pods by using Kubernetes CLI – kubectl get pods.
Ref: https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/

kubectl get deployments –output=pods. is not right.
You can not list pods by listing Kubernetes deployments. You can list pods by using Kubernetes CLI – kubectl get pods.
Ref: https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/

kubectl get pods -l "app=prod". is the right answer.
This command correctly lists pods that have the label app=prod. When creating the deployment, we used the label app=prod so listing pods that have this label retrieve the pods belonging to nginx deployments. You can list pods by using Kubernetes CLI – kubectl get pods.
Ref: https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/
Ref: https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/#list-containers-filtering-by-pod-label

- **47. Question**

- You created a Kubernetes deployment by running kubectl run nginx –image=nginx –replicas=1. After a few days, you decided you no longer want this deployment. You identified the pod and deleted it by running kubectl delete pod. You noticed the pod got recreated.
$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-84748895c4-nqqmt 1/1 Running 0 9m41s

- $ kubectl delete pod nginx-84748895c4-nqqmt
pod "nginx-84748895c4-nqqmt" deleted

.

$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-84748895c4-k6bzl 1/1 Running 0 25s
What should you do to delete the deployment and avoid pod getting recreated?

.

- kubectl delete nginx
- kubectl delete --deployment=nginx
- kubectl delete pod nginx-84748895c4-k6bzl --no-restart
- **kubectl delete deployment nginx**

Unattempted

kubectl delete pod nginx-84748895c4-k6bzl –no-restart. is not right.
kubectl delete pod command does not support the flag –no-restart. The command fails to execute due to the presence of an invalid flag.
$ kubectl delete pod nginx-84748895c4-k6bzl –no-restart
Error: unknown flag: –no-restart
Ref: https://kubernetes.io/docs/reference/kubectl/cheatsheet/#deleting-resources

kubectl delete –deployment=nginx. is not right.
kubectl delete command does not support the parameter –deployment. The command fails to execute due to the presence of an invalid parameter.
$ kubectl delete –deployment=nginx
Error: unknown flag: –deployment
Ref: https://kubernetes.io/docs/reference/kubectl/cheatsheet/#deleting-resources

kubectl delete nginx. is not right.
We haven't provided the kubectl delete command information on what to delete, whether a pod, a service or a deployment. The command syntax is wrong and fails to execute.
$ kubectl delete nginx
error: resource(s) were provided, but no name, label selector, or –all flag specified
Ref: https://kubernetes.io/docs/reference/kubectl/cheatsheet/#deleting-resources

kubectl delete deployment nginx. is the right answer.
This command correctly deletes the deployment. Pods are managed by kubernetes workloads (deployments). When a pod is deleted, the deployment detects the pod is unavailable and brings up another pod to maintain the replica count. The only way to delete the workload is by deleting the deployment itself using the kubectl delete deployment command.
$ kubectl delete deployment nginx

deployment.apps "nginx" deleted
Ref: https://kubernetes.io/docs/reference/kubectl/cheatsheet/#deleting-resources

.

48. Question

.

You created an instance of SQL Server 2017 on Compute Engine to test features in the new version. You want to connect to this instance using the fewest number of steps. What should you do?

.

- Set a Windows password in the GCP Console. Verify that a firewall rule for port 22 exists. Click the RDP button in the GCP Console and supply the credentials to log in.
- Set a Windows username and password in the GCP Console. Verify that a firewall rule for port 3389 exists. Click the RDP button in the GCP Console, and supply the credentials to log in.
- Install an RDP client on your desktop. Verify that a firewall rule for port 3389 exists.
- **Install an RDP client on your desktop. Set a Windows username and password in the GCP Console. Use the credentials to log in to the instance.**

Unattempted

Requirements – Connect to compute instance using fewest steps. The presence of SQL Server 2017 on the instance is a red herring and should be ignored as none of the options provided say anything about the database and all seem to revolve around RDP.

Install a RDP client on your desktop. Verify that a firewall rule for port 3389 exists. is not right.
Although opening port 3389 is essential for serving RDP traffic, we do not have the credentials to RDP so this isn't going to work.

Set a Windows password in the GCP Console. Verify that a firewall rule for port 22 exists. Click the RDP button in the GCP Console and supply the credentials to log in. is not right.
RDP uses port 3389 and not 22.
Ref: https://cloud.google.com/compute/docs/troubleshooting/troubleshooting-rdp

Set a Windows username and password in the GCP Console. Verify that a firewall rule for port 3389 exists. Click the RDP button in the GCP Console, and supply the credentials to log in. is not right.

While this option correctly sets the username and password on the console and verifies a firewall rule is set on port 3389 to allow RDP traffic, you can RDP from console unless you install Chrome RDP for Google Cloud Platform extension in order to RDP from the console. (See Chrome Desktop for GCP tab in https://cloud.google.com/compute/docs/instances/connecting-to-instance#windows). If we assume that installing Chrome RDP for Google Cloud Platform extension is carried out (even though not specified in the option), we end up executing more steps in this option to successfully RDP compare to the correct answer (below)

Install an RDP client on your desktop. Set a Windows username and password in the GCP Console. Use the credentials to log in to the instance. is the right answer.
This option correctly sets the username/password which is essential. In addition, the default VPC comes with port 3389 open to the public. The question doesn't explicitly state the compute engine is in a custom VPC so it is safe to assume we are using default VPC which has default RDP access open to the public. Finally, you install an RDP client on the desktop and use the credentials set up earlier to RDP to the server.

.

49. Question

.

You defined an instance template for a Python web application. When you deploy this application in Google Compute Engine, you want to ensure the service scales up and scales down automatically based on the number of HTTP requests. What should you do?

.

- 1. Create the necessary number of instances based on the instance template to handle peak user traffic. 2. Group the instances together in an unmanaged instance group. 3. Configure the instance group as the Backend Service of an External HTTP(S) load balancer.
- 1. Create an instance from the instance template. 2. Create an image from the instance's disk and export it to Cloud Storage. 3. Create an External HTTP(s) load balancer and add the Cloud Storage bucket as its backend service.
- 1. Create an unmanaged instance group from the instance template. 2. Configure autoscaling on the unmanaged instance group with a scaling policy based on HTTP traffic. 3. Configure the unmanaged instance group as the backend service of an Internal HTTP(S) load balancer.
- 1. Deploy your Python web application instance template to Google Cloud App Engine. 2. Configure autoscaling on the managed instance group with a scaling policy based on HTTP traffic.

- 1. Create a managed instance group from the instance template. 2. Configure autoscaling on the managed instance group with a scaling policy based on HTTP traffic. 3. Configure the instance group as the backend service of an External HTTP(S) load balancer.

Unattempted

1. Create an instance from the instance template.
2. Create an image from the instance's disk and export it to Cloud Storage.
3. Create an External HTTP(s) load balancer and add the Cloud Storage bucket as its backend service. is not right.
You can upload a custom image from instance's boot disk and export it to cloud storage.
https://cloud.google.com/compute/docs/images/export-image
However, this image in the Cloud Storage bucket is unable to handle traffic as it is not a running application. Cloud Storage can not serve requests of the custom image.

1. Create an unmanaged instance group from the instance template.
2. Configure autoscaling on the unmanaged instance group with a scaling policy based on HTTP traffic.
3. Configure the unmanaged instance group as the backend service of an Internal HTTP(S) load balancer. is not right.
An unmanaged instance group does not autoscale. An unmanaged instance group is a collection of virtual machines (VMs) that reside in a single zone, VPC network, and subnet. An unmanaged instance group is useful for grouping together VMs that require individual configuration settings or tuning.
Ref: https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-unmanaged-instances

1. Create the necessary number of instances based on the instance template to handle peak user traffic.
2. Group the instances together in an unmanaged instance group.
3. Configure the instance group as the Backend Service of an External HTTP(S) load balancer. is not right.
An unmanaged instance group does not autoscale. Although we may have enough compute power to handle peak user traffic, it does not automatically scale down when the traffic goes down so it doesn't meet our requirements.
Ref: https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-unmanaged-instances

1. Deploy your Python web application instance template to Google Cloud App Engine.
2. Configure autoscaling on the managed instance group with a scaling policy based on HTTP traffic. is not right.
You can not use compute engine instance templates to deploy applications to Google Cloud

App Engine. Google App Engine lets you deploy applications quickly by providing run time environments for many of the popular languages like Java, PHP, Node.js, Python, C#, .Net, Ruby, and Go. You have an option of using custom runtimes but using compute engine instance templates is not an option.
Ref: https://cloud.google.com/appengine

1. Create a managed instance group from the instance template.
2. Configure autoscaling on the managed instance group with a scaling policy based on HTTP traffic.
3. Configure the instance group as the backend service of an External HTTP(S) load balancer. is the right answer.
The auto-scaling capabilities of Managed instance groups let you automatically add or delete instances from a managed instance group based on increases or decreases in load – this can be set up by configuring scaling policies. In addition, you can configure External HTTP(S) load balancer to send traffic to the managed instance group. The External HTTP(S) load balancer tries to balance requests by using a round-robin algorithm and when the load increases beyond the threshold defined in the scaling policy, autoscaling kicks in and adds more nodes.
Ref: https://cloud.google.com/load-balancing/docs/https
Ref: https://cloud.google.com/compute/docs/instance-groups/creating-groups-of-managed-instances

- **50. Question**

- You deployed a new application inside your Google Kubernetes Engine cluster using the YAML file specified below.
apiVersion: apps/v1
kind: Deployment
metadata:
name: myapp-deployment
spec:
selector:
matchLabels:
app: myapp
replicas: 2
template:
metadata:
labels:
app: myapp
spec:

containers:
– name: myapp
image: myapp:1.1
ports:
– containerPort: 80
—
apiVersion: v1
kind: Service
metadata:
name: myapp-service
spec:
ports:
– port: 8000
targetPort: 80
protocol: TCP
selector:
app: myapp
You check the status of the deployed pods and notice that one of them is still in PENDING status:
kubectl get pods -l app=myapp

.

NAME READY STATUS RESTART AGE
myapp-deployment-58ddbbb995-lp86m 0/1 Pending 0 9m
myapp-deployment-58ddbbb995-qjpkg 1/1 Running 0 9m
You want to find out why the pod is stuck in pending status. What should you do?

.

- Review details of the myapp-service Service object and check for error messages.
- Review details of the myapp-deployment Deployment object and check for error messages.
- **Review details of myapp-deployment-58ddbbb995-lp86m Pod and check for warning messages.**
- View logs of the container in myapp-deployment-58ddbbb995-lp86m pod and check for warning messages.

Unattempted

Review details of the myapp-service Service object and check for error messages. is not right.
The question states we have a problem with the deployment. Checking/Reviewing the status of the service object isn't of much use here.

View logs of the container in myapp-deployment-58ddbbb995-lp86m pod and check for warning messages. is not right.
Since the pod hasn't moved to Running state, the logs of the container would be empty. So running
kubectl logs pod/myapp-deployment-58ddbbb995-lp86m
to check the logs of the pod isn't of much use.

Review details of the myapp-deployment Deployment object and check for error messages. is not right.
Describing the details of the deployment shows us how many of the pods are available and unavailable but does not show errors/warnings related to a specific pod.
Here's a sample output of this use case.
kubectl describe deployment myapp-deployment
Replicas: 3 desired | 3 updated | 3 total | 2 available | 1 unavailable
Events:
Type Reason Age From Message
—- —— —- —- ——-
Normal ScalingReplicaSet 4m54s deployment-controller Scaled up replica set myapp-deployment-869d88c75f to 3

Review details of myapp-deployment-58ddbbb995-lp86m Pod and check for warning messages. is the right answer.
Since the problem is with a specific pod, looking at the details of the pod is the best solution. When you have a deployment with some pods in running and other pods in Pending state, more often than not it is a problem with resources on the nodes. Here's a sample output of this use case. We see that the problem is with insufficient CPU on the Kubernetes nodes so we have to either enable auto-scaling or manually scale up the nodes.
kubectl describe pod myapp-deployment-58ddbbb995-lp86m
Events:
Type Reason Age From Message
—- —— —- —- ——-
Warning FailedScheduling 28s (x4 over 3m1s) default-scheduler 0/1 nodes are available: 1 Insufficient cpu.

- **51. Question**

- You deployed a number of services to Google App Engine Standard. The services are designed as microservices with several interdependencies between them. Most services have few version upgrades but some key services have over 20 version upgrades. You identified an issue with the service pt-createOrder and deployed a new version v3 for this service.

You are confident this works and want this new version to receive all traffic for the service. You want to minimize effort and ensure the availability of service. What should you do?

.

- Execute gcloud app versions stop v2 and gcloud app versions start v3
- Execute gcloud app versions stop v2 --service="pt-createOrder" and gcloud app versions start v3 --service="pt-createOrder"
- Execute gcloud app versions migrate v3
- **Execute gcloud app versions migrate v3 --service="pt-createOrder"**

Unattempted

Execute gcloud app versions migrate v3. is not right.
gcloud app versions migrate v3 migrates all services to version v3. In our scenario, we have multiple services with each service potentially being on a different version. We don't want to migrate all services to v3, instead, we only want to migrate the pt-createOrder service to v3. Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/migrate

Execute gcloud app versions stop v2 –service=''pt-createOrder'' and gcloud app versions start v3 –service=''pt-createOrder''. is not right.
Stopping version v2 and starting version v3 for pt-createOrder service would result in v3 receiving all traffic for pt-createOrder. While this is the intended outcome, stopping version v2 before starting version v3 results in service being unavailable until v3 is ready to receive traffic. As we want to ''ensure availability'', this option is not suitable.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/migrate

Execute gcloud app versions stop v2 and gcloud app versions start v3. is not right.
Stopping version v2 and starting version v3 would result in migrating all services to version v3 which is undesirable. We don't want to migrate all services to v3, instead, we only want to migrate the pt-createOrder service to v3. Moreover, stopping version v2 before starting version v3 results in service being unavailable until v3 is ready to receive traffic. As we want to ''ensure availability'', this option is not suitable.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/migrate

Execute gcloud app versions migrate v3 –service=''pt-createOrder''. is the right answer.
This command correctly migrates the service pt-createOrder to use version 3 and produces the intended outcome while minimizing effort and ensuring the availability of service.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/versions/migrate

.

52. Question

.

You deployed a workload to your GKE cluster by running the command kubectl apply -f app.yaml. You also enabled a LoadBalancer service to expose the deployment by running kubectl apply -f service.yaml. Your pods are struggling due to increased load so you decided to enable horizontal pod autoscaler by running kubectl autoscale deployment [YOUR DEPLOYMENT] –cpu-percent=50 –min=1 –max=10. You noticed the autoscaler has launched several new pods but the new pods have failed with the message "Insufficient cpu". What should you do to resolve this issue?

.

- **Use "gcloud container clusters resize" to add more nodes to the node pool.**
- Use "kubectl container clusters resize" to add more nodes to the node pool.
- Edit the managed instance group of the cluster and enable autoscaling.
- Edit the managed instance group of the cluster and increase the number of VMs by 1.

Unattempted

Use "kubectl container clusters resize" to add more nodes to the node pool. is not right.
kubectl doesn't support the command kubectl container clusters resize. You have to use gcloud container clusters resize to resize a cluster.
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/resize

Edit the managed instance group of the cluster and increase the number of VMs by 1. is not right.
GKE Cluster does not use a managed instance group. Instead, the cluster master (control plan) handles the lifecycle of nodes in the node pools. The cluster master is responsible for managing the workloads' lifecycle, scaling, and upgrades. The master also manages network and storage resources for those workloads.
Ref: https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture

Edit the managed instance group of the cluster and enable autoscaling. is not right.
GKE Cluster does not use a managed instance group. Instead, the cluster master (control plan) handles the lifecycle of nodes in the node pools. The cluster master is responsible for managing the workloads' lifecycle, scaling, and upgrades. The master also manages network and storage resources for those workloads.
Ref: https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture

Use "gcloud container clusters resize" to add more nodes to the node pool. is the right answer.

Your pods are failing with "Insufficient cpu". This is because the existing nodes in the node pool are maxed out, therefore, you need to add more nodes to your node pool. For such scenarios, enabling cluster autoscaling is ideal, however, this is not in any of the answer options. In the absence of cluster autoscaling, the next best approach is to add more nodes to the cluster manually. This is achieved by running the command gcloud container clusters resize which resizes an existing cluster for running containers.
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/resize

.

53. Question

.

You deployed an App Engine application using gcloud app deploy, but it did not deploy to the intended project. You want to find out why this happened and where the application deployed. What should you do?

.

- Check the app YAML file for your application and check the project settings.
- Go to Deployment Manager and review settings for the deployment of application
- Check the web application XML file for your application and check project settings
- **Go to Cloud Shell and run gcloud config list to review the Google Cloud configurations used for deployment.**

Unattempted

Check the app YAML file for your application and check the project settings. is not right.
The Yaml file of application does not hold Google project information.

Check the web application XML file for your application and check project settings. is not right.
The web application file of the application does not hold Google project information.

Go to Deployment Manager and review settings for the deployment of the application. is not right.
Google Cloud Deployment Manager allows you to specify all the resources needed for your application in a declarative format using yaml. In this scenario, we haven't used Cloud Deployment Manager to deploy. The app was deployed using gcloud app deploy so this option is not right.
Ref: https://cloud.google.com/deployment-manager

Go to Cloud Shell and run gcloud config list to review the Google Cloud configurations used for deployment. is the right answer.
If the deployment was successful but it did not deploy to the intended project, it is likely that the gcloud app deploy command deployed the application to a different project. In the same gcloud shell, you can identify the current properties of the configuration by executing gcloud config list. This returns config properties such as project, account etc, as well as app-specific properties such as app/promote_by_default, app/stop_previous_version.
Ref: https://cloud.google.com/sdk/gcloud/reference/config/list

.

54. Question

.

You deployed an LDAP server on Compute Engine. You want to make sure it is reachable by external clients via TLS through port 636 using UDP. What should you do?

.

- Add the network tag allow-udp-636 to the VM instance running the LDAP server.
- Create a route called allow-udp-636 and set the next hop to be the VM instance running the LDAP server.
- **Add a network tag of your choice to the instance. Create a firewall rule to allow ingress on UDP port 636 for that network tag.**
- Add a network tag of your choice to the instance running the LDAP server. Create a firewall rule to allow egress on UDP port 636 for that network tag.

Unattempted

Create a route called allow-udp-636 and set the next hop to be the VM instance running the LDAP server. is not right.
Google Cloud routes define the paths that network traffic takes from a virtual machine (VM) instance to other destinations. These destinations can be inside your Google Cloud Virtual Private Cloud (VPC) network (for example, in another VM) or outside it. Routes aren't a suitable solution for our requirement as we need to enable EXTERNAL clients to reach our VM on port 636 using UDP.
Ref: https://cloud.google.com/vpc/docs/routes

Add the network tag allow-udp-636 to the VM instance running the LDAP server. is not right.
Tags enable you to make firewall rules and routes applicable to specific VM instances but allow-udp-636 is not a network tag that GCP provides. The default network tags provided by GCP are default-allow-icmp, default-allow-internal, default-allow-rdp and default-allow-ssh. In this scenario, we are assigning a tag to the instance with no network

rules so there would be no difference to behavior.
Ref: https://cloud.google.com/vpc/docs/add-remove-network-tags

Add a network tag of your choice to the instance running the LDAP server. Create a firewall rule to allow egress on UDP port 636 for that network tag. is not right.
We are interested in enabling inbound traffic to our VM whereas egress firewall rules control outgoing connections from target instances in your VPC network.
Ref: https://cloud.google.com/vpc/docs/firewalls#egress_cases

Add a network tag of your choice to the instance. Create a firewall rule to allow ingress on UDP port 636 for that network tag. is the right answer.
This fits all our requirements. Ingress firewall rules control incoming connections from a source to target instances in your VPC network. We can create an ingress firewall rule to allow UDP port 636 for a network tag. And when we assign this network tag to the instance, the firewall rule applies to the instances so traffic is accepted on port 636 using UDP. Although not specified in this option, it has to be assumed that the source for the firewall rule is set to 0.0.0.0/0 i.e. all IP ranges so that external clients are allowed to connect to this VM.
Ref: https://cloud.google.com/vpc/docs/firewalls#ingress_cases

.

55. Question

.

You deployed your application to a default node pool on the GKE cluster and you want to configure cluster autoscaling for this GKE cluster. For your application to be profitable, you must limit the number of Kubernetes nodes to 10. You want to start small and scale up as traffic increases and scale down when the traffic goes down. What should you do?

.

- **Update existing GKE cluster to enable autoscaling by running the command gcloud container clusters update [CLUSTER_NAME] --enable-autoscaling --min-nodes=1 --max-nodes=10**
- Create a new GKE cluster by running the command gcloud container clusters create [CLUSTER_NAME] --enable-autoscaling --min-nodes=1 --max-nodes=10. Redeploy your application
- To enable autoscaling, add a tag to the instances in the cluster by running the command gcloud compute instances add-tags [INSTANCE] --tags=enable-autoscaling,min-nodes=1,max-nodes=10

- Set up a stack driver alert to detect slowness in the application. When the alert is triggered, increase nodes in the cluster by running the command gcloud container clusters resize CLUSTER_Name --size .

Unattempted

Set up a stack driver alert to detect slowness in the application. When the alert is triggered, increase nodes in the cluster by running the command gcloud container clusters resize CLUSTER_Name –size . is not right.
The command gcloud container clusters resize command resizes an existing cluster for running containers. While it is possible to manually increase the number of nodes in the cluster by running the command, the scale-up is not automatic, it is a manual process. Also, there is no scale down so it doesn't fit our requirement of "scale up as traffic increases and scale down when the traffic goes down".
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/resize

To enable autoscaling, add a tag to the instances in the cluster by running the command gcloud compute instances add-tags [INSTANCE] –tags=enable-autoscaling,min-nodes=1,max-nodes=10. is not right.
Autoscaling can not be enabled on the GKE cluster by adding tags on compute instances. Autoscaling can be enabled at the time of creating the cluster and can also be enabled for existing clusters by running one of the gcloud container clusters to create/update commands.
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/create
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/update

Create a new GKE cluster by running the command gcloud container clusters create [CLUSTER_NAME] –enable-autoscaling –min-nodes=1 –max-nodes=10. Redeploy your application. is not right.
The command gcloud container clusters create – creates a GKE cluster and the flag –enable-autoscaling enables autoscaling and the parameters –min-nodes=1 –max-nodes=10 define the minimum and maximum number of nodes in the node pool. However, we want to configure cluster autoscaling for the existing GKE cluster; not create a new GKE cluster.
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/create

Update existing GKE cluster to enable autoscaling by running the command gcloud container clusters update [CLUSTER_NAME] –enable-autoscaling –min-nodes=1 –max-nodes=10. is the right answer.
The command gcloud container clusters update – updates an existing GKE cluster. The flag –enable-autoscaling enables autoscaling and the parameters –min-nodes=1 –max-nodes=10 define the minimum and maximum number of nodes in the node pool. This

enables cluster autoscaling which scales up and scales down the nodes automatically between 1 and 10 nodes in the node pool.

.

56. Question

.

You developed a web application that lets users upload and share images. You deployed this application in Google Compute Engine and you have configured Stackdriver Logging. Your application sometimes times out while uploading large images, and your application generates relevant error log entries that are ingested to Stackdriver Logging. You would now like to create alerts based on these metrics. You intend to add more compute resources manually when the number of failures exceeds a threshold. What should you do in order to alert based on these metrics with minimal effort?

.

- In Stackdriver logging, create a new logging metric with the required filters, edit the application code to set the metric value when needed, and create an alert in Stackdriver based on the new metric.
- Create a custom monitoring metric in code, edit the application code to set the metric value when needed, create an alert in Stackdriver based on the new metric.
- **In Stackdriver Logging, create a custom monitoring metric from log data and create an alert in Stackdriver based on the new metric.**
- Add the Stackdriver monitoring and logging agent to the instances running the code.

Unattempted

In Stackdriver logging, create a new logging metric with the required filters, edit the application code to set the metric value when needed, and create an alert in Stackdriver based on the new metric. is not right.
You don't need to edit the application code to send the metric values. The application already pushes error logs whenever the application times out. Since you already have the required entries in the Stackdriver logs, you don't need to edit the application code to send the metric values. You just need to create metrics from log data.
Ref: https://cloud.google.com/logging

Create a custom monitoring metric in code, edit the application code to set the metric value when needed, create an alert in Stackdriver based on the new metric. is not right.
You don't create a custom monitoring metric in code. Stackdriver Logging allows you to easily create metrics from log data. Since the application already pushes error logs to Stackdriver Logging, we just need to create metrics from log data in Stackdriver Logging.
Ref: https://cloud.google.com/logging

Add the Stackdriver monitoring and logging agent to the instances running the code. is not right.
The Stackdriver Monitoring agent gathers system and application metrics from your VM instances and sends them to Monitoring. In order to make use of this approach, you need application metrics but our application doesn't generate metrics. It just logs errors whenever the upload times out and these are then ingested to Stackdriver logging. We can update our application to enable custom metrics for these scenarios, but that is a lot more work than creating metrics from log data in Stackdriver Logging
Ref: https://cloud.google.com/logging

In Stackdriver Logging, create a custom monitoring metric from log data and create an alert in Stackdriver based on the new metric. is the right answer.
Our application adds entries to error logs whenever the application times out during image upload and these logs are ingested to Stackdriver Logging. Since we already have the required data in logs, we just need to create metrics from this log data in Stackdriver Logging. And we can then set up an alert based on this metric. We can trigger an alert if the number of occurrences of the relevant error message is greater than a predefined value. Based on the alert, you can manually add more compute resources.
Ref: https://cloud.google.com/logging

.

57. Question

.

You developed an application that lets users upload statistical files and subsequently run analytics on this data. You chose to use Google Cloud Storage and BigQuery respectively for these requirements as they are highly available and scalable. You have a docker image for your application code, and you plan to deploy on your on-premises Kubernetes clusters. Your on-prem Kubernetes cluster needs to connect to Google Cloud Storage and BigQuery and you want to do this in a secure way following Google recommended practices. What should you do?

.

- Create a new service account, with editor permissions, generate and download a key. Use the key to authenticate inside the application.
- Use the default service account for App Engine, which already has the required permissions.
- Use the default service account for Compute Engine, which already has the required permissions.

- **Create a new service account, grant it the least viable privileges to the required services, generate and download a JSON key. Use the JSON key to authenticate inside the application.**

Unattempted

Use the default service account for Compute Engine, which already has the required permissions. is not right.
The Compute Engine default service account is created with the Cloud IAM project editor role
Ref: https://cloud.google.com/compute/docs/access/service-accounts#default_service_account
The project editor role includes all viewer permissions, plus permissions for actions that modify state, such as changing existing resources. Using a service account that is over-privileged falls foul of the principle of least privilege. Google recommends you enforce the principle of least privilege by ensuring that members have only the permissions that they actually need.
Ref: https://cloud.google.com/iam/docs/understanding-roles

Use the default service account for App Engine, which already has the required permissions. is not right.
App Engine default service account has the Editor role in the project (Same as the default service account for Compute Engine).
Ref: https://cloud.google.com/appengine/docs/standard/python/service-account
The project editor role includes all viewer permissions, plus permissions for actions that modify state, such as changing existing resources. Using a service account that is over-privileged falls foul of the principle of least privilege. Google recommends you enforce the principle of least privilege by ensuring that members have only the permissions that they actually need.
Ref: https://cloud.google.com/iam/docs/understanding-roles

Create a new service account, with editor permissions, generate and download a key. Use the key to authenticate inside the application. is not right.
The project editor role includes all viewer permissions, plus permissions for actions that modify state, such as changing existing resources. Using a service account that is over-privileged falls foul of the principle of least privilege. Google recommends you enforce the principle of least privilege by ensuring that members have only the permissions that they actually need.
Ref: https://cloud.google.com/iam/docs/understanding-roles

Create a new service account, grant it the least viable privileges to the required services, generate and download a JSON key. Use the JSON key to authenticate inside the application. is the right answer.

Using a new service account with just the least viable privileges for the required services follows the principle of least privilege. To use a service account outside of Google Cloud, such as on other platforms or on-premises, you must first establish the identity of the service account. Public/private key pairs provide a secure way of accomplishing this goal. Once you have the key, you can use it in your application to authenticate connections to Cloud Storage and BigQuery.
Ref: https://cloud.google.com/iam/docs/creating-managing-service-account-keys#creating_service_account_keys
Ref: https://cloud.google.com/iam/docs/recommender-overview

.

58. Question

.

You developed an application that reads objects from a cloud storage bucket. You followed GCP documentation and created a service account with just the permissions to read objects from the cloud storage bucket. However, when your application uses this service account, it fails to read objects from the bucket. You suspect this might be an issue with the permissions assigned to the service account. You would like to authenticate a gsutil session with the service account credentials, reproduce the issue yourself and identify the root cause. How can you authenticate gsutil with service account credentials?

.

- **Create JSON keys for the service account and execute gcloud auth activate-service-account --key-file [KEY_FILE]**
- Create JSON keys for the service account and execute gcloud auth service-account --key-file [KEY_FILE]
- Create JSON keys for the service account and execute gcloud authenticate service-account --key-file [KEY_FILE]
- Create JSON keys for the service account and execute gcloud authenticate activate-service-account --key-file [KEY_FILE]

Unattempted

Create JSON keys for the service account and execute gcloud authenticate activate-service-account –key-file [KEY_FILE]. is not right.
gcloud doesn't support using "authenticate" to grant/revoke credentials for Cloud SDK. The correct service is "auth".
Ref: https://cloud.google.com/sdk/gcloud/reference/auth

Create JSON keys for the service account and execute gcloud authenticate service-account –key-file [KEY_FILE]. is not right.
gcloud doesn't support using "authenticate" to grant/revoke credentials for Cloud SDK.

The correct service is "auth".
Ref: https://cloud.google.com/sdk/gcloud/reference/auth

Create JSON keys for the service account and execute gcloud auth service-account –key-file [KEY_FILE]. is not right.
gcloud auth does not support service-account action. The correct action to authenticate a service account is activate-service-account.
Ref: https://cloud.google.com/sdk/gcloud/reference/auth/activate-service-account

Create JSON keys for the service account and execute gcloud auth activate-service-account –key-file [KEY_FILE]. is the right answer.
This command correctly authenticates access to Google Cloud Platform with a service account using its JSON key file. To allow gcloud (and other tools in Cloud SDK) to use service account credentials to make requests, use this command to import these credentials from a file that contains a private authorization key, and activate them for use in gcloud
Ref: https://cloud.google.com/sdk/gcloud/reference/auth/activate-service-account

.

59. Question

.

You developed an application to serve production users and you plan to use Cloud SQL to host user state data which is very critical for the application flow. You want to protect your user state data from zone failures. What should you do?

.

- Create a Failover replica in the same region but in a different zone.
- Create a Read replica in the same region but in a different zone.
- **Configure High Availability (HA) for Cloud SQL and Create a Failover replica in the same region but in a different zone.**
- Configure High Availability (HA) for Cloud SQL and Create a Failover replica in a different region

Unattempted

Create a Read replica in the same region but in a different zone. is not right.
Read replicas do not provide failover capability. To provide failover capability, you need to configure Cloud SQL Instance for High Availability.
Ref: https://cloud.google.com/sql/docs/mysql/replication

Create a Read replica in a different region. is not right.
Read replicas do not provide failover capability. To provide failover capability, you need to configure Cloud SQL Instance for High Availability.
Ref: https://cloud.google.com/sql/docs/mysql/replication

Configure High Availability (HA) for Cloud SQL and Create a Failover replica in a different region. is not right.
A Cloud SQL instance configured for HA is called a regional instance because it's primary and secondary instances are in the same region. They are located in different zones but within the same region. It is not possible to create a Failover replica in a different region.
Ref: https://cloud.google.com/sql/docs/mysql/high-availability

Configure High Availability (HA) for Cloud SQL and Create a Failover replica in the same region but in a different zone. is the right answer.
If a HA-configured instance becomes unresponsive, Cloud SQL automatically switches to serving data from the standby instance. The HA configuration provides data redundancy. A Cloud SQL instance configured for HA has instances in the primary zone (Master node) and secondary zone (standby/failover node) within the configured region. Through synchronous replication to each zone's persistent disk, all writes made to the primary instance are also made to the standby instance. If the primary goes down, the standby/failover node takes over and your data continues to be available to client applications.
Ref: https://cloud.google.com/sql/docs/mysql/high-availability

.

60. Question

.

You have 32 GB of data in a single file that you need to upload to a Nearline Storage bucket. The WAN connection you are using is rated at 1 Gbps, and you are the only one on the connection. You want to use as much of the rated 1 Gbps as possible to transfer the file rapidly. How should you upload the file?

.

- Use the GCP Console to transfer the file instead of gsutil.
- Change the storage class of the bucket from Nearline to Multi-Regional.
- **Enable parallel composite uploads using gsutil on the file transfer.**
- Decrease the TCP window size on the machine initiating the transfer.

Unattempted

Requirements – transfer the file rapidly, use as much of the rated 1 Gbps as possible

Use the GCP Console to transfer the file instead of gsutil. is not right.
GCP Console does not offer any specific features that help in improving the upload speed.

Decrease the TCP window size on the machine initiating the transfer. is not right.
By decreasing the TCP window size, you are reducing the chunks of data sent in the TCP window, and this has the effect of underutilizing your bandwidth and can slow down the upload.

Change the storage class of the bucket from Nearline to Multi-Regional. is not right.
Multi-Regional is not a storage class. It is a bucket location. You can transition between storage classes but that does not improve the upload speed.
https://cloud.google.com/storage/docs/locations
https://cloud.google.com/storage/docs/storage-classes

Enable parallel composite uploads using gsutil on the file transfer. is the right answer.
With cloud storage, Object composition can be used for uploading an object in parallel: you can divide your data into multiple chunks, upload each chunk to a distinct object in parallel, compose your final object, and delete any temporary source objects. This helps maximize your bandwidth usage and ensures the file is uploaded as fast as possible.
Ref: https://cloud.google.com/storage/docs/composite-objects#uploads

.

61. Question

.

You have a collection of audio/video files over 80GB each that you need to migrate to Google Cloud Storage. The files are in your on-premises data center. What migration method can you use to help speed up the transfer process?

.

- **Use parallel uploads to break the file into smaller chunks then transfer it simultaneously.**
- Use multithreaded uploads using the -m option.
- Use the Cloud Transfer Service to transfer.
- Start a recursive upload.

Unattempted

Use parallel uploads to break the file into smaller chunks then transfer it simultaneously. is the right answer.
With cloud storage, Object composition can be used for uploading an object in parallel: you can divide your data into multiple chunks, upload each chunk to a distinct object in

parallel, compose your final object, and delete any temporary source objects. This helps maximize your bandwidth usage and ensures the file is uploaded as fast as possible.
Ref: https://cloud.google.com/storage/docs/composite-objects#uploads

Use multithreaded uploads using the -m option. is not right.
Using the -m option lets you upload multiple files at the same time, but in our case, the individual files are over 80GB each. The best upload speed can be achieved by breaking the file into smaller chunks and transferring it simultaneously.

Use the Cloud Transfer Service to transfer. is not right.
Cloud Transfer Service is used for transferring massive amounts (in the range of petabytes of data) of data to the cloud. While nothing stops us from using Cloud Transfer Service to upload our files, it would be an overkill and very expensive.
Ref: https://cloud.google.com/products/data-transfer

Start a recursive upload. is not right.
In Google Cloud Storage, there is no such thing as a recursive upload.

.

62. Question

.

You have a compute engine instance running a production application. You want to receive an email when the instance consumes more than 90% of its CPU resources for more than 15 minutes. You want to use Google services. What should you do?

.

- 1. Create a Stackdriver Workspace and associate your GCP project with it. 2. Write a script that monitors the CPU usage and sends it as a custom metric to Stackdriver 3. Create an uptime check for the instance in Stackdriver.
- 1. Create a consumer Gmail Account 2. Write a script that monitors the CPU usage. 3. When the CPU usage exceeds the threshold, have the script send an email using the Gmail account and smtp.gmail.com on port 25 as SMTP server.
- **1. Create a Stackdriver Workspace and associate your Google Cloud Platform (GCP) project with it 2. Create an Alerting Policy in Stackdriver that uses the threshold as a trigger condition. 3. Configure your email address in the notification channel.**
- 1. In Stackdriver logging, create a logs based metric to extract the CPU usage by using a regular expression. 2. In Stackdriver Monitoring, create an Alerting Policy based on this metric 3. Configure your email address in the notification channel.

Unattempted

We want to use Google services. So that eliminates the two options where we Write a script. Why would we want to write a script when there is a Google service that does exactly that – with minimal configuration!!

Stackdriver logging does not log CPU usage. (Stackdriver monitoring does that) So that rules out the option In Stackdriver logging, create a logs based metric to extract the CPU usage by using a regular expression.
Ref: https://cloud.google.com/logging/

1. Create a Stackdriver Workspace and associate your Google Cloud Platform (GCP) project with it
2. Create an Alerting Policy in Stackdriver that uses the threshold as a trigger condition.
3. Configure your email address in the notification channel.
is the right answer.

A Workspace is a tool for monitoring resources contained in one or more Google Cloud projects or AWS accounts. In our case, we create a Stackdriver workspace and link our project to this workspace.
Ref: https://cloud.google.com/monitoring/workspaces

Stackdriver monitoring captures the CPU usage. By default, the Monitoring agent collects disk, CPU, network, and process metrics. You can also have the agent send custom metrics to Stackdriver monitoring.
Ref: https://cloud.google.com/monitoring/

You can then set up an alerting policy to alert with CPU utilization exceeds 90% for 15 minutes.
Ref: https://cloud.google.com/monitoring/alerts/. See here for an example of setting up an alerting policy on CPU load. In our case, we'd have to substitute the CPU load for the CPU utilization metric. https://cloud.google.com/monitoring/quickstart-lamp

Stack driver monitoring supports multiple notification options for triggering alerts; email is one of them. Ref: https://cloud.google.com/monitoring/support/notification-options

.

63. Question

.

You have a developer laptop with Cloud SDK installed on Ubuntu. The cloud SDK was installed from Google Cloud Ubuntu package repository. You want to test your application locally on your laptop with Cloud Datastore. What should you do?

.

- Create a Cloud Datastore index using gcloud datastore indexes create
- Install the google-cloud-sdk-datastore-emulator component using the apt get install command.
- Export Cloud Datastore data using gcloud datastore export
- **Install the cloud-datastore-emulator component using the gcloud components install command.**

Unattempted

Export Cloud Datastore data using gcloud datastore export is not right.
By all means, you can export a copy of all or a subset of entities from Google Cloud Datastore to another storage system such as Google Cloud Storage but your application is configured to connect to a Cloud Datastore instance, not another system that stores a raw dump of exported data. So this option is not right.

Create a Cloud Datastore index using gcloud datastore indexes create. is not right.
You could create an index but this doesn't help your application emulate connections to Cloud Datastore on your laptop. So this option is not right.

Install the google-cloud-sdk-datastore-emulator component using the apt get install command. is not right.
There is no such thing as google-cloud-sdk-datastore-emulator; and you don't install gcloud components using apt get. So this option is not right.

Install the cloud-datastore-emulator component using the gcloud components install command. is the right answer.
The Datastore emulator provides local emulation of the production Datastore environment. You can use the emulator to develop and test your application locally
Ref: https://cloud.google.com/datastore/docs/tools/datastore-emulator

.

64. Question

.

You have a development project with appropriate IAM roles defined. You are creating a production project and want to have the same IAM roles on the new project, using the fewest possible steps. What should you do?

.

- Use gcloud iam roles copy and specify your organization as the destination organization.
- **Use gcloud iam roles copy and specify the production project as the destination project.**
- In the Google Cloud Platform Console, use the create role from role functionality.
- In the Google Cloud Platform Console, use the create role functionality and select all applicable permissions.

Unattempted

Our requirements are to create the same iam roles in a different (production) project with the fewest possible steps.

In the Google Cloud Platform Console, use the 'create role from role' functionality. is not right.
This creates a role in the same (development) project, not in the production project. So this doesn't meet our requirement to create same iam roles in production project

In the Google Cloud Platform Console, use the 'create role' functionality and select all applicable permissions. is not right.
This creates a role in the same (development) project, not in the production project. So this doesn't meet our requirement to create same iam roles in production project

Use gcloud iam roles copy and specify your organization as the destination organization. is not right.
We can optionally specify a destination organization but since our requirement is to copy the roles into "production project" (i.e. project, not organization), this option does not meet our requirement to create same iam roles in production project
Ref: https://cloud.google.com/sdk/gcloud/reference/iam/roles/copy

Use gcloud iam roles copy and specify the production project as the destination project. is the right answer.
This is the only option that fits our requirements. You copy the roles into the destination project using gcloud iam roles copy and by specifying the production project destination project.
$gcloud iam roles copy –source "<>" –destination <> –dest-project <>
Ref: https://cloud.google.com/sdk/gcloud/reference/iam/roles/copy

.

65. Question

.

You have a Dockerfile that you need to deploy on Kubernetes Engine. What should you do?

.

- Use kubectl app deploy .
- **Create a docker image from the Dockerfile and upload it to Container Registry. Create a Deployment YAML file to point to that image. Use kubectl to create the deployment with that file.**
- Use gcloud app deploy .
- Create a docker image from the Dockerfile and upload it to Cloud Storage. Create a Deployment YAML file to point to that image. Use kubectl to create the deployment with that file.

Unattempted

Use kubectl app deploy . is not right.
kubectl does not accept app as a verb. Kubectl can deploy a configuration file using kubectl deploy.
Ref: https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#apply

Use gcloud app deploy . is not right.
gcloud app deploy – Deploys the local code and/or configuration of your app to App Engine. gcloud app deploy accepts a flag –image-url which is the docker image but it can't directly use a docker file.
Ref: https://cloud.google.com/sdk/gcloud/reference/app/deploy

Create a docker image from the Dockerfile and upload it to Cloud Storage. Create a Deployment YAML file to point to that image. Use kubectl to create the deployment with that file. is not right.
You can not upload a docker image to cloud storage. They can only be pushed to a Container Registry (e.g. GCR, Dockerhub etc.)
Ref: https://cloud.google.com/container-registry/docs/pushing-and-pulling

Create a docker image from the Dockerfile and upload it to Container Registry. Create a Deployment YAML file to point to that image. Use kubectl to create the deployment with that file. is the right answer.
Once you have a docker image, you can push it to the container register. You can then create a deployment YAML file pointing to this image and use kubectl apply -f to deploy this to the Kubernetes cluster. This assumes you already have a Kubernetes cluster and you gcloud environment is set up to talk to this container by executing gcloud container

clusters get-credentials –zone=
Ref: https://cloud.google.com/container-registry/docs/pushing-and-pulling
Ref: https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#apply
Ref: https://cloud.google.com/sdk/gcloud/reference/container/clusters/get-credentials

.

66. Question

.

You have a Google Cloud Platform account with access to both production and development projects. You need to create an automated process to list all compute instances in development and production projects on a daily basis. What should you do?

.

- **Create two configurations using gcloud config. Write a script that sets configurations as active, individually. For each configuration, use gcloud compute instances list to get a list of compute resources.**
- Create two configurations using gsutil config. Write a script that sets configurations as active, individually. For each configuration, use gsutil compute instances list to get a list of compute resources.
- Go to Cloud Shell and export this information to Cloud Storage on a daily basis.
- Go to GCP Console and export this information to Cloud SQL on a daily basis.

Unattempted

Go to Cloud Shell and export this information to Cloud Storage on a daily basis. is not right.
You want an automated process but this is a manual activity that needs to be executed daily.

Go to GCP Console and export this information to Cloud SQL on a daily basis. is not right.
You want an automated process but this is a manual activity that needs to be executed daily.

Create two configurations using gsutil config. Write a script that sets configurations as active, individually. For each configuration, use gsutil compute instances list to get a list of compute resources. is not right.
The gsutil config command applies to users who have installed gsutil as a standalone tool and is used for obtaining access credentials for Cloud Storage and writes a boto/gsutil configuration file containing the obtained credentials along with a number of other

configuration-controllable values.
Ref: https://cloud.google.com/storage/docs/gsutil/commands/config
It is not used for creating Gcloud configurations. You use gcloud config to do that.
https://cloud.google.com/sdk/gcloud/reference/config/configurations/create

Create two configurations using gcloud config. Write a script that sets configurations as active, individually. For each configuration, use gcloud compute instances list to get a list of compute resources. is the right answer.
You can create two configurations – one for the development project and another for the production project. And you do that by running "gcloud config configurations create" command.
https://cloud.google.com/sdk/gcloud/reference/config/configurations/create
In your custom script, you can load these configurations one at a time and execute gcloud compute instances list to list Google Compute Engine instances in the project that is active in the gcloud configuration.
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/instances/list
Once you have this information, you can export it in a suitable format to a suitable target e.g. export as CSV or export to Cloud Storage/BigQuery/SQL, etc.

.

67. Question

.

You have a large 5-TB AVRO file stored in a Cloud Storage bucket. Your analysts are proficient only in SQL and need access to the data stored in this file. You want to find a cost-effective way to complete their request as soon as possible. What should you do?

.

- Load data in Cloud Datastore and run a SQL query against it.
- Create a BigQuery table and load data in BigQuery. Run a SQL query on this table and drop this table after you complete your request.
- **Create external tables in BigQuery that point to Cloud Storage buckets and run a SQL query on these external tables to complete your request.**
- Create a Hadoop cluster and copy the AVRO file to NDFS by compressing it. Load the file in a hive table and provide access to your analysts so that they can run SQL queries.

Unattempted

Load data in Cloud Datastore and run a SQL query against it. is not right.
Datastore is a highly scalable NoSQL database and although it supports SQL like queries, it doesn't support SQL. Moreover, there is no out of the box way for transforming AVRO file from cloud storage into the Cloud Datastore entity. So we have to do in a bespoke way

which adds to our cost and time.
Ref: https://cloud.google.com/datastore

Create a Hadoop cluster and copy the AVRO file to NDFS by compressing it. Load the file in a hive table and provide access to your analysts so that they can run SQL queries. is not right.
Like Cloud Datastore, Hive doesn't directly support SQL, it provides HiveQL (HQL) which is SQL like. In addition, the process of creating a Hadoop cluster and getting the data eventually into a hive table is time-consuming and adds to our cost and time.

Create a BigQuery table and load data in BigQuery. Run a SQL query on this table and drop this table after you complete your request. is not right.
Like the above two, while it is possible to build a solution that transforms and loads data into the target, BigQuery in this case, is not a trivial process and involves cost and time. GCP provides an out of the box way to query AVRO files from Cloud Storage and this should be preferred.

Create external tables in BigQuery that point to Cloud Storage buckets and run a SQL query on these external tables to complete your request. is the right answer.
BigQuery supports querying Cloud Storage data in a number of formats such as CSV, JSON, AVRO, etc. You do this by creating a Big Query external table that points to a Cloud Storage data source (bucket). This solution works out of the box, involves minimal effort, minimal cost, and is quick.
https://cloud.google.com/bigquery/external-data-cloud-storage

- **68. Question**

- You have a Linux VM that must connect to Cloud SQL. You created a service account with the appropriate access rights. You want to make sure that the VM uses this service account instead of the default Compute Engine service account. What should you do?

-
- Download a JSON Private Key for the service account. On the Custom Metadata of the VM, add that JSON as the value for the key compute-engine-service-account
- Download a JSON Private Key for the service account. After creating the VM, ssh into the VM and save the JSON under ~/.gcloud/compute-engine-service-account.json

- **When creating the VM via the web console, specify the service account under the 'Identity and API Access' section.**
- Download a JSON Private Key for the service account. On the Project Metadata, add that JSON as the value for the key compute-engine-serviceaccount

Unattempted

When creating the VM via the web console, specify the service account under the 'Identity and API Access' section. is the right answer.
You can set the service account at the time of creating the compute instance. You can also update the service account used by the instance – this requires that you stop the instance first and then update the service account. Setting/Updating the service account can be done either via the web console or by executing gcloud command or by the REST API. See below an example for updating the service account through gcloud command.
gcloud compute instances set-service-account instance-1 –zone=us-central1-a –service-account=my-new-service-account@gcloud-gcp-ace-lab-266520.iam.gserviceaccount.com
Updated [https://www.googleapis.com/compute/v1/projects/gcloud-gcp-ace-lab-266520/zones/us-central1-a/instances/instance-1].

Download a JSON Private Key for the service account. On the Project Metadata, add that JSON as the value for the key compute-engine-serviceaccount is not right.
While updating the service account for a compute instance can be done through the console, gcloud or the REST API, they don't do it based on the JSON Private Key.

Download a JSON Private Key for the service account. On the Custom Metadata of the VM, add that JSON as the value for the key compute-engine-service-account. is not right.
While updating the service account for a compute instance can be done through the console, gcloud or the REST API, they don't do it based on the JSON Private Key.

Download a JSON Private Key for the service account. After creating the VM, ssh into the VM and save the JSON under ~/.gcloud/compute-engine-service-account.json is not right.
You can configure a VM to use a certain service account by providing the relevant JSON credentials file, but the procedure is different. Copying the JSON file to a specific path alone is not sufficient, moreover, the path mentioned is wrong as well. See below for a use case where a VM which is unable to list cloud storage buckets is updated to use a service account and it can then list the buckets.
Prior to using a service account. Use gsutil ls to list buckets and it fails.
$ gsutil ls
ServiceException: 401 Anonymous caller does not have storage.buckets.list access to project 393066724129.
Within the VM, execute the command below to use the service account. (Assumes that you have created a service account that provides the necessary permissions and have copied it over the VM)

gcloud auth activate-service-account admin-service-account@gcloud-gcp-ace-266520.iam.gserviceaccount.com –key-file=~/compute-engine-service-account.json
Activated service account credentials for: [admin-service-account@gcloud-gcp-ace-266520.iam.gserviceaccount.com]
The output above doesn't show this, but the credentials are written to the file /home/gcloud_gcp_ace_user/.config/gcloud/legacy_credentials/admin-service-account@gcloud-gcp-ace-266520.iam.gserviceaccount.com/adc.json
Now, use gsutil ls again to list buckets and it works.
$ gsutil ls
gs://test-gcloud-gcp-ace-2020-bucket-1/
gs://test-gcloud-gcp-ace-2020-bucket-2/

.

69. Question

.

You have a number of applications that have bursty workloads and are heavily dependent on topics to decouple publishing systems from consuming systems. Your company would like to go serverless to enable developers to focus on writing code without worrying about infrastructure. Your solution architect has already identified Cloud Pub/Sub as a suitable alternative for decoupling systems. You have been asked to identify a suitable GCP Serverless service that is easy to use with Cloud Pub/Sub. You want the ability to scale down to zero when there is no traffic in order to minimize costs. You want to follow Google recommended practices. What should you suggest?

.

- Cloud Run for Anthos
- **Cloud Functions**
- App Engine Standard
- Cloud Run

Unattempted

GCP serverless compute portfolio includes 4 services, which are all listed in the answer options. Our requirements are to identify a GCP serverless service that
1. Lets us scale down to 0
2. Integrates with Cloud Pub/Sub seamlessly

Cloud Run for Anthos. is not right.
Among the four options, App Engine Standard, Cloud Functions and Cloud Run can all scale down to zero. Cloud Run for Anthos can scale the pods down the zero but the number of nodes per cluster can not scale to zero so these nodes are billed in the absence of requests. This rules out Cloud Run for Anthos.

App Engine Standard. is not right.
App Engine Standard doesn't offer an out of the box integration with Cloud Pub/Sub. We can use the Cloud Client Library to send and receive Pub/Sub messages as described in the note below but the key point to note is the absence of out of the box integration with Cloud Pub/Sub so this rules out App Engine Standard
Ref: https://cloud.google.com/appengine/docs/standard/nodejs/writing-and-responding-to-pub-sub-messages

Cloud Run. is not right.
Cloud Run is an excellent product and integrates with Cloud Pub/Sub for several use cases. For example, every time a new .csv file is created inside a Cloud Storage bucket, an event is fired and delivered via a Pub/Sub subscription to a Cloud Run service. The Cloud Run service extracts data from the file and stores it as structured data into a BigQuery table.
Ref: https://cloud.google.com/run#section-7
At the same time, we want to follow Google recommended practices. Google doesn't list integration with Cloud Pub/Sub as a key feature of Cloud Run. Contrary to this, Google says "If you're building a simple API (a small set of functions to be accessed via HTTP or Cloud Pub/Sub), we recommend using Cloud Functions."

Cloud Functions. is the right answer.
Cloud Functions is Google Cloud's event-driven serverless compute platform that lets you run your code locally or in the cloud without having to provision servers. Cloud Functions scales up or down, so you pay only for compute resources you use. Cloud Functions have excellent integration with Cloud Pub/Sub, lets you scale down to zero and is recommended by Google as the ideal serverless platform to use when dependent on Cloud Pub/Sub.
"If you're building a simple API (a small set of functions to be accessed via HTTP or Cloud Pub/Sub), we recommend using Cloud Functions."
Ref: https://cloud.google.com/serverless-options

- **70. Question**

- You have a number of compute instances belonging to an unmanaged instances group. You need to SSH to one of the Compute Engine instances to run an ad hoc script. You've already authenticated gcloud, however, you don't have an SSH key deployed yet. In the fewest steps possible, what's the easiest way to SSH to the instance?

-

- Create a key with the ssh-keygen command. Upload the key to the instance. Run gcloud compute instances list to get the IP address of the instance, then use the ssh command.
- Run gcloud compute instances list to get the IP address of the instance, then use the ssh command.
- Create a key with the ssh-keygen command. Then use the gcloud compute ssh command.
- **Use the gcloud compute ssh command.**

Unattempted

Create a key with the ssh-keygen command. Upload the key to the instance. Run gcloud compute instances list to get the IP address of the instance, then use the ssh command. is not right.
This approach certainly works. You can create a key pair with ssh-keygen, update the instance metadata with the public key and SSH to the instance. But is it the easiest way to SSH to the instance with the fewest possible steps? Let's explore other options to decide (you will see that there is another option that does the same with less effort). You can find more information about this option here: https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#block-project-keys

Create a key with the ssh-keygen command. Then use the gcloud compute ssh command. is not right.
This works but is more work (having to create the key) than the answer. gcloud compute ssh ensures that the user's public SSH key is present in the project's metadata. If the user does not have a public SSH key, one is generated using ssh-keygen and added to the project's metadata.

Run gcloud compute instances list to get the IP address of the instance, then use the ssh command. is not right.
We can get the IP of the instance by executing the gcloud compute instances list but unless an SSH is generated and updated in project metadata, you would not be able to SSH to the instance. User access to a Linux instance through third-party tools is determined by which public SSH keys are available to the instance. You can control the public SSH keys that are available to a Linux instance by editing metadata, which is where your public SSH keys and related information are stored.
Ref: https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#block-project-keys

Use the gcloud compute ssh command. is the right answer.
gcloud compute ssh ensures that the user's public SSH key is present in the project's metadata. If the user does not have a public SSH key, one is generated using ssh-keygen

and added to the project's metadata. This is similar to the other option where we copy the key explicitly to the project's metadata but here it is done automatically for us. There are also security benefits with this approach. When we use gcloud compute ssh to connect to Linux instances, we are adding a layer of security by storing your host keys as guest attributes. Storing SSH host keys as guest attributes improve the security of your connections by helping to protect against vulnerabilities such as man-in-the-middle (MITM) attacks. On the initial boot of a VM instance, if guest attributes are enabled, Compute Engine stores your generated host keys as guest attributes. Compute Engine then uses these host keys that were stored during the initial boot to verify all subsequent connections to the VM instance.
Ref: https://cloud.google.com/compute/docs/instances/connecting-to-instance
Ref: https://cloud.google.com/sdk/gcloud/reference/compute/ssh