Associate Cloud Engineer Syllabus

Section 1: Setting up a cloud solution environment (~17.5% of the exam)

1.1 Setting up cloud projects and accounts. Activities include:

● Creating a resource hierarchy

● Applying organizational policies to the resource hierarchy

● Granting members IAM roles within a project

● Managing users and groups in Cloud Identity (manually and automated)

● Enabling APIs within projects

● Provisioning and setting up products in Google Cloud’s operations suite

1.2 Managing billing configuration. Activities include:

● Creating one or more billing accounts

● Linking projects to a billing account

● Establishing billing budgets and alerts

● Setting up billing exports

1.3 Installing and configuring the command line interface (CLI), specifically the Cloud SDK (e.g., setting the default project).

Section 2: Planning and configuring a cloud solution (~17.5% of the exam)

2.1 Planning and estimating Google Cloud product use using the Pricing Calculator

2.2 Planning and configuring compute resources. Considerations include:

● Selecting appropriate compute choices for a given workload (e.g., Compute Engine, Google Kubernetes Engine, Cloud Run, Cloud Functions)

● Using preemptible VMs and custom machine types as appropriate

2.3 Planning and configuring data storage options. Considerations include:

● Product choice (e.g., Cloud SQL, BigQuery, Firestore, Spanner, Bigtable)

2 / 16

● Choosing storage options (e.g., Zonal persistent disk, Regional balanced persistent disk, Standard, Nearline, Coldline, Archive)

2.4 Planning and configuring network resources. Tasks include:

● Differentiating load balancing options

● Identifying resource locations in a network for availability

● Configuring Cloud DNS

Section 3: Deploying and implementing a cloud solution (~25% of the exam)

3.1 Deploying and implementing Compute Engine resources. Tasks include:

● Launching a compute instance using the Google Cloud console and Cloud SDK (gcloud) (e.g., assign disks, availability policy, SSH keys)

● Creating an autoscaled managed instance group using an instance template

● Generating/uploading a custom SSH key for instances

● Installing and configuring the Cloud Monitoring and Logging Agent

● Assessing compute quotas and requesting increases

3.2 Deploying and implementing Google Kubernetes Engine resources. Tasks include:

● Installing and configuring the command line interface (CLI) for Kubernetes (kubectl)

● Deploying a Google Kubernetes Engine cluster with different configurations including AutoPilot, regional clusters, private clusters, etc.

● Deploying a containerized application to Google Kubernetes Engine

● Configuring Google Kubernetes Engine monitoring and logging

3.3 Deploying and implementing Cloud Run and Cloud Functions resources. Tasks include, where applicable:

● Deploying an application and updating scaling configuration, versions, and traffic splitting

● Deploying an application that receives Google Cloud events (e.g., Pub/Sub events, Cloud Storage object change notification events)

3.4 Deploying and implementing data solutions. Tasks include:

3 / 16

● Initializing data systems with products (e.g., Cloud SQL, Firestore, BigQuery, Spanner, Pub/Sub, Bigtable, Dataproc, Dataflow, Cloud Storage)

● Loading data (e.g., command line upload, API transfer, import/export, load data from Cloud Storage, streaming data to Pub/Sub)

3.5 Deploying and implementing networking resources. Tasks include:

● Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)

● Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)

● Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)

● Creating a VPN between a Google VPC and an external network using Cloud VPN

● Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

3.6 Deploying a solution using Cloud Marketplace. Tasks include:

● Browsing the Cloud Marketplace catalog and viewing solution details

● Deploying a Cloud Marketplace solution

3.7 Implementing resources via infrastructure as code. Tasks include:

● Building infrastructure via Cloud Foundation Toolkit templates and implementing best practices

● Installing and configuring Config Connector in Google Kubernetes Engine to create, update, delete, and secure resources

Section 4: Ensuring successful operation of a cloud solution (~20% of the exam)

4.1 Managing Compute Engine resources. Tasks include:

● Managing a single VM instance (e.g., start, stop, edit configuration, or delete an instance)

● Remotely connecting to the instance

● Attaching a GPU to a new instance and installing necessary dependencies

● Viewing current running VM inventory (instance IDs, details)

4 / 16

● Working with snapshots (e.g., create a snapshot from a VM, view snapshots, delete a snapshot)

● Working with images (e.g., create an image from a VM or a snapshot, view images, delete an image)

● Working with instance groups (e.g., set autoscaling parameters, assign instance template, create an instance template, remove instance group)

● Working with management interfaces (e.g., Google Cloud console, Cloud Shell, Cloud SDK)

4.2 Managing Google Kubernetes Engine resources. Tasks include:

● Viewing current running cluster inventory (nodes, pods, services)

● Browsing Docker images and viewing their details in the Artifact Registry

● Working with node pools (e.g., add, edit, or remove a node pool)

● Working with pods (e.g., add, edit, or remove pods)

● Working with services (e.g., add, edit, or remove a service)

● Working with stateful applications (e.g. persistent volumes, stateful sets)

● Managing Horizontal and Vertical autoscaling configurations

● Working with management interfaces (e.g., Google Cloud console, Cloud Shell, Cloud SDK, kubectl)

4.3 Managing Cloud Run resources. Tasks include:

● Adjusting application traffic-splitting parameters

● Setting scaling parameters for autoscaling instances

● Determining whether to run Cloud Run (fully managed) or Cloud Run for Anthos

4.4 Managing storage and database solutions. Tasks include:

● Managing and securing objects in and between Cloud Storage buckets

● Setting object life cycle management policies for Cloud Storage buckets

● Executing queries to retrieve data from data instances (e.g., Cloud SQL, BigQuery, Spanner, Datastore, Bigtable)

● Estimating costs of data storage resources

5 / 16

● Backing up and restoring database instances (e.g., Cloud SQL, Datastore)

● Reviewing job status in Dataproc, Dataflow, or BigQuery

4.5 Managing networking resources. Tasks include:

● Adding a subnet to an existing VPC

● Expanding a subnet to have more IP addresses

● Reserving static external or internal IP addresses

● Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

4.6 Monitoring and logging. Tasks include:

● Creating Cloud Monitoring alerts based on resource metrics

● Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)

● Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)

● Configuring log routers

● Viewing and filtering logs in Cloud Logging

● Viewing specific log message details in Cloud Logging

● Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)

● Viewing Google Cloud status

Section 5: Configuring access and security (~20% of the exam)

5.1 Managing Identity and Access Management (IAM). Tasks include:

● Viewing IAM policies

● Creating IAM policies

● Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

5.2 Managing service accounts. Tasks include:

● Creating service accounts

6 / 16

● Using service accounts in IAM policies with minimum permissions

● Assigning service accounts to resources

● Managing IAM of a service account

● Managing service account impersonation

● Creating and managing short-lived service account credentials

5.3 Viewing audit logs

Viewing page 11 out of 54 pages.

Viewing questions 51-55 out of 266 questions

You need to support point-in-time recovery ---- **enable binary logging option**

configure autohealing for network load balancing for a group of Compute Engine instances that run in multiple zones ---- Autohealing health check to healthy (HTTP)

You want to review the configured Kubernetes Engine cluster of an inactive configuration ---------- **kubectl config use-context and kubectl config view**

Cloud Storage to store application backup files for disaster recovery purposes **------- Coldline Storage/Archive Storage**

centralize all these projects under a single, new billing account **------ create a new billing account and set up a payment method**

application has licensing server on the IP 10.0.3.21. Need to deploy the licensing server on Compute Engine without changing the configuration of the application and application must reach the licensing server **------ Reserve the IP 10.0.3.21 as a static internal IP address using gcloud and assign it to the licensing server.**

You are deploying an application to App Engine. You need at least 3 unoccupied instances at all times.

----------- **Automatic Scaling with min\_idle\_instances set to 3**

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 ------ **Use gcloud iam roles copy and specify the production project as the destination project**

You need a dynamic way of provisioning VMs on Compute Engine ---- **Deployment Manager**

Managed Instance Group (option C) and Unmanaged Instance Group (option D) are Compute Engine features that allow you to group related VM instances and manage them as a single entity. However, they do not provide a dynamic way of provisioning VMs based on a configuration file like Deployment Manager does.

Deployment Manager is a configuration management tool that allows you to define and deploy a set of resources, including Compute Engine VMs, in a declarative manner. You can use it to specify the exact

specifications of your VMs in a configuration file, and Deployment Manager will create and manage those VMs for you.

Dockerfile need to deploy on Kubernetes Engine ----- **Create a docker image from the Dockerfile > upload it to Container Registry > Create a Deployment YAML file to point to that image > Use kubectl to create the deployment with that file.**

New Jenkins server for my project. how to deploy in fewest steps possible **------ Use GCP Marketplace to launch the Jenkins solution.**

By using GCP Marketplace to launch the Jenkins solution, you can quickly deploy a Jenkins server with minimal steps.Option B involves creating a new Compute Engine instance and manually installing Jenkins, which also requires more steps.Option C involves creating a Kubernetes cluster and creating a deployment with the Jenkins Docker image, which again involves more steps

Command to update a deployment in Deployment Manager without any resource downtime in the deployment **----- gcloud deployment-manager deployments update --config <deplnt-config-path>**

You want to find out how much it will cost to run the query in BigQuery using on-demand pricing -----**Use the command line to run a dry run query to estimate the number of bytes read. Then convert that bytes estimate to dollars using the Pricing Calculator.**

On-demand pricing for BigQuery : -

Under on-demand pricing, BigQuery charges for queries by using one metric: the number of bytes processed (also referred to as bytes read). You are charged for the number of bytes processed whether the data is stored in BigQuery or in an external data source such as Cloud Storage, Drive, or Cloud Bigtable. On-demand pricing is based solely on usage.

You aren't charged for queries that return an error or for queries that retrieve results from the cache

Canceling a running query job might incur charges up to the full cost. When you run a query, you're charged according to the data processed in the columns you select, even if you set an explicit LIMIT on the results. Partitioning and clustering your tables can help reduce the amount of data processed by queries. As a best practice, use partitioning and clustering whenever possible.

Want my application to run on Google Cloud Platform. That must automatically scale the application based on underlying infrastructure CPU usage using VM directly. **----- Create an instance template, and use the template in a managed instance group with autoscaling configured.**

(Use Elimination Techniques)

You are analyzing Google Cloud Platform service costs from three separate projects. You want to use this information to create service cost estimates by service type, daily and monthly **----- Export your bill to a BigQuery dataset, and then write time window-based SQL queries for analysis**

Key requirements-

1. Analyzing Google Cloud Platform service costs from three separate projects.

2. Using standard query syntax. -> (Relational data and SQL)

Cloud Bigtable,Google Sheets are not feasible for SQL Query.

Cloud billing data can only be exported to a JSON local file and to Bigquery. So, using Cloud Storage to export cloud billing data is not possible to do. BigQuery is feasible for SQL Query

Videos stored in a specific Cloud Storage Regional bucket are moved to Coldline after 90 days, and then deleted after one year from their creation ---- **Use Cloud Storage Object Lifecycle Management using Age conditions with SetStorageClass and Delete actions. Set the SetStorageClass action to 90 days and the Delete action to 365 days.**

There should be no reason to recalculate the time needed to delete after a year.

You can change the storage class of an existing object either by rewriting the object or by using Object Lifecycle Management...Since Object Life Cycle Management was used there was no need to recalculate the expiration date and delete action still remains 365 days.

Cloud Storage Object Lifecycle Management is a feature that allows you to automatically transition objects to different storage classes or delete them based on user-defined rules.

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. ------ **When creating the VM via the web console, specify the service account under the 'Identity and API Access' section.**

you can change access scopes to grant access to a new API, or change an instance so that it runs as a service account that you created, instead of the Compute Engine default service account. However, Google recommends that you use the fine-grained IAM policies instead of relying on access scopes to control resource access for the service account.

To change an instance's service account and access scopes, the instance must be temporarily stopped ... After changing the service account or access scopes, remember to restart the instance." So we can stop the instance, change the service account, then start it up again.

You created an instance of SQL Server 2017 on Compute Engine. You want to connect to this instance in fewest steps **------ Install a RDP client in your desktop. Set a Windows username and password in the GCP Console. Use the credentials to log in to the instance.**

we can't connect using RDP directly in the GCP console. When we click on it, it asks us to install RDP client. There is no option to enter credentials or get an RDP session through the web interface.

You have one GCP account running in your default region and zone and another account running in a non-default region and zone. Want new Compute Engine instance in these two GCP accounts using the CLI ----- Create two configurations using 'gcloud config configurations create [NAME]'. Run 'gcloud config configurations activate [NAME]' to switch between accounts when running the commands to start the Compute Engine instances.

'list' mentioned in all other options does not help create instances.

Demands: Create, Swithch and Start GCP Accounts.

You significantly changed a complex Deployment Manager template and want to confirm that the dependencies of all defined resources are properly met before committing it to the project. You want the most rapid feedback on your changes. ---- **Cloud Pub/Sub, Cloud Dataflow, Cloud Bigtable, BigQuery**

The correct process for building a pipeline to process time-series data. Here's how each of the components is used:

1. Cloud Pub/Sub: receives and distributes time-series data from different sources.

2. Cloud Dataflow: processes the data.

3. Cloud Bigtable: stores and manages the processed data as a NoSQL database.

4. BigQuery: provides a SQL-like interface to analyze the data and extract insights.

I have a project for App Engine application as development environment. The required testing is completed and now want to create new project as production environment. **------- Use gcloud to create the new project, and then deploy your application to the new project.**

B Not Good bcz Requires you to copy the deployed application to the new project. This can be time-consuming and error-prone.

Option C is not as good, because it requires you to create a Deployment Manager configuration file. This can be complex and time-consuming.

need to configure IAM access audit logging in BigQuery for external auditors ---- **Add the auditors group to the 'logging.viewer' and 'bigQuery.dataViewer' predefined IAM roles.**

Creating two new custom IAM roles can be complex and time-consuming.

Access must be on the minimum resources and with the limited access.

AS per Google best practices the roles should be assigned to a group & not to individual users.

need to set up permissions for a set of Compute Engine instances to enable write data into a particular Cloud Storage bucket. ----- **Create a service account and add it to the IAM role 'storage.objectCreator' for that bucket.**

storage.objectCreator contains sufficient privileges to do the job & so admin is not required

A is incorrect as there is no role as write\_only

Always be aware of the Full Control Access.

You are the project owner of a GCP project and want to delegate control to colleagues to manage buckets and files in Cloud Storage **----- Assign Role Storage Admin**

why not storage object admin? Because the objet admin don't have control over buckets and you need it. Question states "Buckets and Objects".

Storage Admin (roles/storage.admin) Grants full control of buckets and objects.

This role allows users to create, manage, and delete buckets and files in Cloud Storage. It also allows users to set permissions on buckets and files.

You have an object in a Cloud Storage bucket that you want to share with an external company. The object contains sensitive data. You want access to the content to be removed after four hours. The external company does not have a Google account to which you can grant specific user-based access privileges. You want to use the most secure method that requires the fewest steps. ------ Create a signed URL with a four-hour expiration and share the URL with the company.

Creating a signed URL with a short expiration time is a secure way to share objects in a Cloud Storage bucket with an external party, especially when the external company does not have a Google account.

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. ----- **Deploy the monitoring pod in a DaemonSet object.**

A DaemonSet ensures that all (or some) nodes in a cluster run a copy of a specific Pod. By deploying the monitoring pod in a DaemonSet object, a copy of the pod will run on each node of the cluster. This ensures that the metrics for all containers running on each node will be sent to the third-party monitoring solution.

A. StatefulSet is used for stateful applications that require unique network identifiers, stable storage, and ordered deployment and scaling.

C. A Deployment object is used to manage a set of replica Pods in a declarative way.

D. Cluster initializers are deprecated and no longer recommended for use in Kubernetes.

There is no automatic enablement of APIs when a service account accesses them. The API needs to be enabled manually in the API Library or through the command-line interface. APIs must be enabled in the project before Deployment Manager can use them. So DM can use Pub/Sub only when Pub/Sub API is enabled.

32 GB single file need to upload to a Nearline Storage bucket. The WAN connection rated at 1 Gbps. You want to transfer the file rapidly ----- **Enable parallel composite uploads using gsutil on the file transfer.**

The most efficient way to upload the large file to Nearline Storage bucket using a single WAN connection rated at 1 Gbps is to enable parallel composite uploads using gsutil. By default, gsutil uses a single thread to upload a single object. But with parallel composite uploads, gsutil will split the file into smaller chunks and upload these chunks in parallel using multiple threads. This will allow the file to be uploaded faster and more efficiently.

N.B.: - Comparison is always wrong. (Instead of, this but not in this)

Database password must not stored in plain text. Where should we store and fetch from ------ tore the database password inside a Secret object

The autoscaling policy is configured so that additional instances are added to the group if the CPU utilization of instances goes above 80%. The initial delay for HTTP health checks against the instances is set to 30 seconds. You observe that when the instance group autoscales, it adds more instances then necessary to support the levels of end-user traffic ---- **Increase the initial delay of the HTTP health check to 200 seconds.**

When autoscaling is enabled, new instances are added based on a metric or metrics (such as CPU utilization) when certain thresholds are met. When adding new instances, it is important to ensure that only the necessary number of instances are added to the instance group and that the group size is properly maintained to prevent overprovisioning and unnecessary costs.

In this scenario, the instance group is adding more instances than necessary when autoscaling due to the initial delay of HTTP health checks. Increasing the initial delay to 200 seconds will ensure that the health check properly reflects the actual availability of the instances and prevent overprovisioning.

need to configure compute resources for a set of batch processing jobs. These jobs take around 2 hours to complete and are run nightly. You want to minimize service costs. ----- **Select Compute Engine. Use preemptible VM instances of the appropriate standard machine type.**

Preemptible VM instances offer the lowest cost for batch processing jobs in the Google Cloud Platform. Preemptible VM instances are computed instances that can run for a maximum of 24 hours and provide no availability guarantees. Preemptible VM instances are up to 80% cheaper than standard compute instances, making them an excellent choice for batch-processing workloads that can be interrupted.

---If some of those instances stop during processing, the job slows but does not completely stop. Preemptible instances complete your batch processing tasks without placing additional workload on your existing instances and without requiring you to pay full price for additional normal instances.

You recently deployed a new version of an application to App Engine and then discovered a bug in the release. You need to immediately revert to the prior version of the application. What should you do? ----- **Go to Cloud Shell and run gcloud config list to review the Google Cloud configuration used for deployment.**

By running gcloud config list in Cloud Shell, you can view the current configuration settings, including the project ID, region, and other relevant settings used for deployment.

Options A and B involve checking the configuration files for the application (app.yaml and web-application.xml), but they may not directly provide information about where the application deployed or why it didn't deploy to the intended project.

You want to configure 10 Compute Engine instances for availability when maintenance occurs. Your requirements state that these instances should attempt to automatically restart if they crash. Also, the instances should be highly available including during system maintenance. ------ **Create an instance template for the instances. Set the 'Automatic Restart' to on. Set the 'On-host maintenance' to Migrate VM instance. Add the instance template to an instance group.**

To ensure the availability of instances during maintenance and restart instances in the event of a crash, you should create an instance group with an instance template that specifies 'Automatic Restart' to on. This will allow your instance to restart in the event of a crash.

Additionally, you should set the 'On-host maintenance' to Migrate VM instance so that VM instances are live migrated to another host in the event of an infrastructure maintenance event.

The instance group should have a health check configured to verify that the instances are healthy. By using an instance group, you can also take advantage of the autoscaling and load-balancing capabilities that come with instance groups.

wWnt the clicked PDF files to be displayed within the browser window directly, without prompting the user to save the file locally ----- **Set Content-Type metadata to application/pdf on the PDF file objects.**

ANSWER A, enabling Cloud CDN on the website frontend, is not relevant to displaying PDF files in the browser. Cloud CDN is a content delivery network that caches content at edge locations around the world to reduce latency and improve website performance.

---- Steps : - In the Google Cloud Console, navigate to the Cloud Storage section and select the "Buckets" page. Select the bucket that contains the static website and the PDF files.

From the "Actions" menu, select "Edit bucket" and then go to the "Website" tab.

In the "Website Configuration" section, select the "Serve objects with this content type" option and enter "application/pdf" in the text field. This will cause PDF files to be served with the correct content type.

Save the changes to the bucket configuration.

After completing these steps, the PDF files on your website will be served with the correct content type and will be displayed directly within the browser window when clicked, without prompting the user to save the file locally.

Machine configured with 2 vCPUs and 4 GB of memory. It is running out of memory. You want to upgrade the virtual machine to have 8 GB of memory. ----- **Stop the VM, increase the memory to 8 GB, and start the VM.**

Production VMs need to be in a different subnet than the test VMs. All the

VMs must be able to reach each other over Internal IP without creating additional routes. You need to set up VPC and the 2 subnets. ------ **Create a single custom VPC with 2 subnets. Create each subnet in a different region and with a different CIDR range.**

ANSWER A meets the requirement because it creates a single custom VPC with 2 subnets, with each subnet in a different region and with a different CIDR range. This ensures that the production and test VMs are in separate subnets and that they can communicate with each other over Internal IP without creating additional routes. Since the subnets are in different regions, they will also have different internal routing tables, which can help isolate the traffic between the two subnets. This configuration provides the necessary network isolation and connectivity required by the production and test workloads.

You have single VPC and a single Subnet in the us-central1 region. You need to deploy a new instance in the same project in the europe-west1 region. ----- **Create a subnetwork in the same VPC, in europe-west1. 2. Create the new instance in the new subnetwork and use the first instance's private address as the endpoint.**

VPC peering only works between VPCs in the same region, so it would not be possible to peer the existing VPC in us-central1 with a new VPC in europe-west1.

You have a website hosted on App Engine standard environment. You want 1% of your users to see a new test version of the website. ------ **Deploy the new version in the same application and use the --splits option to give a weight of 99 to the current version and a weight of 1 to the new version.**

By using the App Engine's traffic splitting feature, we can easily direct a certain percentage of traffic to a specific version of our application.

Application that stores relational data from users across the globe and scaling is required based on size which is unknown --------- **Cloud Spanner**

Cloud SQL for small relational data, scaled manually

Cloud Spanner for relational data, scaled automatically

Cloud Firestore for app-based data.

Cloud Datastore for non-relational data.

Engineering team has the Project Creator role on the organization but should not able to link the projects to the billing account. Only Finance Team could link the projects to billing account and FT should not change to projects ----- **Assign the finance team only the Billing Account User role on the billing account.**

Finance should not provide Organizational level role otherwise it will change to projects.

Want to run a single caching HTTP reverse proxy on GCP for a latency-sensitive website. This specific reverse proxy consumes almost no CPU. I required 30-GB in-memory cache, and an additional 2 GB of memory for the rest of the processes -----

Cloud Memorystore for Redis is a managed service that provides an in-memory cache for your applications. It offers a high throughput and low latency access to the Redis protocol. Cloud Memorystore offers an SLA of 99.9% availability and automatic failover for Redis instances. In this case, a 32-GB Redis instance is sufficient to accommodate the 30-GB cache and the additional 2 GB of memory required for the rest of the processes. This solution is highly scalable and allows you to increase the size of the Redis instance as your needs grow.

----Create a Cloud Memorystore for Redis instance with 32-GB capacity is the recommended option. This option provides the required memory and is cost-effective since the proxy requires almost no CPU. Cloud Memorystore for Redis is designed specifically for in-memory caching, making it the best choice for your use case.

You are hosting an application on bare-metal servers in your own data center. The application needs access to Cloud Storage. ----- **Using Cloud VPN or Interconnect, create a tunnel to a VPC in Google Cloud.**

2. Use 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.

3. In your on-premises network, configure your DNS server to resolve \*.googleapis.com as a CNAME to restricted.googleapis.com.

It provides a secure and direct connection to Cloud Storage without requiring internet access or exposing the servers to public IP addresses.

\* By setting up a VPN or Interconnect tunnel, the on-premises servers can access Google

Cloud resources over a private and encrypted connection.

\* The custom route advertisement for 199.36.153.4/30 ensures that traffic is routed

correctly between the on-premises network and Google Cloud.

\* Configuring the DNS server to resolve \*.googleapis.com as a CNAME to

restricted.googleapis.com ensures that requests are directed to Google Cloud over the

VPN or Interconnect tunnel.

Distributed resources over different projects in Google Cloud Platform. Want to consolidate reporting under the same Stackdriver Monitoring dashboard ---- **Configure a single Stackdriver account, and link all projects to the same account.**

We cannot create a Group for all the projects in the Stockdriver Monitoring, simply create a single Stackdriver account and link all the projects to consolidate.

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

If we set Autoscaling to ON, with min 1 and max 1 instance then if 1 instance is deleted it will create other 1 instance.But if we set Autoscaling to OFF, with min 1 and max 1 instance then if 1 instance is deleted it will not create other 1 instance.

You have one project called proj-sa where you manage all your service accounts. Want to be able to use a service account from this project to take snapshots of VMs running in another project called proj-vm ---- **Grant the service account the IAM Role of Compute Storage Admin in the project called proj-vm.**

Answers A and B are incorrect because they involve downloading and adding the private key of the service account to each VM, which is not necessary and potentially risky.

To take snapshots of VMs running in another project, you need to grant the service account that will take the snapshots the necessary IAM role to perform the action.

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------- **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.**

Because one project can have only one App Engine and the region for App Engine is permanent, so to change the region we need to create new project and then App Engine and then change the region to asia-northeast1.

You need to grant access for three users so that they can view and edit table data on a Cloud Spanner instance ----- **Run gcloud iam roles describe roles/spanner.databaseUser. Add the users to a new group. Add the group to the role.**

Google Recommended - always create a group for providing roles which can simplify the management of permissions in the future.

To make sure Google Kubernetes Engine (GKE) cluster always runs a supported and stable version of Kubernetes ---- **Enable the Node Auto-Upgrades feature for your GKE cluster.**

Google Kubernetes Engine (GKE) supports multiple versions of Kubernetes, and new versions are regularly released. To ensure that your GKE cluster runs a supported and stable version of Kubernetes, it is recommended to enable the Node Auto-Upgrades feature. This feature automatically upgrades the Kubernetes version of each node in the cluster to the latest stable version.