Free Questions for Professional-Cloud-Network-Engineer by certsinside

Shared by Aguirre on 12-12-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

You are responsible for designing a new connectivity solution between your organization's on-premises data center and your Google Cloud Virtual Private Cloud (VPC) network Currently, there Is no end-to-end connectivity. You must ensure a service level agreement (SLA) of 99.99% availability What should you do?

Options:

A- Use one Dedicated Interconnect connection in a single metropolitan area. Configure one Cloud Router and enable global routing in the VPC.

B- Use a Direct Peering connection between your on-premises data center and Google Cloud. Configure Classic VPN with two tunnels and one Cloud Router.

C- Use two Dedicated Interconnect connections in a single metropolitan area. Configure one Cloud Router and enable global routing in the VPC.

D- Use HA VPN. Configure one tunnel from each Interface of the VPN gateway to connect to the corresponding interfaces on the peer gateway on-premises. Configure one Cloud Router and enable global routing in the VPC.

Answer:

B

Question 2

Question Type: MultipleChoice

Your company's logo is published as an image file across multiple websites that are hosted by your company You have implemented Cloud CDN, however, you want to improve the performance of the cache hit ratio associated with this image file. What should you do?

Options:

A- Configure custom cache keys for the backend service that holds the image file, and clear the Host and Protocol checkboxes-

B- Configure Cloud Storage as a custom origin backend to host the image file, and select multi-region as the location type

C- Configure versioned IJRLs for each domain to serve users the *mage file before the cache entry expires

D- Configure the default time to live (TTL) as O for the image file.

Answer:

A

Explanation:

This answer meets the requirement of improving the performance of the cache hit ratio associated with the image file. The reason is:

Custom cache keys allow you to control which parts of the request URL are used to build the cache key.The cache key is a unique identifier that Cloud CDN uses to store and retrieve cached content1.

By default, Cloud CDN uses the complete request URL, including the protocol (http or https) and the host (the domain name), to build the cache key.This means that if the same image file is requested from different domains or protocols, Cloud CDN will cache multiple copies of it, which reduces the cache hit ratio1.

By clearing the Host and Protocol checkboxes, you can tell Cloud CDN to ignore these parts of the request URL when building the cache key.This way, Cloud CDN will cache only one copy of the image file, regardless of which domain or protocol it is requested from, which improves the cache hit ratio1.

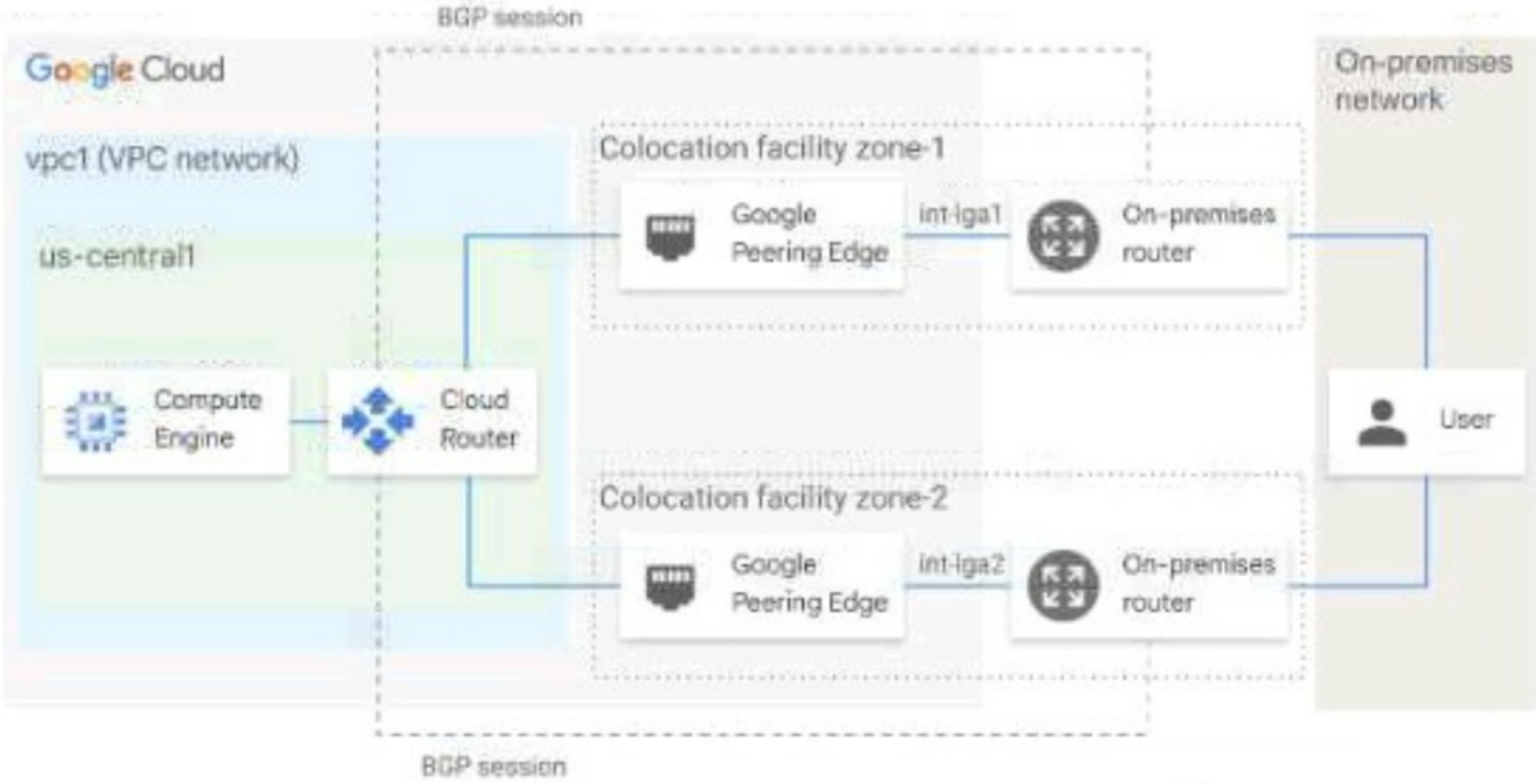
Option B is incorrect because configuring Cloud Storage as a custom origin backend does not affect the cache hit ratio. It only affects how Cloud CDN retrieves the content from the origin if it is not cached. Option C is incorrect because configuring versioned URLs for each domain does not improve the cache hit ratio. It actually worsens it, because it creates more variations of the request URL that Cloud CDN has to cache separately. Option D is incorrect because configuring the default TTL as 0 for the image file means that Cloud CDN will not cache it at all, which defeats the purpose of using Cloud CDN.

Custom cache keys | Cloud CDN | Google Cloud

Question 3

Question Type: MultipleChoice

You have the networking configuration shown in the diagram. A pair of redundant Dedicated Interconnect connections (int-Igal and int-Iga2) terminate on the same Cloud Router. The Interconnect connections terminate on two separate on-premises routers. You are advertising the same prefixes from the Border Gateway Protocol (BGP) sessions associated with the Dedicated Interconnect connections. You need to configure one connection as Active for both ingress and egress traffic. If the active Interconnect connection fails, you want the passive Interconnect connection to automatically begin routing all traffic Which two actions should you take to meet this requirement? (Choose Two)


BGP session
Google Cloud
On-premises network
vpc1 (VPC network)
Colocation facility zone-1
us-central1
Google Peering Edge
int-lga1
On-premises router
Compute Engine
Cloud Router
User
Colocation facility zone-2
Google Peering Edge
int-lga2
On-premises router
BGP session

Options:

A- Configure the advertised route priority > 10,200 on the active Interconnect connection.

B- Advertise a lower MED on the passive Interconnect connection from the on-premises router

C- Configure the advertised route priority as 200 for the BGP session associated with the active Interconnect connection.

D- Configure the advertised route priority as 200 for the BGP session associated with the passive Interconnect connection.

E- Advertise a lower MED on the active Interconnect connection from the on-premises router

Answer:

C, E

Explanation:

This answer meets the requirement of configuring one connection as Active for both ingress and egress traffic, and enabling automatic failover to the passive connection in case of failure. The reason is:

The advertised route priority is a value that Cloud Router uses to set the route priority when advertising routes to your on-premises router.The lower the value, the higher the priority1.By setting the advertised route priority as 200 for the active connection, you ensure that it has a higher priority than the passive connection, which has the default value of 1001. This way, your on-premises router will prefer the routes from the active connection over the passive one for ingress traffic.

The MED (Multi-Exit Discriminator) is a value that your on-premises router uses to indicate its preference for receiving traffic from Cloud Router.The lower the value, the higher the preference2. By advertising a lower MED on the active connection from your on-premises router, you ensure that Cloud Router will prefer sending traffic to the active connection over the passive one for egress traffic.

If the active connection fails, Cloud Router will stop receiving routes from it and will start using the routes from the passive connection for egress traffic. Similarly, your on-premises router will stop receiving routes with priority 200 from the active connection and will start using the routes with priority 100 from the passive connection for ingress traffic. This achieves automatic failover without any manual intervention.

Option A is incorrect because setting the advertised route priority > 10,200 on the active connection would deprioritize it globally in your VPC network, which is not what you want1.Option B is incorrect because advertising a lower MED on the passive connection would make Cloud Router prefer sending traffic to it over the active one, which is not what you want2.Option D is incorrect because setting the advertised route priority as 200 for both connections would make them equally preferred by your on-premises router, which is not what you want1.

Update the base route priority | Cloud Router | Google Cloud

Configuring BGP sessions | Cloud Router | Google Cloud

Question 4

Question Type: **MultipleChoice**

Your company is planning a migration to Google Kubernetes Engine. Your application team informed you that they require a minimum of 60 Pods per node and a maximum of 100 Pods per node Which Pod per node CIDR range should you use?

Options:

A- /24

B- /25

C- /26

D- /28

Answer:

B

Explanation:

To determine the Pod per node CIDR range, you need to calculate how many IP addresses are required for each node, and then choose the smallest CIDR range that can accommodate that number. A CIDR range of /n means that there are 2^(32-n) IP addresses available in that range. For example, a /24 range has 2^(32-24) = 256 IP addresses.

According to the question, the application team requires a minimum of 60 Pods per node and a maximum of 100 Pods per node. Therefore, you need to choose a CIDR range that can provide at least 100 IP addresses per node, but not more than necessary. A /25 range has 2^(32-25) = 128 IP addresses, which is enough for 100 Pods per node. A /26 range has 2^(32-26) = 64 IP addresses, which is not enough for 60 Pods per node. A /24 range has 256 IP addresses, which is more than needed and wastes IP address space. A /28 range has 2^(32-28) = 16 IP addresses, which is far too small for any node.

Therefore, the best option is B. /25.This is also consistent with the Google Kubernetes Engine documentation, which states that each node is allocated a /24 range of IP addresses for Pods by default, but the maximum number of Pods per node is 1101. This means that there are approximately twice as many available IP addresses as possible Pods, which is similar to the ratio of 128 to 100 in the /25 range.

1:Configure maximum Pods per node | Google Kubernetes Engine (GKE) | Google Cloud

Question 5

Question Type: **MultipleChoice**

You are a network administrator at your company planning a migration to Google Cloud and you need to finish the migration as quickly as possible, To ease the transition, you decided to use the same architecture as your on-premises network' a hub-and-spoke model. Your on-premises architecture consists of over 50 spokes. Each spoke does not have connectivity to the other spokes, and all traffic IS sent through the hub for security reasons. You need to ensure that the Google Cloud architecture matches your on-premises architecture. You want to implement a solution that minimizes management overhead and cost, and uses default networking quotas and limits. What should you do?

Options:

A- Connect all the spokes to the hub with Cloud VPN.

B- Connect all the spokes to the hub with VPC Network Peering.

C- Connect all the spokes to the hub With Cloud VPN. Use a third-party network appliance as a default gateway to prevent connectivity between the spokes

D- Connect all the spokes to the hub with VPC Network Peering. Use a third-party network appliance as a default gateway to prevent connectivity between the spokes.

Answer:

D

Explanation:

The correct answer is D because it meets the following requirements:

It matches the hub-and-spoke model of the on-premises network, where each spoke is a separate VPC network that is connected to a central hub VPC network.

It minimizes management overhead and cost, because VPC Network Peering is a simple and low-cost way to connect VPC networks without using any external IP addresses or VPN gateways1.

It uses default networking quotas and limits, because VPC Network Peering does not consume any quota or limit for VPN tunnels, external IP addresses, or forwarding rules2.

It prevents connectivity between the spokes, because VPC Network Peering is non-transitive by default, meaning that a spoke can only communicate with the hub, not with other spokes1.To enforce this restriction, a third-party network appliance can be used as a default gateway in each spoke VPC network, which can filter out any traffic destined for other spokes3.

Option A is incorrect because it does not minimize cost, as Cloud VPN charges for egress traffic and requires external IP addresses for the VPN gateways4.Option B is incorrect because it does not prevent connectivity between the spokes, as VPC Network Peering allows direct communication between peered VPC networks by default1. Option C is incorrect because it does not minimize cost or use default quotas and limits, for the same reasons as option A.

VPC Network Peering overview | VPC

Quotas and limits | VPC

Hub-and-spoke network architecture | Cloud Architecture Center

Cloud VPN overview | Google Cloud

Question 6

Question Type: **MultipleChoice**

Your team is developing an application that will be used by consumers all over the world. Currently, the application sits behind a global external application load balancer You need to protect the application from potential application-level attacks. What should you do?

Options:

A- Enable Cloud CDN on the backend service.

B- Create multiple firewall deny rules to block malicious users, and apply them to the global external application load balancer

C- Create a Google Cloud Armor security policy with web application firewall rules, and apply the security policy to the backend service.

D- Create a VPC Service Controls perimeter with the global external application load balancer as the protected service, and apply it to the backend service

Answer:

C

Explanation:

The correct answer is C because it meets the requirement of protecting the application from potential application-level attacks.Google Cloud Armor security policies are sets of rules that match on attributes from Layer 3 to Layer 7 to protect externally facing applications1.Web application firewall (WAF) rules are predefined rules that detect and mitigate common web attacks such as cross-site scripting (XSS), SQL injection, remote file inclusion, and more2. By applying a Google Cloud Armor security policy with WAF rules to the backend service, you can filter out malicious requests before they reach your application.

Option A is incorrect because Cloud CDN is a content delivery network that caches static content at the edge of Google's network, but it does not provide any protection against application-level attacks3.Option B is incorrect because firewall rules are applied at the VPC network level, not at the load balancer level4.Firewall rules also only match on Layer 3 and 4 attributes, not on Layer 7 attributes that are relevant for application-level attacks4. Option D is incorrect because VPC Service Controls perimeter is a feature that helps you secure

your data from unauthorized access by users outside your organization, but it does not protect your application from external attacks.

Security policy overview | Google Cloud Armor

Web application firewall (WAF) rules | Google Cloud Armor

Cloud CDN overview | Google Cloud

Using firewall rules | VPC

[VPC Service Controls overview | Google Cloud]

Question 7

Question Type: **MultipleChoice**

You ate planning to use Terraform to deploy the Google Cloud infrastructure for your company, The design must meet the following requirements

* Each Google Cloud project must represent an Internal project that your team Will work on

* After an Internal project is finished, the infrastructure must be deleted

* Each Internal project must have Its own Google Cloud project owner to manage the Google Cloud resources.

* You have 10---100 projects deployed at a time

While you are writing the Terraform code, you need to ensure that the deployment is simple and the code is reusable With

centralized management What should you do?

Options:

A- Create a Single project and additional VPCs for each internal project

B- Create a Single Shared VPC and attach each Google Cloud project as a service project

C- dCreate a Single project and Single VPC for each internal project

D- Create a Shared VPC and service project for each internal project

Answer:

D

Explanation:

The correct answer is D because it meets the following requirements:

Each internal project has its own Google Cloud project, which can be easily created and deleted by Terraform using the google_project resource1.

Each internal project has its own Google Cloud project owner, which can be assigned by Terraform using the google_project_iam_member resource1.

The deployment is simple and the code is reusable with centralized management, because the Shared VPC allows you to connect multiple service projects to a single host project that contains the network resources2.This way, you can use Terraform modules to create and manage the network resources in the host project, and then reference them in the service projects3.

Option A is incorrect because it does not create separate Google Cloud projects for each internal project, which makes it harder to delete the infrastructure and assign project owners.Option B is incorrect because it does not create separate Google Cloud projects for each internal project, and also because it attaches the service projects to a Shared VPC, which is not recommended for short-lived projects2. Option C is incorrect because it does not use a Shared VPC, which means that each internal project has to create and manage its own network resources, which increases complexity and reduces reusability.

google_project - Terraform Registry

Managing infrastructure as code with Terraform, Cloud Build, and GitOps | Google Cloud

Automating your automation by Creating Google Cloud Projects Automatically