PROFESSIONAL-CLOUD-NETWORK-ENGINEER Q&As

Professional Cloud Network Engineer

Pass Google PROFESSIONAL-CLOUD-NETWORK-ENGINEER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/professional-cloud-network-engineer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google Official Exam Center

QUESTION 1

You are designing an IP address scheme for new private Google Kubernetes Engine (GKE) clusters. Due to IP address exhaustion of the RFC 1918 address space In your enterprise, you plan to use privately used public IP space for the new clusters. You want to follow Google-recommended practices. What should you do after designing your IP scheme?

A. Create the minimum usable RFC 1918 primary and secondary subnet IP ranges for the clusters. Re-use the secondary address range for the pods across multiple private GKE clusters

B. Create the minimum usable RFC 1918 primary and secondary subnet IP ranges for the clusters Re-use the secondary address range for the services across multiple private GKE clusters

C. Create privately used public IP primary and secondary subnet ranges for the clusters.Create a private GKE cluster with the following options selected and

D. Create privately used public IP primary and secondary subnet ranges for the clusters.Create a private GKE cluster With the following options selected --disable-default-snat, --enable-ip-alias, and--enable-private-nodes

Correct Answer: D

This answer follows the Google-recommended practices for using privately used public IP (PUPI) addresses for GKE Pod address blocks1. The benefits of this approach are:

It allows you to use any public IP addresses that are not owned by Google or your organization for your Pods, which can help mitigate address exhaustion in your enterprise.

It prevents any external traffic from reaching your Pods, as Google Cloud does not route PUPI addresses to the internet or to other VPC networks by default.

It enables you to use VPC Network Peering to connect your GKE cluster to other VPC networks that use different PUPI addresses, as long as you enable the export and import of custom routes for the peering connection.

It preserves the fully integrated network model of GKE, where Pods can communicate with nodes and other resources in the same VPC network without NAT.

The options that you need to select when creating a private GKE cluster with PUPI

addresses are:

–disable-default-snat: This option disables source NAT for outbound traffic from Pods to destinations outside the cluster\\'s VPC network. This is necessary to prevent Pods from using RFC 1918 addresses as their source IP addresses, which

could cause conflicts with other networks that use the same address space2.

–enable-ip-alias: This option enables alias IP ranges for Pods and Services, which allows you to use separate subnet ranges for them. This is required to use PUPI addresses for Pods1.

–enable-private-nodes: This option creates a private cluster, where nodes do not have external IP addresses and can only communicate with the control plane through a private endpoint. This enhances the security and privacy of your

cluster3.

Option A is incorrect because it does not use PUPI addresses for Pods, but rather RFC 1918 addresses. This does not solve the problem of address exhaustion in your enterprise.

Option B is incorrect because it reuses the secondary address range for Services across multiple private GKE clusters, which could cause IP conflicts and routing issues.

Option C is incorrect because it does not specify the options that are needed to create a private GKE cluster with PUPI addresses.

1: Configuring privately used public IPs for GKE | Kubernetes Engine | Google Cloud 2: Using Cloud NAT with GKE | Kubernetes Engine | Google Cloud 3: Private clusters | Kubernetes Engine | Google Cloud

QUESTION 2

You recently deployed two network virtual appliances in us-central1. Your network appliances provide connectivity to your on-premises network, 10.0.0.0/8. You need to configure the routing for your Virtual Private Cloud (VPC). Your design must meet the following requirements:

All access to your on-premises network must go through the network virtual appliances.

Allow on-premises access in the event of a single network virtual appliance failure.

Both network virtual appliances must be used simultaneously.

Which method should you use to accomplish this?

A. Configure two routes for 10.0.0.0/8 with different priorities, each pointing to separate network virtual appliances.

B. Configure an internal HTTP(S) load balancer with the two network virtual appliances as backends. Configure a route for 10.0.0.0/8 with the internal HTTP(S) load balancer as the next hop.

C. Configure a network load balancer for the two network virtual appliances. Configure a route for 10.0.0.0/8 with the network load balancer as the next hop.

D. Configure an internal TCP/UDP load balancer with the two network virtual appliances as backends. Configure a route for 10.0.0.0/8 with the internal load balancer as the next hop.

Correct Answer: B

QUESTION 3

Your organization is implementing a new security policy to control how firewall rules are applied to control flows between virtual machines (VMs). Using Google-recommended practices, you need to set up a firewall rule to enforce strict control of traffic between VM A and VM B. You must ensure that communications flow only from VM A to VM B within the VPC, and no other communication paths are allowed. No other firewall rules exist in the VPC. Which firewall rule should you configure to allow only this communication path?

A. Firewall rule direction: ingress Action: allow Target: VM B service account Source ranges: VM A service account Priority: 1000

B. Firewall rule direction: ingress Action: allow Target: specific VM B tag Source ranges: VM A tag and VM A source IP address Priority: 1000

C. Firewall rule direction: ingress Action: allow Target: VM A service account Source ranges: VM B service account and VM B source IP address Priority: 100

D. Firewall rule direction: ingress Action: allow Target: specific VM A tag Source ranges: VM B tag and VM B source IP address Priority: 100

Correct Answer: D

QUESTION 4

You are deploying an application that runs on Compute Engine instances. You need to determine how to expose your application to a new customer You must ensure that your application meets the following requirements

1.

Maps multiple existing reserved external IP addresses to the Instance

2.

Processes IP Encapsulating Security Payload (ESP) traffic

What should you do?

A. Configure a target pool, and create protocol forwarding rules for each external IP address.

B. Configure a backend service, and create an external network load balancer for each external IP address

C. Configure a target instance, and create a protocol forwarding rule for each external IP address to be mapped to the instance.

D. Configure the Compute Engine Instances\\' network Interface external IP address from None to Ephemeral Add as many external IP addresses as required

Correct Answer: C

The correct answer is C. Configure a target instance, and create a protocol forwarding rule for each external IP address to be mapped to the instance.

This answer is based on the following facts:

A target instance is a Compute Engine instance that handles traffic from one or more forwarding rules1. You can use target instances to forward traffic to a single VM instance from one or more external IP addresses2. A protocol forwarding

rule specifies the IP protocol and port range for the traffic that you want to forward3. You can use protocol forwarding rules to forward traffic of any IP protocol, including ESP4.

The other options are not correct because:

QUESTION 5

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?

A. /24

B. /25

C. /26

D. /28

Correct Answer: B

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