Congratulations! You passed!

Grade received 100%
To pass 80% or higher

Go to next item

Google Kubernetes Engine Networking

Latest Submission Grade 100%

1. Your Pod has been rescheduled and the IP address that was assigned to the Pod when it was originally scheduled is no longer accessible. What is the reason for this? **1 / 1 point**

- (●) The new Pod has received a different IP address.
- ( ) The old Pod IP address is blocked by a firewall.
- ( ) The new Pod IP address is blocked by a firewall.
- ( ) The Pod IP range for the cluster is exhausted.

Correct
That is correct.

2. You have updated your application and deployed a new Pod. What step can you take to ensure that you can access the Pod and and the Pod's application throughout the lifecycle of the Pod using a durable IP address? **1 / 1 point**

- ( ) Add the fully qualified domain name of the application's Pod to your local hostfile.
- (●) Deploy a Kubernetes Service with a selector that locates the application's Pods.

- ◯ Add metadata annotations to the Pod manifest that define a persistent DNS name.
- ◯ Register the fully qualified domain name of the application's Pod in DNS.

✓ **Correct**
That is correct.

3. During testing you cannot find the Google Cloud Load Balancer that should have been configured for your application. You check the manifest for the applicaton and notice that the application's front-end Service type is ClusterIP. How can you correct this? **1 / 1 point**

- ◯ Manually configure the Google Cloud Load Balancer and configure it to direct traffic to the GKE Cluster Nodes.
- ◉ Define spec.type as LoadBalancer in the YAML manifest for the service and redeploy it.
- ◯ Define spec.type as NodePort in the YAML manifest for the service and redeploy it.
- ◯ Make sure the cluster has been configured in VPC native mode as Alias IP is required for Google Cloud Load Balancers.

✓ **Correct**
That is correct.

4. What change can an administrator make to achieve the lowest possible latency when using a Google Cloud load-balancer service to balance network traffic, minimizing double-hop traffic between nodes? Assume that container-native load balancing is not in use. **1 / 1 point**

- ◯ Set the externalTrafficPolicy field to roundRobin in the YAML manifest.

- ◯ Set the spec.type field to Local in the YAML manifest.
- ◉ Set the externalTrafficPolicy field to local in the YAML manifest.
- ◯ Set the spec.type field to LoadBalancer in the YAML manifest.

✓ **Correct**
That is correct.

5. You are designing a GKE solution. One of your requirements is that network traffic load balancing should be directed to Pods directly, instead of balanced across nodes. What change can you make to your environment? **1 / 1 point**

- ◯ Set the externalTrafficPolicy field to local in the YAML manifest for your external services.
- ◉ Configure or migrate your cluster to VPC-Native Mode and deploy a Container-native load balancer.
- ◯ Configure affinity and antiaffinity rules that ensure your application's Pods are distributed across Nodes.
- ◯ Configure external access using the Nodeport Service type working with an external network load balancer.
- ◯ Configure all external access for your application using Ingress resources rather than Services.

✓ **Correct**
That is correct.

6. You need to apply a network policy to five Pods to block ingress traffic from other Pods. Each Pod has a label of app:demo-app. In your network policy manifest, you have specified the label app:demo-app in spec.podSelector. The policy is configured **1 / 1 point**

and when you list the Network Policies on your cluster you can see the policy listed but is not having any effect as you can still ping the Pods from other Pods in the cluster. What is the problem and what action can you take to correct this?

- ○ Network policies are only applied when a Pod is started. Stop all of the Pods in your application and restart them in order to activate the network policy.
- ○ You need to create a matching Google Cloud firewall rule for the network policy. In the Google Cloud Console or Cloud Shell create a firewall rule that matches the Network Policy.
- ○ A network policy must be applied to all Pods in the cluster in order to block ingress traffic. In the network policy manifest, do not define any value for spec.podSelector.
- ◉ You have not enabled network policies on the cluster. Enable network policies on the cluster, and reboot all of the nodes.

✓ **Correct**
That is correct.