Missing Qs

Q42 : You've deployed a microservice called myapp1 to a Google Kubernetes Engine cluster using the YAML file specified below:  
A screenshot of a computer program

Description automatically generated  
You need to refactor this configuration so that the database password is not stored in plain text. You want to follow Google-recommended practices. What should you do?

* A. Store the database password inside the Docker image of the container, not in the YAML file.
* B. Store the database password inside a Secret object. Modify the YAML file to populate the DB\_PASSWORD environment variable from the Secret.
* C. Store the database password inside a ConfigMap object. Modify the YAML file to populate the DB\_PASSWORD environment variable from the ConfigMap.
* D. Store the database password in a file inside a Kubernetes persistent volume, and use a persistent volume claim to mount the volume to the container.

Ans: B

Q53: Your company has a 3-tier solution running on Compute Engine. The configuration of the current infrastructure is shown below.  
A screenshot of a computer

Description automatically generated  
Each tier has a service account that is associated with all instances within it. You need to enable communication on TCP port 8080 between tiers as follows:  
\* Instances in tier #1 must communicate with tier #2.  
\* Instances in tier #2 must communicate with tier #3.  
What should you do?

* A. 1. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances ג€¢ Source filter: IP ranges (with the range set to 10.0.2.0/24) ג€¢ Protocols: allow all 2. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances ג€¢ Source filter: IP ranges (with the range set to 10.0.1.0/24) ג€¢ Protocols: allow all
* B. 1. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances with tier #2 service account ג€¢ Source filter: all instances with tier #1 service account ג€¢ Protocols: allow TCP:8080 2. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances with tier #3 service account ג€¢ Source filter: all instances with tier #2 service account ג€¢ Protocols: allow TCP: 8080
* C. 1. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances with tier #2 service account ג€¢ Source filter: all instances with tier #1 service account ג€¢ Protocols: allow all 2. Create an ingress firewall rule with the following settings: ג€¢ Targets: all instances with tier #3 service account ג€¢ Source filter: all instances with tier #2 service account ג€¢ Protocols: allow all
* D. 1. Create an egress firewall rule with the following settings: ג€¢ Targets: all instances ג€¢ Source filter: IP ranges (with the range set to 10.0.2.0/24) ג€¢ Protocols: allow TCP: 8080 2. Create an egress firewall rule with the following settings: ג€¢ Targets: all instances ג€¢ Source filter: IP ranges (with the range set to 10.0.1.0/24) ג€¢ Protocols: allow TCP: 8080

Ans: B

Q72: You deployed a new application inside your Google Kubernetes Engine cluster using the YAML file specified below.  
A screen shot of a computer

Description automatically generated  
You check the status of the deployed pods and notice that one of them is still in PENDING status:  
A close-up of a white background

Description automatically generated  
You want to find out why the pod is stuck in pending status. What should you do?

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

Ans: C

Q94: 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:  
A close up of a number

Description automatically generated  
What is the most likely cause?

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

Ans: D

Q98: Your customer has implemented a solution that uses Cloud Spanner and notices some read latency-related performance issues on one table. This table is accessed only by their users using a primary key. The table schema is shown below.  
A close-up of a computer code

Description automatically generated  
You want to resolve the issue. What should you do?

* A. Remove the profile\_picture field from the table.
* B. Add a secondary index on the person\_id column.
* C. Change the primary key to not have monotonically increasing values.
* D. Create a secondary index using the following Data Definition Language (DDL): A white background with black text

  Description automatically generated

Ans: C