Ultimate IT Test Preparation Material

Google
GCP-CDOE
Cloud DevOps Engineer

- **Up to Date products, reliable and verified.**
- **Questions and Answers in PDF Format.**

Full Version Features:

- **90 Days Free Updates**
- **30 Days Money Back Guarantee**
- **Instant Download Once Purchased**
- **24 Hours Live Chat Support**

For More Information:

https://www.testsexpert.com/

- **Product Version**

Latest Version: 6.0

Question: 1

You are the on-call Site Reliability Engineer for a microservice that is deployed to a Google Kubernetes Engine (GKE) Autopilot cluster. Your company runs an online store that publishes order messages to Pub/Sub, and a microservice receives these messages and updates stock information in the warehousing system. A sales event caused an increase in orders, and the stock information is not being updated quickly enough. This is causing a large number of orders to be accepted for products that are out of stock. You check the metrics for the microservice and compare them to typical levels:

Microservice metrics	Typical state	Current state
Average CPU across all Pods	20% of Pod limit	30% of Pod limit
Average memory across all Pods	10% of Pod limit	10% of Pod limit
Pub/Sub subscription: Average oldest unacknowledged message age	347 milliseconds	8074 milliseconds
Pub/Sub subscription: Average undelivered messages	5 messages	14705 messages
Pub/Sub subscription: Average acknowledgment latency	312 milliseconds	354 milliseconds

You need to ensure that the warehouse system accurately reflects product inventory at the time orders are placed and minimize the impact on customers. What should you do?

A. Decrease the acknowledgment deadline on the subscription.
B. Add a virtual queue to the online store that allows typical traffic levels.
C. Increase the number of Pod replicas.
D. Increase the Pod CPU and memory limits.

Answer: C

Question: 2

Your team deploys applications to three Google Kubernetes Engine (GKE) environments: development, staging, and production. You use GitHub repositories as your source of truth. You need to ensure that the three environments are consistent. You want to follow Google-recommended practices to enforce and install network policies and a logging DaemonSet on all the GKE clusters in those environments. What should you do?

A. Use Google Cloud Deploy to deploy the network policies and the DaemonSet. Use Cloud Monitoring to trigger an alert if the network policies and DaemonSet drift from your source in the repository.
B. Use Google Cloud Deploy to deploy the DaemonSet and use Policy Controller to configure the network policies. Use Cloud Monitoring to detect drifts from the source in the repository and Cloud Functions to
correct the drifts.

C. Use Cloud Build to render and deploy the network policies and the DaemonSet. Set up Config Sync to sync the configurations for the three environments.
D. Use Cloud Build to render and deploy the network policies and the DaemonSet. Set up a Policy Controller to enforce the configurations for the three environments.

Answer: C

Question: 3

You are using Terraform to manage infrastructure as code within a CI/CD pipeline. You notice that multiple copies of the entire infrastructure stack exist in your Google Cloud project, and a new copy is created each
time a change to the existing infrastructure is made. You need to optimize your cloud spend by ensuring that only a single instance of your infrastructure stack exists at a time. You want to follow Google-recommended
practices. What should you do?

A. Create a new pipeline to delete old infrastructure stacks when they are no longer needed.
B. Confirm that the pipeline is storing and retrieving the terraform.tfstate file from Cloud Storage with the Terraform gcs backend.
C. Verify that the pipeline is storing and retrieving the terraform.tfstate file from a source control.
D. Update the pipeline to remove any existing infrastructure before you apply the latest configuration.

Answer: D

Question: 4

You are creating Cloud Logging sinks to export log entries from Cloud Logging to BigQuery for future analysis. Your organization has a Google Cloud folder named Dev that contains development projects and a folder
named Prod that contains production projects. Log entries for development projects must be exported to dev_dataset, and log entries for production projects must be exported to prod_dataset. You need to minimize
the number of log sinks created, and you want to ensure that the log sinks apply to future projects. What should you do?

A. Create a single aggregated log sink at the organization level.
B. Create a log sink in each project.
C. Create two aggregated log sinks at the organization level, and filter by project ID.
D. Create an aggregated log sink in the Dev and Prod folders.

Answer: A

Question: 5

Your company runs services by using multiple globally distributed Google Kubernetes Engine (GKE) clusters. Your operations team has set up workload monitoring that uses Prometheus-based tooling for metrics,
alerts, and generating dashboards. This setup does not provide a method to view metrics globally across all clusters. You need to implement a scalable solution to support global Prometheus querying and minimize
management overhead. What should you do?

A. Configure Prometheus cross-service federation for centralized data access.
B. Configure workload metrics within Cloud Operations for GKE.
C. Configure Prometheus hierarchical federation for centralized data access.
D. Configure Google Cloud Managed Service for Prometheus.

Answer: A

Question: 6

You need to build a CI/CD pipeline for a containerized application in Google Cloud. Your development team uses a central Git repository for trunk-based development. You want to run all your tests in the pipeline for
any new versions of the application to improve the quality. What should you do?

A. 1. Install a Git hook to require developers to run unit tests before pushing the code to a central repository.
2. Trigger Cloud Build to build the application container. Deploy the application container to a testing environment, and run integration tests.
3. If the integration tests are successful, deploy the application container to your production environment, and run acceptance tests.
B. 1. Install a Git hook to require developers to run unit tests before pushing the code to a central repository. If all tests are successful, build a container.
2. Trigger Cloud Build to deploy the application container to a testing environment, and run integration tests and acceptance tests.
3. If all tests are successful, tag the code as production ready. Trigger Cloud Build to build and deploy the application container to the production environment.
C. 1. Trigger Cloud Build to build the application container, and run unit tests with the container.
2. If unit tests are successful, deploy the application container to a testing environment, and run integration tests.
3. If the integration tests are successful, the pipeline deploys the application container to the production environment. After that, run acceptance tests.
D. 1. Trigger Cloud Build to run unit tests when the code is pushed. If all unit tests are successful, build and push the application container to a central registry.

2. Trigger Cloud Build to deploy the container to a testing environment, and run integration tests and acceptance tests.
3. If all tests are successful, the pipeline deploys the application to the production environment and runs smoke tests

Answer: A

Question: 7

The new version of your containerized application has been tested and is ready to be deployed to production on Google Kubernetes Engine (GKE). You could not fully load-test the new version in your pre-production
environment, and you need to ensure that the application does not have performance problems after deployment. Your deployment must be automated. What should you do?

A. Deploy the application through a continuous delivery pipeline by using canary deployments. Use Cloud Monitoring to look for performance issues, and ramp up traffic as supported by the metrics.
B. Deploy the application through a continuous delivery pipeline by using blue/green deployments. Migrate traffic to the new version of the application and use Cloud Monitoring to look for performance issues.
C. Deploy the application by using kubectl and use Config Connector to slowly ramp up traffic between versions. Use Cloud Monitoring to look for performance issues.
D. Deploy the application by using kubectl and set the spec.updateStrategy.type field to RollingUpdate. Use Cloud Monitoring to look for performance issues, and run the kubectl rollback command if there are any
issues.

Answer: B

Question: 8

You are managing an application that runs in Compute Engine. The application uses a custom HTTP server to expose an API that is accessed by other applications through an internal TCP/UDP load balancer. A
firewall rule allows access to the API port from 0.0.0.0/0. You need to configure Cloud Logging to log each IP address that accesses the API by using the fewest number of steps. What should you do first?

A. Enable Packet Mirroring on the VPC.
B. Install the Ops Agent on the Compute Engine instances.
C. Enable logging on the firewall rule.
D. Enable VPC Flow Logs on the subnet.

Answer: C

Question: 9

Your company runs an ecommerce website built with JVM-based applications and microservice architecture in Google Kubernetes Engine (GKE). The application load increases during the day and decreases during
the night. Your operations team has configured the application to run enough Pods to handle the evening peak load. You want to automate scaling by only running enough Pods and nodes for the load. What should
you do?

A. Configure the Vertical Pod Autoscaler, but keep the node pool size static.
B. Configure the Vertical Pod Autoscaler, and enable the cluster autoscaler.
C. Configure the Horizontal Pod Autoscaler, but keep the node pool size static.
D. Configure the Horizontal Pod Autoscaler, and enable the cluster autoscaler.

Answer: A

Question: 10

Your organization wants to increase the availability target of an application from 99.9% to 99.99% for an investment of $2,000. The application's current revenue is $1,000,000. You need to determine whether the increase in availability is worth the investment for a single year of usage. What should you do?

A. Calculate the value of improved availability to be $900, and determine that the increase in availability is not worth the investment.
B. Calculate the value of improved availability to be $1,000, and determine that the increase in availability is not worth the investment.
C. Calculate the value of improved availability to be $1,000, and determine that the increase in availability is worth the investment.
D. Calculate the value of improved availability to be $9,000, and determine that the increase in availability is worth the investment.

Answer: D

For More Information – **Visit link below:**

https://www.testsexpert.com/

16$ Discount Coupon: 9M2GK4NW

Features:

- Money Back Guarantee……..……...…. 

- 100% Course Coverage…… 

- 90 Days Free Updates…… 

- Instant Email Delivery after Order……