Summary

Exit

Question 1:
The development team has provided you with a Kubernetes Deployment file. You have no infrastructure yet and need to deploy the application. What should you do?

A. Use gcloud to create a Kubernetes cluster. Use Deployment Manager to create the deployment.
B. Use gcloud to create a Kubernetes cluster. Use kubectl to create the deployment.
C. Use kubectl to create a Kubernetes cluster. Use Deployment Manager to create the deployment.
D. Use kubectl to create a Kubernetes cluster. Use kubectl to create the deployment.

Type : SINGLE SELECTION

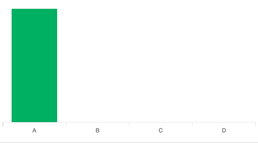
Answered By
1
Correct Answers Given By
1
Percentage
100 %

Question 2:
You have an application deployed on Kubernetes Engine using a Deployment named echo-deployment. The deployment is exposed using a Service called echo- service. You need to perform an update to the application with minimal downtime to the application. What should you do?

A. Use kubectl set image deployment/echo-deployment <new image>
B. Use the rolling update functionality of the Instance Group behind the Kubernetes cluster
C. Update the deployment yaml file with the new container image. Use kubectl delete deployment/echo-deployment and kubectl create "f <yaml-file>
D. Update the service yaml file which the new container image. Use kubectl delete service/echo-service and kubectl create "f <yaml-file>

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %



Question 3:
Your team is developing a web application that will be deployed on Google Kubernetes Engine (GKE). Your CTO expects a successful launch and you need to ensure your application can handle the expected load of tens of thousands of users. You want to test the current deployment to ensure the latency of your application stays below a certain threshold. What should you do?

A. Use a load testing tool to simulate the expected number of concurrent users and total requests to your application, and inspect the results.
B. Enable autoscaling on the GKE cluster and enable horizontal pod autoscaling on your application deployments. Send curl requests to your application, and validate if the auto scaling works.
C. Replicate the application over multiple GKE clusters in every Google Cloud region. Configure a global HTTP(S) load balancer to expose the different clusters over a single global IP address.
D. Use Cloud Debugger in the development environment to understand the latency between the different microservices.

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %

Question 4:
A development team at your company has created a dockerized HTTPS web application. You need to deploy the application on Google Kubernetes Engine (GKE) and make sure that the application scales automatically.How should you deploy to GKE?

A. Use the Horizontal Pod Autoscaler and enable cluster autoscaling. Use an Ingress resource to load-balance the HTTPS traffic.
B. Use the Horizontal Pod Autoscaler and enable cluster autoscaling on the Kubernetes cluster. Use a Service resource of type LoadBalancer to load-balance the HTTPS traffic.
C. Enable autoscaling on the Compute Engine instance group. Use an Ingress resource to load balance the HTTPS traffic.
D. Enable autoscaling on the Compute Engine instance group. Use a Service resource of type LoadBalancer to load-balance the HTTPS traffic.

Type : SINGLE SELECTION

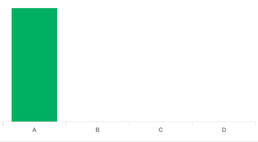
Answered By
1
Correct Answers Given By
1
Percentage
100 %

Question 5:
Your application is deployed in a Google Kubernetes Engine (GKE) cluster. You want to expose this application publicly behind a Cloud Load Balancing HTTP(S) load balancer. What should you do?

A. Configure a GKE Ingress resource.
B. Configure a GKE Service resource.
C. Configure a GKE Ingress resource with type: LoadBalancer.
D. Configure a GKE Service resource with type: LoadBalancer.

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %

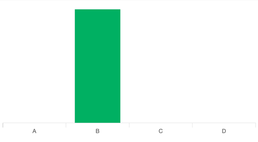


Question 6:
You team needs to create a Google Kubernetes Engine (GKE) cluster to host a newly built application that requires access to third-party services on the internet. Your company does not allow any Compute Engine instance to have a public IP address on Google Cloud. You need to create a deployment strategy that adheres to these guidelines. What should you do?

A. Create a Compute Engine instance, and install a NAT Proxy on the instance. Configure all workloads on GKE to pass through this proxy to access third-party services on the Internet
B. Configure the GKE cluster as a private cluster, and configure Cloud NAT Gateway for the cluster subnet
C. Configure the GKE cluster as a route-based cluster. Configure Private Google Access on the Virtual Private Cloud (VPC)
D. Configure the GKE cluster as a private cluster. Configure Private Google Access on the Virtual Private Cloud (VPC)

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %

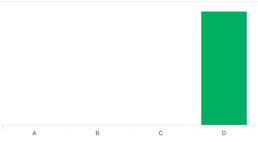


Question 7:
Your company has a stateless web API that performs scientific calculations. The web API runs on a single Google Kubernetes Engine (GKE) cluster. The cluster is currently deployed in us-central1. Your company has expanded to offer your API to customers in Asia. You want to reduce the latency for the users in Asia. What should you do?

A. Use a global HTTP(s) load balancer with Cloud CDN enabled
B. Create a second GKE cluster in asia-southeast1, and expose both API's using a Service of type Load Balancer. Add the public Ips to the Cloud DNS zone
C. Increase the memory and CPU allocated to the application in the cluster
D. Create a second GKE cluster in asia-southeast1, and use kubemci to create a global HTTP(s) load balancer

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %



Question 8:
You are developing an application using different microservices that should remain internal to the cluster. You want to be able to configure each microservice with a specific number of replicas. You want to be able to address a specific microservice from any other microservice in a uniform way regardless of the number of replicas the microservice scales to. You need to implement this solution on Google Kubernetes Engine . What should you do?

A. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using a Service, and use the Service DNS name to address it from other microservices within the cluster
B. Deploy each microservice as a Deployment. Expose the Deployment in the cluster using an ingress. And use the Ingress IP address to address the Deployment from other microservices with in the cluster
C. Deploy each microservice as a POD. Expose the POD in the cluster using a Service, and use the Service DNS name to address the microservice from other microservices within the cluster
D. Deploy each microservice as a POD. Expose the POD in the cluster using an ingress, and use the Ingress IP address to address the pod from other microservices with in the cluster

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %

Question 9:
You want to enable your running Google Kubernetes Engine cluster to scale as demand for your application changes. What should you do?

A. Add additional nodes to your Kubernetes Engine cluster using the following command: gcloud container clusters resize CLUSTER_Name "" -size 10
B. Add a tag to the instances in the cluster with the following command: gcloud compute instances add-tags INSTANCE - -tags enable- autoscaling max-nodes-10
C. Update the existing Kubernetes Engine cluster with the following command: gcloud alpha container clusters update mycluster - -enable- autoscaling - -min-nodes=1 - -max-nodes=10
D. Create a new Kubernetes Engine cluster with the following command: gcloud alpha container clusters create mycluster - -enable- autoscaling - -min-nodes=1 - -max-nodes=10 and redeploy your application

Type : SINGLE SELECTION

Answered By
1
Correct Answers Given By
1
Percentage
100 %