? MENU

?

Google Discussions

### Exam Associate Cloud Engineer All Questions

View all questions & answers for the Associate Cloud Engineer exam

Go to Exam

### **EXAM ASSOCIATE CLOUD ENGINEER TOPIC 1 QUESTION 182 DISCUSSION**

Actual exam question from Google's Associate Cloud Engineer

Question #: 182

Topic #: 1

[All Associate Cloud Engineer Questions]

Your company has developed a new application that consists of multiple microservices. You want to deploy the application to Google Kubernetes Engine (GKE), and you want to ensure that the cluster can scale as more applications are deployed in the future. You want to avoid manual intervention when each new application is deployed. What should you do?

- A. Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment.
- B. Deploy the application on GKE, and add a VerticalPodAutoscaler to the deployment.
- C. Create a GKE cluster with autoscaling enabled on the node pool. Set a minimum and maximum for the size of the node pool.
- D. Create a separate node pool for each application, and deploy each application to its dedicated node pool.

**Show Suggested Answer** 

by ?snkhatri at Sept. 3, 2022, 2:44 p.m.

# Comments

Type your comment...

Submit

| ② ① WendyL C they would ● 1 year, 3 months ago Secoted Answer C C is the right choice See this for reference https://cloud.google.com/architecture/best-practices-for-running-cost-effective-kubenetes-applications-on-gkeffine-tune_gke_autoscaling  A. HorizontalPodAutoscaler - it is best suited for stateless workers that can spin up quickly to react to usage spikes, and shut down gracefully to avoid workload instability. ② ② puyoted 5 times ② ② denno22 the Roomi ② weeks, 4 days ago Secoted Answer C C ② ② ② upvoted 1 times ② ② proted 1 times ② ② proted 1 times ② ② proted 2 times ② ② proted 1 times ② ② proted 1 times ② ② proted 2 times ② ② proted 1 times ② ② proted 2 times ② ② proted 1 times ③ ② proted 2 times ② ② proted 2 times ② ② proted 2 times ② ② proted 1 times ③ ② proted 2 times ② ② proted 3 times ③ ② proted 4 times ③ ② proted 4 times ② ② proted 5 times ③ ② proted 5 times ③ ② proted 5 times ③ ② proted 6 times ④ C the HPA is used in for scaling a deployment (an application), but here, the question is asking to scale the cluster when new applications are being added which have different and independent deployments, we have scaling at cluster level, then at deployment level either horizontally or vertically. ③ ② ② proted 1 times ⑤ ② Proted 1 times ⑥ Secoted Answerr A In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment). However, if salso important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling. ⑤ ② PiperMe 7 months, 3 weeks ago  This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, H     | ?   | Pefar_cloud Highly Voted 1 1 year, 4 months ago  Answer is C  The key point is "ensure that the CLUSTER can scale"  A- HorizontalPodAutoscaler - ensures to scale the number of pods while  C- Create a GKE cluster with autoscaling enabled on the node pool. Set a minimum and maximum for the size of the node pool.  ensures to scale the number of nodes in the cluster.  So the answer is C.   |
|--|-----|--|
| Selected Answer C C is the right choice See this for reference https://cloud.google.com/architecture/best-practices-for-running-cost-effective-kubemetes-applications-on-gke#fine-tune_gke_autoscaling A- HorizontalPodAutoscaler - it is best suited for stateless workers that can spin up quickly to react to usage spikes, and shut down gracefully to avoid workload instability.  ② ① ① upvoted 5 times ② ② den Recent ② 2 weeks, 4 days ago Selected Answer C C C ② ② ① upvoted 1 times ③ ② price 1 in upvoted 1 times ④ ② price 2 weeks, 1 day ago Selected Answer A A is right tans ⑤ ② ① upvoted 1 times ④ ② price 2 workload answer A A is right tans ⑥ ② ② upvoted 1 times Ø ② price 2 workload answer A A is right tans Ø ② upvoted 1 times Ø ② Upvoted 2 times Ø ② upvoted 1 times Ø ② upvoted 2 times Ø ② upvoted 1 times Ø ② upvoted 2 times Ø ② upvoted 1 times Ø ② upvoted 2 times Ø Weekend AlwaerrA It the context of deploying a new application on and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontaliPodAutoscaler to the deployment). H    |     | ? ? upvoted 9 times  |
| C is the right choice See this for reference https://cloud.google.com/architecture/best-practices-for-running-cost-effective-kubemetes-applications-on-gke#fine-tune_gke_autoscaling  A- HorizontalPodAutoscaler - it is best suited for stateless workers that can spin up quickly to react to usage spikes, and shut down gracefully to avoid workload instability.  ②   | ?   |  |
| shut down gracefully to avoid workload instability.  1   |     | C is the right choice See this for reference https://cloud.google.com/architecture/best-practices-for-running-cost-  |
| Part   Description   Descrip   |     | shut down gracefully to avoid workload instability.  |
| Selected Answer C  1 1 2 upvoted 1 times  2 1 2 RKS_2021 4 weeks, 1 day ago  Selected Answer A A is right ans 1 2 1 upvoted 1 times  2 1 1 yehia2221 2 months, 3 weeks ago  Answer is C: the HPA is used in for scaling a deployment (an application), but here, the question is asking to scale the cluster when new applications are being added which have different and independent deployments, we have scaling at cluster level, then at deployment level either horizontally or vertically.  2 1 1 upvoted 1 times  2 1 1 Cynthia2023 9 months, 3 weeks ago  Selected Answer A In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).  However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.  1 2 PiperMe 7 months, 3 weeks ago  This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  1 1 In Jupyoted 1 times  2 1 Mr.Jkr 1 year, 4 months ago  Selected Answer A Its A, When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by  | ?   | ·  |
| Provided 1 times  Provided 2 times  Provided 2 times  Provided 2 times  Provided 2 times  Provided 1 times  Provided 1 times  Provided 1 times  Provided 1 times  Provided 2 times  Provided 1 t |     | Selected Answer: C   |
| Picks 2021 4 weeks, 1 day ago  |     |  |
| Selected Answer: A A is right ans ② ② upvoted 1 times ② ② cynthia2023 9 months, 3 weeks ago ③ upvoted 1 times ② ② cynthia2023 9 months, 3 weeks ago ⑤ upvoted 1 times ② ② cynthia2023 9 months, 3 weeks ago ⑥ upvoted 1 times ② ② upvoted 2 times ② ② upvoted 1 times ② ② upvoted 2 times ② ② upvoted 2 times ② ② upvoted 2 times ② ② upvoted 1 times ② ② upvoted 2 times ③ Upvoted 2 times ③ Upvoted 2 times Ø ② Upvoted 3 times Ø ② Upvoted 3 times Ø ② Upvoted 4 times Ø ② Upvoted 5 times Ø ② Upvoted 5 times Ø ② Upvoted 5 times Ø ② Upvoted 6 times Ø ② Upvoted 7 times Ø ② Upvoted 8 times Ø ② Upvoted 9 times Ø ② Upvoted 9 times Ø ② Upvoted 1 times Ø ② Upvoted 1 times Ø Ø WrJkr 1 year, 4 months ago   | [2] |  |
| A is right ans [?  | ٺ   |  |
| <ul> <li>? 2 yehia2221 2 months, 3 weeks ago</li> <li>Answer is C:</li> <li>the HPA is used in for scaling a deployment (an application), but here, the question is asking to scale the cluster when new applications are being added which have different and independent deployments, we have scaling at cluster level, then at deployment level either horizontally or vertically.</li> <li>? 2 yevoted 1 times</li> <li>? 2 (Cynthia2023 9 months, 3 weeks ago</li> <li>Seiceted Answer A</li> <li>In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).</li> <li>However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.</li> <li>? 2 yeproted 2 times</li> <li>? 2 PiperMe 7 months, 3 weeks ago</li> <li>This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.</li> <li>? 2 yevoted 1 times</li> <li>? 2 yevoted 1 times</li> <li>? 3 yevoted 1 times</li> <li>? 2 yevoted 2 times</li> <li>? 3 yevoted 1 times</li> <li>? 2 yevoted 2 times</li> <li>? 3 yevoted 2 times</li> <li>? 4 yevoted 3 yever A</li> <li>yer 4 yever A</li> <li>yer 5 yever A</li> <li>yer 6 yever A</li> <li>yer 7 yevoted 2 times</li> <li>yer 9 yevoted 2 times</li> <li>? 3 yevoted 2 times</li> </ul>  |     |  |
| <ul> <li>? ? yehia2221 2 months, 3 weeks ago</li></ul>   |     |  |
| Answer is C: the HPA is used in for scaling a deployment (an application), but here, the question is asking to scale the cluster when new applications are being added which have different and independent deployments, we have scaling at cluster level, then at deployment level either horizontally or vertically.  ② ② upvoted 1 times  ② ② Cynthia2023 9 months, 3 weeks ago  Selected Answer.A  In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).  However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.  ② ② PiperMe 7 months, 3 weeks ago  This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  ② ② I upvoted 1 times  ② ② MrJkr 1 year, 4 months ago  Selected Answer.A  Its A,  When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  ② ② I upvoted 2 times  | ?   |  |
| <ul> <li>② Cynthia2023 9 months, 3 weeks ago</li> <li>Selected Answer: A</li> <li>In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).</li> <li>However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.</li> <li>② 1 2 piperMe 7 months, 3 weeks ago</li> <li>This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.</li> <li>② 2 2 upvoted 1 times</li> <li>② 1 MrJkr 1 year, 4 months ago</li> <li>Selected Answer: A</li> <li>Its A,</li> <li>When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.</li> <li>② 2 upvoted 2 times</li> </ul>   |     | Answer is C: the HPA is used in for scaling a deployment (an application), but here, the question is asking to scale the cluster when new applications are being added which have different and independent deployments, we have scaling at cluster level, then at deployment level either horizontally or vertically.   |
| Selected Answer: A In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).  However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.  2 PiperMe 7 months, 3 weeks ago This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  2 P MrJkr 1 year, 4 months ago Selected Answer: A Its A, When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  2 upvoted 2 times  |     |  |
| In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment).  However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling.  PiperMe 7 months, 3 weeks ago  This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  Piper MrJkr 1 year, 4 months ago  Selected Answer: A  Its A,  When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  | ?   | •  |
| This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  ? ? ? upvoted 1 times  ? ? MrJkr 1 year, 4 months ago  Selected Answer: A  Its A,  When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  ? ? ? upvoted 2 times  |     | In the context of deploying a new application and ensuring future scalability with minimal manual intervention, focusing on pod scalability is indeed fundamental. This is accurately addressed by option A (Deploy the application on GKE, and add a HorizontalPodAutoscaler to the deployment). However, it's also important to have node autoscaling enabled (as mentioned in option C) to ensure that the cluster can accommodate the scaling pods. Both pod and node scaling are important for a fully scalable solution, but the immediate focus when deploying a new application is typically on pod configuration and scaling. |
| Selected Answer: A  Its A,  When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  ? ? ? upvoted 2 times  |     | This is incorrect. Horizontal Pod Autoscalers scale based on pod-level metrics such as CPU. While useful, HPAs don't directly address the need to add more nodes if the underlying infrastructure is at capacity. The answer is C which provides the most effective and streamlined way to achieve automatic cluster-level scaling in a GKE environment hosting multiple microservices.  |
| Its A, When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  ? ? upvoted 2 times   | ?   | ? MrJkr 1 year, 4 months ago   |
| ·  |     | Its A, When you first deploy your workload to a Kubernetes cluster, you may not be sure about its resource requirements and how those requirements might change depending on usage patterns, external dependencies, or other factors. Horizontal Pod autoscaling helps to ensure that your workload functions consistently in different situations, and allows you to control costs by only paying for extra capacity when you need it.  |
|  | [7] |  |

i think it is A "you want to ensure that the cluster can scale as more applications are deployed in the future."

Selected Answer: A

? ? unvoted 2 times

| ? sabrinakloud 1 year, 6 months a                                      |  |    |
|--|--|----|
| Selected Answer: C   | igo  |    |
| option C   |  |    |
| ? ? upvoted 2 times  |  |    |
| ? sabrinakloud 1 year, 6 mont  | ns ago   |    |
| option A*  |  |    |
| ? ? upvoted 2 times  |  |    |
| ? dobberzoon 1 year, 6 months a  | go   |    |
| Selected Answer: A  Not knowing how many pods, and ? ? upvoted 1 times | like nooneknows said, chatgpt A is correct.  |    |
| ? nooneknows 1 year, 6 months a  | go   |    |
| Chat GPT said A is the Answer!   |  |    |
| ? ? upvoted 2 times  |  |    |
| ? lummy 1 year, 6 months ago i believe A is the answer, you cant to    | igure out how many nodes you will need in the futurehow you gotta set a maximum          |    |
| ? ? upvoted 4 times  | igate out non many neader you min need in the fatal similar you getta out a maximum      |    |
| ? abdelsha 1 year, 8 months ago  |  |    |
| How do you set the maximum num accurate here.                          | ber of nodes and you do not know how you app will scale in the future? I think A is more | re |
| ? ? upvoted 2 times  |  |    |
| ? xmh5025 1 year, 9 months ago   |  |    |
| Selected Answer: A   |  |    |
| less manual intervention   |  |    |
| ? ? upvoted 4 times  |  |    |
| ? diasporabro 2 years ago  |  |    |
| Selected Answer: C  C is the right choice See this for r               | reference https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscal      | er |
| ? ? upvoted 1 times  |  | 0. |
| ? Charumathi 2 years ago   |  |    |
| Selected Answer: C   |  |    |
| C is the correct answer, you can er                                    | able the cluster autoscaling in node pool by specifying min and max node size.           |    |
| https://cloud.google.com/kubernet ? ? upvoted 1 times                  | es-engine/docs/how-to/cluster-autoscaler#adding_a_node_pool_with_autoscaling             |    |
| ? ale_brd_111 2 years ago  |  |    |
| Selected Answer: C   |  |    |
| it's mentioning "the cluster can sca<br>? ? ? upvoted 2 times          | le" the answer is C  |    |
| Load full discussion   |  |    |
| 2544 fall disodostoffin  |  |    |



## Social Media

Facebook , Twitter
YouTube , Reddit
Pinterest



We are the biggest and most updated IT certification exam material website.

Using our own resources, we strive to strengthen the IT professionals community for free.

? ? ? ?

### © 2024 ExamTopics

ExamTopics doesn't offer Real Microsoft Exam Questions. ExamTopics doesn't offer Real Amazon Exam Questions. ExamTopics

Materials do not contain actual questions and answers from Cisco's Certification Exams.

CFA Institute does not endorse, promote or warrant the accuracy or quality of ExamTopics. CFA® and Chartered Financial Analyst® are registered trademarks owned by CFA Institute.