



- Expert Verified, Online, **Free**.

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 108 DISCUSSION

Actual exam question from Google's Associate Cloud Engineer

Question #: 108

Topic #: 1

[\[All Associate Cloud Engineer Questions\]](#)

A team of data scientists infrequently needs to use a Google Kubernetes Engine (GKE) cluster that you manage. They require GPUs for some long-running, non- restartable jobs. You want to minimize cost. What should you do?

- A. Enable node auto-provisioning on the GKE cluster.
- B. Create a VerticalPodAutscaler for those workloads.
- C. Create a node pool with preemptible VMs and GPUs attached to those VMs.
- D. Create a node pool of instances with GPUs, and enable autoscaling on this node pool with a minimum size of 1.

[Show Suggested Answer](#)

by Polok at June 6, 2020, 5:06 p.m.

## Comments

Type your comment...

[Submit](#)

Polok [Highly Voted](#) 4 years, 4 months ago

If you need something for long-running, non-restartable jobs you don't use preemptible VMs

Think answer is D.

   upvoted 72 times

  **[Removed]** **Highly Voted**  3 years, 6 months ago

Incorrect options are

B. VerticalPodAutoscaler scales PODS based on the app you deploy.

For handle infrequently GPU access, you need infrequently GPU nodes

VerticalAutoscaler Pod deployed on a non GPU node it useless,

[We can't have the node always have GPU for infrequent requests]

C. Preemptible VMs can't last long

D. For infrequent access, you don't want to have a permanent homogenous cluster.

The correct option is "A"

auto-provisioning = Attaches and deletes node pools to cluster based on the requirements.

Hence creating a GPU node pool, and auto-scaling would be better

<https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-provisioning>

   upvoted 25 times

  **kimharsh** 2 years, 11 months ago

A is not correct because you can't add a GPU node to an existing GKE cluster

Limitations

Before using GPUs on GKE, keep in mind the following limitations:

You cannot add GPUs to existing node pools.

GPU nodes cannot be live migrated during maintenance events.

GPUs are only supported with general-purpose N1 machine types.

GPUs are not supported in Windows Server node pools

REF: <https://cloud.google.com/kubernetes-engine/docs/how-to/gpus#limitations>

So the answer should be D

   upvoted 12 times

  **rachee** 2 years, 10 months ago

Your reference says existing "node pools" not GKE cluster. Auto-provisioning creates new "node pools":

<https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-provisioning>

   upvoted 5 times

  **Ridhanya** 2 years, 10 months ago

but node pools are homogenous, so how can we be sure that option A will create a GPU node pool

   upvoted 3 times

  **wjtb** 2 years, 4 months ago

<https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-provisioning>

Node auto-provisioning creates node pools based on the following information:



CPU, memory and ephemeral storage resource requests.

GPU requests

Pending Pods' node affinities and label selectors.

Pending Pods' node taints and tolerations.

   upvoted 9 times

  **dttncl** 3 years ago

I do agree A is the answer. Since this is for infrequent needs, autoscaling in letter D is not cost effective as it will always run min. of 1 instance. If we need to infrequently use a cluster, the nodes should be able to adjust based on the current need.

"With node auto-provisioning, new node pools are created and deleted automatically."

<https://cloud.google.com/kubernetes-engine/docs/how-to/node-auto-provisioning>

   upvoted 7 times

  **ngeorgiev2** 8 months ago

If the answer was "Create auto-provisioning node pool" or demand is not about GPU resources I'll agree with A too, but there is a limitation about existing node pools and GPU, so enabling of auto-provisioning will not create GPU nodes. Need to create separate GPU pool then enable auto-provisioning for it.

   upvoted 4 times

upvoted 1 times

**kyo** 3 years, 3 months ago

I think using NAP is the correct answer.

→Node Auto Provisioning (NAP a.k.a., Nodepool Auto Provisioning)

There is an introduction of NAP described below on the blog.

>The above recommendations optimize for cost. NAP, for instance, reduces costs by taking down nodes during underutilized periods.

<https://cloud.google.com/blog/products/containers-kubernetes/best-practices-for-creating-a-highly-available-gke-cluster>

upvoted 3 times

**JCH760310** 2 years, 10 months ago

they "require GPUs" - so after checking in Udemy practice tests there is similar question there. And the D answer seems to be the best fit for our scenario here.

"This option is the most optimal solution for the requirement. Rather than recreating all nodes, you create a new node pool with GPU enabled. You then modify the pod specification to target particular GPU types by adding node selector to your workload's Pod specification. You still have a single cluster, so you pay Kubernetes cluster management fee for just one cluster, thus minimizing the cost." Still better option than creating new GKE cluster with GPUs.

Ref: <https://cloud.google.com/kubernetes-engine/docs/how-to/gpus>

Ref: <https://cloud.google.com/kubernetes-engine/pricing>

upvoted 7 times

**RKS\_2021** **Most Recent** 1 month ago

**Selected Answer: A**

Changing the answer

upvoted 1 times

**RKS\_2021** 1 month ago

**Selected Answer: D**

D is right answer

upvoted 2 times

**Timfdklfajlksdjlakf** 1 month, 3 weeks ago

**Selected Answer: A**

It's A. Shit gets only auto-provisioned when your devs actually deploy something that requires a GPU. It doesn't run permanently by default thus saves costs since it only gets provisioned when neededn.

upvoted 1 times

**Timfdklfajlksdjlakf** 1 month, 4 weeks ago

**Selected Answer: A**

A is correct. If the application requires a GPU then auto-provisioning will provision a vm with a GPU

upvoted 1 times

**ccpmad** 4 months, 4 weeks ago

**Selected Answer: D**

"Enable node auto-provisioning" with GPU will not works due to limitation "You cannot add GPUs to existing node pools"

upvoted 1 times

**Sheqos** 7 months, 1 week ago

**Selected Answer: D**

Selected Answer: D

upvoted 1 times

**PiperMe** 7 months, 2 weeks ago

**Selected Answer: D**

You are not able to add GPUs to existing node pools. This significantly impacts the viability of option A.

My reasoning for D: A dedicated GPU node pool allows configuring those nodes with specific instance types, disk sizes, etc., ensuring the best fit for the long-running jobs. While it incurs some cost even with a minimum size of 1, it might still be more cost-efficient than full auto-provisioning if the jobs are infrequent but require a predictable baseline capacity. Separating GPU and non-GPU workloads can improve resource scheduling and prevent potential conflicts.

upvoted 1 times

**ngeorgiev2** 8 months ago

In my opinion, more sense has A, but then i read again and again the answer - "Enable node auto-provisioning" with GPU will not works due to limitation "You cannot add GPUs to existing node pools". If "A" was like "Create GPU node pool with

enabled auto-provisioning" this will be correct answer, in in our case should be D

? ? ? upvoted 2 times

? ? JB28 9 months, 1 week ago

The most cost-effective option for your scenario would be \*\*C. Create a node pool with preemptible VMs and GPUs attached to those VMs\*\*.

Preemptible VMs are Google Cloud's excess compute capacity. They are up to 80% cheaper than regular instances, making them a cost-effective choice for fault-tolerant workloads that do not require continuous availability<sup>3</sup>.

However, please note that preemptible VMs are subject to availability and can be preempted if Google Cloud requires access to those resources, but they will be a good choice if the jobs can tolerate occasional preemptions<sup>3</sup>.

While options A, B, and D could also be used in certain scenarios, they may not provide the same level of cost-effectiveness for long-running, non-restartable jobs that require GPUs<sup>5</sup>. Always consider the nature of your workloads and their tolerance for interruptions when choosing the right solution.

? ? ? upvoted 1 times

? ? kaby1987 9 months, 3 weeks ago

**Selected Answer: D**

The ans is D,since they require gpu

? ? ? upvoted 1 times

? ? yash\_1199 9 months, 3 weeks ago

**Selected Answer: D**

The correct option is "D"

? ? ? upvoted 1 times

? ? ogerber 10 months, 3 weeks ago

**Selected Answer: A**

It is A

Node auto-provisioning creates node pools based on the following information:

CPU, memory, and ephemeral storage resource requests.

GPU requests.

Pending Pods' node affinities and label selectors.

Pending Pods' node taints and tolerations.

<https://cloud.google.com/kubernetes-engine/docs/concepts/node-auto-provisioning>

? ? ? upvoted 2 times

? ? kelliott 10 months, 4 weeks ago

**Selected Answer: A**

For me is A

? ? ? upvoted 1 times

? ? vipinnn00980 10 months, 4 weeks ago

**Selected Answer: D**

Think answer is D.

? ? ? upvoted 1 times

? ? sinceronny 9 months, 4 weeks ago

.... any reason?

? ? ? upvoted 1 times

? ? thewalker 11 months ago

**Selected Answer: A**

Best option is A: <https://cloud.google.com/kubernetes-engine/docs/concepts/node-auto-provisioning#how-it-works>

? ? ? upvoted 1 times

[Load full discussion...](#)

Start Learning for free



## 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.