EXAMTOPICS

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

☰ MENU 🔍

⬅ **Google Discussions**

Exam Professional Machine Learning Engineer All Questions
View all questions & answers for the Professional Machine Learning Engineer exam

Go to Exam

📄 **EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 37 DISCUSSIO..**

Actual exam question from Google's Professional Machine Learning Engineer
Question #: 37
Topic #: 1

[All Professional Machine Learning Engineer Questions]

You are developing models to classify customer support emails. You created models with TensorFlow Estimators using small datasets on your on-premises system, but you now need to train the models using large datasets to ensure high performance. You will port your models to Google Cloud and want to minimize code refactoring and infrastructure overhead for easier migration from on-prem to cloud. What should you do?

- A. Use AI Platform for distributed training.

- B. Create a cluster on Dataproc for training.

- C. Create a Managed Instance Group with autoscaling.

- D. Use Kubeflow Pipelines to train on a Google Kubernetes Engine cluster.

Show Suggested Answer

by 👤 **salsabilsf** at *June 9, 2021, 11:49 a.m.*

Comments

Type your comment…

Submit

👤 **maartenalexander** `Highly Voted 👍` 3 years, 4 months ago

A. AI platform provides lower infrastructure overhead and allows you to not have to refactor your code too much (no containerization and such, like in KubeFlow).

👍 ↩ 🚩 upvoted 28 times

👤 **PhilipKoku** `Most Recent ⊘` 5 months ago

`Selected Answer: A`

A) AI Platform

👍 ↩ 🚩 upvoted 1 times

👤 **girgu** 5 months, 1 week ago

The most suitable option for minimizing code refactoring and infrastructure overhead while enabling large-scale training on Google Cloud is:

A. Use AI Platform for distributed training.
* **Simplified Workflow:** AI Platform offers a managed service for training machine learning models. You can train your existing TensorFlow Estimator code with minimal changes, reducing the need for extensive code refactoring.
* **Distributed Training:** AI Platform automatically handles distributing your training job across multiple machines, allowing you to leverage the power of Google's cloud infrastructure to train on large datasets efficiently.
* **Reduced Infrastructure Overhead:** You don't need to manage the underlying infrastructure (e.g., setting up and maintaining a cluster) yourself. AI Platform takes care of all the infrastructure provisioning and management, minimizing the workload on your team.

👍 ↩ 🚩 upvoted 2 times

👤 **fragkris** 11 months ago

`Selected Answer: A`

I chose A. Even though D is a working option, it requires us to create a GKE cluster, which requires more work.

👍 ↩ 🚩 upvoted 2 times

👤 **Sum_Sum** 11 months, 3 weeks ago

`Selected Answer: A`

A - because it has native support for TF

👍 ↩ 🚩 upvoted 1 times

👤 **harithacML** 1 year, 3 months ago

`Selected Answer: A`

A. Use AI Platform for distributed training. : Managed , low infra change migration: yes , although need code refactoring to bigquery sql
B. Create a cluster on Dataproc for training. : only cluster ? what about training?
C. Create a Managed Instance Group with autoscaling. : Same Q?
D. Use Kubeflow Pipelines to train on a Google Kubernetes Engine cluster : only training?

👍 ↩ 🚩 upvoted 2 times

👤 **M25** 1 year, 6 months ago

`Selected Answer: A`

Went with A

👍 ↩ 🚩 upvoted 1 times

👤 **Fatiy** 1 year, 8 months ago

`Selected Answer: A`

Option A is the best choice as AI Platform provides a distributed training framework, enabling you to train large-scale models faster and with less effort

👍 ↩ 🚩 upvoted 1 times

👤 **Mohamed_Mossad** 2 years, 4 months ago

`Selected Answer: A`

using options eliminations answer between A,D will vote for A as it is easier

👍 ↩ 🚩 upvoted 1 times

👤 **Mohamed_Mossad** 2 years, 4 months ago

- using options eliminations answer between A,D will vote for A as it is easier

👍 ↩ 🚩 upvoted 1 times

👤 **David_ml** 2 years, 6 months ago

`Selected Answer: A`

The answer is A. AI platform also contains kubeflow pipelines. you don't need to set up infrastructure to use it. For D you need to set up a kubernetes cluster engine. The question asks us to minimize infrastructure overheard.

👍 ↩ 🚩 upvoted 2 times

mmona19 2 years, 6 months ago

D- Kubeflow pipelines with Vertex ai provides you ability to reuse existing code using a TF conatiner in a pipeline. it helps automate the process. there is a qwiklab walking through this.
A-incorrect, question is asking resuse existing code with minimum changes. distributed deployment does not address that.

👍 ↩ ⚑ upvoted 1 times

David_ml 2 years, 6 months ago

The answer is A. AI platform also contains kubeflow pipelines. you don't need to set up infrastructure to use it. For D you need to set up a kubernetes cluster engine. The question asks us to minimize infrastructure overheard.

👍 ↩ ⚑ upvoted 2 times

A4M 2 years, 9 months ago

A - better to go with managed service and distributed

👍 ↩ ⚑ upvoted 2 times

DHEEPAK 2 years, 9 months ago

I am 100% sure that the answer is D.
Kubeflow pipelines were designed keeping:

A) Portability.
B) Composability.
C) Flexibility in mind.

This is the pain point that the kubeflow pipelines address

👍 ↩ ⚑ upvoted 1 times

David_ml 2 years, 6 months ago

The answer is A. AI platform also contains kubeflow pipelines. you don't need to set up infrastructure to use it. For D you need to set up a kubernetes cluster engine. The question asks us to minimize infrastructure overheard.

👍 ↩ ⚑ upvoted 2 times

NamitSehgal 2 years, 10 months ago

TensorFlow Estimators means require distributed and that is key feature for AI platform or later Vertex AI.

👍 ↩ ⚑ upvoted 3 times

JobQ 2 years, 10 months ago

I think is A

👍 ↩ ⚑ upvoted 1 times

---

q4exam 3 years, 1 month ago

I think the answer is either A or B, but personally think it is likely B because dataproc is a common tool box on GCP used for ML while AI platform might require refactoring. However, I dont really know A or B
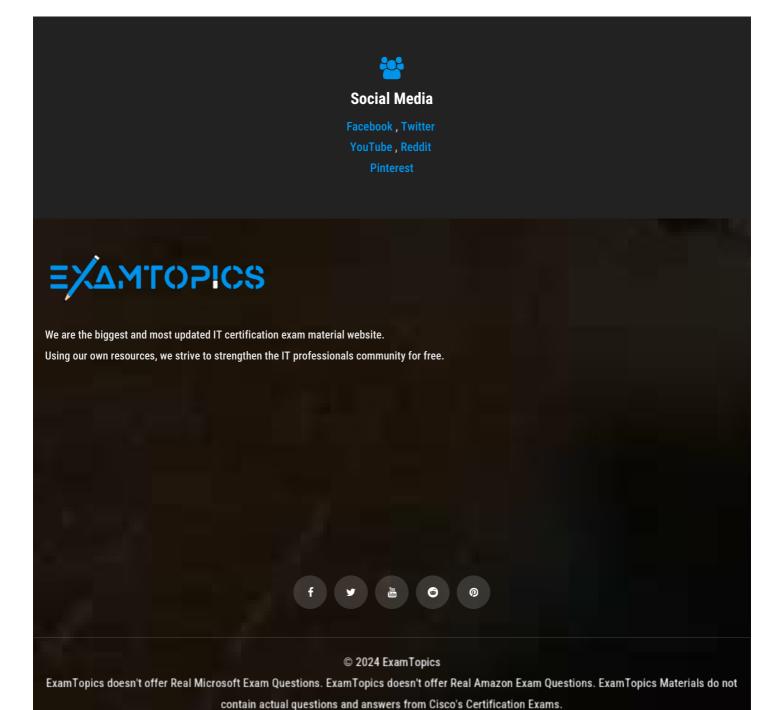
👍 ↩ ⚑ upvoted 3 times

---

george_ognyanov 3 years ago

Another vote for answer A. AI Platform distributed training here.

However, I wanted to share my logic why its not B as well. Dataproc is a managed Hadoop and as such needs a processing engine for ML tasks. Most likely Spark and SparkML. Now Spark code is quite different than pure Python and SparkML is even more different than TFcode. I imagine there might me a way to convert TF code to run on SparkML, but this seems a lot of work. And besides the question specifically wants us to minimize refactoring, so there you have it, we can eliminate option B 100%.

👍 ↩ ⚑ upvoted 5 times

Load full discussion...

Social Media

Facebook , Twitter
YouTube , Reddit
Pinterest

EXAMTOPICS

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.