Data-Engineer-Associate試験の準備方法｜検証するData-Engineer-Associate資格取得試験｜効率的なAWS Certified Data Engineer - Associate (DEA-C01)シュミレーション問題集

2026年JPTestKingの最新Data-Engineer-Associate PDFダンプおよびData-Engineer-Associate試験エンジンの無料共有：https://drive.google.com/open?id=1rOE_XgtjzfbCue6MB3--1ZAE7_J2iTmD

最新のAmazon Data-Engineer-Associateスタディガイドが作成されていることをご注意ください。 これらの試験教材は高い合格率です。 Data-Engineer-Associate学習ガイドは、今後の試験に最適な支援になると確信しています。 「ノーパス全額返金」を保証します。 過去の失敗について落ち込んでいて、有効なData-Engineer-Associate学習ガイドを探したいと思う場合は、間違いなく100％合格として試験資料に返信することをお勧めします。 私たちのData-Engineer-Associate学習ガイドに対する何千もの候補者の選択があなたの賢明な決定です。

JPTestKingのAmazon Data-Engineer-Associate問題集は専門家たちが数年間で過去のデータから分析して作成されて、試験にカバーする範囲は広くて、受験生の皆様のお金と時間を節約します。我々Data-Engineer-Associate問題集の通過率は高いので、90％の合格率を保証します。あなたは弊社の高品質Amazon Data-Engineer-Associate試験資料を利用して、一回に試験に合格します。

>> Data-Engineer-Associate資格取得 <<

Data-Engineer-Associateシュミレーション問題集 & Data-Engineer-Associate合格率

試験の準備をするためにJPTestKingのAmazonのData-Engineer-Associate試験トレーニング資料を買うのは冒険的行為と思ったとしたら、あなたの人生の全てが冒険なことになります。一番遠いところへ行った人はリスクを背負うことを恐れない人です。また、JPTestKingのAmazonのData-Engineer-Associate試験トレーニング資料が信頼できるのは多くの受験生に証明されたものです。JPTestKingのAmazonのData-Engineer-Associate試験トレーニング資料を利用したらきっと成功できますから、JPTestKingを選ばない理由はないです。

Amazon AWS Certified Data Engineer - Associate (DEA-C01) 認定 Data-Engineer-Associate 試験問題 (Q122-Q127):

質問 # 122
A company implements a data mesh that has a central governance account. The company needs to catalog all data in the governance account. The governance account uses AWS Lake Formation to centrally share data and grant access permissions.
The company has created a new data product that includes a group of Amazon Redshift Serverless tables. A data engineer needs to share the data product with a marketing team. The marketing team must have access to only a subset of columns. The data engineer needs to share the same data product with a compliance team.
The compliance team must have access to a different subset of columns than the marketing team needs access to.
Which combination of steps should the data engineer take to meet these requirements? (Select TWO.)

- A. Share the Amazon Redshift data share to the Amazon Redshift Serverless workgroup in the marketing team's account.
- B. Create views of the tables that need to be shared. Include only the required columns.
- C. Share the Amazon Redshift data share to the Lake Formation catalog in the governance account.
- D. Create an Amazon Redshift data than that includes the tables that need to be shared.
- E. Create an Amazon Redshift managed VPC endpoint in the marketing team's account. Grant the marketing team access to the views.

正解：A、B

解説：
The company is using a data mesh architecture with AWS Lake Formation for governance and needs to share specific subsets of data with different teams (marketing and compliance) using Amazon Redshift Serverless.
* Option A: Create views of the tables that need to be shared. Include only the required columns.
Creating views in Amazon Redshift that include only the necessary columns allows for fine-grained access control. This method ensures that each team has access to only the data they are authorized to view.
* Option E: Share the Amazon Redshift data share to the Amazon Redshift Serverless workgroup in the marketing team's account.Amazon Redshift data sharing enables live access to data across Redshift clusters or Serverless workgroups. By sharing data with specific workgroups, you can ensure that the marketing team and compliance team each access the relevant subset of data based on the views created.
* Option B (creating a Redshift data share) is close but does not address the fine-grained column-level access.
* Option C (creating a managed VPC endpoint) is unnecessary for sharing data with specific teams.
* Option D (sharing with the Lake Formation catalog) is incorrect because Redshift data shares do not integrate directly with Lake Formation catalogs; they are specific to Redshift workgroups.
References:
* Amazon Redshift Data Sharing
* AWS Lake Formation Documentation

質問 # 123
A company stores daily records of the financial performance of investment portfolios in .csv format in an Amazon S3 bucket. A data engineer uses AWS Glue crawlers to crawl the S3 data.
The data engineer must make the S3 data accessible daily in the AWS Glue Data Catalog.
Which solution will meet these requirements?

- A. Create an IAM role that includes the AmazonS3FullAccess policy. Associate the role with the crawler.
Specify the S3 bucket path of the source data as the crawler's data store. Create a daily schedule to run the crawler.
Configure the output destination to a new path in the existing S3 bucket.
- B. Create an IAM role that includes the AWSGlueServiceRole policy. Associate the role with the crawler.
Specify the S3 bucket path of the source data as the crawler's data store. Allocate data processing units (DPUs) to run the crawler every day. Configure the output destination to a new path in the existing S3 bucket.
- C. Create an IAM role that includes the AmazonS3FullAccess policy. Associate the role with the crawler.
Specify the S3 bucket path of the source data as the crawler's data store. Allocate data processing units (DPUs) to run the crawler every day. Specify a database name for the output.
- D. Create an IAM role that includes the AWSGlueServiceRole policy. Associate the role with the crawler.
Specify the S3 bucket path of the source data as the crawler's data store. Create a daily schedule to run the crawler. Specify a database name for the output.

正解：D

解説：
To make the S3 data accessible daily in the AWS Glue Data Catalog, the data engineer needs to create a crawler that can crawl the S3 data and write the metadata to the Data Catalog. The crawler also needs to run on a daily schedule to keep the Data Catalog updated with the latest data. Therefore, the solution must include the following steps:
* Create an IAM role that has the necessary permissions to access the S3 data and the Data Catalog. The AWSGlueServiceRole policy is a managed policy that grants these permissions1.
* Associate the role with the crawler.
* Specify the S3 bucket path of the source data as the crawler's data store. The crawler will scan the data and infer the schema and format2.
* Create a daily schedule to run the crawler. The crawler will run at the specified time every day and update the Data Catalog with any changes in the data3.
* Specify a database name for the output. The crawler will create or update a table in the Data Catalog under the specified

database. The table will contain the metadata about the data in the S3 bucket, such as the location, schema, and classification.
Option B is the only solution that includes all these steps. Therefore, option B is the correct answer.
Option A is incorrect because it configures the output destination to a new path in the existing S3 bucket. This is unnecessary and may cause confusion, as the crawler does not write any data to the S3 bucket, only metadata to the Data Catalog.
Option C is incorrect because it allocates data processing units (DPUs) to run the crawler every day. This is also unnecessary, as DPUs are only used for AWS Glue ETL jobs, not crawlers.
Option D is incorrect because it combines the errors of option A and C. It configures the output destination to a new path in the existing S3 bucket and allocates DPUs to run the crawler every day, both of which are irrelevant for the crawler.
References:
* 1: AWS managed (predefined) policies for AWS Glue - AWS Glue
* 2: Data Catalog and crawlers in AWS Glue - AWS Glue
* 3: Scheduling an AWS Glue crawler - AWS Glue
* [4]: Parameters set on Data Catalog tables by crawler - AWS Glue
* [5]: AWS Glue pricing - Amazon Web Services (AWS)

質問 # 124
A company has a frontend ReactJS website that uses Amazon API Gateway to invoke REST APIs. The APIs perform the functionality of the website. A data engineer needs to write a Python script that can be occasionally invoked through API Gateway. The code must return results to API Gateway.
Which solution will meet these requirements with the LEAST operational overhead?

- A. Create an AWS Lambda function. Ensure that the function is warm byscheduling an Amazon EventBridge rule to invoke the Lambda function every 5 minutes by usingmock events.
- B. Deploy a custom Python script on an Amazon Elastic Container Service (Amazon ECS) cluster.
- C. Create an AWS Lambda Python function with provisioned concurrency.
- D. Deploy a custom Python script that can integrate with API Gateway on Amazon Elastic Kubernetes Service (Amazon EKS).

正解：C

解説：
AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers.
You can use Lambda to create functions that perform custom logic and integrate with other AWS services, such as API Gateway. Lambda automatically scales your application by running code in response to each trigger. You pay only for the compute time you consume1.
Amazon ECS is a fully managed container orchestration service that allows you to run and scale containerized applications on AWS. You can use ECS to deploy, manage, and scale Docker containers using either Amazon EC2 instances or AWS Fargate, a serverless compute engine for containers2.
Amazon EKS is a fully managed Kubernetes service that allows you to run Kubernetes clusters on AWS without needing to install, operate, or maintain your own Kubernetes control plane. You can use EKS to deploy, manage, and scale containerized applications using Kubernetes on AWS3.
The solution that meets the requirements with the least operational overhead is to create an AWS Lambda Python function with provisioned concurrency. This solution has the following advantages:
It does not require you to provision, manage, or scale any servers or clusters, as Lambda handles all the infrastructure for you. This reduces the operational complexity and cost of running your code.
It allows you to write your Python script as a Lambda function and integrate it with API Gateway using a simple configuration. API Gateway can invoke your Lambda function synchronously or asynchronously, and return the results to the frontend website.
It ensures that your Lambda function is ready to respond to API requests without any cold start delays, by using provisioned concurrency. Provisioned concurrency is a feature that keeps your function initialized and hyper-ready to respond in double-digit milliseconds. You can specify the number of concurrent executions that you want to provision for your function.
Option A is incorrect because it requires you to deploy a custom Python script on an Amazon ECS cluster.
This solution has the following disadvantages:
It requires you to provision, manage, and scale your own ECS cluster, either using EC2 instances or Fargate. This increases the operational complexity and cost of running your code.
It requires you to package your Python script as a Docker container image and store it in a container registry, such as Amazon ECR or Docker Hub. This adds an extra step to your deployment process.
It requires you to configure your ECS cluster to integrate with API Gateway, either using an Application Load Balancer or a Network Load Balancer. This adds another layer of complexity to your architecture.
Option C is incorrect because it requires you to deploy a custom Python script that can integrate with API Gateway on Amazon EKS. This solution has the following disadvantages:

It requires you to provision, manage, and scale your own EKS cluster, either using EC2 instances or Fargate. This increases the operational complexity and cost of running your code.
It requires you to package your Python script as a Docker container image and store it in a container registry, such as Amazon ECR or Docker Hub. This adds an extra step to your deployment process.
It requires you to configure your EKS cluster to integrate with API Gateway, either using an Application Load Balancer, a Network Load Balancer, or a service of type LoadBalancer. This adds another layer of complexity to your architecture.
Option D is incorrect because it requires you to create an AWS Lambda function and ensure that the function is warm by scheduling an Amazon EventBridge rule to invoke the Lambda function every 5 minutes by using mock events. This solution has the following disadvantages:
It does not guarantee that your Lambda function will always be warm, as Lambda may scale down your function if it does not receive any requests for a long period of time. This may cause cold start delays when your function is invoked by API Gateway.
It incurs unnecessary costs, as you pay for the compute time of your Lambda function every time it is invoked by the EventBridge rule, even if it does not perform any useful work1.
References:
1: AWS Lambda - Features
2: Amazon Elastic Container Service - Features
3: Amazon Elastic Kubernetes Service - Features
[4]: Building API Gateway REST API with Lambda integration - Amazon API Gateway
[5]: Improving latency with Provisioned Concurrency - AWS Lambda
[6]: Integrating Amazon ECS with Amazon API Gateway - Amazon Elastic Container Service
[7]: Integrating Amazon EKS with Amazon API Gateway - Amazon Elastic Kubernetes Service
[8]: Managing concurrency for a Lambda function - AWS Lambda

質問 # 125
A company needs to set up a data catalog and metadata management for data sources that run in the AWS Cloud. The company will use the data catalog to maintain the metadata of all the objects that are in a set of data stores. The data stores include structured sources such as Amazon RDS and Amazon Redshift. The data stores also include semistructured sources such as JSON files and .xml files that are stored in Amazon S3.
The company needs a solution that will update the data catalog on a regular basis. The solution also must detect changes to the source metadata.
Which solution will meet these requirements with the LEAST operational overhead?

- A. Use the AWS Glue Data Catalog as the central metadata repository. Extract the schema for Amazon RDS and Amazon Redshift sources, and build the Data Catalog. Use AWS Glue crawlers for data that is in Amazon S3 to infer the schema and to automatically update the Data Catalog.
- B. Use Amazon Aurora as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the Aurora data catalog. Schedule the Lambda functions to run periodically.
- C. Use Amazon DynamoDB as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the DynamoDB data catalog. Schedule the Lambda functions to run periodically.
- D. Use the AWS Glue Data Catalog as the central metadata repository. Use AWS Glue crawlers to connect to multiple data stores and to update the Data Catalog with metadata changes. Schedule the crawlers to run periodically to update the metadata catalog.

正解：D

解説：
This solution will meet the requirements with the least operational overhead because it uses the AWS Glue Data Catalog as the central metadata repository for data sources that run in the AWS Cloud. The AWS Glue Data Catalog is a fully managed service that provides a unified view of your data assets across AWS and on-premises data sources. It stores the metadata of your data in tables, partitions, and columns, and enables you to access and query your data using various AWS services, such as Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum. You can use AWS Glue crawlers to connect to multiple data stores, such as Amazon RDS, Amazon Redshift, and Amazon S3, and to update the Data Catalog with metadata changes.
AWS Glue crawlers can automatically discover the schema and partition structure of your data, and create or update the corresponding tables in the Data Catalog. You can schedule the crawlers to run periodically to update the metadata catalog, and configure them to detect changes to the source metadata, such as new columns, tables, or partitions12.
The other options are not optimal for the following reasons:
A: Use Amazon Aurora as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the Aurora data catalog. Schedule the

Lambda functions to run periodically. This option is not recommended, as it would require more operational overhead to create and manage an Amazon Aurora database as the data catalog, and to write and maintain AWS Lambda functions to gather and update the metadata information from multiple sources. Moreover, this option would not leverage the benefits of the AWS Glue Data Catalog, such as data cataloging, data transformation, and data governance.
C: Use Amazon DynamoDB as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the DynamoDB data catalog. Schedule the Lambda functions to run periodically. This option is also not recommended, as it would require more operational overhead to create and manage an Amazon DynamoDB table as the data catalog, and to write and maintain AWS Lambda functions to gather and update the metadata information from multiple sources. Moreover, this option would not leverage the benefits of the AWS Glue Data Catalog, such as data cataloging, data transformation, and data governance.
D: Use the AWS Glue Data Catalog as the central metadata repository. Extract the schema for Amazon RDS and Amazon Redshift sources, and build the Data Catalog. Use AWS Glue crawlers for data that is in Amazon S3 to infer the schema and to automatically update the Data Catalog. This option is not optimal, as it would require more manual effort to extract the schema for Amazon RDS and Amazon Redshift sources, and to build the Data Catalog. This option would not take advantage of the AWS Glue crawlers' ability to automatically discover the schema and partition structure of your data from various data sources, and to create or update the corresponding tables in the Data Catalog.
References:
1: AWS Glue Data Catalog
2: AWS Glue Crawlers
3: Amazon Aurora
4: AWS Lambda
5: Amazon DynamoDB

質問 # 126
A company plans to use Amazon Kinesis Data Firehose to store data in Amazon S3. The source data consists of 2 MB csv files. The company must convert the .csv files to JSON format. The company must store the files in Apache Parquet format.
Which solution will meet these requirements with the LEAST development effort?

- A. Use Kinesis Data Firehose to invoke an AWS Lambda function that transforms the .csv files to JSON. Use Kinesis Data Firehose to store the files in Parquet format.
- B. Use Kinesis Data Firehose to invoke an AWS Lambda function that transforms the .csv files to JSON and stores the files in Parquet format.
- C. Use Kinesis Data Firehose to convert the csv files to JSON. Use an AWS Lambda function to store the files in Parquet format.
- D. Use Kinesis Data Firehose to convert the csv files to JSON and to store the files in Parquet format.

正解：D

解説：
The company wants to use Amazon Kinesis Data Firehose to transform CSV files into JSON format and store the files in Apache Parquet format with the least development effort.
Option B: Use Kinesis Data Firehose to convert the CSV files to JSON and to store the files in Parquet format.
Kinesis Data Firehose supports data format conversion natively, including converting incoming CSV data to JSON format and storing the resulting files in Parquet format in Amazon S3. This solution requires the least development effort because it uses built-in transformation features of Kinesis Data Firehose.
Other options (A, C, D) involve invoking AWS Lambda functions, which would introduce additional complexity and development effort compared to Kinesis Data Firehose's native format conversion capabilities.
Reference:
Amazon Kinesis Data Firehose Documentation

質問 # 127
.....

Amazon Data-Engineer-Associate資格取得 私たちは一緒に進歩し、より良くなります、Amazon Data-Engineer-Associate資格取得 実際、私たちはあなたを私たちの練習教材からブロックする障壁を取り除きます、Amazon Data-Engineer-Associate資格取得 最もプロな人々が注目しているIT専門家になりたかったら、後悔しないように速くショッピングカートを入れましょう、JPTestKingはAmazonのData-Engineer-Associateの認定試験を真実に、全面的に研究したサイトです、Amazon Data-Engineer-Associate資格取得 あなたが激しい競争から目立つようにするためのいくつかのスキルが必要です、Amazon Data-Engineer-Associate 資格取得 話と行動の距離はどのぐらいありますか。

そして、下におろされる、それから、源吉は立ちどまるData-Engineer-Associate合格率と、しばらく、かうやつてるんだ、私たちは一緒に進歩し、より良くなります、実際、私たちはあなたを私たちの練習教材からブロックする障壁を取り除きます、最もプData-Engineer-Associateロな人々が注目しているIT専門家になりたかったら、後悔しないように速くショッピングカートを入れましょう。

実際的Amazon Data-Engineer-Associate｜更新するData-Engineer-Associate資格取得試験｜試験の準備方法AWS Certified Data Engineer - Associate (DEA-C01)シュミレーション問題集

JPTestKingはAmazonのData-Engineer-Associateの認定試験を真実に、全面的に研究したサイトです、あなたが激しい競争から目立つようにするためのいくつかのスキルが必要です。

- ハイパスレートのData-Engineer-Associate資格取得 - 合格スムーズData-Engineer-Associateシュミレーション問題集 | 高品質なData-Engineer-Associate合格率 □ ➡ www.mogiexam.com □□□は、《 Data-Engineer-Associate 》を無料でダウンロードするのに最適なサイトですData-Engineer-Associate基礎訓練
- Data-Engineer-Associateファンデーション □ Data-Engineer-Associate試験解答 □ Data-Engineer-Associate資格練習 □ （ www.goshiken.com ）で☀ Data-Engineer-Associate □☀□を検索して、無料で簡単にダウンロードできますData-Engineer-Associate受験対策解説集
- Data-Engineer-Associate受験対策解説集 □ Data-Engineer-Associate認定試験トレーリング □ Data-Engineer-Associate復習資料 □ ✔ www.mogiexam.com □✔□で使える無料オンライン版➽ Data-Engineer-Associate □の試験問題Data-Engineer-Associate認定試験トレーリング
- Data-Engineer-Associate日本語独学書籍 □ Data-Engineer-Associate受験料過去問 □ Data-Engineer-Associate PDF □ □ www.goshiken.com □サイトにて✔ Data-Engineer-Associate □✔□問題集を無料で使おうData-Engineer-Associate日本語受験攻略
- Data-Engineer-Associate資料的中率 □ Data-Engineer-Associate復習対策 □ Data-Engineer-Associate資格取得 □ □ { Data-Engineer-Associate }を無料でダウンロード➽ www.passtest.jp □で検索するだけData-Engineer-Associate受験料過去問
- Data-Engineer-Associate試験解答 □ Data-Engineer-Associate復習テキスト □ Data-Engineer-Associate復習テキスト □ □ www.goshiken.com □サイトにて最新“ Data-Engineer-Associate ”問題集をダウンロードData-Engineer-Associate復習テキスト
- Data-Engineer-Associate基礎訓練 □ Data-Engineer-Associate受験対策解説集 □ Data-Engineer-Associate受験対策解説集 □ ➥ www.topexam.jp □に移動し、☀ Data-Engineer-Associate □☀□を検索して、無料でダウンロード可能な試験資料を探しますData-Engineer-Associate日本語独学書籍
- 真実的なData-Engineer-Associate資格取得一回合格-信頼的なData-Engineer-Associateシュミレーション問題集 □ Open Webサイト▹ www.goshiken.com ◃検索☀ Data-Engineer-Associate □☀□無料ダウンロードData-Engineer-Associate練習問題
- 真実的なData-Engineer-Associate資格取得一回合格-信頼的なData-Engineer-Associateシュミレーション問題集 ↖ [www.goshiken.com]の無料ダウンロード⇒ Data-Engineer-Associate ⇐ページが開きますData-Engineer-Associate PDF
- Data-Engineer-Associate練習問題 □ Data-Engineer-Associate復習対策 □ Data-Engineer-Associate基礎訓練 □ [www.goshiken.com]で「 Data-Engineer-Associate 」を検索して、無料で簡単にダウンロードできますData-Engineer-Associate復習解答例
- 効率的Data-Engineer-Associate資格取得 - 資格試験のリーダー - 素晴らしいAmazon AWS Certified Data Engineer - Associate (DEA-C01) □ （ jp.fast2test.com ）サイトで「 Data-Engineer-Associate 」の最新問題が使えるData-Engineer-Associate基礎訓練
- writeablog.net, liberationmeditation.org, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, peopleoffaithbiblecollege.org, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

2026年JPTestKingの最新Data-Engineer-Associate PDFダンプおよびData-Engineer-Associate試験エンジンの無料共有：https://drive.google.com/open?id=1rOE_XgtjzfbCue6MB3--1ZAE7_J2iTmD